EXHIBIT C-1
Defendant's Preliminary Invalidity Contentions
Orckit Corporation v. Cisco Systems, Inc., 2:22-cv-00276-JRG-RSP

Chart for U.S. Patent 7,545,740 ("the '740 Patent")
U.S. Patent Publication No. 2006/0221974 to Hilla et al. ("Hilla")

As shown in the chart below, all Asserted Claims of the '740 Patent are invalid under (1) 35 U.S.C. § 102 (a), (e), and (g) because Hilla meets each element of those claims, and/or (2) 35 U.S.C. § 103 because Hilla renders those claims obvious either alone, or in combination with the knowledge of a person having ordinary skill in the art, and in further combination with the references specifically identified below and in the following claim chart and/or one or more references identified in Defendant's Preliminary Invalidity Contentions. The following quotations and diagrams come from Hilla titled "Method And Apparatus For Dynamic Load Balancing Over A Network Link Bundle", which was filed on April 2, 2005, and published on October 5, 2006.

Motivations to combine the disclosures in Hilla with disclosures in other publications known in the art, as explained in this chart, include at least the similarity in subject matter between the references to the extent they concern methods of data communication systems, and specifically to methods and systems for link aggregation in a data communication network. Insofar as the references cite other patents or publications, or suggest additional changes, one of ordinary skill in the art would look beyond a single reference to other references in the field.

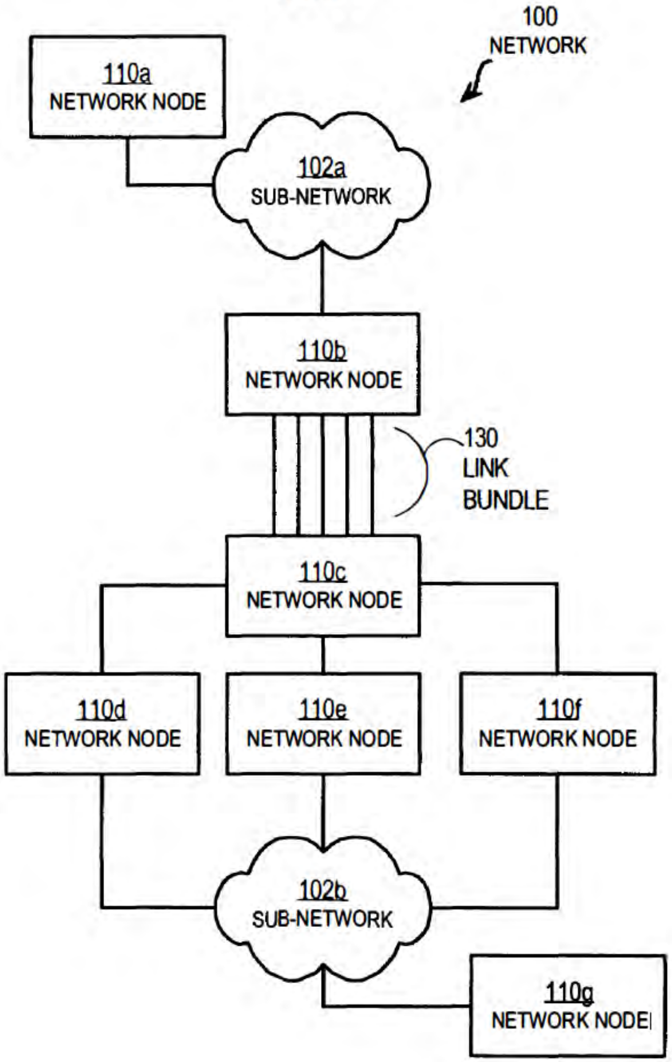
These invalidity contentions are based on Defendant's present understanding of the asserted claims, and Orckit's apparent construction of the claims in its November 3, 2022 Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1, and Orckit's January 19, 2023 First Amended Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1 (Orckit's "Infringement Disclosures"), which is deficient at least insofar as it fails to cite any documents or identify accused structures, acts, or materials in the Accused Products with particularity. Defendant does not agree with Orckit's application of the claims, or that the claims satisfy the requirements of 35 U.S.C. § 112. Defendant's contentions herein are not, and should in no way be seen as, admissions or adoptions as to any particular claim scope or construction, or as any admission that any particular element is met by any accused product in any particular way. Defendant objects to any attempt to imply claim construction from this chart. Defendant's prior art invalidity contentions are made in a variety of alternatives and do not represent Defendant's agreement or view as to the meaning, definiteness, written description support for, or enablement of any claim contained therein.

The following contentions are subject to revision and amendment pursuant to Federal Rule of Civil Procedure 26(e), the Local Rules, and the Orders of record in this matter subject to further investigation and discovery regarding the prior art and the Court's construction of the claims at issue.

No.	'740 Patent Claim 1	Hilla
1[preamble]	A method for communication, comprising:	Hilla discloses a method for communication. For example, Hilla discloses communication techniques for distributing data packets between nodes over a network link bundle. Hilla at Abstract ("Techniques for distributing data packets over a network link bundle include storing an output data packet in a data flow queue based on a flow identification associated with the output data packet. The flow identification indicates a set of one or more data packets, including the output data packet, which are to be sent in the same sequence as received. State data is also received. The state data indicates a physical status of a first port of multiple active egress ports that are connected to a corresponding bundle of communication links with one particular network device. A particular data flow queue is determined based at least in part on the state data. A next data packet is directed from the particular data flow queue to a second port of the active egress ports. These techniques allow a more efficient use of a network link bundle.") Hilla at [0006] ("Communications between nodes are typically effected by exchanging discrete packets of data. Each packet typically comprises 1] header information associated with a particular protocol, and 2] payload information that follows the header information and contains information to be processed, often independently of that particular protocol. In some protocols, the packet includes 3] trailer information following the payload and indicating the end of the payload information. The header includes information such as the source of the packet, its destination, the length of the payload, and other properties used by the protocol. Often, the data in the payload for the particular protocol includes a header and payload for a different protocol associated with a different, higher layer of the OSI Reference Model. The higher layer protocol is said to be encapsulated in the lower layer protocol. The headers included in a packet traversing multiple heterogeneous networks, such as the Internet, typi-cally include a physical (layer 1) header, a data-link (layer 2) header, an internetwork

No.	’740 Patent Claim 1	Hilla
		(layer 3) header and a transport (layer 4) header, as defined by the Open Systems Intercon-nection (OSI) Reference Model.”) Hilla at [0033] (“A method and apparatus are described for dynamic balancing of data packet traffic loads over a link bundle in a network. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present inven-tion. It will be apparent, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.”) Hilla at [0034] (“The invention is described in the following sections in the context of an Ethernet 802.3ad link bundle between two routers in the core or backbone of an enterprise network, but the invention is not limited to this context. In other embodiments, the link bundle is at a network edge, for example connecting an end node performing as a host for a high-throughput network server process of a client-server application. In other embodiments, different communication links are bundled between two network nodes, such as Packet over SONET (POS), and High-level Data Link Control (HDLC) links, among others.”) Hilla at [0036] (“FIG. 1 illustrates a link bundle 130 that includes five communication links between intermediate network node 110c and network node 110b. Network node 110b may be an intermediate network node or an end node that is connected both to intermediate network node 11 Oc and subnetwork 102a. For purposes of illustration, intermediate network node 110c is connected by a bundle 130 of five communication links to network node 110b and thereby to subnetwork 102a and end node 11 Oa. Similarly, intermediate network node 110c is connected by three unbundled com-munication links to network nodes 110d, ll0e, 110/, respec-tively, and thence to sub-network 102b and end node 110g. In other embodiments an intermediate network node 11 Oc may be connected to more or fewer network nodes with more or fewer links in each of one or more link bundles as part of a network with the same or more end nodes.”)

No.	’740 Patent Claim 1	Hilla
		Hilla at [0065] (“FIG. 4A is a flow diagram that illustrates a method 400 for dynamically balancing data packet traffic load on a link bundle, according to an embodiment. Although steps are shown in FIG. 4A and subsequent flow diagrams 4B and 5 in a particular order for purposes of illustration, in other embodiments one or more steps are performed in a different order or overlapping in time or are omitted, or changed in some combination of ways. For example, in a preferred embodiment shown in FIG. 4B, step 490 to advance age counters is performed in a separate aging process 244 on DLB block 240, and thus overlaps in time any of the other steps depicted in FIG. 4A.”) Hilla at Figure 1

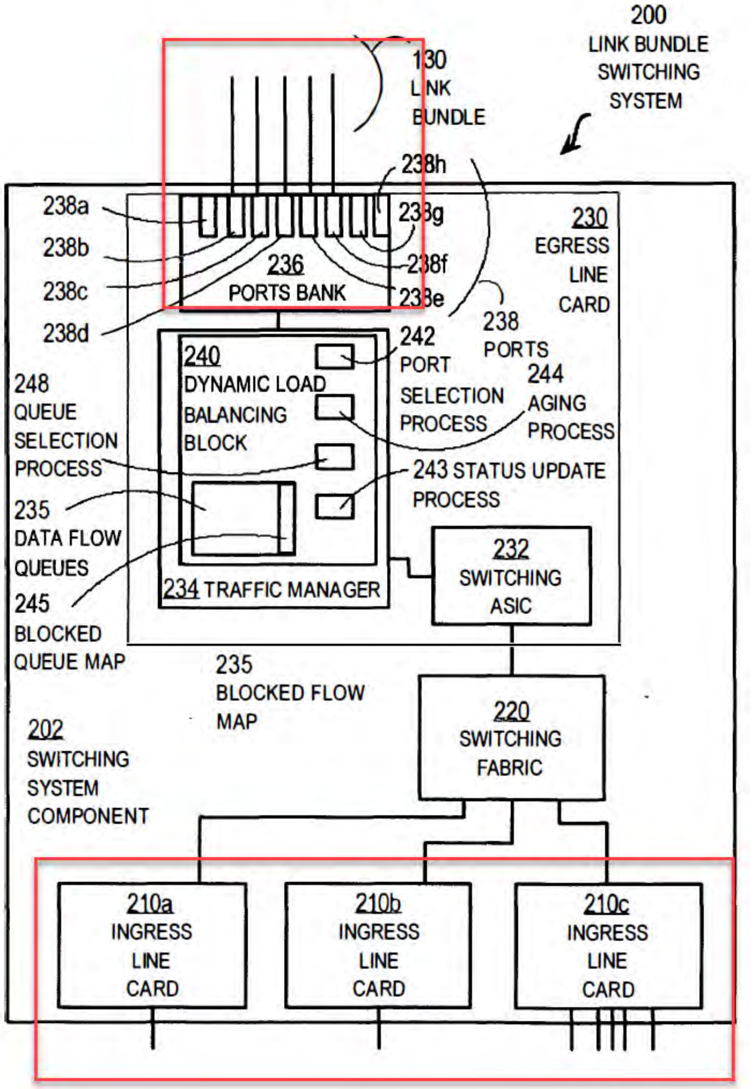
No.	’740 Patent Claim 1	Hilla
		FIG. 1  100 NETWORK 110a NETWORK NODE 102a SUB-NETWORK 110b NETWORK NODE 130 LINK BUNDLE 110c NETWORK NODE 110d NETWORK NODE 110e NETWORK NODE 110f NETWORK NODE 102b SUB-NETWORK 110g NETWORK NODE

No.	’740 Patent Claim 1	Hilla
1[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel,	Hilla discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. For example, Hilla discloses connecting network nodes to line cards using communication links such as ports arranged in parallel. Hilla at [0004] (“Networks of general purpose computer systems connected by external communication links are well known and widely used in commerce. The networks often include one or more network devices that facilitate the passage of information between the computer systems. A network node is a network device or computer system connected by the communication links.”) Hilla at [0005] (“Information is exchanged between network nodes according to one or more of many well known, new or still developing protocols. In this context, a protocol consists of a set of rules defining how the nodes interact with each other based on information sent over the communication links. The protocols are effective at different layers of operation within each node, from generating and receiving physical signals of various types, to selecting a link for transferring those signals, to the format of information indicated by those signals, to identifying which software application executing on a computer system sends or receives the information. The conceptually different layers of protocols for exchanging information over a network are described in the Open Systems Interconnection (OSI) Reference Model. The OSI Reference Model is generally described in more detail in Section 1.1 of the reference book entitled Interconnections Second Edition, by Radia Perlman, published September 1999, which is hereby incorporated by reference as though fully set forth herein.”) Hilla at [0008] (“Routers and switches are network devices that determine which communication link or links to employ to support the progress of packets through the network. For example, Ethernet switches forward packets according to the Ethernet protocol. Some current routers implement sophis-ticated algorithms that provide high performance forwarding of packets based on combining layer 2 and layer 3 header information, or some other combination. For example, instead of making forwarding decisions separately on each packet in a stream of related packets (called a "packet flow" or simply a "flow"), such as a stream

No.	’740 Patent Claim 1	Hilla
		directed from the same source node to the same destination node, these routers identify the packet flow from a unique signature derived from the layer 2 or layer 3 header information and forward each member of the flow according to the same decision made for the first packet in the flow.”) Hilla at [0012] (“A load-balancing process is used on the sending network node of the pair connected by a bundle of commu-nication links for the purpose of determining which com-munication link to use for sending one or more data packets to the receiving network node of the pair. Current balancing algorithms use a fixed mapping to associate data packets with a specific port in a set of ports connected to the communication links in the bundle. Typically, information in a header portion of a data packet is used to derive a value that is associated with one port of the set. The algorithm is designed to generate a value in a range of values that are associated with the full set of ports. Thus data packets directed to the receiving node are distributed over all com-munication links in the bundle by the load balancing pro-cess. Many load-balancing processes are designed so that all data packets in the same data flow are sent through the same port.”) Hilla at [0035] (“FIG. 1 is a block diagram that illustrates a network with a link bundle, according to an embodiment. A computer network is a geographically distributed collection of inter-connected sub-networks (e.g, sub-networks 102a, 102b, collectively referenced hereinafter as sub-networks 102) for transporting data between network nodes (e.g., network nodes 110a, 110b, 110c, 110d, ll0e, 110/, 110g, collectively referenced hereinafter as network nodes 110). A local area network (LAN) is an example of such a sub-network 102. The network's topology is defined by an arrangement of end nodes that communicate with one another, typically through one or more intermediate network nodes such as a router or switch, which facilitates routing data between end nodes. As used herein, an end node is a node that is configured to originate or terminate communications over the network. In contrast, an intermediate network node facilitates the pas-sage of data between end nodes. In FIG. 1, the network nodes 110 include both end nodes (e.g., nodes 110a, 110g) and intermediate nodes (e.g., network node 110c). Each sub-network 102 includes zero or more intermediate net-work nodes.”)

No.	’740 Patent Claim 1	Hilla
		Hilla at [0036] (“FIG. 1 illustrates a link bundle 130 that includes five communication links between intermediate network node 110c and network node 110b. Network node 110b may be an intermediate network node or an end node that is connected both to intermediate network node 11 Oc and subnetwork 102a. For purposes of illustration, intermediate network node 110c is connected by a bundle 130 of five communication links to network node 110b and thereby to subnetwork 102a and end node 11 Oa. Similarly, intermediate network node 110c is connected by three unbundled com-munication links to network nodes 110d, ll0e, 110/, respec-tively, and thence to sub-network 102b and end node 110g. In other embodiments an intermediate network node 11 Oc may be connected to more or fewer network nodes with more or fewer links in each of one or more link bundles as part of a network with the same or more end nodes.”) Hilla at [0038] (“FIG. 2 is a block diagram that illustrates a link-bundle switching system 200 in a router, which can be used as intermediate network node 11 Oc, according to an embodi-ment. The switching system 200 includes a switching system component 202 connected to communication links, includ-ing link bundle 130. The switching system component 202 includes multiple ingress line cards, including cards 210a, 210b, 210c, for receiving inbound data packets, switching fabric component 220, and multiple egress line cards for sending outbound data packets, including egress line card 230 for link bundle 130.”) Hilla at [0039] (“An ingress line card (e.g., 210a) receives a data packet, inspects the appropriate header fields to make a forwarding decision, and forwards the packet toward the egress line card via the switching fabric component 220. The switch fabric component 220 forwards the packet to the appropriate egress line card 230. Multiple ingress line cards can forward traffic toward a single egress line card 230. In this example, the link bundle 130 is capable of transmitting more packets simultaneously because of the increased band-width achieved by bundling. Some ingress line cards (e.g., 210c) are connected to a link bundle with multiple commu-nication links. All communication links in the link bundle are considered a single ingress logical link. In the illustrated embodiment, the five communication links connected to ingress line card 210c are connected to network node 110b. The ingress line card 210c handles incoming traffic on the link bundle 130, while the egress line card 230 handles the outbound traffic on the link bundle 130.”)

No.	’740 Patent Claim 1	Hilla
		Hilla at [0040] (“The ingress line cards 210 determine an outbound physical or logical link to use to forward the data packet, for example using data in a routing table stored on the router, and sends information to the egress line card associated with the outbound physical or logical link. All communication links in the link bundle 130 are considered a single logical link. Information about data packets to be output on the link bundle 130 is sent to the egress line card 230.”) Hilla at [0041] (“The egress line card 230 includes a switching application specific integrated circuit (ASIC) 232, a traffic manager block 234, a bank 236 of physical ports, at least some of which are connected to the communication links in the link bundle. In the illustrated embodiment, a dynamic load balancing (DLB) block 240 is included in the traffic manager 234. In some embodiments, the switching ASIC 232, traffic manager 234 and ports bank 236 are standard components of conventional egress cards for link bundles, and the DLB block 240 is external to the traffic manager 234.”) Hilla at [0054] (“In some embodiments, the flow control status table 344 is in the same memory block as the port buffers 340. In some embodiments, the flow control status table 344 is in a different memory block. For example, in some embodi-ments, the port buffers 340 are in a main memory for the router and the flow control status table 344 is on egress line card 230, such as on ports bank 236 or in DLB block 240. In an illustrated embodiment, the port buffers 340 are in the ports bank memory on the egress line card 230. The contents of the flow control status bits are generated by the ports bank memory fill levels and fed directly to a register or small memory for the flow control status records 342 in the DLB 240.”) Hilla at Figure 2 (annotations added)

No.	’740 Patent Claim 1	Hilla
		FIG. 2  200 LINK BUNDLE SWITCHING SYSTEM 130 LINK BUNDLE 238h 238a 238g 238b 236 PORTS BANK 238f 238c 238e 238d 230 EGRESS LINE CARD 238 PORTS 242 PORT SELECTION PROCESS 240 DYNAMIC LOAD BALANCING BLOCK 244 AGING PROCESS 248 QUEUE SELECTION PROCESS 243 STATUS UPDATE PROCESS 235 DATA FLOW QUEUES 245 BLOCKED QUEUE MAP 234 TRAFFIC MANAGER 232 SWITCHING ASIC 235 BLOCKED FLOW MAP 220 SWITCHING FABRIC 202 SWITCHING SYSTEM COMPONENT 210a INGRESS LINE CARD 210b INGRESS LINE CARD 210c INGRESS LINE CARD

No.	’740 Patent Claim 1	Hilla
1[b]	at least one of said first physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction	Hilla discloses at least one of said first physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction. For example, Hilla discloses sending inbound and outbound packets over communications link such as ports to communicate in both an upstream direction and a downstream direction. Hilla at [0035] (“FIG. 1 is a block diagram that illustrates a network with a link bundle, according to an embodiment. A computer network is a geographically distributed collection of inter-connected sub-networks (e.g, sub-networks 102a, 102b, collectively referenced hereinafter as sub-networks 102) for transporting data between network nodes (e.g., network nodes 110a, 110b, 110c, 110d, ll0e, 110/, 110g, collectively referenced hereinafter as network nodes 110). A local area network (LAN) is an example of such a sub-network 102. The network's topology is defined by an arrangement of end nodes that communicate with one another, typically through one or more intermediate network nodes such as a router or switch, which facilitates routing data between end nodes. As used herein, an end node is a node that is configured to originate or terminate communications over the network. In contrast, an intermediate network node facilitates the pas-sage of data between end nodes. In FIG. 1, the network nodes 110 include both end nodes (e.g., nodes 110a, 110g) and intermediate nodes (e.g., network node 110c). Each sub-network 102 includes zero or more intermediate net-work nodes.”) Hilla at [0036] (“FIG. 1 illustrates a link bundle 130 that includes five communication links between intermediate network node 110c and network node 110b. Network node 110b may be an intermediate network node or an end node that is connected both to intermediate network node 11 Oc and subnetwork 102a. For purposes of illustration, intermediate network node 110c is connected by a bundle 130 of five communication links to network node 110b and thereby to subnetwork 102a and end node 11 Oa. Similarly, intermediate network node 110c is connected by three unbundled com-munication links to network nodes 110d, ll0e, 110/, respec-tively, and thence to sub-network 102b and end node 110g. In other embodiments an intermediate network node 11 Oc may be connected to more or fewer network nodes with more or fewer links in each of one or more link bundles as part of a network with the same or more end nodes.”)

No.	’740 Patent Claim 1	Hilla
		Hilla at [0038] (“FIG. 2 is a block diagram that illustrates a link-bundle switching system 200 in a router, which can be used as intermediate network node 11 Oc, according to an embodi-ment. The switching system 200 includes a switching system component 202 connected to communication links, includ-ing link bundle 130. The switching system component 202 includes multiple ingress line cards, including cards 210a, 210b, 210c, for receiving inbound data packets, switching fabric component 220, and multiple egress line cards for sending outbound data packets, including egress line card 230 for link bundle 130.”) Hilla at [0039] (“An ingress line card (e.g., 210a) receives a data packet, inspects the appropriate header fields to make a forwarding decision, and forwards the packet toward the egress line card via the switching fabric component 220. The switch fabric component 220 forwards the packet to the appropriate egress line card 230. Multiple ingress line cards can forward traffic toward a single egress line card 230. In this example, the link bundle 130 is capable of transmitting more packets simultaneously because of the increased band-width achieved by bundling. Some ingress line cards (e.g., 210c) are connected to a link bundle with multiple commu-nication links. All communication links in the link bundle are considered a single ingress logical link. In the illustrated embodiment, the five communication links connected to ingress line card 210c are connected to network node 110b. The ingress line card 210c handles incoming traffic on the link bundle 130, while the egress line card 230 handles the outbound traffic on the link bundle 130.”) Hilla at [0040] (“The ingress line cards 210 determine an outbound physical or logical link to use to forward the data packet, for example using data in a routing table stored on the router, and sends information to the egress line card associated with the outbound physical or logical link. All communication links in the link bundle 130 are considered a single logical link. Information about data packets to be output on the link bundle 130 is sent to the egress line card 230.”) Hilla at [0044] (“The ports bank 236 is a bank of ports which includes multiple ports that are treated by a router as a single logical port for purposes of communicating through link bundle 130 as a single logical link. Each communication link in the link bundle 130 is connected to a

No.	’740 Patent Claim 1	Hilla
		different port of the ports bank 236. In the illustrated embodiment, ports bank 236 includes eight ports 238a, 238b, 238c, 238d, 238e, 238/, 238g, 238h (collectively referenced hereinafter as ports 238). For purposes of illustration, it is assumed that these eight ports are identified by a three bit code that represents the eight values from O through 7 to correspond to ports 238a through 238h, respectively. Each three bit code is called a port number (port #), herein. Only five of the ports (e.g., 238b, 238c, 238d, 238e, 238/, identified by code values 1 through 5, respectively) are connected to communication links of the link bundle 130 in the illustrated embodiment. These are called active ports. The remaining three ports (e.g., 238a, 238g, 238h, identified by code values 0, 6, 7, respectively) are inactive ports. In some embodiments, a port becomes inactive by virtue of a communication link that is attached to the port going down. When a port goes down, it is no longer eligible to send traffic and the link is removed from the route table. In an example embodiment, the hard-ware described in more detail below would detect the link's inability to forward traffic and dynamically shift traffic over to an active link. Such a hardware shift can occur much faster than waiting for a routing table update.”)
1[c]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel,	Hilla discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. For example, Hilla discloses connecting the line cards to a network via a switching system including a switch fabric and bus using parallel connections. A person of ordinary skill would understand that the switching system is connected to ingress and egress line cards through parallel connections that comprise the second ground of second physical links. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Hilla is found to not meet this limitation, coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel would have been obvious to a person having ordinary skill in the art, as explained below. Hilla at [0038] (“FIG. 2 is a block diagram that illustrates a link-bundle switching system 200 in a router, which can be used as intermediate network node 11 Oc, according to an embodi-ment. The switching system 200 includes a switching system component 202

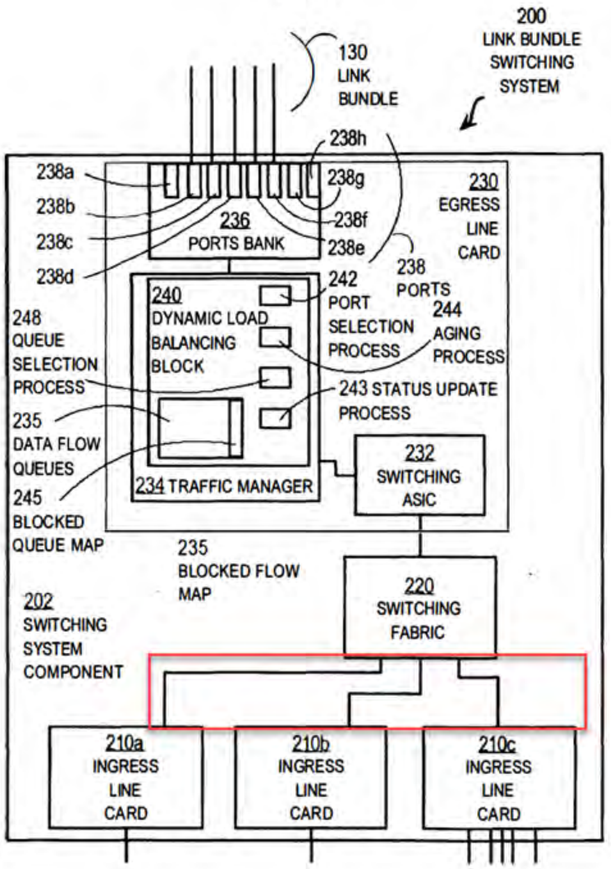
No.	’740 Patent Claim 1	Hilla
		connected to communication links, includ-ing link bundle 130. The switching system component 202 includes multiple ingress line cards, including cards 210a, 210b, 210c, for receiving inbound data packets, switching fabric component 220, and multiple egress line cards for sending outbound data packets, including egress line card 230 for link bundle 130.”) Hilla at [0039] (“An ingress line card (e.g., 210a) receives a data packet, inspects the appropriate header fields to make a forwarding decision, and forwards the packet toward the egress line card via the switching fabric component 220. The switch fabric component 220 forwards the packet to the appropriate egress line card 230. Multiple ingress line cards can forward traffic toward a single egress line card 230. In this example, the link bundle 130 is capable of transmitting more packets simultaneously because of the increased band-width achieved by bundling. Some ingress line cards (e.g., 210c) are connected to a link bundle with multiple commu-nication links. All communication links in the link bundle are considered a single ingress logical link. In the illustrated embodiment, the five communication links connected to ingress line card 210c are connected to network node 110b. The ingress line card 210c handles incoming traffic on the link bundle 130, while the egress line card 230 handles the outbound traffic on the link bundle 130.”) Hilla at [0042] (“The switching ASIC performs various standard functions well known in the art, such as retrieving data packet contents from router memory based on information received from the switching fabric 220 and appending a revised header to the beginning of the packet.”) Hilla at [0067] (“In step 412 a current flow ID is determined for the current data packet. Any method known in the art at the time the method 400 is implemented may be used. In some embodiments, step 412 includes determining and combining the bit values in one or more fields in the header portions of one or more protocols. In some embodiments, step 412 involves receiving the current flow ID from another com-ponent, e.g., from switching fabric block 220, or switching ASIC 232 or traffic manager block 234. In some embodi-ments, the flow ID is the same as the particular queue 332 where the data packet is stored by the traffic manager 234. In some embodiments in which packet sequence within a data flow is not important and commit depth is not used to select ports, step 412 is omitted.)

No.	’740 Patent Claim 1	Hilla
		Hilla at [0121] (“Computer system 600 includes a communication mechanism such as a bus 610 for passing information between other internal and external components of the computer system 600. Information is represented as physical signals of a measurable phenomenon, typically electric voltages, but including, in other embodiments, such phe-nomena as magnetic, electromagnetic, pressure, chemical, molecular atomic and quantum interactions. For example, north and south magnetic fields, or a zero and non zero electric voltage, represent two states (0, 1) of a binary digit (bit). A sequence of binary digits constitutes digital data that is used to represent a number or code for a character. A bus 610 includes many parallel conductors of information so that information is transferred quickly among devices coupled to the bus 610. One or more processors 602 for processing information are coupled with the bus 610. A processor 602 performs a set of operations on information. The set of operations include bringing information in from the bus 610 and placing information on the bus 610. The set of opera-tions also typically include comparing two or more units of information, shifting positions of units of information, and combining two or more units of information, such as by addition or multiplication. A sequence of operations to be executed by the processor 602 constitute computer instruc-tions.”) Hilla at [0122] (“Computer system 600 also includes a memory 604 coupled to bus 610. The memory 604, such as a random access memory (RAM) or other dynamic storage device, stores information including computer instructions. Dynamic memory allows information stored therein to be changed by the computer system 600. RAM allows a unit of information stored at a location called a memory address to be stored and retrieved independently of information at neighboring addresses. The memory 604 is also used by the processor 602 to store temporary values during execution of computer instructions. The computer system 600 also includes a read only memory (ROM) 606 or other static storage device coupled to the bus 610 for storing static information, including instructions, that is not changed by the computer system 600. Also coupled to bus 610 is a non-volatile (persistent) storage device 608, such as a mag-netic disk or optical disk, for storing information, including instructions, that persists even when the computer system 600 is turned off or otherwise loses power.”) Hilla at [0126] (“Computer system 600 also includes one or more instances of a communications interface 670 coupled to bus 610. Communication interface 670 provides a

No.	’740 Patent Claim 1	Hilla
		two-way communication coupling to a variety of external devices that operate with their own processors, such as printers, scanners, external disks, and terminal 612. Firmware or software running in the computer system 600 provides a terminal interface or character-based command interface so that external commands can be given to the computer system. For example, communication interface 670 may be a parallel port or a serial port such as an RS-232 or RS-422 interface, or a universal serial bus (USB) port on a personal computer. In some embodiments, communications interface 670 is an integrated services digital network (ISDN) card or a digital subscriber line (DSL) card or a telephone modem that provides an information communication connection to a corresponding type of telephone line. In some embodiments, a communication interface 670 is a cable modem that converts signals on bus 610 into signals for a communica-tion connection over a coaxial cable or into optical signals for a communication connection over a fiber optic cable. As another example, communications interface 670 may be a local area network (LAN) card to provide a data communi-cation connection to a compatible LAN, such as Ethernet. Wireless links may also be implemented. For wireless links, the communications interface 670 sends and receives elec-trical, acoustic or electromagnetic signals, including infrared and optical signals, which carry information streams, such as digital data. Such signals are examples of carrier waves”) Hilla at [0128] (“n the illustrated computer used as a router, the computer system 600 includes switching system 630 as special purpose hardware for switching information for flow over a network. Switching system 630 typically includes multiple communications interfaces, such as communica-tions interface 670, for coupling to multiple other devices. In general, each coupling is with a network link 632 that is connected to another device in or attached to a network, such as local network 680 in the illustrated embodiment, to which a variety of external devices with their own processors are connected. In some embodiments an input interface or an output interface or both are linked to each of one or more external network elements. Although three network links 632a, 632b, 632c are included in network links 632 in the illustrated embodiment, in other embodiments, more or fewer links are connected to switching system 630. Network links 632 typically provides information communication through one or more networks to other devices that use or process the information. For example, network link 632b may provide a connection through local network 680 to a host computer 682 or to equipment 684 operated by an Internet Service Provider (ISP). ISP

No.	’740 Patent Claim 1	Hilla
		equipment 684 in tum provides data communication services through the public, world-wide packet-switching communication network of networks now commonly referred to as the Internet 690. A computer called a server 692 connected to the Internet provides a service in response to information received over the Internet. For example, server 692 provides routing information for use with switching system 630.”) Hilla at [0129] (“The switching system 630 includes logic and cir-cuitry configured to perform switching functions associated with passing information among elements of network 680, including passing information received along one network link, e.g. 632a, as output on the same or different network link, e.g., 632c. The switching system 630 switches infor-mation traffic arriving on an input interface to an output interface according to pre-determined protocols and conven-tions that are well known. In some embodiments, switching system 630 includes its own processor and memory to perform some of the switching functions in software. In some embodiments, switching system 630 relies on proces-sor 602, memory 604, ROM 606, storage 608, or some combination, to perform one or more switching functions in software. For example, switching system 630, in coopera-tion with processor 604 implementing a particular protocol, can determine a destination of a packet of data arriving on input interface on link 632a and send it to the correct destination using output interface on link 632c. The desti-nations may include host 682, server 692, other terminal devices connected to local network 680 or Internet 690, or other routing and switching devices in local network 680 or Internet 690.”) Hilla at [0130] (“The invention is related to the use of computer system 600 for implementing the techniques described herein. According to one embodiment of the invention, those techniques are performed by computer system 600 in response to processor 602 executing one or more sequences of one or more instructions contained in memory 604. Such instructions, also called software and program code, may be read into memory 604 from another computer-readable medium such as storage device 608. Execution of the sequences of instructions contained in memory 604 causes processor 602 to perform the method steps described herein. In alternative embodiments, hardware, such as application specific integrated circuit 620 and circuits in switching system 630, may be used in place of or in combination with software to

No.	’740 Patent Claim 1	Hilla
		implement the invention. Thus, embodiments of the invention are not limited to any specific combination of hardware and software.”) Hilla at [0131] (“The signals transmitted over network link 632 and other networks through communications interfaces such as interface 670, which carry information to and from computer system 600, are exemplary forms of carrier waves. Com-puter system 600 can send and receive information, includ-ing program code, through the networks 680, 690 among others, through network links 632 and communications interfaces such as interface 670. In an example using the Internet 690, a server 692 transmits program code for a particular application, requested by a message sent from computer 600, through Internet 690, ISP equipment 684, local network 680 and network link 632b through commu-nications interface in switching system 630. The received code may be executed by processor 602 or switching system 630 as it is received, or may be stored in storage device 608 or other non-volatile storage for later execution, or both. In this manner, computer system 600 may obtain application program code in the form of a carrier wave.) Hilla at Figure 2 (annotation added)

No.	’740 Patent Claim 1	Hilla
		FIG. 2  200 LINK BUNDLE SWITCHING SYSTEM 130 LINK BUNDLE 238h 238a 238b 238c 238d 238g 238f 238e 236 PORTS BANK 230 EGRESS LINE CARD 238 PORTS 242 PORT SELECTION PROCESS 244 AGING PROCESS 240 DYNAMIC LOAD BALANCING BLOCK 248 QUEUE SELECTION PROCESS 243 STATUS UPDATE PROCESS 235 DATA FLOW QUEUES 245 BLOCKED QUEUE MAP 234 TRAFFIC MANAGER 232 SWITCHING ASIC 235 BLOCKED FLOW MAP 220 SWITCHING FABRIC 202 SWITCHING SYSTEM COMPONENT 210a INGRESS LINE CARD 210b INGRESS LINE CARD 210c INGRESS LINE CARD Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Hilla in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 1[c] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of three such references.

No.	’740 Patent Claim 1	Hilla
		For example, Ghosh discloses connecting the line cards to the active supervisor via the backplane using parallel interface circuitry. Ghosh at [0059] (“Line cards 803, 805, and 807 can communicate with an active supervisor 811 through interface circuitry 883, 885, and 887 and the backplane 815. According to various embodiments, each line card includes a plurality of ports that can act as either input ports or output ports for communication with external fibre channel network entities 851 and 853. The backplane 815 can provide a communi-cations channel for all traffic between line cards and super- visors. Individual line cards 803 and 807 can also be coupled to external fibre channel network entities 851 and 853 through fibre channel ports 843 and 847.”) As another example, Bruckman discloses connecting the line cards to the network using traces comprising a backplane. Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”) For example, Basso discloses coupling the blades to the communication network via link connections to the switch fabric.

No.	’740 Patent Claim 1	Hilla
		Basso at [0009] (“A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more ports. These ports may be connected to another one or more network devices. The switch fabric may be configured to direct incoming packets of data to particular blades where one or more of the network proces-sors in the recipient blade may be configured to process the received packets.”)
1[d]	at least one of said second physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction;	Hilla discloses at least one of said second physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction. For example, Hilla discloses sending inbound and outbound packets over two-way communications connections connecting the switching system to the line cards to communicate in both an upstream direction and a downstream direction. Hilla at [0035] (“FIG. 1 is a block diagram that illustrates a network with a link bundle, according to an embodiment. A computer network is a geographically distributed collection of inter-connected sub-networks (e.g., sub-networks 102a, 102b, collectively referenced hereinafter as sub-networks 102) for transporting data between network nodes (e.g., network nodes 110a, 110b, 110c, 110d, ll0e, 110/, 110g, collectively referenced hereinafter as network nodes 110). A local area network (LAN) is an example of such a sub-network 102. The network's topology is defined by an arrangement of end nodes that communicate with one another, typically through one or more intermediate network nodes such as a router or switch, which facilitates routing data between end nodes. As used herein, an end node is a node that is configured to originate or terminate communications over the network. In contrast, an intermediate network node facilitates the pas-sage of data between end nodes. In FIG. 1, the network nodes 110 include both end nodes (e.g., nodes 110a, 110g) and intermediate nodes (e.g., network node 110c). Each sub-network 102 includes zero or more intermediate net-work nodes.”) Hilla at [0036] (“FIG. 1 illustrates a link bundle 130 that includes five communication links between intermediate network node 110c and network node 110b. Network node 110b may be an intermediate network node or an end node that is connected both to intermediate network

No.	’740 Patent Claim 1	Hilla
		node 11 Oc and subnetwork 102a. For purposes of illustration, intermediate network node 110c is connected by a bundle 130 of five communication links to network node 110b and thereby to subnetwork 102a and end node 11 Oa. Similarly, intermediate network node 110c is connected by three unbundled com-munication links to network nodes 110d, ll0e, 110/, respec-tively, and thence to sub-network 102b and end node 110g. In other embodiments an intermediate network node 11 Oc may be connected to more or fewer network nodes with more or fewer links in each of one or more link bundles as part of a network with the same or more end nodes.”) Hilla at [0038] (“FIG. 2 is a block diagram that illustrates a link-bundle switching system 200 in a router, which can be used as intermediate network node 11 Oc, according to an embodi-ment. The switching system 200 includes a switching system component 202 connected to communication links, includ-ing link bundle 130. The switching system component 202 includes multiple ingress line cards, including cards 210a, 210b, 210c, for receiving inbound data packets, switching fabric component 220, and multiple egress line cards for sending outbound data packets, including egress line card 230 for link bundle 130.”) Hilla at [0039] (“An ingress line card (e.g., 210a) receives a data packet, inspects the appropriate header fields to make a forwarding decision, and forwards the packet toward the egress line card via the switching fabric component 220. The switch fabric component 220 forwards the packet to the appropriate egress line card 230. Multiple ingress line cards can forward traffic toward a single egress line card 230. In this example, the link bundle 130 is capable of transmitting more packets simultaneously because of the increased band-width achieved by bundling. Some ingress line cards (e.g., 210c) are connected to a link bundle with multiple commu-nication links. All communication links in the link bundle are considered a single ingress logical link. In the illustrated embodiment, the five communication links connected to ingress line card 210c are connected to network node 110b. The ingress line card 210c handles incoming traffic on the link bundle 130, while the egress line card 230 handles the outbound traffic on the link bundle 130.”) Hilla at [0040] (“The ingress line cards 210 determine an outbound physical or logical link to use to forward the data packet, for example using data in a routing table stored on the router,

No.	’740 Patent Claim 1	Hilla
		and sends information to the egress line card associated with the outbound physical or logical link. All communication links in the link bundle 130 are considered a single logical link. Information about data packets to be output on the link bundle 130 is sent to the egress line card 230.”) Hilla at [0044] (“The ports bank 236 is a bank of ports which includes multiple ports that are treated by a router as a single logical port for purposes of communicating through link bundle 130 as a single logical link. Each communication link in the link bundle 130 is connected to a different port of the ports bank 236. In the illustrated embodiment, ports bank 236 includes eight ports 238a, 238b, 238c, 238d, 238e, 238/, 238g, 238h (collectively referenced hereinafter as ports 238). For purposes of illustration, it is assumed that these eight ports are identified by a three bit code that represents the eight values from O through 7 to correspond to ports 238a through 238h, respectively. Each three bit code is called a port number (port #), herein. Only five of the ports (e.g., 238b, 238c, 238d, 238e, 238/, identified by code values 1 through 5, respectively) are connected to communication links of the link bundle 130 in the illustrated embodiment. These are called active ports. The remaining three ports (e.g., 238a, 238g, 238h, identified by code values 0, 6, 7, respectively) are inactive ports. In some embodiments, a port becomes inactive by virtue of a communication link that is attached to the port going down. When a port goes down, it is no longer eligible to send traffic and the link is removed from the route table. In an example embodiment, the hard-ware described in more detail below would detect the link's inability to forward traffic and dynamically shift traffic over to an active link. Such a hardware shift can occur much faster than waiting for a routing table update.”) Hilla at [0121] (“Computer system 600 includes a communication mechanism such as a bus 610 for passing information between other internal and external components of the computer system 600. Information is represented as physical signals of a measurable phenomenon, typically electric voltages, but including, in other embodiments, such phe-nomena as magnetic, electromagnetic, pressure, chemical, molecular atomic and quantum interactions. For example, north and south magnetic fields, or a zero and non zero electric voltage, represent two states (0, 1) of a binary digit (bit). A sequence of binary digits constitutes digital data that is used to represent a number or code for a character. A bus 610 includes many parallel conductors of information so that information is transferred quickly among devices coupled to the bus 610.

No.	’740 Patent Claim 1	Hilla
		One or more processors 602 for processing information are coupled with the bus 610. A processor 602 performs a set of operations on information. The set of operations include bringing information in from the bus 610 and placing information on the bus 610. The set of opera-tions also typically include comparing two or more units of information, shifting positions of units of information, and combining two or more units of information, such as by addition or multiplication. A sequence of operations to be executed by the processor 602 constitute computer instruc-tions.”) Hilla at [0122] (“Computer system 600 also includes a memory 604 coupled to bus 610. The memory 604, such as a random access memory (RAM) or other dynamic storage device, stores information including computer instructions. Dynamic memory allows information stored therein to be changed by the computer system 600. RAM allows a unit of information stored at a location called a memory address to be stored and retrieved independently of information at neighboring addresses. The memory 604 is also used by the processor 602 to store temporary values during execution of computer instructions. The computer system 600 also includes a read only memory (ROM) 606 or other static storage device coupled to the bus 610 for storing static information, including instructions, that is not changed by the computer system 600. Also coupled to bus 610 is a non-volatile (persistent) storage device 608, such as a mag-netic disk or optical disk, for storing information, including instructions, that persists even when the computer system 600 is turned off or otherwise loses power.”) Hilla at [0126] (“Computer system 600 also includes one or more instances of a communications interface 670 coupled to bus 610. Communication interface 670 provides a two-way communication coupling to a variety of external devices that operate with their own processors, such as printers, scanners, external disks, and terminal 612. Firmware or software running in the computer system 600 provides a terminal interface or character-based command interface so that external commands can be given to the computer system. For example, communication interface 670 may be a parallel port or a serial port such as an RS-232 or RS-422 interface, or a universal serial bus (USB) port on a personal computer. In some embodiments, communications interface 670 is an integrated services digital network (ISDN) card or a digital subscriber line (DSL) card or a telephone modem that provides an information communication connection to a corresponding type of telephone line. In some embodiments,

No.	’740 Patent Claim 1	Hilla
		a communication interface 670 is a cable modem that converts signals on bus 610 into signals for a communica-tion connection over a coaxial cable or into optical signals for a communication connection over a fiber optic cable. As another example, communications interface 670 may be a local area network (LAN) card to provide a data communi-cation connection to a compatible LAN, such as Ethernet. Wireless links may also be implemented. For wireless links, the communications interface 670 sends and receives elec-trical, acoustic or electromagnetic signals, including infrared and optical signals, which carry information streams, such as digital data. Such signals are examples of carrier waves”)
1[e]	receiving a data frame having frame attributes sent between the communication network and the network node:	Hilla discloses receiving a data frame having frame attributes sent between the communication network and the network node. For example, Hilla discloses data packets sent between a network and network node, with packet information including header information and payload information. Hilla at [0006] (“Communications between nodes are typically effected by exchanging discrete packets of data. Each packet typically comprises 1] header information associated with a particular protocol, and 2] payload information that follows the header information and contains information to be processed, often independently of that particular protocol. In some protocols, the packet includes 3] trailer information following the payload and indicating the end of the payload information. The header includes information such as the source of the packet, its destination, the length of the payload, and other properties used by the protocol. Often, the data in the payload for the particular protocol includes a header and payload for a different protocol associated with a different, higher layer of the OSI Reference Model. The higher layer protocol is said to be encapsulated in the lower layer protocol. The headers included in a packet traversing multiple heterogeneous networks, such as the Internet, typi-cally include a physical (layer 1) header, a data-link (layer 2) header, an internetwork (layer 3) header and a transport (layer 4) header, as defined by the Open Systems Intercon-nection (OSI) Reference Model.”)

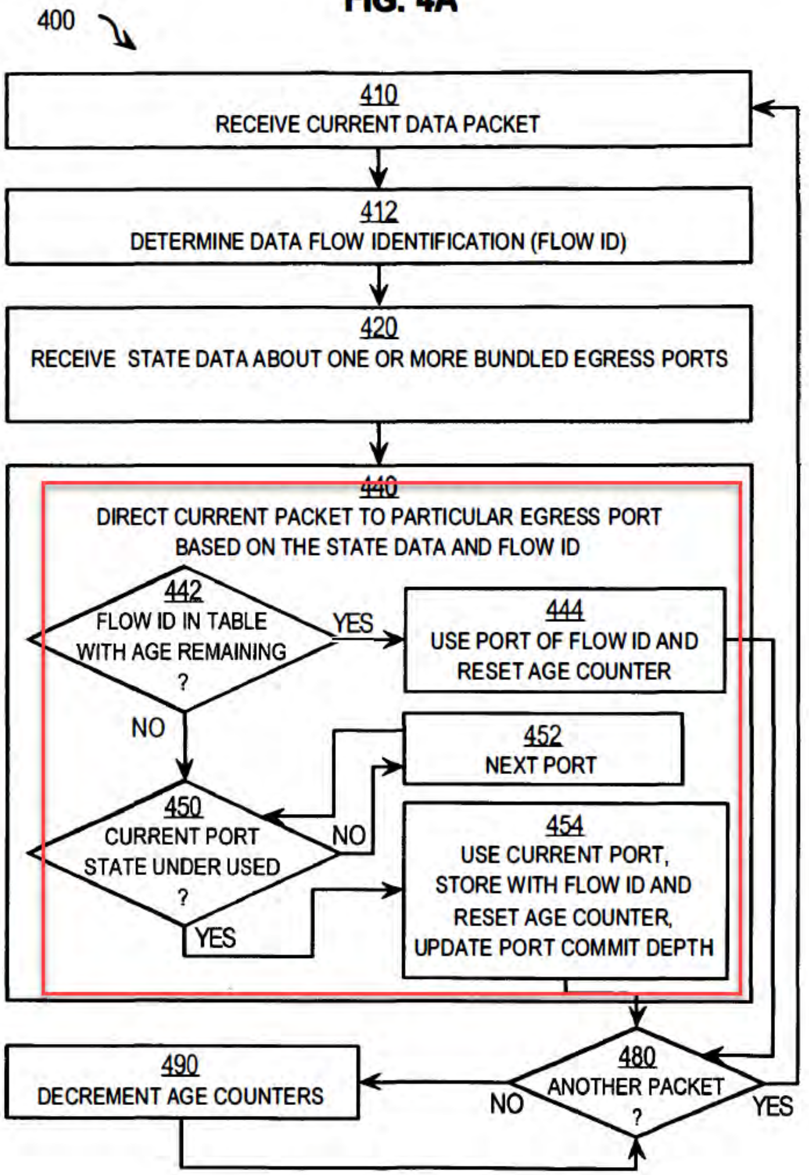
No.	’740 Patent Claim 1	Hilla
		Hilla at [0012] (“A load-balancing process is used on the sending network node of the pair connected by a bundle of commu-nication links for the purpose of determining which com-munication link to use for sending one or more data packets to the receiving network node of the pair. Current balancing algorithms use a fixed mapping to associate data packets with a specific port in a set of ports connected to the communication links in the bundle. Typically, information in a header portion of a data packet is used to derive a value that is associated with one port of the set. The algorithm is designed to generate a value in a range of values that are associated with the full set of ports. Thus data packets directed to the receiving node are distributed over all com-munication links in the bundle by the load balancing pro-cess. Many load-balancing processes are designed so that all data packets in the same data flow are sent through the same port.”) Hilla at [0039] (“An ingress line card (e.g., 210a) receives a data packet, inspects the appropriate header fields to make a forwarding decision, and forwards the packet toward the egress line card via the switching fabric component 220. The switch fabric component 220 forwards the packet to the appropriate egress line card 230. Multiple ingress line cards can forward traffic toward a single egress line card 230. In this example, the link bundle 130 is capable of transmitting more packets simultaneously because of the increased band-width achieved by bundling. Some ingress line cards (e.g., 210c) are connected to a link bundle with multiple commu-nication links. All communication links in the link bundle are considered a single ingress logical link. In the illustrated embodiment, the five communication links connected to ingress line card 210c are connected to network node 110b. The ingress line card 210c handles incoming traffic on the link bundle 130, while the egress line card 230 handles the outbound traffic on the link bundle 130.”) Hilla at [0040] (“The ingress line cards 210 determine an outbound physical or logical link to use to forward the data packet, for example using data in a routing table stored on the router, and sends information to the egress line card associated with the outbound physical or logical link. All communication links in the link bundle 130 are considered a single logical link. Information about data packets to be output on the link bundle 130 is sent to the egress line card 230.”)

No.	’740 Patent Claim 1	Hilla
		Hilla at [0067] (“In step 412 a current flow ID is determined for the current data packet. Any method known in the art at the time the method 400 is implemented may be used. In some embodiments, step 412 includes determining and combining the bit values in one or more fields in the header portions of one or more protocols. In some embodiments, step 412 involves receiving the current flow ID from another com-ponent, e.g., from switching fabric block 220, or switching ASIC 232 or traffic manager block 234. In some embodi-ments, the flow ID is the same as the particular queue 332 where the data packet is stored by the traffic manager 234. In some embodiments in which packet sequence within a data flow is not important and commit depth is not used to select ports, step 412 is omitted.”)
1[f]	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	Hilla discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. For example, Hilla discloses using packet information such as header information to select, a port and a communication connector between the switching system and line cards. A person of ordinary skill in the art would understand in order to forward a data packet, a port and other communication links such as communication connectors between the switching system and line cards must be selected. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Hilla is found to not meet this limitation, selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group would have been obvious to a person having ordinary skill in the art, as explained below. Hilla at [0005] (“Information is exchanged between network nodes according to one or more of many well known, new or still developing protocols. In this context, a protocol consists of a set of rules defining how the nodes interact with each other based on information sent over the communication links. The protocols are effective at different layers of operation within each node, from generating and receiving physical signals of various types, to selecting a link for transferring those signals, to the format of information indicated by those signals, to identifying which software application executing on a computer system sends or receives the information. The conceptually different layers of protocols for exchanging information over a

No.	’740 Patent Claim 1	Hilla
		network are described in the Open Systems Interconnection (OSI) Reference Model. The OSI Reference Model is generally described in more detail in Section 1.1 of the reference book entitled Interconnections Second Edition, by Radia Perlman, published September 1999, which is hereby incorporated by reference as though fully set forth herein.”) Hilla at [0014] (“Typically, the fixed-mapping takes several bits from one or more fields in layer 2 or layer 3 headers, or both, and inputs those bits to a hash function that produces an output with a certain number of bits. The output is then used directly or indirectly to select a port among the set connected to the bundle of communication links. By judicious choice of the fields, data packets from the same flow may be mapped to the same port.”) Hilla at [0021] (“In a more recent approach, load balancing selects a port based on the degree to which buffers that hold data being sent out on each port are filled. A data packet from a new flow is directed to a port that has a buffer that is not full. While this approach tends to distribute data packets from new flows to ports more able to handle the new flows, it experiences some problems. For example, if a long sequence of data packets from the same flow are directed to the ports, they are all directed to the same port to preserve sequence order. The port receiving the data packets from this flow can become full. The next data packet from that flow cannot be placed until the port buffer fill level drops. Data transmission on the entire bundle halts until that next data packet can be placed in its target port buffer. The bundle thus does not perform at advertised capacity.”) Hilla at [0045] (“According to various embodiments, dynamic load balancing (DLB) block 240 selects the next packet from the data flow queues and directs the next data packet to one of the ports 238 in ports bank 236 based on the state of one or more ports 238 in the ports bank 236, or time gaps between packets of the same flow, or both. In the illustrated embodi-ment, DLB block 240 includes logic for a port selection process 242 (also called an arbiter process), logic for a port status update process 243, logic for an aging process 244, and logic for a queue selection process 248, all of which are described in greater detail in a later section.”) Hilla at [0056] (“In the illustrated embodiment, each record 342 also includes a commit depth field (e.g., 347a, 347b, collectively referenced hereinafter as commit depth field 347). A

No.	’740 Patent Claim 1	Hilla
		par-ticular commit depth value indicates an amount of data in the one or more data flow queues 332 that have been directed to the corresponding port by the port selection process, as is described in more detail below. Any units or granularity in data amount may be used. In some embodiments, the amount of data is expressed as a number of data packets; in some embodiments the amount of data is expressed as a number of octets. For example, if there are 12 data packets in one data flow queue directed to the port #1 and 25 data packets in another data flow queue also directed to the same port, and data packets are the units for expressing amount, then the value in commit depth field 347a for port #1 is 37. The commit depth is a measure of the amount of data expected to be transmitted through the corresponding port; and is independent of the fill level of the port buffer. The commit field is used in some embodiments to determine whether a particular port is under-utilized compared to another port, as described in more detail below. In some embodiments, the commit depth field 247 is omitted.”) Hilla at [0089] (“In step 450 it is determined if the state data associated with a current port indicates that the current port is underused. The first current port may be selected in any manner. For example, in some embodiments, the first current port is selected to be the first active port (e.g., port #1). In some embodiments, the first current port is always a least used port, e.g., one of the starving ports, or the port with the emptiest port buffer.”) Hilla at [0090] (“In step 450, any method may be used to determine that the current port is underused. For example, in some embodiments the current port is underused if the flow control status code for the port is 1, indicating a starving port buffer. In some embodiments, the current port is underused if the flow control status code for the port is 1 or 2, indicating a starving or hungry port buffer. In some embodiments, the current port is NOT underused if the flow control status code for the port is 3, indicating a satisfied port buffer. In some embodiments, the current port is determined to be not underused unless it is the least used of all the active ports. For example, the port selection process 242 uses the port buffer fill level to determine whether or not it should send data to a particular port. The port selection process 242 inspects the fill level fields 346 of all the flow control status records 342 and uses this information to select which port to assign to the unassigned flow. In an illustrated embodiment, the port with the least amount of fill is determined to be the best choice. If the current port is underused,

No.	’740 Patent Claim 1	Hilla
		then control passes to step 454 to use that port and update the mapping table. If the current port is not underused, then control passes to step 452.”) Hilla at [0108] (“As described above, the port selection process directs a data packet to a particular port. The port selection process does not determine the data packet that it receives; but reacts to the data packet sent. The selection of the next data packet is a function of the traffic manager. In some embodiments, the DLB block includes a queue selection process 248 in the traffic manager 234 to determine which data flow queue should be the source of the next data packet directed to a port.”) Hilla at Figure 4A (annotations added)

		FIG. 4A  400 410 RECEIVE CURRENT DATA PACKET 412 DETERMINE DATA FLOW IDENTIFICATION (FLOW ID) 420 RECEIVE STATE DATA ABOUT ONE OR MORE BUNDLED EGRESS PORTS 440 DIRECT CURRENT PACKET TO PARTICULAR EGRESS PORT BASED ON THE STATE DATA AND FLOW ID 442 FLOW ID IN TABLE WITH AGE REMAINING ? YES 444 USE PORT OF FLOW ID AND RESET AGE COUNTER NO 452 NEXT PORT 450 CURRENT PORT STATE UNDER USED ? NO YES 454 USE CURRENT PORT, STORE WITH FLOW ID AND RESET AGE COUNTER, UPDATE PORT COMMIT DEPTH 490 DECREMENT AGE COUNTERS NO 480 ANOTHER PACKET ? YES

No.	’740 Patent Claim 1	Hilla
		Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Hilla in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 1[f] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Basso discloses using a hash function and index table to select for blade/port combinations over which to send the packet over a user port and a switch fabric link. Basso further discloses that this selection is performed based upon packet information. Basso at [0010] (“Upon a network processor receiving a packet of data, the network processor may index into a table, commonly referred to as a forwarding table, to determine the table associated with a particular logical interface as well as the next destination address. The forwarding table may comprise a plurality of entries where each entry may comprise information indicating a particular table associated with a particular logical interface as well as the next destination address. Each logical interface may be associated with a table storing a plurality of entries containing blade/ port combinations as discussed further below. In one embodiment, an entry may be indexed in the forwarding table using a destination address in the received packet header.”) Basso at [0011] (“A hash function may then be performed on the received packet to generate a hash value. In one embodiment, a hash function may be performed on the source and destination address in the packet header to generate a hash value.”) Basso at [0012] (“The hash value generated may be used to index into the table associated with a particular logical interface. Upon indexing into the table associated with the logical interface, an appropriate blade/port combination may be identified to transmit the received packet of data. In one embodiment, a blade/port combination may be selected in the indexed entry of the table associated with the logical interface by using a portion of the bits of the hashed value. The received packet may then be transmitted through the identified blade/port combination to

No.	’740 Patent Claim 1	Hilla
		the next destination (next destination previously identified by the next destination address in the forwarding table).”) Basso at [0035] (“By logically grouping a plurality of ports 404 coupled to a particular network device into a logical interface 405, network processor 403 may be configured to transmit processed packets to that particular network device via any blade 402/port 404 combination grouped in that logical interface 405. For example, referring to FIG. 4, ports 404A-404I are physically connected to router 104B. If ports 404A-404I were logically grouped into logical interface 405, then a particular network processor 403, e.g., network processor 403A, may be configured to transmit processed packets that are determined to be transmitted to router 104B through any of ports 404A-404I in blades 402A-C, respectively. Network processor 403, e.g., network processor 403A, may be configured to transmit the processed packets to router 104B through ports 404, e.g., ports 404D-I, not in its blade 402, e.g., blade 402A, by forwarding the processed packets to switch fabric 401 which may then direct the processed packets to another appropriate physical blade 402/port 404 combination. Network processor 403, e.g., network processor 403A, may further be configured to transmit the processed packets to router 104B through any ports 404, e.g., ports 404A-C, in its blade 402, e.g., blade402A, instead of just one physical port 404 in its blade 402, e.g., blade 402A. A more detailed description of routing packets implementing logical interface(s) 405 is provided below in FIG. 5.”) Basso at [0040] (“In step 502, network processor 403, e.g., network processor 403A, may receive a packet of data from switch fabric 401. Upon receiving the packet of data, network processor 403, in step 503, may index into a table, commonly referred to as a forwarding table, to determine the table associated with a particular logical interface 405 as well as the next destination address, i.e., the next hop address. The forwarding table may comprise a plurality of entries where each entry may comprise information indicating a particular table associated with a particular logical interface 405 as well as the next destination address. Each logical interface 405 may be associated with a table storing a plurality of entries containing blade 402/port 404 combinations as discussed further below. In one embodiment, an entry may be indexed in the forwarding table using a destination address in the received packet header. It is noted that an entry may be indexed in the forwarding table usingother means and that such means would be recognized by an artisan of ordinary skill in the art. It is further noted that

No.	’740 Patent Claim 1	Hilla
		embodiments implementing such means would fall within the scope of the present invention.”) Basso at [0041] (“In step 504, a hash function may be performed on the received packet to generate a hash value. In one embodiment, a hash function may be performed on the source and destination address in the packet header to generate a hash value. It is noted that in other embodiments a hash function may be performed on different fields, e.g., port, type of service, in the received packet to generate a hash value.”) Basso at [0042] (“In step 505, the hash value generated in step 504 may be used to index into the table associated with a particular logical interface 405 determined in step 503. Upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be identified in step 506 to transmit the received packet of data as explained below.”) Basso at [0043] (“As stated above, the table associated with a particular logical interface 405 may comprise a plurality of entries where each entry may comprise a threshold value associated with a particular blade 402/port 404 combination. The threshold value may represent a percentage of the total number of packets received by router 104A that may be transmitted through the blade 402/port 404 combination associated with that threshold value. In one embodiment, the threshold value may be updated periodically by a user, e.g., system administrator, in control of router 104, e.g., router 104A. For example, the threshold value, e.g., twenty percent of the number of packets received by router 104A, associated with a particular blade 402/port 404 combination may be updated by lowering the threshold value by one percent during each update. An example of an entry of the table associated with a particular logical interface 405 is shown in Table 1 below:

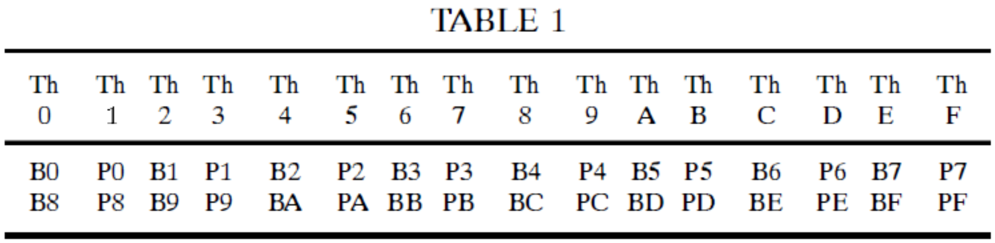
<table>
<tr><th>No.</th><th>’740 Patent Claim 1</th><th>Hilla</th></tr>
<tr><td></td><td></td><td>


TABLE 1

<table>
<tr><th>Th
0</th><th>Th
1</th><th>Th
2</th><th>Th
3</th><th>Th
4</th><th>Th
5</th><th>Th
6</th><th>Th
7</th><th>Th
8</th><th>Th
9</th><th>Th
A</th><th>Th
B</th><th>Th
C</th><th>Th
D</th><th>Th
E</th><th>Th
F</th></tr>
<tr><td>B0
B8</td><td>P0
P8</td><td>B1
B9</td><td>P1
P9</td><td>B2
BA</td><td>P2
PA</td><td>B3
BB</td><td>P3
PB</td><td>B4
BC</td><td>P4
PC</td><td>B5
BD</td><td>P5
PD</td><td>B6
BE</td><td>P6
PE</td><td>B7
BF</td><td>P7
PF</td></tr>
</table>

Basso at [0044] (“Table 1 above illustrates an exemplary entry in the table associated with a particular logical interface 405. Each entry may comprise a plurality of threshold values (16 threshold values in exemplary Table 1) where each threshold value is associated with a particular blade 402/port 404 combination. For example, threshold value (Th0) is associated with blade B0/port P0 combination where blade BO may refer to a particular blade 402, e.g., blade 402B, and port PO may refer to a particular port 404, e.g., port 404E. Threshold value (Thl) is associated with blade Bl/port Pl combination and so forth. As stated above, each threshold value may represent a percentage of the total number of packets received by router 104A that may be transmitted through the blade 402/port 404 combination associated with that threshold value. For example, threshold value (Th0) may represent a percentage of the total number of packets received by router 104A that may be transmitted through port PO in blade BO. If port PO refers to port 404D and blade BO refers to blade 402B, then if Th0 has a value of twenty percent, a maximum of twenty percent of the total packets received by router 104A may be transmitted through port 404D in blade 402B.”)

Basso at [0045] (“As stated above, upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be identified in step 506 to transmit the received packet of data. In one embodiment, the hash value generated in step 504 may be used to select a particular threshold value and hence a blade 402/port 404 combination associated with the selected threshold value. In one embodiment, a portion of the bits of the hash value, e.g., most significant bits, may be used to select a particular threshold value in the entry indexed in step 505. For example, referring to Table 1, since there are 16 different threshold values in each entry of the table associated with
</td></tr>
</table>

No.	’740 Patent Claim 1	Hilla
		logical interface 405, only four bits of the hash value generated in step 504 may be used to select a threshold value. Upon selecting a threshold value, the blade 402/port 404 combination associated with the selected threshold value may be used to transmit the received packet.”) As another example, Wiher discloses using cell header information to at each node to select and route the ATM data cell over a data link and selected backplane. Wiher at 3:43-65 (“In general, in another aspect, the invention features an apparatus for communicating data cells between a data link and a backplane. The apparatus includes transceiver circuitry to transmit and receive data cells over a data link and a plurality of backplane interfaces each including at least one cell signal terminal. Each of the backplane interface is coupled to a backplane interconnection circuit. Each backplane interconnection circuit transmits and receives cells over the cell signal terminals of its associated backplane interface. The apparatus also includes de-multiplexing circuitry coupling the transceiver circuitry to each of the backplane interconnection circuits. The de-multiplexing circuitry receives a data cell from the transceiver circuitry, select a backplane interconnection circuit associated with the data cell, and provide the data cell to the selected backplane interconnection circuit for transmission over the cell signal terminals of the associated backplane interface. The apparatus also includes multiplexing circuitry coupling the plurality of backplane interconnection circuits to the transceiver circuitry. The multiplexing circuitry receives data cells from each of the backplane interconnection circuits and provide the received data cells to the transceiver circuitry.”) Wiher at 3:66-4:22 (“Implementations of the invention may include one or more of the following features. The backplane interconnection circuits may independently receive and transmit data cells over the plurality of backplane interfaces. The de-multiplexing circuitry may select a backplane interface based on data in the header field of the data cell. The apparatus may include header translation circuitry to alter header data in cells sent between the plurality of backplane interfaces and the transceiver circuitry. Each of the plurality of backplane interfaces may include separate terminals to receive cells and separate terminals to transmit cells. The terminals to transmit cells may include a first and second control terminal and at

No.	’740 Patent Claim 1	Hilla
		least one outgoing cell data terminal. A backplane interface's backplane interconnection circuitry may accepts a signal on the first control terminal as indicating that a cell may be sent over the interface, asserts a 15 signal on the second control terminal to indicate that a cell is being transmitted, and transmits data bits of the cell on the outgoing cell data terminal. Each backplane interface may include a single outgoing cell data terminal and each bit of the cell may be serially transmitted over the single outgoing cell data terminal. Each backplane interface may include multiple outgoing cell data terminals and bits of the cell may be sent in parallel over the eight outgoing cell data terminals.”)
1[g]	sending the data frame over the selected first and second physical links,	Hilla discloses sending the data frame over the selected first and second physical links. For example, Hilla discloses sending received data packets over ports and communication connectors between the switching system and line cards. Hilla at [0039] (“An ingress line card (e.g., 210a) receives a data packet, inspects the appropriate header fields to make a forwarding decision, and forwards the packet toward the egress line card via the switching fabric component 220. The switch fabric component 220 forwards the packet to the appropriate egress line card 230. Multiple ingress line cards can forward traffic toward a single egress line card 230. In this example, the link bundle 130 is capable of transmitting more packets simultaneously because of the increased band-width achieved by bundling. Some ingress line cards (e.g., 210c) are connected to a link bundle with multiple commu-nication links. All communication links in the link bundle are considered a single ingress logical link. In the illustrated embodiment, the five communication links connected to ingress line card 210c are connected to network node 110b. The ingress line card 210c handles incoming traffic on the link bundle 130, while the egress line card 230 handles the outbound traffic on the link bundle 130.”) Hilla at [0040] (“The ingress line cards 210 determine an outbound physical or logical link to use to forward the data packet, for example using data in a routing table stored on the router, and sends information to the egress line card associated with the outbound physical or logical link. All communication links in the link bundle 130 are considered a single logical link.

No.	’740 Patent Claim 1	Hilla
		Information about data packets to be output on the link bundle 130 is sent to the egress line card 230.”) Hilla at [0045] (“According to various embodiments, dynamic load balancing (DLB) block 240 selects the next packet from the data flow queues and directs the next data packet to one of the ports 238 in ports bank 236 based on the state of one or more ports 238 in the ports bank 236, or time gaps between packets of the same flow, or both. In the illustrated embodi-ment, DLB block 240 includes logic for a port selection process 242 (also called an arbiter process), logic for a port status update process 243, logic for an aging process 244, and logic for a queue selection process 248, all of which are described in greater detail in a later section.”) Hilla at [0057] (“In the illustrated embodiment, each record 342 also includes a paused field (e.g., 348a, 348b, collectively ref-erenced hereinafter as paused field 348). A particular paused value indicates whether a pause message was received on the corresponding physical link. In some protocols, such as Ethernet, a network node receiving data over a link may indicate to the sending node that the receiving node can not process additional data packets at the present time. In response to receiving the pause message, the sending net-work node completes sending the current data packet, but does not send another data packet until certain conditions are satisfied. In some embodiments, the condition is the passage of a prescribed amount of time. In some embodiments, the pause message specifies the amount of time. In some embodiments, the condition is receipt of a subsequent mes-sage to resume sending data packets. In some embodiments, the paused field 248 is omitted, and the pause option of the protocol is not enabled.”) Hilla at [0106] (“Using the steps of method 400 and 490* in some embodiments, data packets are distributed to ports that are not satisfied before being directed to ports that are satisfied and thus heavily utilized. In other illustrated embodiments, data packets are distributed to ports that have the smallest commit depth, such as port #4 before being assigned to other ports. Data packets from a single data flow are directed to use the same port, no matter how heavily utilized, until a gap in data packets occurs that is long enough to send the later packets on less busy ports without having the packets arrive out of order at the other end of the link bundle.”)

No.	’740 Patent Claim 1	Hilla
1[h]	said sending comprising communicating along at least one of said bi-directional links.	Hilla discloses said sending comprising communicating along at least one of said bi-directional links. *See supra at* 1[b], 1[d], 1[g].

No.	’740 Patent Claim 2	Hilla
2[a]	The method according to claim 1, wherein the network node comprises a user node, and	Hilla discloses the method according to claim 1, wherein the network node comprises a user node. For example, Hilla discloses network nodes as network devices or computer systems used by users. Hilla at [0004] (“Networks of general purpose computer systems connected by external communication links are well known and widely used in commerce. The networks often include one or more network devices that facilitate the passage of information between the computer systems. A network node is a network device or computer system connected by the communication links.”) Hilla at [0005] (“Information is exchanged between network nodes according to one or more of many well known, new or still developing protocols. In this context, a protocol consists of a set of rules defining how the nodes interact with each other based on information sent over the communication links. The protocols are effective at different layers of operation within each node, from generating and receiving physical signals of various types, to selecting a link for transferring those signals, to the format of information indicated by those signals, to identifying which software application executing on a computer system sends or receives the information. The conceptually different layers of protocols for exchanging information over a network are described in the Open Systems Interconnection (OSI) Reference Model. The OSI Reference Model is generally described in more detail in Section 1.1 of the reference book

No.	’740 Patent Claim 2	Hilla
		entitled Interconnections Second Edition, by Radia Perlman, published September 1999, which is hereby incorporated by reference as though fully set forth herein.”) Hilla at [0008] (“Routers and switches are network devices that determine which communication link or links to employ to support the progress of packets through the network. For example, Ethernet switches forward packets according to the Ethernet protocol. Some current routers implement sophis-ticated algorithms that provide high performance forwarding of packets based on combining layer 2 and layer 3 header information, or some other combination. For example, instead of making forwarding decisions separately on each packet in a stream of related packets (called a "packet flow" or simply a "flow"), such as a stream directed from the same source node to the same destination node, these routers identify the packet flow from a unique signature derived from the layer 2 or layer 3 header information and forward each member of the flow according to the same decision made for the first packet in the flow.”) Hilla at [0012] (“A load-balancing process is used on the sending network node of the pair connected by a bundle of commu-nication links for the purpose of determining which com-munication link to use for sending one or more data packets to the receiving network node of the pair. Current balancing algorithms use a fixed mapping to associate data packets with a specific port in a set of ports connected to the communication links in the bundle. Typically, information in a header portion of a data packet is used to derive a value that is associated with one port of the set. The algorithm is designed to generate a value in a range of values that are associated with the full set of ports. Thus data packets directed to the receiving node are distributed over all com-munication links in the bundle by the load balancing pro-cess. Many load-balancing processes are designed so that all data packets in the same data flow are sent through the same port.”) Hilla at [0035] (“FIG. 1 is a block diagram that illustrates a network with a link bundle, according to an embodiment. A computer network is a geographically distributed collection of inter-connected sub-networks (e.g, sub-networks 102a, 102b, collectively referenced hereinafter as sub-networks 102) for transporting data between network nodes (e.g., network nodes 110a, 110b, 110c, 110d, ll0e, 110/, 110g, collectively referenced hereinafter as network

No.	’740 Patent Claim 2	Hilla
		nodes 110). A local area network (LAN) is an example of such a sub-network 102. The network's topology is defined by an arrangement of end nodes that communicate with one another, typically through one or more intermediate network nodes such as a router or switch, which facilitates routing data between end nodes. As used herein, an end node is a node that is configured to originate or terminate communications over the network. In contrast, an intermediate network node facilitates the pas-sage of data between end nodes. In FIG. 1, the network nodes 110 include both end nodes (e.g., nodes 110a, 110g) and intermediate nodes (e.g., network node 110c). Each sub-network 102 includes zero or more intermediate net-work nodes.”) Hilla at [0036] (“FIG. 1 illustrates a link bundle 130 that includes five communication links between intermediate network node 110c and network node 110b. Network node 110b may be an intermediate network node or an end node that is connected both to intermediate network node 11 Oc and subnetwork 102a. For purposes of illustration, intermediate network node 110c is connected by a bundle 130 of five communication links to network node 110b and thereby to subnetwork 102a and end node 11 Oa. Similarly, intermediate network node 110c is connected by three unbundled com-munication links to network nodes 110d, ll0e, 110/, respec-tively, and thence to sub-network 102b and end node 110g. In other embodiments an intermediate network node 11 Oc may be connected to more or fewer network nodes with more or fewer links in each of one or more link bundles as part of a network with the same or more end nodes.”) Hilla at [0038] (“FIG. 2 is a block diagram that illustrates a link-bundle switching system 200 in a router, which can be used as intermediate network node 11 Oc, according to an embodi-ment. The switching system 200 includes a switching system component 202 connected to communication links, includ-ing link bundle 130. The switching system component 202 includes multiple ingress line cards, including cards 210a, 210b, 210c, for receiving inbound data packets, switching fabric component 220, and multiple egress line cards for sending outbound data packets, including egress line card 230 for link bundle 130.”) Hilla at [0039] (“An ingress line card (e.g., 210a) receives a data packet, inspects the appropriate header fields to make a forwarding decision, and forwards the packet toward the

No.	’740 Patent Claim 2	Hilla
		egress line card via the switching fabric component 220. The switch fabric component 220 forwards the packet to the appropriate egress line card 230. Multiple ingress line cards can forward traffic toward a single egress line card 230. In this example, the link bundle 130 is capable of transmitting more packets simultaneously because of the increased band-width achieved by bundling. Some ingress line cards (e.g., 210c) are connected to a link bundle with multiple commu-nication links. All communication links in the link bundle are considered a single ingress logical link. In the illustrated embodiment, the five communication links connected to ingress line card 210c are connected to network node 110b. The ingress line card 210c handles incoming traffic on the link bundle 130, while the egress line card 230 handles the outbound traffic on the link bundle 130.”) Hilla at [0040] (“The ingress line cards 210 determine an outbound physical or logical link to use to forward the data packet, for example using data in a routing table stored on the router, and sends information to the egress line card associated with the outbound physical or logical link. All communication links in the link bundle 130 are considered a single logical link. Information about data packets to be output on the link bundle 130 is sent to the egress line card 230.”)
2[b]	wherein sending the data frame comprises establishing a communication service between the user node and the communication network.	Hilla discloses wherein sending the data frame comprises establishing a communication service between the user node and the communication network. For example, Hilla discloses establishing a communication system and service between network devices and a network in which information is data traffic is exchanged between the user node and the communication network. Hilla at [0004] (“Networks of general purpose computer systems connected by external communication links are well known and widely used in commerce. The networks often include one or more network devices that facilitate the passage of information between the computer systems. A network node is a network device or computer system connected by the communication links.”)

No.	’740 Patent Claim 2	Hilla
		Hilla at [0006] (“Communications between nodes are typically effected by exchanging discrete packets of data. Each packet typically comprises 1] header information associated with a particular protocol, and 2] payload information that follows the header information and contains information to be processed, often independently of that particular protocol. In some protocols, the packet includes 3] trailer information following the payload and indicating the end of the payload information. The header includes information such as the source of the packet, its destination, the length of the payload, and other properties used by the protocol. Often, the data in the payload for the particular protocol includes a header and payload for a different protocol associated with a different, higher layer of the OSI Reference Model. The higher layer protocol is said to be encapsulated in the lower layer protocol. The headers included in a packet traversing multiple heterogeneous networks, such as the Internet, typi-cally include a physical (layer 1) header, a data-link (layer 2) header, an internetwork (layer 3) header and a transport (layer 4) header, as defined by the Open Systems Intercon-nection (OSI) Reference Model.”) Hilla at [0010] (“In some circumstances, the bandwidth needed between two nodes does not match one of the readily available bandwidths. In such circumstances, some net-works bundle multiple communication links between a pair of network nodes. For example, if network traffic between a particular server and an Ethernet switch in an office building needs bandwidth up to 500 Mbps, then it might be more cost-effective to connect five Fast Ethernet ports on each device rather than to install a Gigabit Ethernet port on each device and string a Gigabit cable in the walls between them. The five Fast Ethernet links in this example constitute a bundle of communication links. Similarly, if network traffic needs exceed 10 Gbps, then these needs can be met with a bundle of two or more 10 Gigabit Ethernet communication links. Link Aggregation Control Protocol (LACP) is part of an IEEE specification (802.3ad) that allows several physical ports to be bundled together to form a single logical channel. LACP allows a switch to negotiate an automatic bundle by sending LACP packets to the peer.”) Hilla at [0035] (“FIG. 1 is a block diagram that illustrates a network with a link bundle, according to an embodiment. A computer network is a geographically distributed collection of inter-connected sub-networks (e.g, sub-networks 102a, 102b, collectively referenced hereinafter as sub-networks 102) for transporting data between network nodes (e.g., network

No.	’740 Patent Claim 2	Hilla
		nodes 110a, 110b, 110c, 110d, ll0e, 110/, 110g, collectively referenced hereinafter as network nodes 110). A local area network (LAN) is an example of such a sub-network 102. The network's topology is defined by an arrangement of end nodes that communicate with one another, typically through one or more intermediate network nodes such as a router or switch, which facilitates routing data between end nodes. As used herein, an end node is a node that is configured to originate or terminate communications over the network. In contrast, an intermediate network node facilitates the pas-sage of data between end nodes. In FIG. 1, the network nodes 110 include both end nodes (e.g., nodes 110a, 110g) and intermediate nodes (e.g., network node 110c). Each sub-network 102 includes zero or more intermediate net-work nodes.”) Hilla at [0036] (“FIG. 1 illustrates a link bundle 130 that includes five communication links between intermediate network node 110c and network node 110b. Network node 110b may be an intermediate network node or an end node that is connected both to intermediate network node 11 Oc and subnetwork 102a. For purposes of illustration, intermediate network node 110c is connected by a bundle 130 of five communication links to network node 110b and thereby to subnetwork 102a and end node 11 Oa. Similarly, intermediate network node 110c is connected by three unbundled com-munication links to network nodes 110d, ll0e, 110/, respec-tively, and thence to sub-network 102b and end node 110g. In other embodiments an intermediate network node 11 Oc may be connected to more or fewer network nodes with more or fewer links in each of one or more link bundles as part of a network with the same or more end nodes.”) Hilla at [0128] (“In the illustrated computer used as a router, the computer system 600 includes switching system 630 as special purpose hardware for switching information for flow over a network. Switching system 630 typically includes multiple communications interfaces, such as communica-tions interface 670, for coupling to multiple other devices. In general, each coupling is with a network link 632 that is connected to another device in or attached to a network, such as local network 680 in the illustrated embodiment, to which a variety of external devices with their own processors are connected. In some embodiments an input interface or an output interface or both are linked to each of one or more external network elements. Although three network links 632a, 632b, 632c are included in network links 632 in

No.	’740 Patent Claim 2	Hilla
		the illustrated embodiment, in other embodiments, more or fewer links are connected to switching system 630. Network links 632 typically provides information communication through one or more networks to other devices that use or process the information. For example, network link 632b may provide a connection through local network 680 to a host computer 682 or to equipment 684 operated by an Internet Service Provider (ISP). ISP equipment 684 in tum provides data communication services through the public, world-wide packet-switching communication network of networks now commonly referred to as the Internet 690. A computer called a server 692 connected to the Internet provides a service in response to information received over the Internet. For example, server 692 provides routing information for use with switching system 630.”)

No.	’740 Patent Claim 3	Hilla
3	The method according to claim 1, wherein the second physical links comprise backplane traces formed on a backplane to which the one or more interface modules are coupled.	Hilla discloses the method according to claim 1, wherein the second physical links comprise backplane traces formed on a backplane to which the one or more interface modules are coupled. For example, Hilla discloses a switching system comprised of a switch fabric and bus in which communication connectors are formed and couple the switching system to line cards. A switch fabric and bus are also known as a backplane, and connections to the switch fabric and bus constitute traces formed on a backplane. Hilla at [0038] (“FIG. 2 is a block diagram that illustrates a link-bundle switching system 200 in a router, which can be used as intermediate network node 11 Oc, according to an embodi-ment. The switching system 200 includes a switching system component 202 connected to communication links, includ-ing link bundle 130. The switching system component 202 includes multiple ingress line cards, including cards 210a, 210b, 210c, for receiving inbound data packets, switching fabric component 220, and multiple egress line cards for sending outbound data packets, including egress line card 230 for link bundle 130.”) Hilla at [0039] (“An ingress line card (e.g., 210a) receives a data packet, inspects the appropriate header fields to make a forwarding decision, and forwards the packet toward the egress line card via the switching fabric component 220. The switch fabric component 220

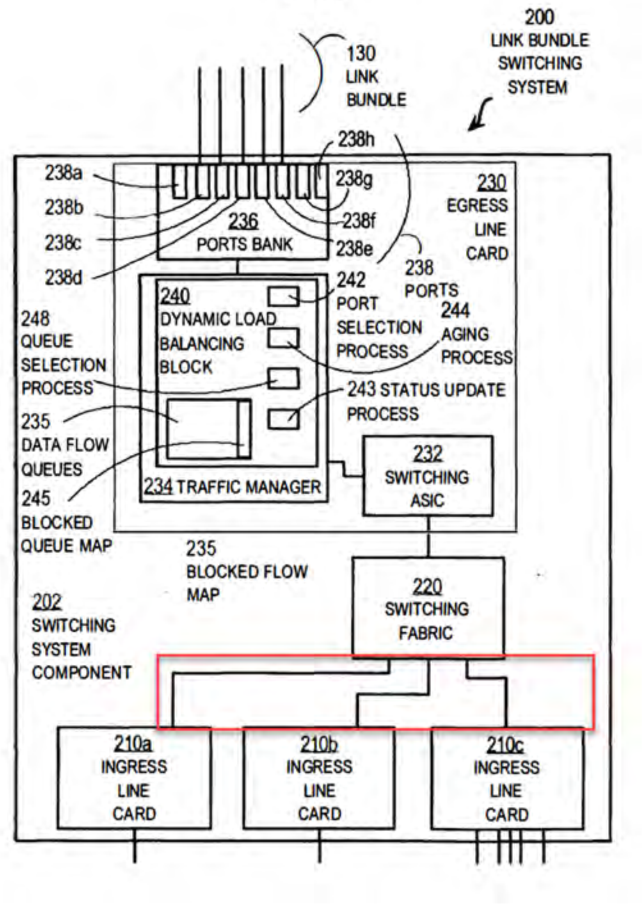
No.	’740 Patent Claim 3	Hilla
		forwards the packet to the appropriate egress line card 230. Multiple ingress line cards can forward traffic toward a single egress line card 230. In this example, the link bundle 130 is capable of transmitting more packets simultaneously because of the increased band-width achieved by bundling. Some ingress line cards (e.g., 210c) are connected to a link bundle with multiple commu-nication links. All communication links in the link bundle are considered a single ingress logical link. In the illustrated embodiment, the five communication links connected to ingress line card 210c are connected to network node 110b. The ingress line card 210c handles incoming traffic on the link bundle 130, while the egress line card 230 handles the outbound traffic on the link bundle 130.”) Hilla at [0040] (“The ingress line cards 210 determine an outbound physical or logical link to use to forward the data packet, for example using data in a routing table stored on the router, and sends information to the egress line card associated with the outbound physical or logical link. All communication links in the link bundle 130 are considered a single logical link. Information about data packets to be output on the link bundle 130 is sent to the egress line card 230.”) Hilla at [0044] (“The ports bank 236 is a bank of ports which includes multiple ports that are treated by a router as a single logical port for purposes of communicating through link bundle 130 as a single logical link. Each communication link in the link bundle 130 is connected to a different port of the ports bank 236. In the illustrated embodiment, ports bank 236 includes eight ports 238a, 238b, 238c, 238d, 238e, 238/, 238g, 238h (collectively referenced hereinafter as ports 238). For purposes of illustration, it is assumed that these eight ports are identified by a three bit code that represents the eight values from O through 7 to correspond to ports 238a through 238h, respectively. Each three bit code is called a port number (port #), herein. Only five of the ports (e.g., 238b, 238c, 238d, 238e, 238/, identified by code values 1 through 5, respectively) are connected to communication links of the link bundle 130 in the illustrated embodiment. These are called active ports. The remaining three ports (e.g., 238a, 238g, 238h, identified by code values 0, 6, 7, respectively) are inactive ports. In some embodiments, a port becomes inactive by virtue of a communication link that is attached to the port going down. When a port goes down, it is no longer eligible to send traffic and the link is removed from the route table. In an example embodiment, the hard-ware described in more detail below

No.	’740 Patent Claim 3	Hilla
		would detect the link's inability to forward traffic and dynamically shift traffic over to an active link. Such a hardware shift can occur much faster than waiting for a routing table update.”) Hilla at [0042] (“The switching ASIC performs various standard functions well known in the art, such as retrieving data packet contents from router memory based on information received from the switching fabric 220 and appending a revised header to the beginning of the packet.”) Hilla at [0067] (“In step 412 a current flow ID is determined for the current data packet. Any method known in the art at the time the method 400 is implemented may be used. In some embodiments, step 412 includes determining and combining the bit values in one or more fields in the header portions of one or more protocols. In some embodiments, step 412 involves receiving the current flow ID from another com-ponent, e.g., from switching fabric block 220, or switching ASIC 232 or traffic manager block 234. In some embodi-ments, the flow ID is the same as the particular queue 332 where the data packet is stored by the traffic manager 234. In some embodiments in which packet sequence within a data flow is not important and commit depth is not used to select ports, step 412 is omitted.”) Hilla at [0121] (“Computer system 600 includes a communication mechanism such as a bus 610 for passing information between other internal and external components of the computer system 600. Information is represented as physical signals of a measurable phenomenon, typically electric voltages, but including, in other embodiments, such phe-nomena as magnetic, electromagnetic, pressure, chemical, molecular atomic and quantum interactions. For example, north and south magnetic fields, or a zero and non zero electric voltage, represent two states (0, 1) of a binary digit (bit). A sequence of binary digits constitutes digital data that is used to represent a number or code for a character. A bus 610 includes many parallel conductors of information so that information is transferred quickly among devices coupled to the bus 610. One or more processors 602 for processing information are coupled with the bus 610. A processor 602 performs a set of operations on information. The set of operations include bringing information in from the bus 610 and placing information on the bus 610. The set of opera-tions also typically include comparing two or more units of information, shifting positions of units of information, and combining two or more units of information, such as by

No.	’740 Patent Claim 3	Hilla
		addition or multiplication. A sequence of operations to be executed by the processor 602 constitute computer instruc-tions.”) Hilla at [0122] (“Computer system 600 also includes a memory 604 coupled to bus 610. The memory 604, such as a random access memory (RAM) or other dynamic storage device, stores information including computer instructions. Dynamic memory allows information stored therein to be changed by the computer system 600. RAM allows a unit of information stored at a location called a memory address to be stored and retrieved independently of information at neighboring addresses. The memory 604 is also used by the processor 602 to store temporary values during execution of computer instructions. The computer system 600 also includes a read only memory (ROM) 606 or other static storage device coupled to the bus 610 for storing static information, including instructions, that is not changed by the computer system 600. Also coupled to bus 610 is a non-volatile (persistent) storage device 608, such as a mag-netic disk or optical disk, for storing information, including instructions, that persists even when the computer system 600 is turned off or otherwise loses power.”) Hilla at [0126] (“Computer system 600 also includes one or more instances of a communications interface 670 coupled to bus 610. Communication interface 670 provides a two-way communication coupling to a variety of external devices that operate with their own processors, such as printers, scanners, external disks, and terminal 612. Firmware or software running in the computer system 600 provides a terminal interface or character-based command interface so that external commands can be given to the computer system. For example, communication interface 670 may be a parallel port or a serial port such as an RS-232 or RS-422 interface, or a universal serial bus (USB) port on a personal computer. In some embodiments, communications interface 670 is an integrated services digital network (ISDN) card or a digital subscriber line (DSL) card or a telephone modem that provides an information communication connection to a corresponding type of telephone line. In some embodiments, a communication interface 670 is a cable modem that converts signals on bus 610 into signals for a communica-tion connection over a coaxial cable or into optical signals for a communication connection over a fiber optic cable. As another example, communications interface 670 may be a local area network (LAN) card to provide a data communi-cation connection to a compatible LAN, such as Ethernet. Wireless links may also be implemented.

No.	’740 Patent Claim 3	Hilla
		For wireless links, the communications interface 670 sends and receives elec-trical, acoustic or electromagnetic signals, including infrared and optical signals, which carry information streams, such as digital data. Such signals are examples of carrier waves”) Hilla at [0128] (“In the illustrated computer used as a router, the computer system 600 includes switching system 630 as special purpose hardware for switching information for flow over a network. Switching system 630 typically includes multiple communications interfaces, such as communica-tions interface 670, for coupling to multiple other devices. In general, each coupling is with a network link 632 that is connected to another device in or attached to a network, such as local network 680 in the illustrated embodiment, to which a variety of external devices with their own processors are connected. In some embodiments an input interface or an output interface or both are linked to each of one or more external network elements. Although three network links 632a, 632b, 632c are included in network links 632 in the illustrated embodiment, in other embodiments, more or fewer links are connected to switching system 630. Network links 632 typically provides information communication through one or more networks to other devices that use or process the information. For example, network link 632b may provide a connection through local network 680 to a host computer 682 or to equipment 684 operated by an Internet Service Provider (ISP). ISP equipment 684 in tum provides data communication services through the public, world-wide packet-switching communication network of networks now commonly referred to as the Internet 690. A computer called a server 692 connected to the Internet provides a service in response to information received over the Internet. For example, server 692 provides routing information for use with switching system 630.”) Hilla at [0129] (“The switching system 630 includes logic and cir-cuitry configured to perform switching functions associated with passing information among elements of network 680, including passing information received along one network link, e.g. 632a, as output on the same or different network link, e.g., 632c. The switching system 630 switches infor-mation traffic arriving on an input interface to an output interface according to pre-determined protocols and conven-tions that are well known. In some embodiments, switching system 630 includes its own processor and memory to perform some of the switching functions in software. In some embodiments, switching system 630 relies on proces-sor 602, memory 604,

No.	’740 Patent Claim 3	Hilla
		ROM 606, storage 608, or some combination, to perform one or more switching functions in software. For example, switching system 630, in coopera-tion with processor 604 implementing a particular protocol, can determine a destination of a packet of data arriving on input interface on link 632a and send it to the correct destination using output interface on link 632c. The desti-nations may include host 682, server 692, other terminal devices connected to local network 680 or Internet 690, or other routing and switching devices in local network 680 or Internet 690.”) Hilla at [0130] (“The invention is related to the use of computer system 600 for implementing the techniques described herein. According to one embodiment of the invention, those techniques are performed by computer system 600 in response to processor 602 executing one or more sequences of one or more instructions contained in memory 604. Such instructions, also called software and program code, may be read into memory 604 from another computer-readable medium such as storage device 608. Execution of the sequences of instructions contained in memory 604 causes processor 602 to perform the method steps described herein. In alternative embodiments, hardware, such as application specific integrated circuit 620 and circuits in switching system 630, may be used in place of or in combination with software to implement the invention. Thus, embodiments of the invention are not limited to any specific combination of hardware and software.”) Hilla at [0131] (“The signals transmitted over network link 632 and other networks through communications interfaces such as interface 670, which carry information to and from computer system 600, are exemplary forms of carrier waves. Com-puter system 600 can send and receive information, includ-ing program code, through the networks 680, 690 among others, through network links 632 and communications interfaces such as interface 670. In an example using the Internet 690, a server 692 transmits program code for a particular application, requested by a message sent from computer 600, through Internet 690, ISP equipment 684, local network 680 and network link 632b through commu-nications interface in switching system 630. The received code may be executed by processor 602 or switching system 630 as it is received, or may be stored in storage device 608 or other non-volatile storage for later execution, or both. In this manner, computer system 600 may obtain application program code in the form of a carrier wave.”)

No.	’740 Patent Claim 3	Hilla
		Hilla at Figure 2 (annotation added) FIG. 2  200 LINK BUNDLE SWITCHING SYSTEM 130 LINK BUNDLE 238h 238a 238b 238c 238d 238g 238f 238e 236 PORTS BANK 230 EGRESS LINE CARD 238 PORTS 242 PORT SELECTION PROCESS 244 AGING PROCESS 240 DYNAMIC LOAD BALANCING BLOCK 248 QUEUE SELECTION PROCESS 243 STATUS UPDATE PROCESS 235 DATA FLOW QUEUES 245 BLOCKED QUEUE MAP 234 TRAFFIC MANAGER 232 SWITCHING ASIC 235 BLOCKED FLOW MAP 202 SWITCHING SYSTEM COMPONENT 220 SWITCHING FABRIC 210a INGRESS LINE CARD 210b INGRESS LINE CARD 210c INGRESS LINE CARD In addition, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Hilla in combination with (1) the knowledge of a person of ordinary skill in the

No.	’740 Patent Claim 3	Hilla
		art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 3[a] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Ghosh discloses connecting the line cards to the active supervisor via the backplane using parallel interface circuitry. Ghosh at [0059] (“Line cards 803, 805, and 807 can communicate with an active supervisor 811 through interface circuitry 883, 885, and 887 and the backplane 815. According to various embodiments, each line card includes a plurality of ports that can act as either input ports or output ports for communication with external fibre channel network entities 851 and 853. The backplane 815 can provide a communi-cations channel for all traffic between line cards and super- visors. Individual line cards 803 and 807 can also be coupled to external fibre channel network entities 851 and 853 through fibre channel ports 843 and 847.”) For example, Bruckman discloses connecting the line cards to the network using traces comprising a backplane. Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”)

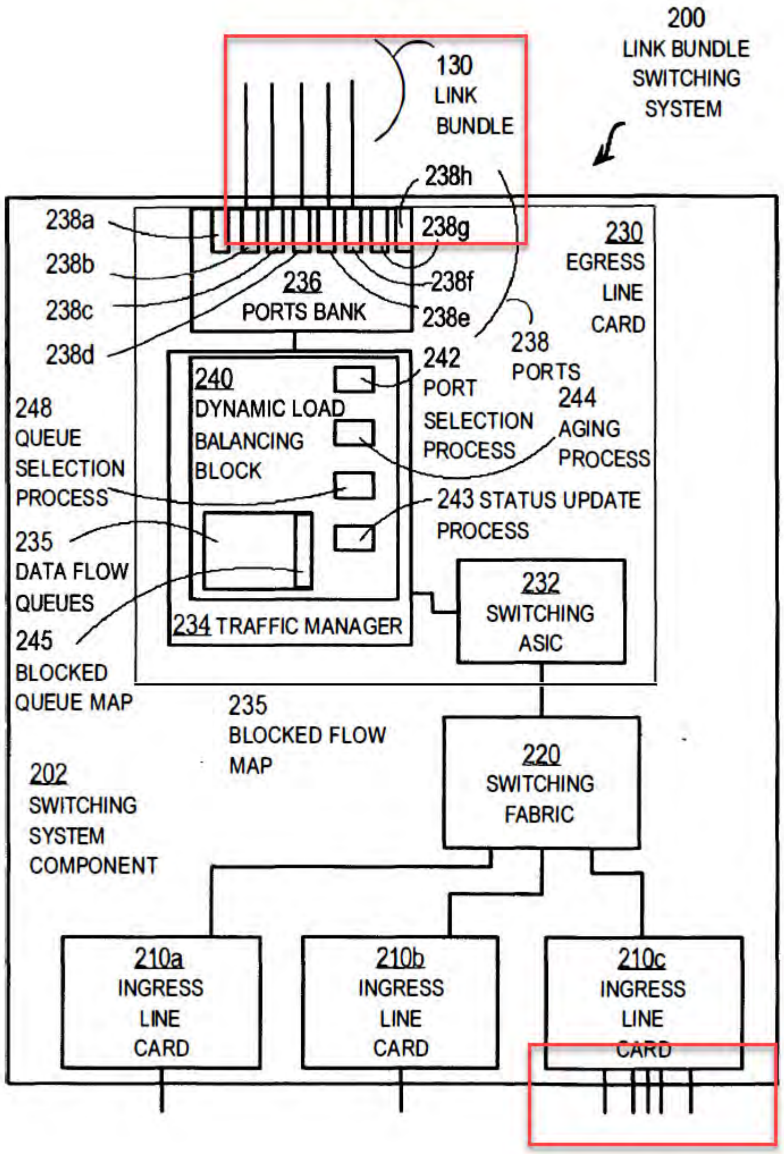
No.	’740 Patent Claim 4	Hilla
4[preamble]	A method for communication, comprising:	Hilla discloses a method for communication. *See supra* at 1[preamble].
4[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	Hilla discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].
4[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Hilla discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra* at 1[c].
4[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	Hilla discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra* at 1[e].
4[d]	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link	Hilla discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. *See supra* at 1[f].

No.	’740 Patent Claim 4	Hilla
	out of the second group; and	
4[e]	sending the data frame over the selected first and second physical links,	Hilla discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].
4[f]	at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group.	Hilla discloses at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group. For example, Hilla discloses link bundles comprised of multiple communication links such as ports, constituting an Ethernet link aggregation group. A person of ordinary skill in the art would understands that communications connectors between the switching system and line cards could also be aggregated into link bundles. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Hilla is found to not meet this limitation, at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group would have been obvious to a person having ordinary skill in the art, as explained below. Hilla at Abstract (“Techniques for distributing data packets over a network link bundle include storing an output data packet in a data flow queue based on a flow identification associated with the output data packet. The flow identification indicates a set of one or more data packets, including the output data packet, which are to be sent in the same sequence as received. State data is also received. The state data indicates a physical status of a first port of multiple active egress ports that are connected to a corresponding bundle of communication links with one particular network device. A particular data flow queue is determined based at least in part on the state data. A next data packet is directed from the particular data flow queue to a second port of the active egress ports. These techniques allow a more efficient use of a network link bundle.”) Hilla at [0010] (“In some circumstances, the bandwidth needed between two nodes does not match one of the readily available bandwidths. In such circumstances, some net-works bundle

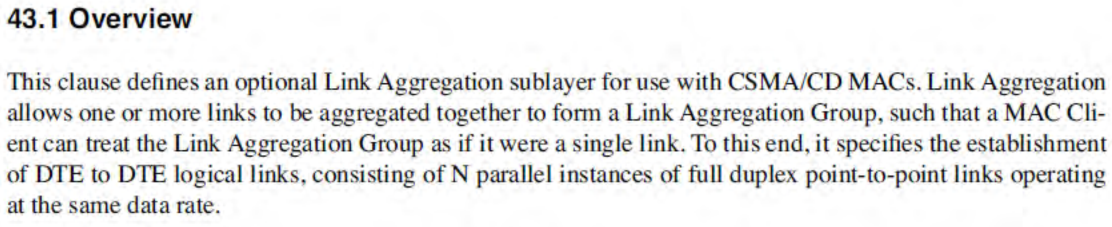
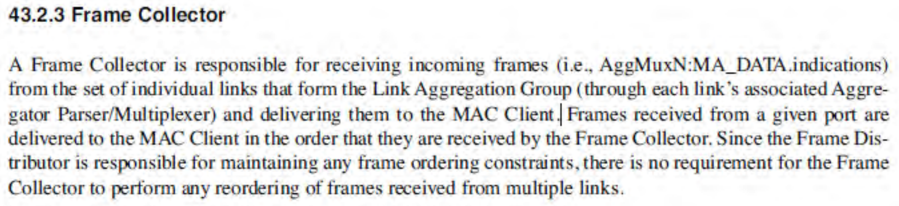
No.	’740 Patent Claim 4	Hilla
		multiple communication links between a pair of network nodes. For example, if network traffic between a particular server and an Ethernet switch in an office building needs bandwidth up to 500 Mbps, then it might be more cost-effective to connect five Fast Ethernet ports on each device rather than to install a Gigabit Ethernet port on each device and string a Gigabit cable in the walls between them. The five Fast Ethernet links in this example constitute a bundle of communication links. Similarly, if network traffic needs exceed 10 Gbps, then these needs can be met with a bundle of two or more 10 Gigabit Ethernet communication links. Link Aggregation Control Protocol (LACP) is part of an IEEE specification (802.3ad) that allows several physical ports to be bundled together to form a single logical channel. LACP allows a switch to negotiate an automatic bundle by sending LACP packets to the peer.”) Hilla at [0011] (“Bundled communication links are commercially available. For example bundled Ethernet links are available from Cisco Systems, Inc. of San Jose, Calif. as ETHER-CHANNEL™ capabilities on Ethernet switches and routers. Bundled links are also available on routers for use with a Synchronous Optical Network (SONET) for optical com-munication links as part of packet over SONET (POS) technology from Cisco Systems.”) Hilla at [0033] (“A method and apparatus are described for dynamic balancing of data packet traffic loads over a link bundle in a network. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present inven-tion. It will be apparent, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.”) Hilla at [0034] (“The invention is described in the following sections in the context of an Ethernet 802.3ad link bundle between two routers in the core or backbone of an enterprise network, but the invention is not limited to this context. In other embodiments, the link bundle is at a network edge, for example connecting an end node performing as a host for a high-throughput network server process of a client-server application. In other embodiments,

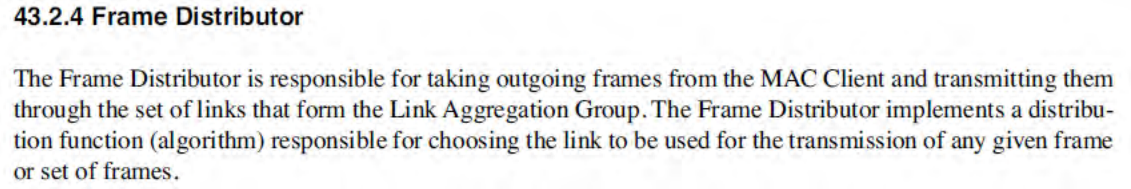
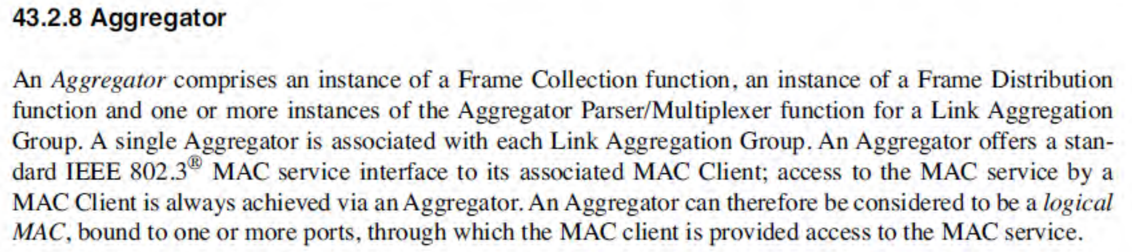
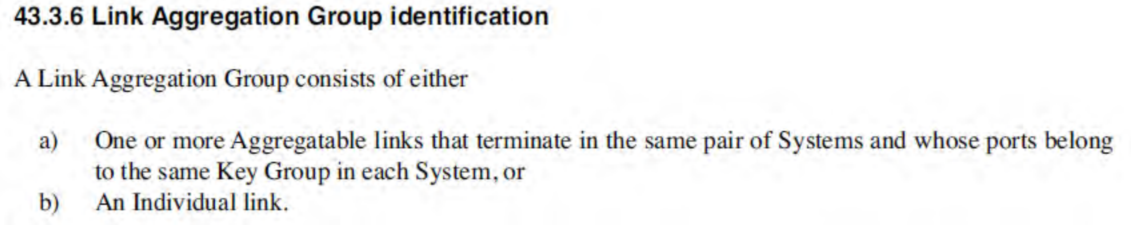
No.	’740 Patent Claim 4	Hilla
		different communication links are bundled between two network nodes, such as Packet over SONET (POS), and High-level Data Link Control (HDLC) links, among others.”) Hilla at [0036] (“FIG. 1 illustrates a link bundle 130 that includes five communication links between intermediate network node 110c and network node 110b. Network node 110b may be an intermediate network node or an end node that is connected both to intermediate network node 11 Oc and subnetwork 102a. For purposes of illustration, intermediate network node 110c is connected by a bundle 130 of five communication links to network node 110b and thereby to subnetwork 102a and end node 11 Oa. Similarly, intermediate network node 110c is connected by three unbundled com-munication links to network nodes 110d, ll0e, 110/, respec-tively, and thence to sub-network 102b and end node 110g. In other embodiments an intermediate network node 11 Oc may be connected to more or fewer network nodes with more or fewer links in each of one or more link bundles as part of a network with the same or more end nodes.”) Hilla at [0038] (“FIG. 2 is a block diagram that illustrates a link-bundle switching system 200 in a router, which can be used as intermediate network node 11 Oc, according to an embodi-ment. The switching system 200 includes a switching system component 202 connected to communication links, includ-ing link bundle 130. The switching system component 202 includes multiple ingress line cards, including cards 210a, 210b, 210c, for receiving inbound data packets, switching fabric component 220, and multiple egress line cards for sending outbound data packets, including egress line card 230 for link bundle 130.”) Hilla at [0039] (“An ingress line card (e.g., 210a) receives a data packet, inspects the appropriate header fields to make a forwarding decision, and forwards the packet toward the egress line card via the switching fabric component 220. The switch fabric component 220 forwards the packet to the appropriate egress line card 230. Multiple ingress line cards can forward traffic toward a single egress line card 230. In this example, the link bundle 130 is capable of transmitting more packets simultaneously because of the increased band-width achieved by bundling. Some ingress line cards (e.g., 210c) are connected to a link bundle with multiple commu-nication links. All communication links in the link bundle are considered a single ingress logical link. In the illustrated embodiment, the five communication

No.	’740 Patent Claim 4	Hilla
		links connected to ingress line card 210c are connected to network node 110b. The ingress line card 210c handles incoming traffic on the link bundle 130, while the egress line card 230 handles the outbound traffic on the link bundle 130.”) Hilla at [0040] (“The ingress line cards 210 determine an outbound physical or logical link to use to forward the data packet, for example using data in a routing table stored on the router, and sends information to the egress line card associated with the outbound physical or logical link. All communication links in the link bundle 130 are considered a single logical link. Information about data packets to be output on the link bundle 130 is sent to the egress line card 230.”) Hilla at [0044] (“The ports bank 236 is a bank of ports which includes multiple ports that are treated by a router as a single logical port for purposes of communicating through link bundle 130 as a single logical link. Each communication link in the link bundle 130 is connected to a different port of the ports bank 236. In the illustrated embodiment, ports bank 236 includes eight ports 238a, 238b, 238c, 238d, 238e, 238/, 238g, 238h (collectively referenced hereinafter as ports 238). For purposes of illustration, it is assumed that these eight ports are identified by a three bit code that represents the eight values from O through 7 to correspond to ports 238a through 238h, respectively. Each three bit code is called a port number (port #), herein. Only five of the ports (e.g., 238b, 238c, 238d, 238e, 238/, identified by code values 1 through 5, respectively) are connected to communication links of the link bundle 130 in the illustrated embodiment. These are called active ports. The remaining three ports (e.g., 238a, 238g, 238h, identified by code values 0, 6, 7, respectively) are inactive ports. In some embodiments, a port becomes inactive by virtue of a communication link that is attached to the port going down. When a port goes down, it is no longer eligible to send traffic and the link is removed from the route table. In an example embodiment, the hard-ware described in more detail below would detect the link's inability to forward traffic and dynamically shift traffic over to an active link. Such a hardware shift can occur much faster than waiting for a routing table update.”) Hilla at Figure 2 (annotations added)

		FIG. 2 

130
LINK
BUNDLE
200
LINK BUNDLE
SWITCHING
SYSTEM
238h
238a
238g
238b
230
EGRESS
LINE
CARD
236
PORTS BANK
238f
238c
238e
238d
238
PORTS
242
PORT
SELECTION
PROCESS
240
DYNAMIC LOAD
BALANCING
BLOCK
244
AGING
PROCESS
248
QUEUE
SELECTION
PROCESS
243 STATUS UPDATE
PROCESS
235
DATA FLOW
QUEUES
232
SWITCHING
ASIC
245
BLOCKED
QUEUE MAP
234 TRAFFIC MANAGER
235
BLOCKED FLOW
MAP
220
SWITCHING
FABRIC
202
SWITCHING
SYSTEM
COMPONENT
210a
INGRESS
LINE
CARD
210b
INGRESS
LINE
CARD
210c
INGRESS
LINE
CARD

No.	’740 Patent Claim 4	Hilla
		Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Hilla in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 4[f] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, IEEE 802.3 discloses the aggregation of one or more links together to form a Link Aggregation Group. IEEE 802.3 at 1465  **43.1 Overview** This clause defines an optional Link Aggregation sublayer for use with CSMA/CD MACs. Link Aggregation allows one or more links to be aggregated together to form a Link Aggregation Group, such that a MAC Client can treat the Link Aggregation Group as if it were a single link. To this end, it specifies the establishment of DTE to DTE logical links, consisting of N parallel instances of full duplex point-to-point links operating at the same data rate. IEEE 802.3 at 1470  **43.2.3 Frame Collector** A Frame Collector is responsible for receiving incoming frames (i.e., AggMuxN:MA_DATA.indications) from the set of individual links that form the Link Aggregation Group (through each link’s associated Aggregator Parser/Multiplexer) and delivering them to the MAC Client. Frames received from a given port are delivered to the MAC Client in the order that they are received by the Frame Collector. Since the Frame Distributor is responsible for maintaining any frame ordering constraints, there is no requirement for the Frame Collector to perform any reordering of frames received from multiple links. IEEE 802.3 at 1471

No.	’740 Patent Claim 4	Hilla
		 **43.2.4 Frame Distributor** The Frame Distributor is responsible for taking outgoing frames from the MAC Client and transmitting them through the set of links that form the Link Aggregation Group. The Frame Distributor implements a distribution function (algorithm) responsible for choosing the link to be used for the transmission of any given frame or set of frames. IEEE 802.3 at 1474  **43.2.8 Aggregator** An *Aggregator* comprises an instance of a Frame Collection function, an instance of a Frame Distribution function and one or more instances of the Aggregator Parser/Multiplexer function for a Link Aggregation Group. A single Aggregator is associated with each Link Aggregation Group. An Aggregator offers a standard IEEE 802.3® MAC service interface to its associated MAC Client; access to the MAC service by a MAC Client is always achieved via an Aggregator. An Aggregator can therefore be considered to be a *logical MAC*, bound to one or more ports, through which the MAC client is provided access to the MAC service. IEEE 802.3 at 1481  **43.3.6 Link Aggregation Group identification** A Link Aggregation Group consists of either a) One or more Aggregatable links that terminate in the same pair of Systems and whose ports belong to the same Key Group in each System, or b) An Individual link. For example, Ghosh discloses aggregating physical links, including ports, into aggregate port channels that form a single logical link to increase bandwidth. Ghosh at Abstract (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggregation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels.

No.	’740 Patent Claim 4	Hilla
		Ports can be aggregated synchronously to allow consistent gen-eration of port channel map tables.”) Ghosh at [0004] (“Neighboring nodes in a fibre channel network are typically interconnected through multiple physical links. For example, a local fibre channel switch may be connected to a remote fibre channel switch through four physical links. In many instances, it may be beneficial to aggregate some of the physical links into logical links. That is, multiple physi-cal links can be combined to form a logical interface to provide higher aggregate bandwidth, load balancing, and link redundancy. When a frame is being transmitted over a logical link, it does not matter what particular physical link is being used as long as all the frames of a given flow are transmitted through the same link. If a constituent physical link goes down, the logical link can still remain operational.”) Ghosh at [0007] (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggre-gation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent generation of port channel map tables.”) Ghosh at [0008] (“In one embodiment, a method for aggregating ports in a fibre channel fabric is provided. It is determined that a plurality of local ports at a local fibre channel switch are compatible. Identifiers for the plurality of local ports are sent to a remote fibre channel switch. The remote fibre channel switch determines if a plurality of remote ports are compat-ible, the plurality of remote ports corresponding to the plurality of local ports. An indication that one or more of the remote physical ports are compatible is received. A port channel including one or more of the local ports correspond-ing to the compatible remote ports is created.”) Ghosh at [0010] (“In another embodiment, a fibre channel network is described. The fibre channel network includes a local fibre channel switch and a remote fibre channel switch. The local fibre channel switch aggregates a compatible subset of the plurality of local ports and sends identifiers for the compat-ible subset of the plurality of local ports to the remote fibre

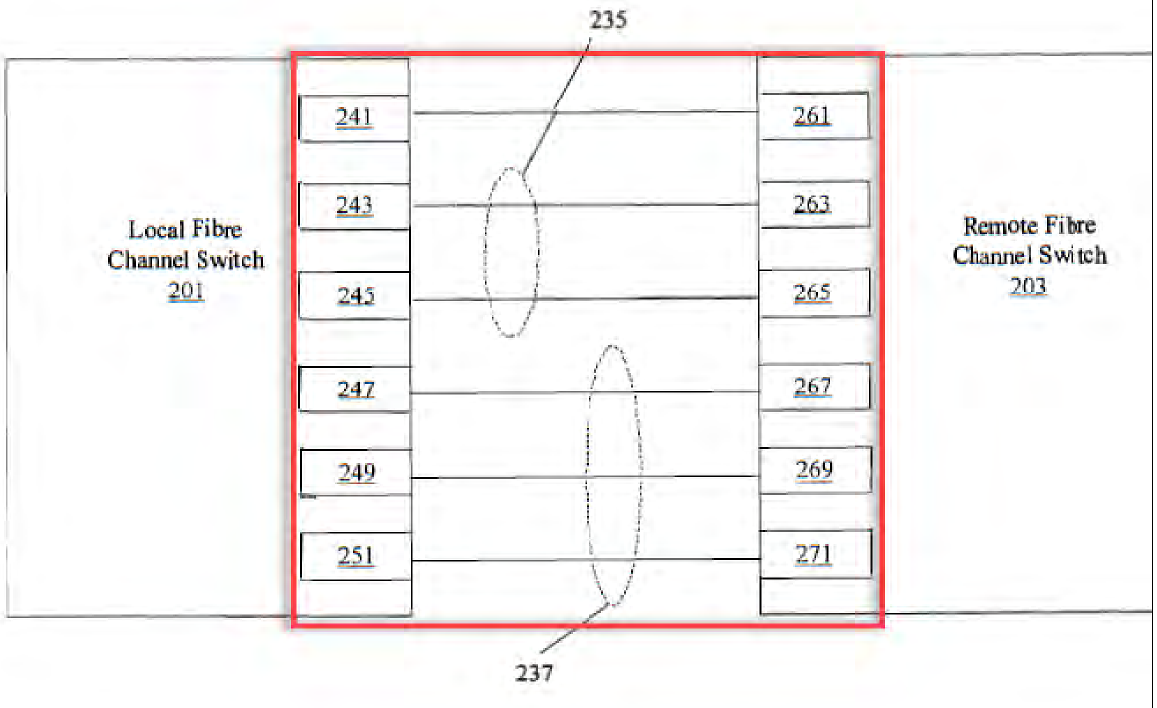
No.	’740 Patent Claim 4	Hilla
		channel switch. The remote fibre channel switch determines if a subset of the plurality of remote ports are compatible. The subset of the plurality of remote ports corresponds to the compatible subset of the plurality of local ports.”) Ghosh at [0022] (“Switches in a fibre channel network are typically interconnected using multiple physical links. The physical links connecting a pair of switches allows transmission of data and control signals. In some instances, it is useful to aggregate multiple physical links into a logical link. Physi-cal links are also referred to herein as physical interfaces and channels while logical links are also referred to herein as logical interfaces and port channels. For example, a local switch may be connected to a remote switch through four physical links. Instead of having to transmit data through a particular physical link, the physical links can be aggregated to form one or more logical links. In one example, all four physical links are aggregated into a single logical link. Instead of having data transmitted through a particular physical link, the data can merely be transmitted over a particular logical link without regard to the particular physi-cal interface used. Aggregating physical links into a logical link allows for higher aggregated bandwidth, load balancing, and link redundancy. For example, if a particular physical link fails or is overloaded, data can still be transmitted over the logical link.”) Ghosh at [0029] (“FIG. 2 is a diagrammatic representation showing links between two switches, such as two fibre channel switches shown in FIG. 1. A local fibre channel switch 201 includes local ports 241, 243, 245, 247, 249, and 251. A remote fibre channel switch 203 includes remote ports 261, 263, 265, 267, 269, and 271. Local port 241 is coupled to remote port 261 through an individual physical link or channel. Connected ports are also referred to herein as peer ports. Local port 243 is coupled to remote port 263 and local port 245 is coupled to remote port 265. The two resulting physical links are aggregated to form port channel 235. Local ports 247, 249, and 251 are coupled to remote ports 267, 269, and 271 respectively. The three resulting physical links are aggregated to form port channel 237.”) Ghosh at [0030] (“According to various embodiments, local fibre channel switch 201 and remote fibre channel switch both have associated identifiers. In some examples, the identifiers are globally unique identifiers such as a global switch world wide names (WWNs). Each local

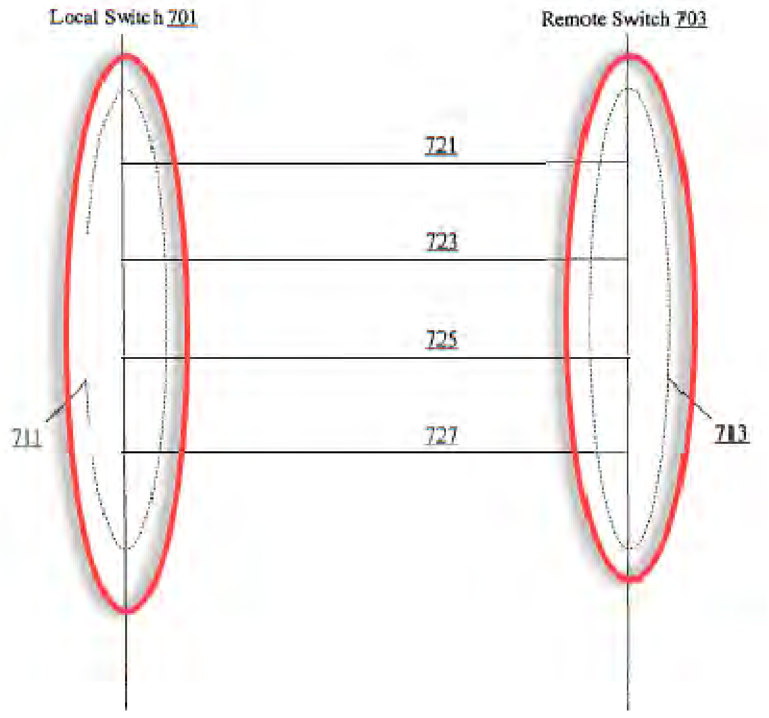
No.	’740 Patent Claim 4	Hilla
		port 241, 243, 245, 247, 249, and 251 and each remote port 261,263,265,267, 269, and 271 can also be associated with identifiers. In some examples, the identifiers are port WWNs. The port WWNs are typically used for debugging or identifying the peer port in alert or warning messages. However, according to various embodiments, the techniques of the present invention use WWNs as globally unique identifiers to aggregate ports instead of using compatibility keys which are only locally unique. Compatibility keys are mechanisms typically used by other protocols such as Ethernet for aggregation.”) Ghosh at [0033] (“A variety of parameters can be used to aggregate physical ports. FIG. 3 is a flow process diagram showing one technique for aggregating physical ports into a logical port. And 301, it is determined if auto create functionality is enabled. According to various embodiments, auto create functionality allows automatic configuration and detection of compatible physical ports as well as aggregation into one or more logical ports. Auto creation does not require user intervention. In other examples, administrators can manu-ally arrange ports for aggregation.”) Ghosh at [0037] (“FIG. 4 is an exchange diagram showing one example of a bring up procedure used for a port creating a new port channel. A local switch 401 is coupled to a remote switch 403. The local switch 401 includes a physical port Al coupled to physical port Bl included in remote switch 403. When two peer ports Al and Bl are being aggregated into a port channel, the peer switches 401 and 403 typically already know the world wide names of the individual physical peer ports. However, the peer switches only know the world wide name of their own logical port or port channel. That is, both switches have the individual physical link configured, but the link is not yet part of a port channel. At 421, a local switch 401 sends a synchronize (sync) message 411 to the remote switch 403 to begin the process of creating a port channel including ports Al and Bl.”) Ghosh at [0038] (“In some examples, the sync message 411 includes a local port channel identifier and a remote port channel identifier. In one particular example, the local port channel identifier is set to the world wide name of the local port channel assigned by the local switch 401. The remote port channel identifier is left blank to indicate that the port Al is being

No.	’740 Patent Claim 4	Hilla
		aggregated as part of a new port channel. The sync message 411 can also include other parameters such as channel status, channel model, or channel intent.”) Ghosh at [0042] (“When two peer ports A2 and B2 are aggregated into a port channel Cl, the peer switches 501 and 503 typically already know the world wide names of the indi-vidual physical peer ports A2 and B2 as well as the world wide name information of the port channel Cl. Conse-quently, the port channel is already successfully established. According to various embodiments, local switch 501 and remote switch 503 perform parameter checking to ensure that the new physical port A2 and B2 can be safely added to the existing port channel Cl. At 521, a local switch can check configuration parameters to ensure that physical ports Al and A2 at the local switch 501 are compatible. The compatibility checking can be performed anytime. In some examples, compatibility checking is checked before a local switch 501 sends a synchronize (sync) message 511 to the remote switch 503 to begin the process of aggregating ports A2 and B2 into the port channel.) Ghosh at [0043] (“In some examples, the sync message 511 includes local port channel identifier and a remote port channel identifier. In one particular example, the local port channel identifier is set to the world wide name of the local port channel assigned by the local switch 501. The remote port channel identifier is filled with the existing port channel identifier to indicate that the port A2 is being aggregated into existing port channel C2. The sync message 511 can also include other parameters such as channel status, channel model, or channel intent.”) Ghosh at [0044] (“At 531, remote switch 503 uses the information received from the local switch 501 to verify port B2 is compatible with other port in port channel C2. In one example, configuration parameters associated with B2 are checked against configuration parameters associated with Bl. The remote switch 503 can also check if the port B2 is already assigned to a different port channel. If the port B2 is compatible with port Bl, the remote switch 503 can proceed and send a sync accept message 513 in response to the sync message 511 to indicate that the port B2 can be aggregated into the port channel. The sync accept message indicates that a port channel can now be modified. At 523, local switch 501 uses the information to update its own port channel database. However, the port channel may not yet

No.	’740 Patent Claim 4	Hilla
		be fully operational until the hardware configuration is completed. The local switch 501 continues hardware configuration such as line card configuration to make the port A2 part of the port channel Cl. An acknowledgment 527 is sent and received by remote switch 503 at 529. In some examples, the local switch 501 sends a commit signal 515 when hardware configuration is complete.”) Ghosh at [0045] (“The remote switch 503 receives the commit signal at 533 and begins its own hardware configuration. On completion of its hardware configuration, remote switch 503 sends out a commit accept signal 517 to indicate to local switch 501 that hardware configuration is completed. According to various embodiments, local switch 501 receives the commit accept signal 517 and notifies relevant applications that the port channel is now fully operational at 525 and that port A2 has been aggregated into port channel Cl. The local switch 501 can also send out an acknowledge message 519. When the remote switch 503 receives the acknowledge, it notifies relevant applications that the port channel is operational at 535 and that port B2 has been aggregated into port channel Cl. In one embodiments, the techniques of the present invention contemplate using a two phase SYNC and COMMIT mechanism similar to the mechanism used in EPP.”) Ghosh at [0046] (“FIGS. 4 and 5 show examples of ports being aggregated into a port channel. At a particular switch, ports can be selected for aggregation into a port channel in a variety of manners. FIG. 6 is an exchange diagram showing automatic selection of ports at a switch for aggregation into a port channel. A local switch 601 is coupled to a remote switch 603. In one example, the local switch 601 includes physical ports Al, A2, A3, and A4 while remote switch 603 includes physical ports Bl, B2, B3, and B4. No port chan-nels have been formed.”) Ghosh at [0049] (“At 631, remote switch 603 uses the information received from the local switch 601 to verify that the peer ports of Al, A2, and A4 are compatible. That is, ports Bl, B2, and B4 are checked for compatibility. In one example, only ports Bl and B2 may be compatible, and consequently only ports Al, A2, Bl, and B2 can be included in the port channel. In another example, ports Bl, B2, and B4 are compatible, so ports Al, A2, A4, Bl, B2, and B4 can be aggregated into port channel Cl. According to various embodiments, if the port B2 is compatible with port Bl, the remote switch 603 can proceed and send a sync accept

No.	’740 Patent Claim 4	Hilla
		message 613 in response to the sync message 611 to indicate that the port B2 can be aggregated into the port channel. It should be noted that remote switch 603 can send a list indicating that ports B2 and B4 are compatible with Bl. However, the remote switch 603 sends only one compatible port B2 back for several reasons, and in the process of selection compatible port channels get priority over com-patible individual ports.”) Ghosh at [0050] (“One reason is that aggregation mechanisms and techniques can be implemented more elegantly by handling ports on an individual basis. Any individual port will either start a new port channel, be added to an existing port channel, or operate stand alone. There is no need to keep track of groups of ports to be aggregated. Another reason is that fewer ports need to be locked if only a single port is being aggregated at any one time. The sync accept message indicates that a port channel can now be modified. At 623, local switch 601 receives the information and recognizes that Al and A2 can now be aggregated into port channel Cl. However, the port channel may not yet be fully operational until the hardware configuration is completed. An acknowl-edgment 627 is sent and received by remote switch 603 at 629. In some examples, the local switch 601 sends a commit signal 615 when hardware configuration is complete.”) Ghosh at [0051] (“The remote switch 603 receives the commit signal at 633 to create port channel Cl including ports Bl and B2. Hardware configuration can now be performed. On comple-tion of its hardware configuration, remote switch 603 sends out a commit accept signal 617 to indicate to local switch 601 that hardware configuration is completed. According to various embodiments, local switch 601 receives the commit accept signal 617 and notifies relevant applications that the port channel is now fully operational at 625 and that ports Al and A2 have been aggregated into port channel Cl. The local switch 601 can also send out an acknowledge message 619. When the remote switch 603 receives the acknowledge, it notifies relevant applications that the port channel is fully operational at 635 and that ports Bl and B2 have been aggregated into port channel Cl.”) Ghosh at [0053] (“FIG. 7 is a diagrammatic representation showing synchronous aggregation of ports into a port channel. A local switch 701 is coupled to a remote switch 703 through links 721, 723, 725, and 727. According to various embodiments, the links are being

No.	’740 Patent Claim 4	Hilla
		aggregated into port channel 711 at the local switch 701 and port channel 713 at the remote switch 703 in a synchronous manner. That is the peer ports corre-sponding to each link are brought up in the same order at both the local switch 701 and the remote switch 703.”) Ghosh at Figure 2 (annotation added)  235 241 243 245 247 249 251 261 263 265 267 269 271 Local Fibre Channel Switch 201 Remote Fibre Channel Switch 203 237 Figure 2

No.	’740 Patent Claim 4	Hilla
		Ghosh at Figure 7 (annotation added)  Local Switch 701 Remote Switch 703 721 723 725 727 711 713 Figure 7

No.	’740 Patent Claim 5	Hilla
5[preamble]	A method for communication, comprising:	Hilla discloses a method for communication. *See supra at* 1[preamble].

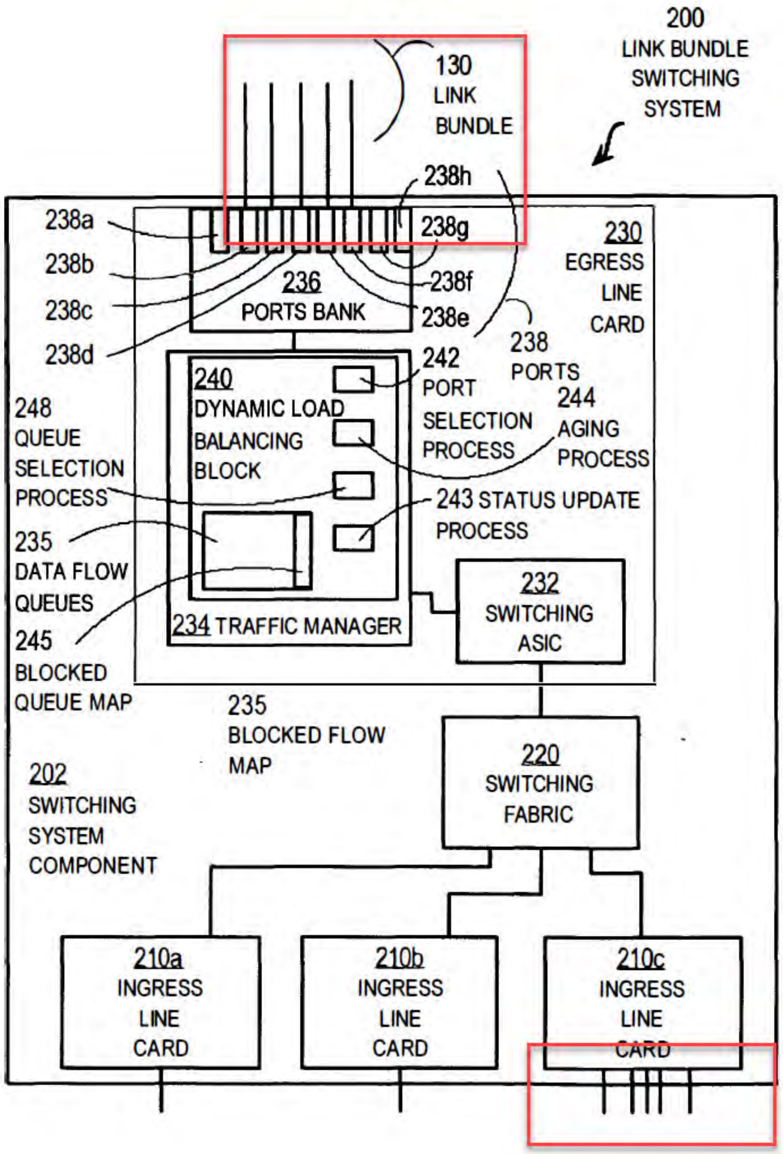
No.	’740 Patent Claim 5	Hilla
5[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	Hilla discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].
5[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Hilla discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra at* 1[c].
5[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	Hilla discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra at* 1[e].
5[d]	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	Hilla discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. *See supra at* 1[f].
5[e]	sending the data frame over the selected first	Hilla discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].

No.	’740 Patent Claim 5	Hilla
	and second physical links,	
5[f]	coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node.	Hilla discloses coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node. For example, Hilla discloses link bundles comprised of multiple communication links such as ports in, for example, an Ethernet switch which connect the network node with line cards. Hilla further discloses aggregating ports into link bundles to increase the bandwidth for network traffic provide to the network node. Hilla at Abstract (“Techniques for distributing data packets over a network link bundle include storing an output data packet in a data flow queue based on a flow identification associated with the output data packet. The flow identification indicates a set of one or more data packets, including the output data packet, which are to be sent in the same sequence as received. State data is also received. The state data indicates a physical status of a first port of multiple active egress ports that are connected to a corresponding bundle of communication links with one particular network device. A particular data flow queue is determined based at least in part on the state data. A next data packet is directed from the particular data flow queue to a second port of the active egress ports. These techniques allow a more efficient use of a network link bundle.”) Hilla at [0010] (“In some circumstances, the bandwidth needed between two nodes does not match one of the readily available bandwidths. In such circumstances, some net-works bundle multiple communication links between a pair of network nodes. For example, if network traffic between a particular server and an Ethernet switch in an office building needs bandwidth up to 500 Mbps, then it might be more cost-effective to connect five Fast Ethernet ports on each device rather than to install a Gigabit Ethernet port on each device and string a Gigabit cable in the walls between them. The five Fast Ethernet links in this example constitute a bundle of communication links. Similarly, if network traffic needs exceed 10 Gbps, then these needs can be met with a bundle of two or more 10 Gigabit Ethernet communication links. Link Aggregation Control Protocol (LACP) is part of an IEEE

No.	’740 Patent Claim 5	Hilla
		specification (802.3ad) that allows several physical ports to be bundled together to form a single logical channel. LACP allows a switch to negotiate an automatic bundle by sending LACP packets to the peer.”) Hilla at [0011] (“Bundled communication links are commercially available. For example bundled Ethernet links are available from Cisco Systems, Inc. of San Jose, Calif. as ETHER-CHANNEL™ capabilities on Ethernet switches and routers. Bundled links are also available on routers for use with a Synchronous Optical Network (SONET) for optical com-munication links as part of packet over SONET (POS) technology from Cisco Systems.”) Hilla at [0033] (“A method and apparatus are described for dynamic balancing of data packet traffic loads over a link bundle in a network. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present inven-tion. It will be apparent, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.”) Hilla at [0034] (“The invention is described in the following sections in the context of an Ethernet 802.3ad link bundle between two routers in the core or backbone of an enterprise network, but the invention is not limited to this context. In other embodiments, the link bundle is at a network edge, for example connecting an end node performing as a host for a high-throughput network server process of a client-server application. In other embodiments, different communication links are bundled between two network nodes, such as Packet over SONET (POS), and High-level Data Link Control (HDLC) links, among others.”) Hilla at [0036] (“FIG. 1 illustrates a link bundle 130 that includes five communication links between intermediate network node 110c and network node 110b. Network node 110b may be an intermediate network node or an end node that is connected both to intermediate network node 11 Oc and subnetwork 102a. For purposes of illustration, intermediate network node 110c is connected by a bundle 130 of five communication links to network node 110b

No.	’740 Patent Claim 5	Hilla
		and thereby to subnetwork 102a and end node 11 Oa. Similarly, intermediate network node 110c is connected by three unbundled com-munication links to network nodes 110d, ll0e, 110/, respec-tively, and thence to sub-network 102b and end node 110g. In other embodiments an intermediate network node 11 Oc may be connected to more or fewer network nodes with more or fewer links in each of one or more link bundles as part of a network with the same or more end nodes.”) Hilla at [0038] (“FIG. 2 is a block diagram that illustrates a link-bundle switching system 200 in a router, which can be used as intermediate network node 11 Oc, according to an embodi-ment. The switching system 200 includes a switching system component 202 connected to communication links, includ-ing link bundle 130. The switching system component 202 includes multiple ingress line cards, including cards 210a, 210b, 210c, for receiving inbound data packets, switching fabric component 220, and multiple egress line cards for sending outbound data packets, including egress line card 230 for link bundle 130.”) Hilla at [0039] (“An ingress line card (e.g., 210a) receives a data packet, inspects the appropriate header fields to make a forwarding decision, and forwards the packet toward the egress line card via the switching fabric component 220. The switch fabric component 220 forwards the packet to the appropriate egress line card 230. Multiple ingress line cards can forward traffic toward a single egress line card 230. In this example, the link bundle 130 is capable of transmitting more packets simultaneously because of the increased band-width achieved by bundling. Some ingress line cards (e.g., 210c) are connected to a link bundle with multiple commu-nication links. All communication links in the link bundle are considered a single ingress logical link. In the illustrated embodiment, the five communication links connected to ingress line card 210c are connected to network node 110b. The ingress line card 210c handles incoming traffic on the link bundle 130, while the egress line card 230 handles the outbound traffic on the link bundle 130.”) Hilla at [0040] (“The ingress line cards 210 determine an outbound physical or logical link to use to forward the data packet, for example using data in a routing table stored on the router, and sends information to the egress line card associated with the outbound physical or logical link. All communication links in the link bundle 130 are considered a single logical link.

No.	’740 Patent Claim 5	Hilla
		Information about data packets to be output on the link bundle 130 is sent to the egress line card 230.”) Hilla at [0044] (“The ports bank 236 is a bank of ports which includes multiple ports that are treated by a router as a single logical port for purposes of communicating through link bundle 130 as a single logical link. Each communication link in the link bundle 130 is connected to a different port of the ports bank 236. In the illustrated embodiment, ports bank 236 includes eight ports 238a, 238b, 238c, 238d, 238e, 238/, 238g, 238h (collectively referenced hereinafter as ports 238). For purposes of illustration, it is assumed that these eight ports are identified by a three bit code that represents the eight values from O through 7 to correspond to ports 238a through 238h, respectively. Each three bit code is called a port number (port #), herein. Only five of the ports (e.g., 238b, 238c, 238d, 238e, 238/, identified by code values 1 through 5, respectively) are connected to communication links of the link bundle 130 in the illustrated embodiment. These are called active ports. The remaining three ports (e.g., 238a, 238g, 238h, identified by code values 0, 6, 7, respectively) are inactive ports. In some embodiments, a port becomes inactive by virtue of a communication link that is attached to the port going down. When a port goes down, it is no longer eligible to send traffic and the link is removed from the route table. In an example embodiment, the hard-ware described in more detail below would detect the link's inability to forward traffic and dynamically shift traffic over to an active link. Such a hardware shift can occur much faster than waiting for a routing table update.”) Hilla at Figure 2 (annotations added)

		FIG. 2 

200
LINK BUNDLE
SWITCHING
SYSTEM
130
LINK
BUNDLE
238h
238g
238a
238b
238c
238d
236
PORTS BANK
238f
238e
238
PORTS
230
EGRESS
LINE
CARD
240
DYNAMIC LOAD
BALANCING
BLOCK
242
PORT
SELECTION
PROCESS
244
AGING
PROCESS
248
QUEUE
SELECTION
PROCESS
243 STATUS UPDATE
PROCESS
235
DATA FLOW
QUEUES
245
BLOCKED
QUEUE MAP
234 TRAFFIC MANAGER
232
SWITCHING
ASIC
235
BLOCKED FLOW
MAP
202
SWITCHING
SYSTEM
COMPONENT
220
SWITCHING
FABRIC
210a
INGRESS
LINE
CARD
210b
INGRESS
LINE
CARD
210c
INGRESS
LINE
CARD

No.	’740 Patent Claim 6	Hilla
6	The method according to claim 1, wherein coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node.	Hilla discloses the method according to claim 1, wherein coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node. For example, Hilla discloses queuing data packet flows between line cards and a network in inbound and outbound directions. A person of ordinary skill in the art would understand that communication systems employ multiplexers and demultiplexers to queue and route upstream and downstream data packets. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Hilla is found to not meet this limitation, wherein coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node would have been obvious to a person having ordinary skill in the art, as explained below. *See supra* Claim 1. Hilla at Abstract (“Techniques for distributing data packets over a network link bundle include storing an output data packet in a data flow queue based on a flow identification associated with the output data packet. The flow identification indicates a set of one or more data packets, including the output data packet, which are to be sent in the same sequence as received. State data is also received. The state data indicates a physical status of a first port of multiple active egress ports that are connected to a corresponding bundle of communication links with one particular network device. A particular data flow queue is determined based at least in part on the state data. A next data packet is directed from the particular data flow queue to a second port of the active egress ports. These techniques allow a more efficient use of a network link bundle.”)

No.	’740 Patent Claim 6	Hilla
		Hilla at [0045] (“According to various embodiments, dynamic load balancing (DLB) block 240 selects the next packet from the data flow queues and directs the next data packet to one of the ports 238 in ports bank 236 based on the state of one or more ports 238 in the ports bank 236, or time gaps between packets of the same flow, or both. In the illustrated embodi-ment, DLB block 240 includes logic for a port selection process 242 (also called an arbiter process), logic for a port status update process 243, logic for an aging process 244, and logic for a queue selection process 248, all of which are described in greater detail in a later section.”) Hilla at [0046] (“The traffic manager 234 has a data bucket, e.g., data flow queues 235 to hold all data flows received by the traffic manager 234 from the switching ASIC 232. Any type of memory may be used for data flow queues. In an example embodiment, the data flow queues 235 are first in first out (FIFO) buffers, well known in the art. The traffic manager 234 stores data packets from the same packet flow in the same data flow queue, and stores multiple packet flows over one or more queues. A large number of queues are included in data flow queues 235 in order to allow most packet flows to be stored in separate queues.”) Hilla at [0047] (“In the illustrated embodiment, the data flow queues 235 include a mapping of blocked data flow queues called a blocked queue map 245. The blocked queue map holds data that indicates whether data packets from a particular queue are directed to a port with a status unsuitable for rapidly placing that packet on the corresponding physical link. The blocked queue map 245 is manipulated by the dynamic load balancing block 240, as described in more detail below. Thus, in the illustrated embodiment, the data flow queues 235 are depicted in the DLB block 240. In other embodi-ments, the blocked queue map 245 is separate from the data flow queues 235, and the data flow queues are in the traffic manager 234 but external to the DLB block 240.”) Hilla at [0048] (“FIG. 3A is a block diagram that illustrates data flow queues 235 in more detail, according to an embodi-ment. The queues 235 includes multiple queues, including queues 332a, 332b, and other queues indicated by ellipsis 339, collectively referenced as queues 332. In the illustrated embodiment, each queue 332 includes a blocked bit (e.g., 336a, 336b, collectively referenced hereinafter as blocked bits 336), a queue depth field (e.g., 337a, 337b, collectively referenced hereinafter as queue depth field 337), and data packets that share

No.	’740 Patent Claim 6	Hilla
		the same flow ID (e.g., 338a, 338b, collectively referenced hereinafter as data packets 337). The blocked bits 336 constitute the blocked queue map 245, described above. One value indicates that the packet data in a queue is directed to a port that is blocked. A different value indicates the data is not directed to a port that is blocked (e.g., the port is not blocked or the data is not yet assigned to a port). Also as described above, in some embodiments, the blocked bits 336 of the blocked queue map 245 are stored separately from the queues 235.”) Hilla at [0049] (“The data packets 338 hold the data packets that are received by the traffic manager 234 from the ASIC 232. The data packets are assigned to a particular queue 332 of the queues 235 using any method known in the art, as described in more detail below. The queue depth field 337 holds a value that indicates the amount of data in the rest of the queue. The amount may be expressed in any units. In an illustrated embodiment, the amount is expressed as a num-ber of data packets. In some embodiments, the amount is expressed as a number of octets. An octet is eight binary digits, i.e., eight bits. The value in the depth field 337 increases as data packets are added to a queue from the ASIC 232. The value in the depth field 337 decreases as data packets are moved to a buffer for the port to which the queue is assigned, as described in more detail below. In some embodiments, the depth fields 337 are stored separately from the queues 235, e.g., in a vector of depth values with one element for each queue 332.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Hilla in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 6 of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Wiher discloses multiplexing and demultiplexing circuitry to transmit and receive ATM data cells over a data link between the ATM network and network access equipment. Wiher at 3:43-65 (“In general, in another aspect, the invention features an apparatus for communicating data cells between a data link and a backplane. The apparatus includes

No.	’740 Patent Claim 6	Hilla
		transceiver circuitry to transmit and receive data cells over a data link and a plurality of backplane interfaces each including at least one cell signal terminal. Each of the backplane interface is coupled to a backplane interconnection circuit. Each backplane interconnection circuit transmits and receives cells over the cell signal terminals of its associated backplane interface. The apparatus also includes de-multiplexing circuitry coupling the transceiver circuitry to each of the backplane interconnection circuits. The de-multiplexing circuitry receives a data cell from the transceiver circuitry, select a backplane interconnection circuit associated with the data cell, and provide the data cell to the selected backplane interconnection circuit for transmission over the cell signal terminals of the associated backplane interface. The apparatus also includes multiplexing circuitry coupling the plurality of backplane interconnection circuits to the transceiver circuitry. The multiplexing circuitry receives data cells from each of the backplane interconnection circuits and provide the received data cells to the transceiver circuitry.”) Wiher at 3:66-4:22 (“Implementations of the invention may include one or more of the following features. The backplane interconnection circuits may independently receive and transmit data cells over the plurality of backplane interfaces. The de-multiplexing circuitry may select a backplane interface based on data in the header field of the data cell. The apparatus may include header translation circuitry to alter header data in cells sent between the plurality of backplane interfaces and the transceiver circuitry. Each of the plurality of backplane interfaces may include separate terminals to receive cells and separate terminals to transmit cells. The terminals to transmit cells may include a first and second control terminal and at least one outgoing cell data terminal. A backplane interface's backplane interconnection circuitry may accepts a signal on the first control terminal as indicating that a cell may be sent over the interface, asserts a 15 signal on the second control terminal to indicate that a cell is being transmitted, and transmits data bits of the cell on the outgoing cell data terminal. Each backplane interface may include a single outgoing cell data terminal and each bit of the cell may be serially transmitted over the single outgoing cell data terminal. Each backplane interface may include multiple outgoing cell data terminals and bits of the cell may be sent in parallel over the eight outgoing cell data terminals.”)

No.	’740 Patent Claim 6	Hilla
		For example, Lebizay discloses an optical add/drop multiplexer that multiplexers and demultiplexes data packets sent between the network boards and network.. Lebizay at [0043] (“InfiniBand offers link layer Virtual Lanes (VLs) to support multiple logical channels (i.e. multiplexing) on the same physical link. Infiniband offers up to 16 virtual lanes per link. VLs provide a mechanism to avoid head-of-line blocking and the ability to support Quality of Service (QoS). The difference between a Virtual Lane and a Service Level (SL) is that a Virtual Lane is the actual logical lane (mul-tiplexed) used on a given point-to-point link. The Service Level stays constant as a packet traverses the fabric, and specifies the desired service level within a subnet. The SL (AF, EF or BE) is included in the link header, and each switch maps the SL to a VL supported by the destination link. A switch supporting a limited number of virtual lanes will map the SL field to a VL it supports. Without preserving the SL, the desired SL (AF, EF or BE) would be lost in this mapping, and later in the path, a switch supporting more VLs would be unable to recover finer granularity of SLs between two packets mapped to the same VL.”) Lebizay at [0050] (“The issue with using a ring, however, is how to map the addressing of multiple boards across these fibers. One solution is to employ Wavelength Division Multiplex-ing (WDM). A WDM optical mesh defines a meshed-topology in the wavelength space as opposed to the physical fiber space. By utilizing multiple discrete lambda-waves as optical carriers such that by meshing dedicated optical wavelengths between every two boards, layer 2 protocols are eliminated, thereby creating a dramatic improvement in the efficiency of the transport. Today, every packet transport requires a protocol that allows the end point (and interme-diate points) to decipher the intended path (or consumer) of the packet. This protocol increases the amount of overhead required in the packet bus, allowing less room for actual data to be sent. By moving the protocol into the wavelength of the actual optical signal, the destination is implied by the wavelength and no additional bandwidth needs to be sur-rendered on the signal to provide this information. This makes the efficiency of the transport better and also speeds the routing of the packet through the network. In addition, the use of optical interconnects in a backplane environment greatly increases chassis bandwidth as well as reducing electrical radiation that often accompanies copper intercon-nects. The components involved include an optical back-plane in a physical ring topology, and the necessary

No.	’740 Patent Claim 6	Hilla
		trans-mitters and receivers for the size of the installation (i.e., number of slots in the chassis). In addition, optical add/drop multiplexer devices are required.”)

No.	’740 Patent Claim 7	Hilla
7	The method according to claim 1, wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the first and second physical links.	Hilla discloses the method according to claim 1, wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the first and second physical links. For example, Hilla discloses selecting ports and connectors involving balancing data flow queues among the ports and connectors based on use and availability. *See supra* Claim 1 Hilla at [0017] (“Even if the mapping of data flows to ports is uniform, underutilization of a bundle can occur. For example, consider the situation in which two packet flows include 10,000 data packets and 10 other packet flows each include 100 packets, all packets of the same size. If there are three communication links in the bundle, then the fixed-mapping process is likely to send four packet flows to each of the three ports connected to corresponding communica-tion links in the bundle. Two of the three ports might carry 10,300 data packets, the third will carry 400 data packets. The first two ports might become overused even while the third port is underused, leading to a reduction in the rate at which the 10,300 data packets are sent over the first and second communication links. The bundle as a whole per-forms at a rate less than its advertised capability. The situation could be even worse if both large data flows are sent over the same port; then as many as 20,200 packets are sent over the first port while the second and third ports carry only 400 packets each. It would likely be preferable to send one flow of 10,000 data packets over each of the first two ports and ten flows totaling 1,000 data packets over the third port.”) Hilla at [0033] (“A method and apparatus are described for dynamic balancing of data packet traffic loads over a link bundle in a network. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough

No.	’740 Patent Claim 7	Hilla
		understanding of the present inven-tion. It will be apparent, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.”) Hilla at [0041] (“The egress line card 230 includes a switching application specific integrated circuit (ASIC) 232, a traffic manager block 234, a bank 236 of physical ports, at least some of which are connected to the communication links in the link bundle. In the illustrated embodiment, a dynamic load balancing (DLB) block 240 is included in the traffic manager 234. In some embodiments, the switching ASIC 232, traffic manager 234 and ports bank 236 are standard components of conventional egress cards for link bundles, and the DLB block 240 is external to the traffic manager 234.”) Hilla at [0045] (“According to various embodiments, dynamic load balancing (DLB) block 240 selects the next packet from the data flow queues and directs the next data packet to one of the ports 238 in ports bank 236 based on the state of one or more ports 238 in the ports bank 236, or time gaps between packets of the same flow, or both. In the illustrated embodi-ment, DLB block 240 includes logic for a port selection process 242 (also called an arbiter process), logic for a port status update process 243, logic for an aging process 244, and logic for a queue selection process 248, all of which are described in greater detail in a later section.”) Hilla at [0047] (“In the illustrated embodiment, the data flow queues 235 include a mapping of blocked data flow queues called a blocked queue map 245. The blocked queue map holds data that indicates whether data packets from a particular queue are directed to a port with a status unsuitable for rapidly placing that packet on the corresponding physical link. The blocked queue map 245 is manipulated by the dynamic load balancing block 240, as described in more detail below. Thus, in the illustrated embodiment, the data flow queues 235 are depicted in the DLB block 240. In other embodi-ments, the blocked queue map 245 is separate from the data flow queues 235, and the data flow queues are in the traffic manager 234 but external to the DLB block 240.”)

No.	’740 Patent Claim 7	Hilla
		Hilla at [0065] (“FIG. 4A is a flow diagram that illustrates a method 400 for dynamically balancing data packet traffic load on a link bundle, according to an embodiment. Although steps are shown in FIG. 4A and subsequent flow diagrams 4B and 5 in a particular order for purposes of illustration, in other embodiments one or more steps are performed in a different order or overlapping in time or are omitted, or changed in some combination of ways. For example, in a preferred embodiment shown in FIG. 4B, step 490 to advance age counters is performed in a separate aging process 244 on DLB block 240, and thus overlaps in time any of the other steps depicted in FIG. 4A.”) Hilla at [0091] (“In some embodiments, step 450 determines that the current port is underused if the port is indicated to be the port with the smallest value in the commit depth field 347. In some of these embodiments, a status update process 243 continually cycles through the records of the flow control status table 344 and updates the contents of the fill level field 346, and determines the one port with the smallest value in the commit depth field 347 at any one time. The port# with the smallest commit depth is reported in a particular register where it is accessed during step 450. For example, using the information in Table 1, port #4 is the underused port to which the next data flow is directed. As various queues are assigned to various ports, the values in the commit depth fields 347 change, and the port with the smallest commit depth changes. The commit depth value for a port buffer is increased whenever a flow queue is assigned to a port. That port's commit depth is the sum of all the packets assigned to that particular port. In the some embodiments, the commit depth is reduced as packets are played out of the flow queues an onto the port buffer. Thus, as the flow queue depth decreases, the commit depth value follows and also decreases. The update of the commit depth occurs when the flow queue update occurs; so decreases ripple to the commit depth from the flow queue. If the flow queue ages out, then the commit depth also gets adjusted, as described below in more detail with reference to step 499 in FIG. 4B. In some embodiments, the value in the commit depth field 347 is decremented as data packets are moved out of the port buffer onto the corresponding port.”) Hilla at [0108] (“As described above, the port selection process directs a data packet to a particular port. The port selection process does not determine the data packet that it receives; but reacts to the data packet sent. The selection of the next data packet is a function of the

No.	’740 Patent Claim 7	Hilla
		traffic manager. In some embodiments, the DLB block includes a queue selection process 248 in the traffic manager 234 to determine which data flow queue should be the source of the next data packet directed to a port.”) Hilla at [0109] (“In some embodiments, the traffic manager 234 determines not to send data packets from data flow queues that are directed to ports that are not suited to accepting more data at the current time. For example, in some embodiments DLB block 240 is included in traffic manager 234, so that traffic manager 234 has access to flow control status table 344. A queue selection process 248 in the traffic manager determines not to send any further data packets from flows directed to satisfied port buffers, until those status bits change to reflect a hungry or starving port buffer. This process is described further in FIG. 5.”) Hilla at [0110] (“FIG. 5 is a flow diagram that illustrates a queue selection process 500 in a traffic manager of an egress line card, according to an embodiment. Prior to step 510, data is received for a packet flow, and placed in the data packets field 338 of a data flow queue 332 of the data flow queues 235, as in a standard traffic manager. In the illustrated embodiment, a data queue exists in queues 235 for each flow ID. In some embodiments, the flow ID is one of a particular number of hash values, as described above. The data packet is placed in the queue associated with the flow ID of the packet. Processing continues at step 510.”)

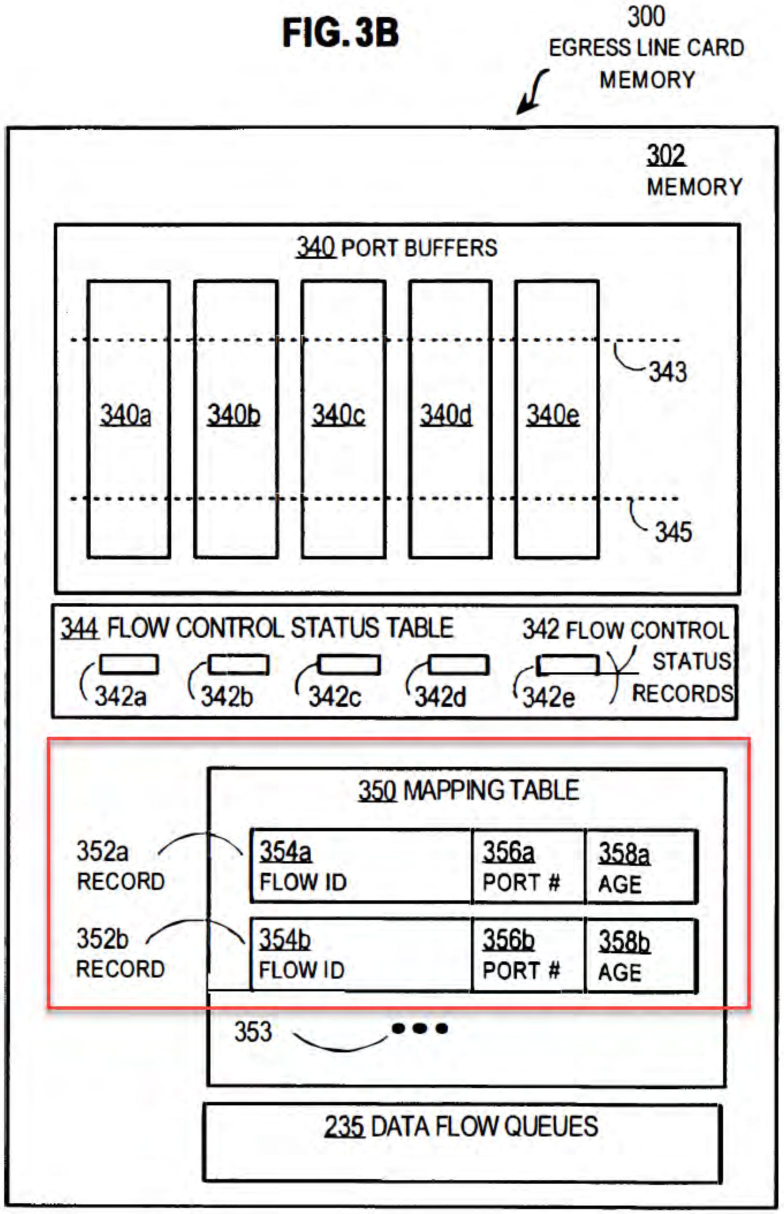
No.	’740 Patent Claim 8	Hilla
8	The method according to claim 1, wherein selecting the first and second physical links comprises applying a mapping function to the at least one of the frame attributes.	Hilla discloses the method according to claim 1, wherein selecting the first and second physical links comprises applying a mapping function to the at least one of the frame attributes. For example, Hilla discloses applying a selection mechanism for ports and communication connectors by using a mapping table to the data packet information. *See supra* Claim 1 Hilla at [0012] (“A load-balancing process is used on the sending network node of the pair connected by a bundle of commu-nication links for the purpose of determining which

No.	’740 Patent Claim 8	Hilla
		com-munication link to use for sending one or more data packets to the receiving network node of the pair. Current balancing algorithms use a fixed mapping to associate data packets with a specific port in a set of ports connected to the communication links in the bundle. Typically, information in a header portion of a data packet is used to derive a value that is associated with one port of the set. The algorithm is designed to generate a value in a range of values that are associated with the full set of ports. Thus data packets directed to the receiving node are distributed over all com-munication links in the bundle by the load balancing pro-cess. Many load-balancing processes are designed so that all data packets in the same data flow are sent through the same port.”) Hilla at [0015] (“For example, if there are eight communication links in a bundle, some fixed-mapping load-balancing pro-cesses map different data packets to one of eight values, such as by using a hash function with a three-bit output. Three bits represent eight different values (0 to 7) which are associated with the eight different ports in the set connected to the eight communication links. While such an approach may cause data packets with similar values in their layer 2 and layer 3 headers to be directed to different ports of the set, there is no guarantee that the process will distribute traffic uniformly across the set of ports. For example, a disproportionate number of data packets might be mapped to the value 5. Thus some ports may still become overused, causing a reduction in the effective bandwidth.”) Hilla at [0047] (“In the illustrated embodiment, the data flow queues 235 include a mapping of blocked data flow queues called a blocked queue map 245. The blocked queue map holds data that indicates whether data packets from a particular queue are directed to a port with a status unsuitable for rapidly placing that packet on the corresponding physical link. The blocked queue map 245 is manipulated by the dynamic load balancing block 240, as described in more detail below. Thus, in the illustrated embodiment, the data flow queues 235 are depicted in the DLB block 240. In other embodi-ments, the blocked queue map 245 is separate from the data flow queues 235, and the data flow queues are in the traffic manager 234 but external to the DLB block 240.”) Hilla at [0051] (“In the illustrated embodiment, the data structures in memory 302 include the data flow queues 235, port buffers 340, flow control status table 344, and mapping table 350.

No.	’740 Patent Claim 8	Hilla
		In other embodiments, other data structures are used to contain data corresponding to the data described here. For example, in some embodiments, some data shown in tables 344 and 350 are spread across multiple different data structures, such as one or more bitmap vectors.”) Hilla at [0058] (“The mapping table 350 includes multiple records, including record 352a, 352b, and other records indicated by ellipsis 353, collectively referenced hereinafter as records 352. In the illustrated embodiment, each record 352 includes a flow identification (flow ID) field (e.g., 354a, 354b, collectively referenced hereinafter as flow ID field 354). A particular flow ID corresponds to a particular stream of related data packets as determined, for example, by exam-ining source or destination fields, or both, in the headers of one or more protocols. Many methods for determining a flow ID are known and practiced in the art. In some embodiments, the flow ID field 354 is omitted, and the position of the record 352 in the mapping table 350 is related to a flow ID. For example, in some embodiments a flow ID is input to a hash function that outputs a particular record number in table 350 that corresponds to the flow ID input to the hash function. Each record 352 in the mapping table 350 corresponds to a queue 332 in the data flow queues 235.”) Hilla at [0061] (“The mapping table 350 is used in some embodi-ments in which the sequence of data packets received at the network node connected by the link bundle to the egress line card 230 (e.g., network node 110b connected by link bundle 130 to an egress line card in network node 110c) should be preserved. The use of table 350 to ensure that data packets arrive in correct sequence is described in more detail in the next section.”) Hilla at [0063] (“The number of records 352 in record table 350, and the corresponding number of queues 332 in data flow queues 235, is a design choice made at the time an embodiment of the invention is implemented. It is advantageous that the number of records 352 in mapping table 350 be large compared to the number of data flows being directed through a link bundle at one time if a hash function is used. If the number of records is large, then the chances are small that two data flows will be mapped by a hash value to the same record. In an example embodiment, mapping table 350 includes 1024 records 352 and data flow queues 235 includes 1024 queues 332. If multiple data flows are hashed to the same hash value, then those data flows are not distinguished in such embodiments, and are treated as a single data flow.”)

No.	’740 Patent Claim 8	Hilla
		Hilla at [0064] (“In some embodiments the mapping table 350 is in the same memory block as the port buffers 340, or flow control status bits 342, or both. In some embodiments, the mapping table is in a different memory block. For example, in some embodiments, the port buffers 340 are on ports bank 236, and both the flow control status table 344 and the mapping table 350 are on DLB block 240. In some embodi-ments, some fields are stored in different data structures.”) Hilla at [0072] (“In step 442, it is determined whether a mapping table (e.g., mapping table 350) already has an entry (e.g., a record 352) for the flow ID of the current data packet with age remaining. Age (e.g., a value stored in the age counter field 358) is an indication of the time that has passed since the most recent data packet of the flow associated with the entry was sent through the port indicated in the entry (e.g., the value in port # field 356). If age remains, as the term is used in step 442, then too little time has passed since the previous packet to allow the current packet to be sent through a different port. As a result, control passes to step 444 to use the same port as used for the previous data packet of the same flow. The use of the terms "age" and "too little" time in an example embodiment are described in more detail below.”) Hilla at [0073] (“Any method may be used to determine the entry in the mapping table associated with the current flow ID. For purposes of illustration, it is assumed that the mapping table 350 includes 1024 records, to distinguish up to 1024 differ-ent flow IDs. In some embodiments, such as embodiments using contents addressable memory (CAM), the flow ID determined in step 412 is compared to each entry in table 350 until a match is found. In other embodiments using CAM, the records 352 in mapping table 350 are sorted by flow ID and a binary search is made of the table until the record with the matching flow ID is found. In a preferred embodiment, at least a portion of the flow ID is input to a hash function, which produces a hash value between 0 and 1023. The queue 332 and record 352 at the position of the hash value+! is then the queue 332 and record 352 associ-ated with the current flow ID. For purposes of illustration, it is assumed that the flow ID of the current data packet hashes to a hash value such that the record 352 at the position of the hash value+ 1 is record 352b.”)

No.	’740 Patent Claim 8	Hilla
		Hilla at [0098] (“In addition, in step 454, the port # of the current port and particular initial value for the age are stored in association with the flow ID in the mapping table. In the illustrated embodiment, in step 442 the current flow ID had been input to the hash function to obtain a hash value between 0 and 1023 and a record number equal to the hash value+!, which corresponds to record 352b. Thus the con-tents of record 352b are updated with the port # of the current port and particular initial value for the age. In the examples described above, including Table 1, the current flow ID is stored in the flow ID field 354b, the value 5 is stored in the port # field 356b, and the value 40 is stored in the age counter field 358b. Control then passes to step 480.”) Hilla at [0107] (“In other embodiments other data structures are used as mapping table 350, or to represent the physical state of the active ports connected to the link bundle. In other embodi-ments other choices are made for the particular initial value of a timing field that corresponds to the age counter field used in the illustrated embodiment. In other embodiments, other methods are used to determine which port to use based on either the state or the gap in time between data packets of the same flow.”) Hilla at [0119] (“In the illustrated embodiment, the traffic manager receives flow control information for the port FIFO buffers to avoid sending packets toward a full or paused port buffer. In the illustrated embodiment, the traffic manager 234 has a queue 332 (data bucket) for each hash result (record 352 in the map table). The traffic manager 234 makes a queue selection knowing which queue maps to which port based on the data received in step 510 and the state of those ports. This is desirable to avoid a blocking situation in which the traffic manager 234 tries to send a packet to the ports bank when the associated port is not suitable for accepting more data, such as when the port buffer is full, or the port has received a pause message.”) Hilla at Figure 3b (annotations added)

		FIG. 3B 

300
EGRESS LINE CARD
MEMORY
302
MEMORY
340 PORT BUFFERS
340a
340b
340c
340d
340e
343
345
344 FLOW CONTROL STATUS TABLE
342 FLOW CONTROL
STATUS
RECORDS
342a
342b
342c
342d
342e
350 MAPPING TABLE
352a
RECORD
352b
RECORD
354a
FLOW ID
356a
PORT #
358a
AGE
354b
FLOW ID
356b
PORT #
358b
AGE
353
235 DATA FLOW QUEUES

No.	’740 Patent Claim 9	Hilla
9	The method according to claim 8, wherein applying the mapping function comprises applying a hashing function.	Hilla discloses the method according to claim 8, wherein applying the mapping function comprises applying a hashing function. For example, Hilla discloses applying a hash function as part of the mapping of data packets. *See supra* Claim 8 Hilla at [0012] (“A load-balancing process is used on the sending network node of the pair connected by a bundle of commu-nication links for the purpose of determining which com-munication link to use for sending one or more data packets to the receiving network node of the pair. Current balancing algorithms use a fixed mapping to associate data packets with a specific port in a set of ports connected to the communication links in the bundle. Typically, information in a header portion of a data packet is used to derive a value that is associated with one port of the set. The algorithm is designed to generate a value in a range of values that are associated with the full set of ports. Thus data packets directed to the receiving node are distributed over all com-munication links in the bundle by the load balancing pro-cess. Many load-balancing processes are designed so that all data packets in the same data flow are sent through the same port.”) Hilla at [0014] (“Typically, the fixed-mapping takes several bits from one or more fields in layer 2 or layer 3 headers, or both, and inputs those bits to a hash function that produces an output with a certain number of bits. The output is then used directly or indirectly to select a port among the set connected to the bundle of communication links. By judicious choice of the fields, data packets from the same flow may be mapped to the same port.”) Hilla at [0015] (“For example, if there are eight communication links in a bundle, some fixed-mapping load-balancing pro-cesses map different data packets to one of eight values, such as by using a hash function with a three-bit output. Three bits represent eight different values (0 to 7) which are associated with the eight different ports in the set connected to the eight communication links. While such an approach may cause data packets with similar values in their layer 2 and layer 3 headers to be directed to different ports of the set, there is no guarantee

No.	’740 Patent Claim 9	Hilla
		that the process will distribute traffic uniformly across the set of ports. For example, a disproportionate number of data packets might be mapped to the value 5. Thus some ports may still become overused, causing a reduction in the effective bandwidth.”) Hilla at [0016] (“In some approaches, the fixed-mapping is adjusted to accommodate different bundle sizes. For example, if bundle sizes are allowed to vary from two to eight commu-nication links per bundle, some fixed-mapping load-balanc-ing processes map different data packets to one of eight values, such as by using a hash function with a three-bit output, as described above, to accommodate the largest bundle. In smaller bundles, the 8 possible output values are then mapped to the three active communication links. For example, output values 0,3,6 are mapped to the first port, values 1,4,7 are mapped to the second port, and values 2,5 are mapped to the third port. Because there are three values mapped to the first two ports compared to two values mapped to the third port, the third port will be underutilized compared to the first two. The underutilization of the third port occurs even if the distribution of the original eight values is uniform.”) Hilla at [0058] (“The mapping table 350 includes multiple records, including record 352a, 352b, and other records indicated by ellipsis 353, collectively referenced hereinafter as records 352. In the illustrated embodiment, each record 352 includes a flow identification (flow ID) field (e.g., 354a, 354b, collectively referenced hereinafter as flow ID field 354). A particular flow ID corresponds to a particular stream of related data packets as determined, for example, by exam-ining source or destination fields, or both, in the headers of one or more protocols. Many methods for determining a flow ID are known and practiced in the art. In some embodiments, the flow ID field 354 is omitted, and the position of the record 352 in the mapping table 350 is related to a flow ID. For example, in some embodiments a flow ID is input to a hash function that outputs a particular record number in table 350 that corresponds to the flow ID input to the hash function. Each record 352 in the mapping table 350 corresponds to a queue 332 in the data flow queues 235.”) Hilla at [0063] (“The number of records 352 in record table 350, and the corresponding number of queues 332 in data flow queues 235, is a design choice made at the time an embodiment of the invention is implemented. It is advantageous that the number of records 352 in mapping

No.	’740 Patent Claim 9	Hilla
		table 350 be large compared to the number of data flows being directed through a link bundle at one time if a hash function is used. If the number of records is large, then the chances are small that two data flows will be mapped by a hash value to the same record. In an example embodiment, mapping table 350 includes 1024 records 352 and data flow queues 235 includes 1024 queues 332. If multiple data flows are hashed to the same hash value, then those data flows are not distinguished in such embodiments, and are treated as a single data flow.”) Hilla at [0073] (“Any method may be used to determine the entry in the mapping table associated with the current flow ID. For purposes of illustration, it is assumed that the mapping table 350 includes 1024 records, to distinguish up to 1024 differ-ent flow IDs. In some embodiments, such as embodiments using contents addressable memory (CAM), the flow ID determined in step 412 is compared to each entry in table 350 until a match is found. In other embodiments using CAM, the records 352 in mapping table 350 are sorted by flow ID and a binary search is made of the table until the record with the matching flow ID is found. In a preferred embodiment, at least a portion of the flow ID is input to a hash function, which produces a hash value between 0 and 1023. The queue 332 and record 352 at the position of the hash value+! is then the queue 332 and record 352 associ-ated with the current flow ID. For purposes of illustration, it is assumed that the flow ID of the current data packet hashes to a hash value such that the record 352 at the position of the hash value+ 1 is record 352b.”) Hilla at [0076] (“In some embodiments, the number of records in the mapping table is so large compared to the number of data flows received at one time that there are rarely any "colli-sions," e.g. multiple different concurrent data flows hashing to the same hash value and therefore to the same record 352. In some of these embodiments, the record number indicated by the hash value is treated as if it belongs to the flow ID of the current data packet, and the flow ID field 354 is omitted from the records 352. Thus, in the example above, record 352b is assumed to belong to the flow ID of the current data packet. The flow ID becomes an index into the record table; i.e., the hash value is the address of the record entry. In this case, also, the value in the age counter field (e.g., 358b) is checked to determine if there is age remaining. To ensure that no age remains in a record that has not yet been used, the mapping table 350 is initialized with null values for the age counter field 358 or the port # field 356 or both.”)

No.	’740 Patent Claim 9	Hilla
		Hilla at [0098] (“In addition, in step 454, the port # of the current port and particular initial value for the age are stored in association with the flow ID in the mapping table. In the illustrated embodiment, in step 442 the current flow ID had been input to the hash function to obtain a hash value between 0 and 1023 and a record number equal to the hash value+!, which corresponds to record 352b. Thus the con-tents of record 352b are updated with the port # of the current port and particular initial value for the age. In the examples described above, including Table 1, the current flow ID is stored in the flow ID field 354b, the value 5 is stored in the port # field 356b, and the value 40 is stored in the age counter field 358b. Control then passes to step 480.”) Hilla at [0115] (“In step 530, a data flow is determined to be blocked based on the data received during step 510. For example, it is determined that all flow IDs that hash to a hash value of 1 or 2 are blocked, because the flow control status bits associated with the hash values of 1 and 2 indicate the associated port buffers are satisfied. In some embodiments, step 530 includes determining a data flow directed to the most underused port as the unblocked data flow.”) Hilla at [0116] (“In step 540, the traffic manager 234 sends a data packet from a data queue for a flow ID that does not hash to the hash value of a heavily used link, e.g., hash values 1 and 2. For example, the traffic manager 234 sends a data packet from a queue for a flow ID that hashes to the values 3, 4 or 5. If no packet data resides in the data queue for these flow ID's, then no packet is sent and processing continues at step 510. The flow control status bits associated with these hash values indicate the associated port buffers are hungry or starving, as shown in Table 1. In some embodiments, the traffic manager 234 sends a data packet from a queue for a flow ID that hashes to the value 5 because the flow control status bits associated with hash value 5 indicate the associ-ated port buffer is starving. If no packet data resides in the data queue for flow ID hashed to value 5, then a packet may be chosen for flow IDs that hash to values 3 or 4 (whose associated port buffers are hungry).”)

No.	’740 Patent Claim 10	Hilla
10[a]	The method according to claim 9, wherein applying the hashing	Hilla discloses the method according to claim 9, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second physical links.

No.	’740 Patent Claim 10	Hilla
	function comprises determining a hashing size responsively to a number of at least some of the first and second physical links,	For example, Hilla discloses a hash function, which includes determining the number of ports. A person of ordinary skill in the art would understand that a hashing size would include the number of communication connectors in addition to the number or ports. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. *See supra at* Claim 9. Hilla at [0014] (“Typically, the fixed-mapping takes several bits from one or more fields in layer 2 or layer 3 headers, or both, and inputs those bits to a hash function that produces an output with a certain number of bits. The output is then used directly or indirectly to select a port among the set connected to the bundle of communication links. By judicious choice of the fields, data packets from the same flow may be mapped to the same port.”) Hilla at [0015] (“For example, if there are eight communication links in a bundle, some fixed-mapping load-balancing pro-cesses map different data packets to one of eight values, such as by using a hash function with a three-bit output. Three bits represent eight different values (0 to 7) which are associated with the eight different ports in the set connected to the eight communication links. While such an approach may cause data packets with similar values in their layer 2 and layer 3 headers to be directed to different ports of the set, there is no guarantee that the process will distribute traffic uniformly across the set of ports. For example, a disproportionate number of data packets might be mapped to the value 5. Thus some ports may still become overused, causing a reduction in the effective bandwidth.”) Hilla at [0016] (“In some approaches, the fixed-mapping is adjusted to accommodate different bundle sizes. For example, if bundle sizes are allowed to vary from two to eight commu-nication links per bundle, some fixed-mapping load-balanc-ing processes map different data packets to one of eight values, such as by using a hash function with a three-bit output, as described above, to accommodate the largest bundle. In smaller bundles, the 8 possible output values are then mapped to the three active communication links. For example, output values 0,3,6 are mapped to the first port, values 1,4,7 are mapped to the second port, and values 2,5 are mapped to the third port. Because there are three values mapped to the first

No.	’740 Patent Claim 10	Hilla
		two ports compared to two values mapped to the third port, the third port will be underutilized compared to the first two. The underutilization of the third port occurs even if the distribution of the original eight values is uniform.”) Hilla at [0017] (“Even if the mapping of data flows to ports is uniform, underutilization of a bundle can occur. For example, consider the situation in which two packet flows include 10,000 data packets and 10 other packet flows each include 100 packets, all packets of the same size. If there are three communication links in the bundle, then the fixed-mapping process is likely to send four packet flows to each of the three ports connected to corresponding communica-tion links in the bundle. Two of the three ports might carry 10,300 data packets, the third will carry 400 data packets. The first two ports might become overused even while the third port is underused, leading to a reduction in the rate at which the 10,300 data packets are sent over the first and second communication links. The bundle as a whole per-forms at a rate less than its advertised capability. The situation could be even worse if both large data flows are sent over the same port; then as many as 20,200 packets are sent over the first port while the second and third ports carry only 400 packets each. It would likely be preferable to send one flow of 10,000 data packets over each of the first two ports and ten flows totaling 1,000 data packets over the third port.”) Hilla at [0056] (“In the illustrated embodiment, each record 342 also includes a commit depth field (e.g., 347a, 347b, collectively referenced hereinafter as commit depth field 347). A par-ticular commit depth value indicates an amount of data in the one or more data flow queues 332 that have been directed to the corresponding port by the port selection process, as is described in more detail below. Any units or granularity in data amount may be used. In some embodiments, the amount of data is expressed as a number of data packets; in some embodiments the amount of data is expressed as a number of octets. For example, if there are 12 data packets in one data flow queue directed to the port #1 and 25 data packets in another data flow queue also directed to the same port, and data packets are the units for expressing amount, then the value in commit depth field 347a for port #1 is 37. The commit depth is a measure of the amount of data expected to be transmitted through the corresponding port; and is independent of the fill level of the port buffer. The commit field is used in some embodiments to determine whether a particular port is under-utilized compared to another

No.	’740 Patent Claim 10	Hilla
		port, as described in more detail below. In some embodiments, the commit depth field 247 is omitted.”) Hilla at [0058] (“The mapping table 350 includes multiple records, including record 352a, 352b, and other records indicated by ellipsis 353, collectively referenced hereinafter as records 352. In the illustrated embodiment, each record 352 includes a flow identification (flow ID) field (e.g., 354a, 354b, collectively referenced hereinafter as flow ID field 354). A particular flow ID corresponds to a particular stream of related data packets as determined, for example, by exam-ining source or destination fields, or both, in the headers of one or more protocols. Many methods for determining a flow ID are known and practiced in the art. In some embodiments, the flow ID field 354 is omitted, and the position of the record 352 in the mapping table 350 is related to a flow ID. For example, in some embodiments a flow ID is input to a hash function that outputs a particular record number in table 350 that corresponds to the flow ID input to the hash function. Each record 352 in the mapping table 350 corresponds to a queue 332 in the data flow queues 235.”) Hilla at [0059] (“Each record 352 includes a port number (port #) field (e.g., 356a, 356b, collectively referenced hereinafter as port # field 356). The port # field contains data that indicates a particular port on the ports bank, if any, which is used to send data packets for the flow ID associated with the record 352. The value in the port# field indicates the port to which the corresponding queue is assigned.”) Hilla at [0063] (“The number of records 352 in record table 350, and the corresponding number of queues 332 in data flow queues 235, is a design choice made at the time an embodiment of the invention is implemented. It is advantageous that the number of records 352 in mapping table 350 be large compared to the number of data flows being directed through a link bundle at one time if a hash function is used. If the number of records is large, then the chances are small that two data flows will be mapped by a hash value to the same record. In an example embodiment, mapping table 350 includes 1024 records 352 and data flow queues 235 includes 1024 queues 332. If multiple data flows are hashed to the same hash value, then those data flows are not distinguished in such embodiments, and are treated as a single data flow.”)

No.	’740 Patent Claim 10	Hilla
		Hilla at [0072] (“In step 442, it is determined whether a mapping table (e.g., mapping table 350) already has an entry (e.g., a record 352) for the flow ID of the current data packet with age remaining. Age (e.g., a value stored in the age counter field 358) is an indication of the time that has passed since the most recent data packet of the flow associated with the entry was sent through the port indicated in the entry (e.g., the value in port # field 356). If age remains, as the term is used in step 442, then too little time has passed since the previous packet to allow the current packet to be sent through a different port. As a result, control passes to step 444 to use the same port as used for the previous data packet of the same flow. The use of the terms "age" and "too little" time in an example embodiment are described in more detail below.”) Hilla at [0073] (“Any method may be used to determine the entry in the mapping table associated with the current flow ID. For purposes of illustration, it is assumed that the mapping table 350 includes 1024 records, to distinguish up to 1024 differ-ent flow IDs. In some embodiments, such as embodiments using contents addressable memory (CAM), the flow ID determined in step 412 is compared to each entry in table 350 until a match is found. In other embodiments using CAM, the records 352 in mapping table 350 are sorted by flow ID and a binary search is made of the table until the record with the matching flow ID is found. In a preferred embodiment, at least a portion of the flow ID is input to a hash function, which produces a hash value between 0 and 1023. The queue 332 and record 352 at the position of the hash value+! is then the queue 332 and record 352 associ-ated with the current flow ID. For purposes of illustration, it is assumed that the flow ID of the current data packet hashes to a hash value such that the record 352 at the position of the hash value+ 1 is record 352b.”) Hilla at [0075] (“If there is a value in the flow ID field (e.g., 354b) that matches the current flow ID, then a previous data packet of the same flow was sent through the port indicated by the value in the port # field of this record (e.g., 356b). In this case, the value in the age counter field (e.g., 358b) is checked to determine if there is age remaining.”)

No.	’740 Patent Claim 10	Hilla
		Hilla at [0076] (“In some embodiments, the number of records in the mapping table is so large compared to the number of data flows received at one time that there are rarely any "colli-sions," e.g. multiple different concurrent data flows hashing to the same hash value and therefore to the same record 352. In some of these embodiments, the record number indicated by the hash value is treated as if it belongs to the flow ID of the current data packet, and the flow ID field 354 is omitted from the records 352. Thus, in the example above, record 352b is assumed to belong to the flow ID of the current data packet. The flow ID becomes an index into the record table; i.e., the hash value is the address of the record entry. In this case, also, the value in the age counter field (e.g., 358b) is checked to determine if there is age remaining. To ensure that no age remains in a record that has not yet been used, the mapping table 350 is initialized with null values for the age counter field 358 or the port # field 356 or both.”) Hilla at [0078] (“If there is age remaining, then the time between the current data packet and the previous data packet of the same flow ID is too short to ensure that the current packet will arrive after the earlier packet if the current packet is sent through a different port. Thus control passes to step 444 to send the current data packet through the same port as the previous data packet with the same flow ID. That port is indicated by the value in the port # field (e.g., field 356b) of the record (e.g., record 352b) associated with this flow ID.”) Hilla at [0080] (“If it is determined in step 442 that there is age remaining in a record associated with the current flow ID, then control passes to step 444. In step 444, the current data packet is directed to the port indicated by the port number in the record of the current flow ID. For example, the current data packet is placed in the port buffer for the port indicated by the value of the port # field 346b.”) Hilla at [0096] (“In another example embodiment, step 450 uses the port with the smallest value of commit depth as determined by the background status update process 243. Using the data of Table 1, the status update process 243 determines that the smallest commit depth at the current time belongs to port #4, with 11 data packets in the data flow queues already directed to port 4 and places the port # for this port in a register used during step 450. Port #4, which is merely hungry, could have a smaller commit depth than port #5, which is starving, for example, right after a data flow queue with 30 data packets is directed to port #5. During

No.	’740 Patent Claim 10	Hilla
		step 450, the register is read, and the value of port# is determined to be 4. Therefore, port 4 is determined to be underused. In some embodiments, the port with the smallest commit depth that is not paused is determined to be underused. For example, using the infor-mation in Table 1, port #3 is determined to be underused because port #4 is paused.”) Hilla at [0098] (“In addition, in step 454, the port # of the current port and particular initial value for the age are stored in association with the flow ID in the mapping table. In the illustrated embodiment, in step 442 the current flow ID had been input to the hash function to obtain a hash value between 0 and 1023 and a record number equal to the hash value+!, which corresponds to record 352b. Thus the con-tents of record 352b are updated with the port # of the current port and particular initial value for the age. In the examples described above, including Table 1, the current flow ID is stored in the flow ID field 354b, the value 5 is stored in the port # field 356b, and the value 40 is stored in the age counter field 358b. Control then passes to step 480.”)
10[b]	applying the hashing function to the at least one of the frame attributes to produce a hashing key,	Hilla discloses applying the hashing function to the at least one of the frame attributes to produce a hashing key. For example, Hilla discloses a hash function including determining a hash value based on the data packet information. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. Hilla at [0063] (“The number of records 352 in record table 350, and teh corresponding number of queues 332 in data flow queues 235, is a design choice made at the time an embodiment of the invention is implemented. It is advantageous that the number of records 352 in mapping table 350 be large compared to the number of data flows being directed through a link bundle at one time if a hash function is used. If the number of records is large, then the chances are small that two data flows will be mapped by a hash value to the same record. In an example embodiment, mapping table 350 includes 1024 records 352 and data flow queues 235 includes 1024 queues 332. If multiple data flows are hashed to the same hash

No.	’740 Patent Claim 10	Hilla
		value, then those data flows are not distinguished in such embodiments, and are treated as a single data flow.”) Hilla at [0069] (“In some embodiments, step 420 includes sending to the traffic manager 234 data that indicates flow control status. Any method may be used. For example, data is sent to a register on the traffic manager 234 or to a register on the DLB block 240 that is readable by the traffic manager. In an illustrated embodiment, the flow control status table 344 and mapping table 350 are in an area of memory accessible to the traffic manager 234. In some embodiments, the data sent to the traffic manager includes both a packet flow ID and the flow control status record 342 of the port buffer of the port associated with the packet flow ID. For example, the data sent to the traffic manager includes a hash value of 2 to which the flow ID hashes and the control status fill level equal to 3, which indicates that the port buffer to which this data flow maps is satisfied.”) Hilla at [0073] (“Any method may be used to determine the entry in the mapping table associated with the current flow ID. For purposes of illustration, it is assumed that the mapping table 350 includes 1024 records, to distinguish up to 1024 differ-ent flow IDs. In some embodiments, such as embodiments using contents addressable memory (CAM), the flow ID determined in step 412 is compared to each entry in table 350 until a match is found. In other embodiments using CAM, the records 352 in mapping table 350 are sorted by flow ID and a binary search is made of the table until the record with the matching flow ID is found. In a preferred embodiment, at least a portion of the flow ID is input to a hash function, which produces a hash value between 0 and 1023. The queue 332 and record 352 at the position of the hash value+! is then the queue 332 and record 352 associ-ated with the current flow ID. For purposes of illustration, it is assumed that the flow ID of the current data packet hashes to a hash value such that the record 352 at the position of the hash value+ 1 is record 352b.”) Hilla at [0076] (“In some embodiments, the number of records in the mapping table is so large compared to the number of data flows received at one time that there are rarely any "colli-sions," e.g. multiple different concurrent data flows hashing to the same hash value and therefore to the same record 352. In some of these embodiments, the record number indicated

No.	’740 Patent Claim 10	Hilla
		by the hash value is treated as if it belongs to the flow ID of the current data packet, and the flow ID field 354 is omitted from the records 352. Thus, in the example above, record 352b is assumed to belong to the flow ID of the current data packet. The flow ID becomes an index into the record table; i.e., the hash value is the address of the record entry. In this case, also, the value in the age counter field (e.g., 358b) is checked to determine if there is age remaining. To ensure that no age remains in a record that has not yet been used, the mapping table 350 is initialized with null values for the age counter field 358 or the port # field 356 or both.”) Hilla at [0098] (“In addition, in step 454, the port # of the current port and particular initial value for the age are stored in association with the flow ID in the mapping table. In the illustrated embodiment, in step 442 the current flow ID had been input to the hash function to obtain a hash value between 0 and 1023 and a record number equal to the hash value+!, which corresponds to record 352b. Thus the con-tents of record 352b are updated with the port # of the current port and particular initial value for the age. In the examples described above, including Table 1, the current flow ID is stored in the flow ID field 354b, the value 5 is stored in the port # field 356b, and the value 40 is stored in the age counter field 358b. Control then passes to step 480.”) Hilla at [0110] (“FIG. 5 is a flow diagram that illustrates a queue selection process 500 in a traffic manager of an egress line card, according to an embodiment. Prior to step 510, data is received for a packet flow, and placed in the data packets field 338 of a data flow queue 332 of the data flow queues 235, as in a standard traffic manager. In the illustrated embodiment, a data queue exists in queues 235 for each flow ID. In some embodiments, the flow ID is one of a particular number of hash values, as described above. The data packet is placed in the queue associated with the flow ID of the packet. Processing continues at step 510.”) Hilla at [0111] (“In step 510 state data is received that indicates a blocked data flow. Any method may be used to indicate the blocked data flow. For example, in some embodiments, data indicating a flow ID and flow control status records for the associated buffer are sent by the port selection process 242 during step 420 and received during step 510. In some embodiments, data indicating the flow ID hash value is received instead of data indicating the

No.	’740 Patent Claim 10	Hilla
		flow ID. In the illustrated embodiment, data is retrieved from the mapping table 350 and flow control status table 344 during 510.”) Hilla at [0115] (“In step 530, a data flow is determined to be blocked based on the data received during step 510. For example, it is determined that all flow IDs that hash to a hash value of 1 or 2 are blocked, because the flow control status bits associated with the hash values of 1 and 2 indicate the associated port buffers are satisfied. In some embodiments, step 530 includes determining a data flow directed to the most underused port as the unblocked data flow.”) Hilla at [0116] (“In step 540, the traffic manager 234 sends a data packet from a data queue for a flow ID that does not hash to the hash value of a heavily used link, e.g., hash values 1 and 2. For example, the traffic manager 234 sends a data packet from a queue for a flow ID that hashes to the values 3, 4 or 5. If no packet data resides in the data queue for these flow ID's, then no packet is sent and processing continues at step 510. The flow control status bits associated with these hash values indicate the associated port buffers are hungry or starving, as shown in Table 1. In some embodiments, the traffic manager 234 sends a data packet from a queue for a flow ID that hashes to the value 5 because the flow control status bits associated with hash value 5 indicate the associ-ated port buffer is starving. If no packet data resides in the data queue for flow ID hashed to value 5, then a packet may be chosen for flow IDs that hash to values 3 or 4 (whose associated port buffers are hungry).”) Hilla at [0118] (“During step 540, in these embodiments, a data flow queue is selected that is not blocked, based on the blocked queue map 245. The next data packet in the selected data flow queue is then sent for processing by the port selection process 242. In the embodiment using Table 1, it is deter-mined that the next data packet should be sent from a data flow queue directed to port #3 or #5. It is assumed for purposes of illustration that a data flow queue having a hash value directed to record 352a is associated with port #3 in mapping table 350. Therefore the next data packet in this data flow is sent to the port selection process 242.”)

No.	’740 Patent Claim 10	Hilla
10[c]	calculating a modulo of a division operation of the hashing key by the hashing size, and	Hilla discloses calculating a modulo of a division operation of the hashing key by the hashing size. For example, Hilla discloses a hash function including determining a hash value based on the data packet information and using that hash value to direct data flows. A person of ordinary skill in the art would understand that applying a hash function includes determining parameters responsive to the system and packet features, and generating a result. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Hilla is found to not meet this limitation, calculating a modulo of a division operation of the hashing key by the hashing size would have been obvious to a person having ordinary skill in the art, as explained below. Hilla at [0063] (“The number of records 352 in record table 350, and the corresponding number of queues 332 in data flow queues 235, is a design choice made at the time an embodiment of the invention is implemented. It is advantageous that the number of records 352 in mapping table 350 be large compared to the number of data flows being directed through a link bundle at one time if a hash function is used. If the number of records is large, then the chances are small that two data flows will be mapped by a hash value to the same record. In an example embodiment, mapping table 350 includes 1024 records 352 and data flow queues 235 includes 1024 queues 332. If multiple data flows are hashed to the same hash value, then those data flows are not distinguished in such embodiments, and are treated as a single data flow.”) Hilla at [0069] (“In some embodiments, step 420 includes sending to the traffic manager 234 data that indicates flow control status. Any method may be used. For example, data is sent to a register on the traffic manager 234 or to a register on the DLB block 240 that is readable by the traffic manager. In an illustrated embodiment, the flow control status table 344 and mapping table 350 are in an area of memory accessible to the traffic manager 234. In some embodiments, the data sent to the traffic manager includes both a packet flow ID and the flow control status record 342 of the port buffer of the port associated with the packet flow ID. For example, the data sent to the traffic manager includes a hash value of 2 to which the flow ID

No.	’740 Patent Claim 10	Hilla
		hashes and the control status fill level equal to 3, which indicates that the port buffer to which this data flow maps is satisfied.”) Hilla at [0073] (“Any method may be used to determine the entry in the mapping table associated with the current flow ID. For purposes of illustration, it is assumed that the mapping table 350 includes 1024 records, to distinguish up to 1024 differ-ent flow IDs. In some embodiments, such as embodiments using contents addressable memory (CAM), the flow ID determined in step 412 is compared to each entry in table 350 until a match is found. In other embodiments using CAM, the records 352 in mapping table 350 are sorted by flow ID and a binary search is made of the table until the record with the matching flow ID is found. In a preferred embodiment, at least a portion of the flow ID is input to a hash function, which produces a hash value between 0 and 1023. The queue 332 and record 352 at the position of the hash value+! is then the queue 332 and record 352 associ-ated with the current flow ID. For purposes of illustration, it is assumed that the flow ID of the current data packet hashes to a hash value such that the record 352 at the position of the hash value+ 1 is record 352b.”) Hilla at [0076] (“In some embodiments, the number of records in the mapping table is so large compared to the number of data flows received at one time that there are rarely any "colli-sions," e.g. multiple different concurrent data flows hashing to the same hash value and therefore to the same record 352. In some of these embodiments, the record number indicated by the hash value is treated as if it belongs to the flow ID of the current data packet, and the flow ID field 354 is omitted from the records 352. Thus, in the example above, record 352b is assumed to belong to the flow ID of the current data packet. The flow ID becomes an index into the record table; i.e., the hash value is the address of the record entry. In this case, also, the value in the age counter field (e.g., 358b) is checked to determine if there is age remaining. To ensure that no age remains in a record that has not yet been used, the mapping table 350 is initialized with null values for the age counter field 358 or the port # field 356 or both.”) Hilla at [0098] (“In addition, in step 454, the port # of the current port and particular initial value for the age are stored in association with the flow ID in the mapping table. In the illustrated embodiment, in step 442 the current flow ID had been input to the hash function

No.	’740 Patent Claim 10	Hilla
		to obtain a hash value between 0 and 1023 and a record number equal to the hash value+!, which corresponds to record 352b. Thus the con-tents of record 352b are updated with the port # of the current port and particular initial value for the age. In the examples described above, including Table 1, the current flow ID is stored in the flow ID field 354b, the value 5 is stored in the port # field 356b, and the value 40 is stored in the age counter field 358b. Control then passes to step 480.”) Hilla at [0110] (“FIG. 5 is a flow diagram that illustrates a queue selection process 500 in a traffic manager of an egress line card, according to an embodiment. Prior to step 510, data is received for a packet flow, and placed in the data packets field 338 of a data flow queue 332 of the data flow queues 235, as in a standard traffic manager. In the illustrated embodiment, a data queue exists in queues 235 for each flow ID. In some embodiments, the flow ID is one of a particular number of hash values, as described above. The data packet is placed in the queue associated with the flow ID of the packet. Processing continues at step 510.”) Hilla at [0111] (“In step 510 state data is received that indicates a blocked data flow. Any method may be used to indicate the blocked data flow. For example, in some embodiments, data indicating a flow ID and flow control status records for the associated buffer are sent by the port selection process 242 during step 420 and received during step 510. In some embodiments, data indicating the flow ID hash value is received instead of data indicating the flow ID. In the illustrated embodiment, data is retrieved from the mapping table 350 and flow control status table 344 during 510.”) Hilla at [0115] (“In step 530, a data flow is determined to be blocked based on the data received during step 510. For example, it is determined that all flow IDs that hash to a hash value of 1 or 2 are blocked, because the flow control status bits associated with the hash values of 1 and 2 indicate the associated port buffers are satisfied. In some embodiments, step 530 includes determining a data flow directed to the most underused port as the unblocked data flow.”) Hilla at [0116] (“In step 540, the traffic manager 234 sends a data packet from a data queue for a flow ID that does not hash to the hash value of a heavily used link, e.g., hash values 1

No.	’740 Patent Claim 10	Hilla
		and 2. For example, the traffic manager 234 sends a data packet from a queue for a flow ID that hashes to the values 3, 4 or 5. If no packet data resides in the data queue for these flow ID's, then no packet is sent and processing continues at step 510. The flow control status bits associated with these hash values indicate the associated port buffers are hungry or starving, as shown in Table 1. In some embodiments, the traffic manager 234 sends a data packet from a queue for a flow ID that hashes to the value 5 because the flow control status bits associated with hash value 5 indicate the associ-ated port buffer is starving. If no packet data resides in the data queue for flow ID hashed to value 5, then a packet may be chosen for flow IDs that hash to values 3 or 4 (whose associated port buffers are hungry).”) Hilla at [0118] (“During step 540, in these embodiments, a data flow queue is selected that is not blocked, based on the blocked queue map 245. The next data packet in the selected data flow queue is then sent for processing by the port selection process 242. In the embodiment using Table 1, it is deter-mined that the next data packet should be sent from a data flow queue directed to port #3 or #5. It is assumed for purposes of illustration that a data flow queue having a hash value directed to record 352a is associated with port #3 in mapping table 350. Therefore the next data packet in this data flow is sent to the port selection process 242.”)
10[d]	selecting the first and second physical links responsively to the modulo.	Hilla discloses selecting the first and second physical links responsively to the modulo. For example, Hilla discloses a hash function including determining a hash value based on the data packet information. A person of ordinary skill in the art would understand that applying a hash function includes determining parameters responsive to the system and packet features, and generating a result. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Hilla is found to not meet this limitation, selecting the first and second physical links responsively to the modulo would have been obvious to a person having ordinary skill in the art, as explained below. Hilla at [0063] (“The number of records 352 in record table 350, and the corresponding number of queues 332 in data flow queues 235, is a design choice made at the time an embodiment of the invention is implemented. It is advantageous that the number of records 352 in mapping table 350 be large compared to the number of data flows being directed

No.	’740 Patent Claim 10	Hilla
		through a link bundle at one time if a hash function is used. If the number of records is large, then the chances are small that two data flows will be mapped by a hash value to the same record. In an example embodiment, mapping table 350 includes 1024 records 352 and data flow queues 235 includes 1024 queues 332. If multiple data flows are hashed to the same hash value, then those data flows are not distinguished in such embodiments, and are treated as a single data flow.”) Hilla at [0069] (“In some embodiments, step 420 includes sending to the traffic manager 234 data that indicates flow control status. Any method may be used. For example, data is sent to a register on the traffic manager 234 or to a register on the DLB block 240 that is readable by the traffic manager. In an illustrated embodiment, the flow control status table 344 and mapping table 350 are in an area of memory accessible to the traffic manager 234. In some embodiments, the data sent to the traffic manager includes both a packet flow ID and the flow control status record 342 of the port buffer of the port associated with the packet flow ID. For example, the data sent to the traffic manager includes a hash value of 2 to which the flow ID hashes and the control status fill level equal to 3, which indicates that the port buffer to which this data flow maps is satisfied.”) Hilla at [0073] (“Any method may be used to determine the entry in the mapping table associated with the current flow ID. For purposes of illustration, it is assumed that the mapping table 350 includes 1024 records, to distinguish up to 1024 differ-ent flow IDs. In some embodiments, such as embodiments using contents addressable memory (CAM), the flow ID determined in step 412 is compared to each entry in table 350 until a match is found. In other embodiments using CAM, the records 352 in mapping table 350 are sorted by flow ID and a binary search is made of the table until the record with the matching flow ID is found. In a preferred embodiment, at least a portion of the flow ID is input to a hash function, which produces a hash value between 0 and 1023. The queue 332 and record 352 at the position of the hash value+! is then the queue 332 and record 352 associ-ated with the current flow ID. For purposes of illustration, it is assumed that the flow ID of the current data packet hashes to a hash value such that the record 352 at the position of the hash value+ 1 is record 352b.”)

No.	’740 Patent Claim 10	Hilla
		Hilla at [0076] (“In some embodiments, the number of records in the mapping table is so large compared to the number of data flows received at one time that there are rarely any "colli-sions," e.g. multiple different concurrent data flows hashing to the same hash value and therefore to the same record 352. In some of these embodiments, the record number indicated by the hash value is treated as if it belongs to the flow ID of the current data packet, and the flow ID field 354 is omitted from the records 352. Thus, in the example above, record 352b is assumed to belong to the flow ID of the current data packet. The flow ID becomes an index into the record table; i.e., the hash value is the address of the record entry. In this case, also, the value in the age counter field (e.g., 358b) is checked to determine if there is age remaining. To ensure that no age remains in a record that has not yet been used, the mapping table 350 is initialized with null values for the age counter field 358 or the port # field 356 or both.”) Hilla at [0098] (“In addition, in step 454, the port # of the current port and particular initial value for the age are stored in association with the flow ID in the mapping table. In the illustrated embodiment, in step 442 the current flow ID had been input to the hash function to obtain a hash value between 0 and 1023 and a record number equal to the hash value+!, which corresponds to record 352b. Thus the con-tents of record 352b are updated with the port # of the current port and particular initial value for the age. In the examples described above, including Table 1, the current flow ID is stored in the flow ID field 354b, the value 5 is stored in the port # field 356b, and the value 40 is stored in the age counter field 358b. Control then passes to step 480.”) Hilla at [0110] (“FIG. 5 is a flow diagram that illustrates a queue selection process 500 in a traffic manager of an egress line card, according to an embodiment. Prior to step 510, data is received for a packet flow, and placed in the data packets field 338 of a data flow queue 332 of the data flow queues 235, as in a standard traffic manager. In the illustrated embodiment, a data queue exists in queues 235 for each flow ID. In some embodiments, the flow ID is one of a particular number of hash values, as described above. The data packet is placed in the queue associated with the flow ID of the packet. Processing continues at step 510.”) Hilla at [0111] (“In step 510 state data is received that indicates a blocked data flow. Any method may be used to indicate the blocked data flow. For example, in some embodiments,

No.	’740 Patent Claim 10	Hilla
		data indicating a flow ID and flow control status records for the associated buffer are sent by the port selection process 242 during step 420 and received during step 510. In some embodiments, data indicating the flow ID hash value is received instead of data indicating the flow ID. In the illustrated embodiment, data is retrieved from the mapping table 350 and flow control status table 344 during 510.”) Hilla at [0115] (“In step 530, a data flow is determined to be blocked based on the data received during step 510. For example, it is determined that all flow IDs that hash to a hash value of 1 or 2 are blocked, because the flow control status bits associated with the hash values of 1 and 2 indicate the associated port buffers are satisfied. In some embodiments, step 530 includes determining a data flow directed to the most underused port as the unblocked data flow.”) Hilla at [0116] (“In step 540, the traffic manager 234 sends a data packet from a data queue for a flow ID that does not hash to the hash value of a heavily used link, e.g., hash values 1 and 2. For example, the traffic manager 234 sends a data packet from a queue for a flow ID that hashes to the values 3, 4 or 5. If no packet data resides in the data queue for these flow ID's, then no packet is sent and processing continues at step 510. The flow control status bits associated with these hash values indicate the associated port buffers are hungry or starving, as shown in Table 1. In some embodiments, the traffic manager 234 sends a data packet from a queue for a flow ID that hashes to the value 5 because the flow control status bits associated with hash value 5 indicate the associ-ated port buffer is starving. If no packet data resides in the data queue for flow ID hashed to value 5, then a packet may be chosen for flow IDs that hash to values 3 or 4 (whose associated port buffers are hungry).”) Hilla at [0118] (“During step 540, in these embodiments, a data flow queue is selected that is not blocked, based on the blocked queue map 245. The next data packet in the selected data flow queue is then sent for processing by the port selection process 242. In the embodiment using Table 1, it is deter-mined that the next data packet should be sent from a data flow queue directed to port #3 or #5. It is assumed for purposes of illustration that a data flow queue having a hash value directed to record 352a is associated with port #3 in mapping table 350. Therefore the next data packet in this data flow is sent to the port selection process 242.”)

No.	’740 Patent Claim 10	Hilla
		Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Hilla in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[d] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses distributing data frames over physical links and traces based on a hash function involving a division operation (%). Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0025] (“Typically, setting the protection policy includes determining a maximum number of the physical links that may fail while the logical link continues to provide at least the guaranteed bandwidth for the connection. In one embodiment, the guaranteed bandwidth is a bandwidth B, and the plurality of physical links consists of N links, and the maximum number is an integer P, and the link bandwidth allocated to each of the links is no less than B/(N-P). Conveying the data may further include managing the transmission of the data responsively to an actual number X of the physical links that have failed so that the guaranteed bandwidth on each of the links is limited to B/(N-X), X P, and an excess bandwidth on the physical links over the guaranteed bandwidth is available for other connections.”) Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link

<table>
<tr><th>No.</th><th>’740 Patent Claim 10</th><th>Hilla</th></tr>
<tr><td></td><td></td><td>

include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”)

Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:

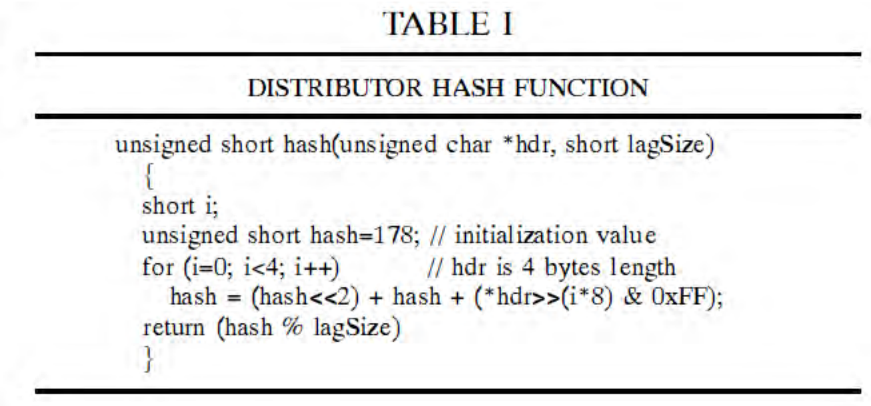


TABLE I

DISTRIBUTOR HASH FUNCTION

```
unsigned short hash(unsigned char *hdr, short lagSize)
  {
  short i;
  unsigned short hash=178; // initialization value
  for (i=0; i<4; i++)          // hdr is 4 bytes length
     hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
  return (hash % lagSize)
  }
```

”)

Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)

Bruckman at [0067] (“A similar problem may arise if there is a failure in a link in an aggregation group or in one of a number of line cards serving the aggregation group. In this case, to maintain the bandwidth allocation B made by CAC 44, each of the remaining links in the group must now carry, on average, B/(N-M) traffic, wherein M is the number of links in the group that are out of service. If only BIN has been allocated to each link, the remaining active links may not have sufficient bandwidth to continue to provide the bandwidth that has

</td></tr>
</table>

No.	’740 Patent Claim 10	Hilla
		been guaranteed to the connections that they are required to carry. A similar problem arises with respect to loading of traces 52. For example, if there is a failure in LC2 or in one of links 30 in group 36 that connect to LC2, the trace connecting the multiplexer 50 in LCl will have to carry a substantially larger share of the bandwidth, or even all of the bandwidth, that is allocated to the connection in ques-tion.”) Bruckman at [0068] (“FIG. 3 is a flow chart that schematically illustrates a method for dealing with these problems of fluctuating bandwidth requirements, in accordance with an embodiment of the present invention. In order to provide sufficient bandwidth for failure protection, CAC 44 uses a safety margin based on a protection parameter P, which is assigned at a protection setting step 60. P represents the maximum number of links in the group that can be out of service while still permitting the aggregation group to provide a given connection with the bandwidth that has been guaranteed to the connection. CAC 44 will then allocate at least B/(N-P) bandwidth to each link in the group, so that if P links fail, the group still provides total bandwidth of (N-P)*B/(N-P)= B. Setting P=l is equivalent to 1:N protection, so that the group will be unaffected by failure of a single link. In the example of group 36, shown in FIG. 2, setting P=2 will give both facility and equipment protection, i.e., the group will be unaffected not only by failure of a link, but also by failure of one of line cards 34. In the extreme case, in which P=N-1, CAC 44 will allocate the full bandwidth B on each link in the group.”) As another example, Singh discloses determining a ratio between the number of ingress and egress links and the number of links carrying data to the backplane and using a modulo to correspond to the channel’s link number. Singh at 9:30-43 (“The ratio between the number of line ingress links and the number of links carrying data to the backplane gives the backplane speedup for the system. In this example, there are 10 ingress links into the MS and 20 links (2 backplane channels) carrying that data to the backplane. This gives a backplane speedup of 2x. As another example, with 8 ingress links and 12 backplane links, there is a speedup of 1.5x. It should be noted that in addition to the backplane speedup, there is also an ingress/egress speedup. With 10 ingress links capable of carrying 2 Gbps each of raw data, this presents a 20 Gbps interface to the MS. An OC-192

No.	’740 Patent Claim 10	Hilla
		only has approximately 10 Gbps worth of data. Taking into account cell overhead and cell quantization inefficiencies, there still remains excess capacity in the links.”) Singh at 11:29-38 (“FIG. 9 is a diagram illustrating link to channel assignments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a channel, whether it is a backplane channel or an ingress/egress channel. There is no logical relationship between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.”) Singh at 15:15-39 (“The number of crossbars that are required in a system is dependent on how many links are being used to create the backplane channels. There should be an even number of crossbars and they would be divided evenly across the switch cards. The following equation, for most cases, pro-vides the correct number of crossbars: # of Crossbars=(# links per ingress channelx# of ingress channels per portx# of port cardsx speedup)/32. For the 8x8 configuration, the # of crossbars should be multiplied by (4x# of iMS)/(# backplane channels per port card). The number of port cards should be rounded up to the nearest supported configuration, i.e. 8, 16, or 32. The speedup in the case of crossbars should be the fractional speedup that is desired. Example to determine the number of arbiters and cross-bars for the following system: 4 channel port cards (40 Gbps) 8 links per channel 16 port cards Speedup=l.5 # of arbiters=(4x2x2)/2=8 # of crossbars=(8x4x16xl.5)/32=24. This would give 3crossbars per arbiter.”)

No.	’740 Patent Claim 10	Hilla
		Singh at 16:28-44 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 16x16 and 32x32 is the organization of the switchplane. The port card remains the same. Backplane channels 1 and 2 are used for the backplane connectivity. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 16, correspond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”) Singh at 17:31-49 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 8x8 and 16x16 is the organi-zation of the switchplane. The port card remains the same. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Links 0-7 and 24-31 on the arbiters would not be used and would be powered off. Links 0-7 and 24-31 on the crossbars would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Backplane channels 1 and 2 are used for the backplane connectivity. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 8, corre-spond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”)

No.	’740 Patent Claim 11	Hilla
11	The method according to claim 10, wherein selecting the first and	Hilla discloses the method according to claim 10, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical

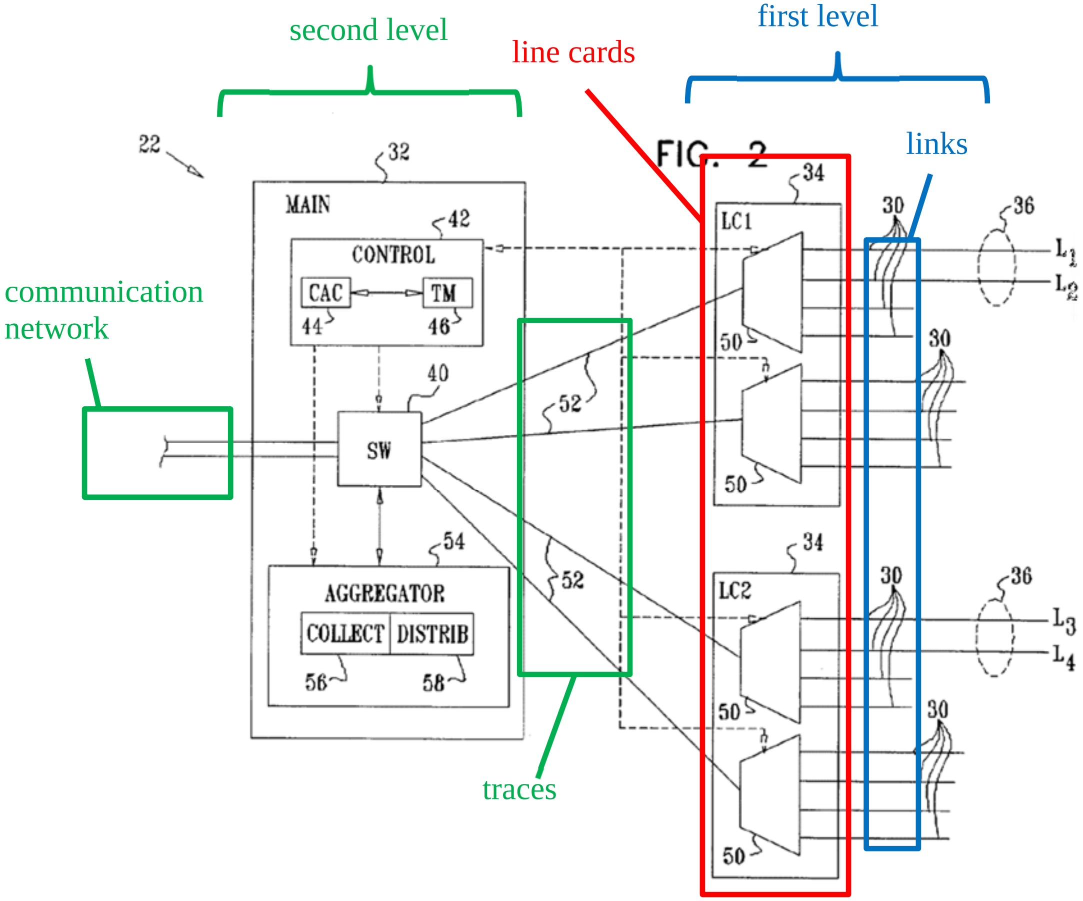
No.	’740 Patent Claim 11	Hilla
	second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo.	links responsively to respective first and second subsets of bits in a binary representation of the modulo. For example, Hilla discloses determining flow control status bits associated with hash values that indicate the availability of ports when selection occurs. A person of ordinary skill in the art would understand that selecting ports and communication connectors includes associating with the subset of bits of the flow control status. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Hilla is found to not meet this limitation, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo would have been obvious to a person having ordinary skill in the art, as explained below. *See supra* Claim 10 Hilla at [0015] (“For example, if there are eight communication links in a bundle, some fixed-mapping load-balancing pro-cesses map different data packets to one of eight values, such as by using a hash function with a three-bit output. Three bits represent eight different values (0 to 7) which are associated with the eight different ports in the set connected to the eight communication links. W hile such an approach may cause data packets with similar values in their layer 2 and layer 3 headers to be directed to different ports of the set, there is no guarantee that the process will distribute traffic uniformly across the set of ports. For example, a disproportionate number of data packets might be mapped to the value 5. Thus some ports may still become overused, causing a reduction in the effective bandwidth.”) Hilla at [0016] (“In some approaches, the fixed-mapping is adjusted to accommodate different bundle sizes. For example, if bundle sizes are allowed to vary from two to eight commu-nication links per bundle, some fixed-mapping load-balanc-ing processes map different data packets to one of eight values, such as by using a hash function with a three-bit output, as described above, to accommodate the largest bundle. In smaller bundles, the 8 possible output values are then mapped to the three active communication links. For example,

No.	’740 Patent Claim 11	Hilla
		output values 0,3,6 are mapped to the first port, values 1,4,7 are mapped to the second port, and values 2,5 are mapped to the third port. Because there are three values mapped to the first two ports compared to two values mapped to the third port, the third port will be underutilized compared to the first two. The underutilization of the third port occurs even if the distribution of the original eight values is uniform.”) Hilla at [0044] (“The ports bank 236 is a bank of ports which includes multiple ports that are treated by a router as a single logical port for purposes of communicating through link bundle 130 as a single logical link. Each communication link in the link bundle 130 is connected to a different port of the ports bank 236. In the illustrated embodiment, ports bank 236 includes eight ports 238a, 238b, 238c, 238d, 238e, 238/, 238g, 238h (collectively referenced hereinafter as ports 238). For purposes of illustration, it is assumed that these eight ports are identified by a three bit code that represents the eight values from O through 7 to correspond to ports 238a through 238h, respectively. Each three bit code is called a port number (port #), herein. Only five of the ports (e.g., 238b, 238c, 238d, 238e, 238/, identified by code values 1 through 5, respectively) are connected to communication links of the link bundle 130 in the illustrated embodiment. These are called active ports. The remaining three ports (e.g., 238a, 238g, 238h, identified by code values 0, 6, 7, respectively) are inactive ports. In some embodiments, a port becomes inactive by virtue of a communication link that is attached to the port going down. When a port goes down, it is no longer eligible to send traffic and the link is removed from the route table. In an example embodiment, the hard-ware described in more detail below would detect the link's inability to forward traffic and dynamically shift traffic over to an active link. Such a hardware shift can occur much faster than waiting for a routing table update.”) Hilla at [0049] (“The data packets 338 hold the data packets that are received by the traffic manager 234 from the ASIC 232. The data packets are assigned to a particular queue 332 of the queues 235 using any method known in the art, as described in more detail below. The queue depth field 337 holds a value that indicates the amount of data in the rest of the queue. The amount may be expressed in any units. In an illustrated embodiment, the amount is expressed as a num-ber of data packets. In some embodiments, the amount is expressed as a number of octets. An octet is eight binary digits, i.e., eight bits. The value in the depth field

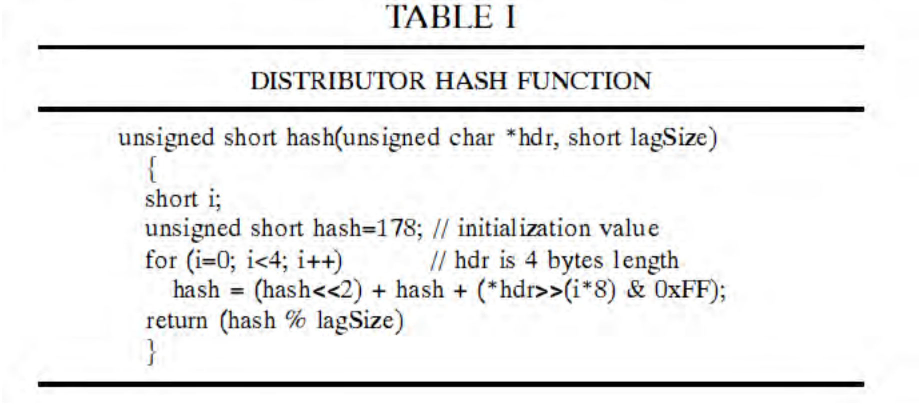
No.	’740 Patent Claim 11	Hilla
		337 increases as data packets are added to a queue from the ASIC 232. The value in the depth field 337 decreases as data packets are moved to a buffer for the port to which the queue is assigned, as described in more detail below. In some embodiments, the depth fields 337 are stored separately from the queues 235, e.g., in a vector of depth values with one element for each queue 332.”) Hilla at [0064] (“In some embodiments the mapping table 350 is in the same memory block as the port buffers 340, or flow control status bits 342, or both. In some embodiments, the mapping table is in a different memory block. For example, in some embodiments, the port buffers 340 are on ports bank 236, and both the flow control status table 344 and the mapping table 350 are on DLB block 240. In some embodi-ments, some fields are stored in different data structures.”) Hilla at [0067] (“In step 412 a current flow ID is determined for the current data packet. Any method known in the art at the time the method 400 is implemented may be used. In some embodiments, step 412 includes determining and combining the bit values in one or more fields in the header portions of one or more protocols. In some embodiments, step 412 involves receiving the current flow ID from another com-ponent, e.g., from switching fabric block 220, or switching ASIC 232 or traffic manager block 234. In some embodi-ments, the flow ID is the same as the particular queue 332 where the data packet is stored by the traffic manager 234. In some embodiments in which packet sequence within a data flow is not important and commit depth is not used to select ports, step 412 is omitted.”) Hilla at [0068] (“In step 420, state data about one or more of the egress ports in a link bundle is received. Any method for receiving data may be employed, including those described above for step 410. For example, in some embodiments, one or more of the status codes in the flow control status bits 342 are retrieved from a known memory block on ports bark 236. In other embodiments, other kinds of state data that indicate the physical status of one or more egress ports are received. In some embodiments, a status update process 243 stores data in a common area of memory to be retrieved by the port selection process 242. In some embodiments, step 420 is omitted.”)

No.	’740 Patent Claim 11	Hilla
		Hilla at [0073] (“Any method may be used to determine the entry in the mapping table associated with the current flow ID. For purposes of illustration, it is assumed that the mapping table 350 includes 1024 records, to distinguish up to 1024 differ-ent flow IDs. In some embodiments, such as embodiments using contents addressable memory (CAM), the flow ID determined in step 412 is compared to each entry in table 350 until a match is found. In other embodiments using CAM, the records 352 in mapping table 350 are sorted by flow ID and a binary search is made of the table until the record with the matching flow ID is found. In a preferred embodiment, at least a portion of the flow ID is input to a hash function, which produces a hash value between 0 and 1023. The queue 332 and record 352 at the position of the hash value+! is then the queue 332 and record 352 associ-ated with the current flow ID. For purposes of illustration, it is assumed that the flow ID of the current data packet hashes to a hash value such that the record 352 at the position of the hash value+ 1 is record 352b.”) Hilla at [0116] (“In step 540, the traffic manager 234 sends a data packet from a data queue for a flow ID that does not hash to the hash value of a heavily used link, e.g., hash values 1 and 2. For example, the traffic manager 234 sends a data packet from a queue for a flow ID that hashes to the values 3, 4 or 5. If no packet data resides in the data queue for these flow ID's, then no packet is sent and processing continues at step 510. The flow control status bits associated with these hash values indicate the associated port buffers are hungry or starving, as shown in Table 1. In some embodiments, the traffic manager 234 sends a data packet from a queue for a flow ID that hashes to the value 5 because the flow control status bits associated with hash value 5 indicate the associ-ated port buffer is starving. If no packet data resides in the data queue for flow ID hashed to value 5, then a packet may be chosen for flow IDs that hash to values 3 or 4 (whose associated port buffers are hungry).”) Hilla at [0121] (“Computer system 600 includes a communication mechanism such as a bus 610 for passing information between other internal and external components of the computer system 600. Information is represented as physical signals of a measurable phenomenon, typically electric voltages, but including, in other embodiments, such phe-nomena as magnetic, electromagnetic, pressure, chemical, molecular atomic and quantum interactions. For example, north and south magnetic fields, or a zero and non zero electric voltage,

No.	’740 Patent Claim 11	Hilla
		represent two states (0, 1) of a binary digit (bit). A sequence of binary digits constitutes digital data that is used to represent a number or code for a character. A bus 610 includes many parallel conductors of information so that information is transferred quickly among devices coupled to the bus 610. One or more processors 602 for processing information are coupled with the bus 610. A processor 602 performs a set of operations on information. The set of operations include bringing information in from the bus 610 and placing information on the bus 610. The set of opera-tions also typically include comparing two or more units of information, shifting positions of units of information, and combining two or more units of information, such as by addition or multiplication. A sequence of operations to be executed by the processor 602 constitute computer instruc-tions.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Hilla in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 11 of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses distributing data frames over physical links and traces based on a division operation in the hash function involving specific byte lengths of the frame information. Bruckman at Figure 2 (annotated)

No.	’740 Patent Claim 11	Hilla
		 FIG. 2 second level line cards first level links communication network traces 22 32 MAIN 42 CONTROL CAC TM 44 46 40 SW 54 AGGREGATOR COLLECT DISTRIB 56 58 52 52 34 LC1 50 50 30 30 36 L_1 L_2 34 LC2 50 50 30 30 36 L_3 L_4 Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:

No.	’740 Patent Claim 11	Hilla

		
		TABLE I
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		”)
		
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		
		As another example, Solomon discloses using a subset of bits to encode for the selected physical port, involving specific byte lengths of the frame information.
		
		Solomon at [0054] (“Having selected a physical port, RSVP-TE proces-sor 30 of switch A now generates a suitable MPLS label, at a label generation step 64. The preceding node upstream of switch A will subsequently attach this MPLS label to all MPLS packets transmitted through tunnel 28 to switch A. The label is assigned, in conjunction with the mapping function of mapper 34, so as to ensure that all MPLS packets carrying this label are switched through the physical port that was selected for this tunnel at step 62. For this purpose, RSVP-TE processor 30 of switch A dedicates a sub-set of the bits of MPLS label 52 to encode the serial number of the selected physical port. For example, the four least-signifi-cant bits of MPLS label 52 may be used for encoding the selected port number. This configuration is suitable for representing LAG groups having up to 16 physical ports (N<16). The remaining bits of MPLS label 52 may be chosen at random or using any suitable method known in the art.”)

No.	’740 Patent Claim 11	Hilla
		Solomon at [0056] (“Mapper 34 of switch A maps the received packets belonging to tunnel 28 to the selected physical Ethernet port at a mapping step 70. For this purpose, mapper 34 extracts the MPLS label from each received packet and decodes the selected physical port number from the dedicated sub-set of bits, such as the four LSB, as described in step 64 above. The decoded value is used for mapping the packet to the selected physical port, which was allocated by the CAC processor at step 62 above. In the four-bit example described above, the mapping function may be written explicitly as: Selected port number=((MPLS label) and (0x0000F)), wherein "and" denotes the "bitwise and" operator.”)

No.	’740 Patent Claim 12	Hilla
12	The method according to claim 1, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port	Hilla discloses the method according to claim 1, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port. For example, Hilla discloses data packets with information such as headers, that includes the source of the packet, its destination, the length of the payload, and other properties used by the protocol. *See supra* Claim 1 Hilla at [0006] (“Communications between nodes are typically effected by exchanging discrete packets of data. Each packet typically comprises 1] header information associated with a particular protocol, and 2] payload information that follows the header information and contains information to be processed, often independently of that particular protocol. In some protocols, the packet includes 3] trailer information following the payload and indicating the end of the payload information. The header includes information such as the source of the packet, its destination, the length of the payload, and other properties used by the protocol.

No.	’740 Patent Claim 12	Hilla
	and a destination TCP port.	Often, the data in the payload for the particular protocol includes a header and payload for a different protocol associated with a different, higher layer of the OSI Reference Model. The higher layer protocol is said to be encapsulated in the lower layer protocol. The headers included in a packet traversing multiple heterogeneous networks, such as the Internet, typi-cally include a physical (layer 1) header, a data-link (layer 2) header, an internetwork (layer 3) header and a transport (layer 4) header, as defined by the Open Systems Intercon-nection (OSI) Reference Model.”) Hilla at [0008] (“Routers and switches are network devices that determine which communication link or links to employ to support the progress of packets through the network. For example, Ethernet switches forward packets according to the Ethernet protocol. Some current routers implement sophis-ticated algorithms that provide high performance forwarding of packets based on combining layer 2 and layer 3 header information, or some other combination. For example, instead of making forwarding decisions separately on each packet in a stream of related packets (called a "packet flow" or simply a "flow"), such as a stream directed from the same source node to the same destination node, these routers identify the packet flow from a unique signature derived from the layer 2 or layer 3 header information and forward each member of the flow according to the same decision made for the first packet in the flow.”) Hilla at [0012] (“A load-balancing process is used on the sending network node of the pair connected by a bundle of commu-nication links for the purpose of determining which com-munication link to use for sending one or more data packets to the receiving network node of the pair. Current balancing algorithms use a fixed mapping to associate data packets with a specific port in a set of ports connected to the communication links in the bundle. Typically, information in a header portion of a data packet is used to derive a value that is associated with one port of the set. The algorithm is designed to generate a value in a range of values that are associated with the full set of ports. Thus data packets directed to the receiving node are distributed over all com-munication links in the bundle by the load balancing pro-cess. Many load-balancing processes are designed so that all data packets in the same data flow are sent through the same port.”)

No.	’740 Patent Claim 12	Hilla
		Hilla at [0014] (“Typically, the fixed-mapping takes several bits from one or more fields in layer 2 or layer 3 headers, or both, and inputs those bits to a hash function that produces an output with a certain number of bits. The output is then used directly or indirectly to select a port among the set connected to the bundle of communication links. By judicious choice of the fields, data packets from the same flow may be mapped to the same port.) Hilla at [0015] (“For example, if there are eight communication links in a bundle, some fixed-mapping load-balancing pro-cesses map different data packets to one of eight values, such as by using a hash function with a three-bit output. Three bits represent eight different values (0 to 7) which are associated with the eight different ports in the set connected to the eight communication links. While such an approach may cause data packets with similar values in their layer 2 and layer 3 headers to be directed to different ports of the set, there is no guarantee that the process will distribute traffic uniformly across the set of ports. For example, a disproportionate number of data packets might be mapped to the value 5. Thus some ports may still become overused, causing a reduction in the effective bandwidth.”) Hilla at [0039] (“An ingress line card (e.g., 210a) receives a data packet, inspects the appropriate header fields to make a forwarding decision, and forwards the packet toward the egress line card via the switching fabric component 220. The switch fabric component 220 forwards the packet to the appropriate egress line card 230. Multiple ingress line cards can forward traffic toward a single egress line card 230. In this example, the link bundle 130 is capable of transmitting more packets simultaneously because of the increased band-width achieved by bundling. Some ingress line cards (e.g., 210c) are connected to a link bundle with multiple commu-nication links. All communication links in the link bundle are considered a single ingress logical link. In the illustrated embodiment, the five communication links connected to ingress line card 210c are connected to network node 110b. The ingress line card 210c handles incoming traffic on the link bundle 130, while the egress line card 230 handles the outbound traffic on the link bundle 130.”) Hilla at [0058] (“The mapping table 350 includes multiple records, including record 352a, 352b, and other records indicated by ellipsis 353, collectively referenced hereinafter as records 352. In the illustrated embodiment, each record 352 includes a flow identification

No.	’740 Patent Claim 12	Hilla
		(flow ID) field (e.g., 354a, 354b, collectively referenced hereinafter as flow ID field 354). A particular flow ID corresponds to a particular stream of related data packets as determined, for example, by exam-ining source or destination fields, or both, in the headers of one or more protocols. Many methods for determining a flow ID are known and practiced in the art. In some embodiments, the flow ID field 354 is omitted, and the position of the record 352 in the mapping table 350 is related to a flow ID. For example, in some embodiments a flow ID is input to a hash function that outputs a particular record number in table 350 that corresponds to the flow ID input to the hash function. Each record 352 in the mapping table 350 corresponds to a queue 332 in the data flow queues 235.”) Hilla at [0067] (“In step 412 a current flow ID is determined for the current data packet. Any method known in the art at the time the method 400 is implemented may be used. In some embodiments, step 412 includes determining and combining the bit values in one or more fields in the header portions of one or more protocols. In some embodiments, step 412 involves receiving the current flow ID from another com-ponent, e.g., from switching fabric block 220, or switching ASIC 232 or traffic manager block 234. In some embodi-ments, the flow ID is the same as the particular queue 332 where the data packet is stored by the traffic manager 234. In some embodiments in which packet sequence within a data flow is not important and commit depth is not used to select ports, step 412 is omitted.”)

No.	’740 Patent Claim 13	Hilla
13[preamble]	A method for communication, comprising:	Hilla discloses a method for communication. *See supra at* 1[preamble].
13[a]	coupling a network node to one or more interface modules using a first group of	Hilla discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].

No.	’740 Patent Claim 13	Hilla
	first physical links arranged in parallel;	
13[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Hilla discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra at* 1[c].
13[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	Hilla discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra at* 1[e].
13[d]	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	Hilla discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. *See supra at* 1[f].
13[e]	sending the data frame over the selected first and second physical links,	Hilla discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].
13[f]	coupling the network node to the one or	Hilla discloses coupling the network node to the one or more interface modules. *See supra at* 1[a].

No.	’740 Patent Claim 13	Hilla
	more interface modules and	
13[g]	coupling each of the one or more interface modules to the communication network comprising	Hilla discloses coupling each of the one or more interface modules to the communication network. *See supra at* 1[c].
13[h]	specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node, and	Hilla discloses specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node. For example, Hilla discloses connecting line cards to the network, in which the bandwidth of a particular link is limited by its specific requirements and characteristics. A person of ordinary skill in the art would understand that specific bandwidth limitations of a communication service may be a committed information rate (CIR), a peak information rate (PIR) or an excess information rate (EIR). Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Hilla is found to not meet this limitation, coupling each of the one or more interface modules to the communication network comprising specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node would have been obvious to a person having ordinary skill in the art, as explained below. Hilla at [0009] (“The number of bits that are carried over a commu-nication link in a unit time is called the speed or bandwidth of the communication link. The bandwidth of a particular link is limited by the physical characteristics of the cable and the port on each network node to which the cable is connected. As used here, a port is a physical interface on a network device that is, or can be, connected to a cable to serve as a communication link with a port on another network device. For example, three types of widely used Ethernet links have three different bandwidths of 100 Mega-bits per second (Mbps, where 1 Megabit=l06 binary digits called bits), 1 Gigabit per second (Gbps, where 1 Gigabit= 109 bits), or 10 Gbps. These three

No.	’740 Patent Claim 13	Hilla
		bandwidths are termed Fast Ethernet, Gigabit Ethernet and 10 Gigabit Ethernet, respectively.”) Hilla at [0010] (“In some circumstances, the bandwidth needed between two nodes does not match one of the readily available bandwidths. In such circumstances, some net-works bundle multiple communication links between a pair of network nodes. For example, if network traffic between a particular server and an Ethernet switch in an office building needs bandwidth up to 500 Mbps, then it might be more cost-effective to connect five Fast Ethernet ports on each device rather than to install a Gigabit Ethernet port on each device and string a Gigabit cable in the walls between them. The five Fast Ethernet links in this example constitute a bundle of communication links. Similarly, if network traffic needs exceed 10 Gbps, then these needs can be met with a bundle of two or more 10 Gigabit Ethernet communication links. Link Aggregation Control Protocol (LACP) is part of an IEEE specification (802.3ad) that allows several physical ports to be bundled together to form a single logical channel. LACP allows a switch to negotiate an automatic bundle by sending LACP packets to the peer.”) Hilla at [0039] (“An ingress line card (e.g., 210a) receives a data packet, inspects the appropriate header fields to make a forwarding decision, and forwards the packet toward the egress line card via the switching fabric component 220. The switch fabric component 220 forwards the packet to the appropriate egress line card 230. Multiple ingress line cards can forward traffic toward a single egress line card 230. In this example, the link bundle 130 is capable of transmitting more packets simultaneously because of the increased band-width achieved by bundling. Some ingress line cards (e.g., 210c) are connected to a link bundle with multiple commu-nication links. All communication links in the link bundle are considered a single ingress logical link. In the illustrated embodiment, the five communication links connected to ingress line card 210c are connected to network node 110b. The ingress line card 210c handles incoming traffic on the link bundle 130, while the egress line card 230 handles the outbound traffic on the link bundle 130.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Hilla in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in

No.	’740 Patent Claim 13	Hilla
		further combination with (2) each (individually, as well as one or more together) of the references identified in element 13[h] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses specifying certain committed, excess, and guaranteed bandwidths, including CIR, EIR, and PIR, respectively. Bruckman at [0013] (“Service level agreements between network service providers and customers commonly specify a certain com-mitted bandwidth, or committed information rate (CIR), which the service provider guarantees to provide to the customer at all times, regardless of bandwidth stress on the network. Additionally or alternatively, the agreement may specify an excess bandwidth, which is available to the customer when network traffic permits. The excess band-width is typically used by customers for lower-priority services, which do not require committed bandwidth. The network service provider may guarantee the customer a certain minimum excess bandwidth, or excess information rate (EIR), in order to avoid starvation of such services in case of bandwidth stress. In general, the bandwidth guaran-teed by a service provider, referred to as the peak informa-tion rate (PIR), may include either CIR, or EIR, or both CIR and EIR (in which case PIR=CIR+EIR). The term "guaran-teed bandwidth," as used in the context of the present patent application and in the claims, includes all these types of guaranteed bandwidth.”) As another example, Solomon discloses a service property of a guaranteed bandwidth, sometimes denoted as CIR-Committed Information Rate and PIR-Peak Information Rate. Solomon at [0023] (“In another embodiment, establishing the path includes receiving an indication of a requested service property of the flow, and selecting the port includes assign-ing the port to the flow so as to comply with the requested service property. In a disclosed embodiment, the requested service property includes at least one of a guaranteed bandwidth, a peak bandwidth and a class-of-service. Addi-tionally or alternatively, assigning the port includes selecting the port having a maximum available bandwidth out of the plurality of aggregated ports. Further additionally or alter-natively, assigning the port includes

No.	’740 Patent Claim 13	Hilla
		selecting the port hav-ing a minimum available bandwidth out of the plurality of aggregated ports, which is still greater than or equal to the guaranteed bandwidth.”) Solomon at [0050] (“The method of FIG. 3 begins when the preceding node asks to establish a part of tunnel 28 (comprising one or more hops) for sending MPLS packets to MPLS/LAG switch 26 A. The preceding node requests and then receives the MPLS label, which it will subsequently attach to all packets that are sent to MPLS/LAG switch 26 labeledA. The preceding node sends downstream an RSVP-TE PATH mes-sage augmented with a LABEL_REQUEST object, as defined by RSVP-TE, to MPLS/LAG switch A, at a label requesting step 60. The PATH message typically comprises information regarding service properties that are requested for tunnel 28. The service properties may comprise a guar-anteed bandwidth (sometimes denoted CIR-Committed Information Rate) and a peak bandwidth (sometimes denoted PIR-Peak Information Rate), as well as a requested CoS (Class of Service-a measure of packet priority).”)
13[i]	allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements.	Hilla discloses allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements. For examples, Hilla discloses distributing data traffic of a communication service over the ports and communication connectors based on bandwidth requirements and physical characteristics. Hilla at [0009] (“The number of bits that are carried over a commu-nication link in a unit time is called the speed or bandwidth of the communication link. The bandwidth of a particular link is limited by the physical characteristics of the cable and the port on each network node to which the cable is connected. As used here, a port is a physical interface on a network device that is, or can be, connected to a cable to serve as a communication link with a port on another network device. For example, three types of widely used Ethernet links have three different bandwidths of 100 Mega-bits per second (Mbps, where 1 Megabit=l06 binary digits called bits), 1 Gigabit per second (Gbps, where 1 Gigabit= 109 bits), or 10 Gbps. These three bandwidths are termed Fast Ethernet, Gigabit Ethernet and 10 Gigabit Ethernet, respectively.”)

No.	’740 Patent Claim 13	Hilla
		Hilla at [0010] (“In some circumstances, the bandwidth needed between two nodes does not match one of the readily available bandwidths. In such circumstances, some net-works bundle multiple communication links between a pair of network nodes. For example, if network traffic between a particular server and an Ethernet switch in an office building needs bandwidth up to 500 Mbps, then it might be more cost-effective to connect five Fast Ethernet ports on each device rather than to install a Gigabit Ethernet port on each device and string a Gigabit cable in the walls between them. The five Fast Ethernet links in this example constitute a bundle of communication links. Similarly, if network traffic needs exceed 10 Gbps, then these needs can be met with a bundle of two or more 10 Gigabit Ethernet communication links. Link Aggregation Control Protocol (LACP) is part of an IEEE specification (802.3ad) that allows several physical ports to be bundled together to form a single logical channel. LACP allows a switch to negotiate an automatic bundle by sending LACP packets to the peer.”) Hilla at [0039] (“An ingress line card (e.g., 210a) receives a data packet, inspects the appropriate header fields to make a forwarding decision, and forwards the packet toward the egress line card via the switching fabric component 220. The switch fabric component 220 forwards the packet to the appropriate egress line card 230. Multiple ingress line cards can forward traffic toward a single egress line card 230. In this example, the link bundle 130 is capable of transmitting more packets simultaneously because of the increased band-width achieved by bundling. Some ingress line cards (e.g., 210c) are connected to a link bundle with multiple commu-nication links. All communication links in the link bundle are considered a single ingress logical link. In the illustrated embodiment, the five communication links connected to ingress line card 210c are connected to network node 110b. The ingress line card 210c handles incoming traffic on the link bundle 130, while the egress line card 230 handles the outbound traffic on the link bundle 130.”)

No.	’740 Patent Claim 14	Hilla
14[preamble]	A method for connecting user ports to a communication network, comprising:	Hilla discloses a method for connecting user ports to a communication network. For example, Hilla discloses techniques for connecting ports to a network using communication links of a network device. Hilla at Abstract (“Techniques for distributing data packets over a network link bundle include storing an output data packet in a data flow queue based on a flow identification associated with the output data packet. The flow identification indicates a set of one or more data packets, including the output data packet, which are to be sent in the same sequence as received. State data is also received. The state data indicates a physical status of a first port of multiple active egress ports that are connected to a corresponding bundle of communication links with one particular network device. A particular data flow queue is determined based at least in part on the state data. A next data packet is directed from the particular data flow queue to a second port of the active egress ports. These techniques allow a more efficient use of a network link bundle.”) Hilla at [0006] (“Communications between nodes are typically effected by exchanging discrete packets of data. Each packet typically comprises 1] header information associated with a particular protocol, and 2] payload information that follows the header information and contains information to be processed, often independently of that particular protocol. In some protocols, the packet includes 3] trailer information following the payload and indicating the end of the payload information. The header includes information such as the source of the packet, its destination, the length of the payload, and other properties used by the protocol. Often, the data in the payload for the particular protocol includes a header and payload for a different protocol associated with a different, higher layer of the OSI Reference Model. The higher layer protocol is said to be encapsulated in the lower layer protocol. The headers included in a packet traversing multiple heterogeneous networks, such as the Internet, typi-cally include a physical (layer 1) header, a data-link (layer 2) header, an internetwork (layer 3) header and a transport (layer 4) header, as defined by the Open Systems Intercon-nection (OSI) Reference Model.”)

No.	’740 Patent Claim 14	Hilla
		Hilla at [0033] (“A method and apparatus are described for dynamic balancing of data packet traffic loads over a link bundle in a network. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present inven-tion. It will be apparent, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.”) Hilla at [0034] (“The invention is described in the following sections in the context of an Ethernet 802.3ad link bundle between two routers in the core or backbone of an enterprise network, but the invention is not limited to this context. In other embodiments, the link bundle is at a network edge, for example connecting an end node performing as a host for a high-throughput network server process of a client-server application. In other embodiments, different communication links are bundled between two network nodes, such as Packet over SONET (POS), and High-level Data Link Control (HDLC) links, among others.”) Hilla at [0036] (“FIG. 1 illustrates a link bundle 130 that includes five communication links between intermediate network node 110c and network node 110b. Network node 110b may be an intermediate network node or an end node that is connected both to intermediate network node 11 Oc and subnetwork 102a. For purposes of illustration, intermediate network node 110c is connected by a bundle 130 of five communication links to network node 110b and thereby to subnetwork 102a and end node 11 Oa. Similarly, intermediate network node 110c is connected by three unbundled com-munication links to network nodes 110d, ll0e, 110/, respec-tively, and thence to sub-network 102b and end node 110g. In other embodiments an intermediate network node 11 Oc may be connected to more or fewer network nodes with more or fewer links in each of one or more link bundles as part of a network with the same or more end nodes.”) Hilla at [0065] (“FIG. 4A is a flow diagram that illustrates a method 400 for dynamically balancing data packet traffic load on a link bundle, according to an embodiment. Although steps are shown in FIG. 4A and subsequent flow diagrams 4B and 5 in a particular order for purposes of illustration, in other embodiments one or more steps are performed in a different

No.	’740 Patent Claim 14	Hilla
		order or overlapping in time or are omitted, or changed in some combination of ways. For example, in a preferred embodiment shown in FIG. 4B, step 490 to advance age counters is performed in a separate aging process 244 on DLB block 240, and thus overlaps in time any of the other steps depicted in FIG. 4A.”)
14[a]	coupling the user ports to one or more user interface modules;	Hilla discloses coupling the user ports to one or more user interface modules. *See supra at* 1[a].
14[b]	coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel,	Hilla discloses coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel. *See supra at* 1[c], 3.
14[c]	at least one of said backplane traces being bi-directional and operative to communicate in both an upstream direction and a downstream direction;	Hilla discloses at least one of said backplane traces being bi-directional and operative to communicate in both an upstream direction and a downstream direction. *See supra at* 14[b], 1[d].
14[d]	receiving data frames sent between the user ports and the communication network, the data frames having	Hilla discloses receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes. *See supra at* 14[a], 1[e].

No.	’740 Patent Claim 14	Hilla
	respective frame attributes;	
14[e]	for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces; and	Hilla discloses for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces. *See supra at* 14[b], 1[f].
14[f]	sending the data frame over the selected backplane trace;	Hilla discloses sending the data frame over the selected backplane trace. *See supra at* 14[e], 1[g].
14[g]	said sending comprising communicating along said at least one of said backplane traces.	Hilla discloses said sending comprising communicating along said at least one of said backplane traces. *See supra at* 14[f], 1[h].

No.	’740 Patent Claim 15	Hilla
15[preamble]	A method for connecting user ports to a communication network, comprising:	Hilla discloses a method for connecting user ports to a communication network. *See supra at* 14[preamble].
15[a]	coupling the user ports to one or more user interface modules;	Hilla discloses coupling the user ports to one or more user interface modules. *See supra at* 14[a].
15[b]	coupling each user interface module to the communication	Hilla discloses coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel.

No.	’740 Patent Claim 15	Hilla
	network via a backplane using two or more backplane traces arranged in parallel;	*See supra at* 14[b].
15[c]	receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes;	Hilla discloses receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes. *See supra at* 14[d].
15[d]	for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces; and	Hilla discloses for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces. *See supra at* 14[e].
15[e]	sending the data frame over the selected backplane trace,	Hilla discloses sending the data frame over the selected backplane trace. *See supra at* 14[f].
15[f]	at least some of the backplane traces being aggregated into an Ethernet link aggregation (LAG) group.	Hilla discloses at least some of the backplane traces being aggregated into an Ethernet link aggregation (LAG) group. *See supra at* 15[e], 4[f], 3.

No.	’740 Patent Claim 16	Hilla
16	The method according to claim 14, wherein selecting the backplane trace comprises applying a hashing function to the at least one of the frame attributes.	Hilla discloses the method according to claim 14, wherein selecting the backplane trace comprises applying a hashing function to the at least one of the frame attributes. *See supra* at 14, 9, 8.

No.	’740 Patent Claim 17	Hilla
17[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Hilla discloses apparatus for connecting a network node with a communication network. For example, Hilla discloses a system for connecting network nodes such as a network device or computer system to a network. Hilla at Abstract (“Techniques for distributing data packets over a network link bundle include storing an output data packet in a data flow queue based on a flow identification associated with the output data packet. The flow identification indicates a set of one or more data packets, including the output data packet, which are to be sent in the same sequence as received. State data is also received. The state data indicates a physical status of a first port of multiple active egress ports that are connected to a corresponding bundle of communication links with one particular network device. A particular data flow queue is determined based at least in part on the state data. A next data packet is directed from the particular data flow queue to a second port of the active egress ports. These techniques allow a more efficient use of a network link bundle.”) Hilla at [0006] (“Communications between nodes are typically effected by exchanging discrete packets of data. Each packet typically comprises 1] header information associated with a particular protocol, and 2] payload information that follows the header information and contains information to be processed, often independently of that particular protocol. In some protocols, the packet includes 3] trailer information following the payload and indicating the end of the payload information. The header includes information such as the source of the

No.	’740 Patent Claim 17	Hilla
		packet, its destination, the length of the payload, and other properties used by the protocol. Often, the data in the payload for the particular protocol includes a header and payload for a different protocol associated with a different, higher layer of the OSI Reference Model. The higher layer protocol is said to be encapsulated in the lower layer protocol. The headers included in a packet traversing multiple heterogeneous networks, such as the Internet, typi-cally include a physical (layer 1) header, a data-link (layer 2) header, an internetwork (layer 3) header and a transport (layer 4) header, as defined by the Open Systems Intercon-nection (OSI) Reference Model.”) Hilla at [0033] (“A method and apparatus are described for dynamic balancing of data packet traffic loads over a link bundle in a network. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present inven-tion. It will be apparent, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.”) Hilla at [0034] (“The invention is described in the following sections in the context of an Ethernet 802.3ad link bundle between two routers in the core or backbone of an enterprise network, but the invention is not limited to this context. In other embodiments, the link bundle is at a network edge, for example connecting an end node performing as a host for a high-throughput network server process of a client-server application. In other embodiments, different communication links are bundled between two network nodes, such as Packet over SONET (POS), and High-level Data Link Control (HDLC) links, among others.”) Hilla at [0035] (“FIG. 1 is a block diagram that illustrates a network with a link bundle, according to an embodiment. A computer network is a geographically distributed collection of inter-connected sub-networks (e.g, sub-networks 102a, 102b, collectively referenced hereinafter as sub-networks 102) for transporting data between network nodes (e.g., network nodes 110a, 110b, 110c, 110d, ll0e, 110/, 110g, collectively referenced hereinafter as network nodes 110). A local area network (LAN) is an example of such a sub-network 102. The network's topology is defined by an arrangement of end nodes that communicate with one

No.	’740 Patent Claim 17	Hilla
		another, typically through one or more intermediate network nodes such as a router or switch, which facilitates routing data between end nodes. As used herein, an end node is a node that is configured to originate or terminate communications over the network. In contrast, an intermediate network node facilitates the pas-sage of data between end nodes. In FIG. 1, the network nodes 110 include both end nodes (e.g., nodes 110a, 110g) and intermediate nodes (e.g., network node 110c). Each sub-network 102 includes zero or more intermediate net-work nodes.”) Hilla at [0036] (“FIG. 1 illustrates a link bundle 130 that includes five communication links between intermediate network node 110c and network node 110b. Network node 110b may be an intermediate network node or an end node that is connected both to intermediate network node 11 Oc and subnetwork 102a. For purposes of illustration, intermediate network node 110c is connected by a bundle 130 of five communication links to network node 110b and thereby to subnetwork 102a and end node 11 Oa. Similarly, intermediate network node 110c is connected by three unbundled com-munication links to network nodes 110d, ll0e, 110/, respec-tively, and thence to sub-network 102b and end node 110g. In other embodiments an intermediate network node 11 Oc may be connected to more or fewer network nodes with more or fewer links in each of one or more link bundles as part of a network with the same or more end nodes.”) Hilla at [0065] (“FIG. 4A is a flow diagram that illustrates a method 400 for dynamically balancing data packet traffic load on a link bundle, according to an embodiment. Although steps are shown in FIG. 4A and subsequent flow diagrams 4B and 5 in a particular order for purposes of illustration, in other embodiments one or more steps are performed in a different order or overlapping in time or are omitted, or changed in some combination of ways. For example, in a preferred embodiment shown in FIG. 4B, step 490 to advance age counters is performed in a separate aging process 244 on DLB block 240, and thus overlaps in time any of the other steps depicted in FIG. 4A.”)
17[a]	one or more interface modules, which are arranged to process	Hilla discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network.

No.	’740 Patent Claim 17	Hilla
	data frames having frame attributes sent between the network node and the communication network,	For example, Hilla discloses line cards that inspect appropriate header fields to make forwarding decisions for data packets sent between a network node and a network. Hilla at [0035] (“FIG. 1 is a block diagram that illustrates a network with a link bundle, according to an embodiment. A computer network is a geographically distributed collection of inter-connected sub-networks (e.g, sub-networks 102a, 102b, collectively referenced hereinafter as sub-networks 102) for transporting data between network nodes (e.g., network nodes 110a, 110b, 110c, 110d, ll0e, 110/, 110g, collectively referenced hereinafter as network nodes 110). A local area network (LAN) is an example of such a sub-network 102. The network's topology is defined by an arrangement of end nodes that communicate with one another, typically through one or more intermediate network nodes such as a router or switch, which facilitates routing data between end nodes. As used herein, an end node is a node that is configured to originate or terminate communications over the network. In contrast, an intermediate network node facilitates the pas-sage of data between end nodes. In FIG. 1, the network nodes 110 include both end nodes (e.g., nodes 110a, 110g) and intermediate nodes (e.g., network node 110c). Each sub-network 102 includes zero or more intermediate net-work nodes.”) Hilla at [0038] (“FIG. 2 is a block diagram that illustrates a link-bundle switching system 200 in a router, which can be used as intermediate network node 11 Oc, according to an embodi-ment. The switching system 200 includes a switching system component 202 connected to communication links, includ-ing link bundle 130. The switching system component 202 includes multiple ingress line cards, including cards 210a, 210b, 210c, for receiving inbound data packets, switching fabric component 220, and multiple egress line cards for sending outbound data packets, including egress line card 230 for link bundle 130.”) Hilla at [0039] (“An ingress line card (e.g., 210a) receives a data packet, inspects the appropriate header fields to make a forwarding decision, and forwards the packet toward the egress line card via the switching fabric component 220. The switch fabric component 220 forwards the packet to the appropriate egress line card 230. Multiple ingress line cards can forward traffic toward a single egress line card 230. In this example, the link bundle 130 is capable of transmitting more packets simultaneously because of the increased band-width

No.	’740 Patent Claim 17	Hilla
		achieved by bundling. Some ingress line cards (e.g., 210c) are connected to a link bundle with multiple commu-nication links. All communication links in the link bundle are considered a single ingress logical link. In the illustrated embodiment, the five communication links connected to ingress line card 210c are connected to network node 110b. The ingress line card 210c handles incoming traffic on the link bundle 130, while the egress line card 230 handles the outbound traffic on the link bundle 130.”) Hilla at [0040] (“The ingress line cards 210 determine an outbound physical or logical link to use to forward the data packet, for example using data in a routing table stored on the router, and sends information to the egress line card associated with the outbound physical or logical link. All communication links in the link bundle 130 are considered a single logical link. Information about data packets to be output on the link bundle 130 is sent to the egress line card 230.”) Hilla at [0041] (“The egress line card 230 includes a switching application specific integrated circuit (ASIC) 232, a traffic manager block 234, a bank 236 of physical ports, at least some of which are connected to the communication links in the link bundle. In the illustrated embodiment, a dynamic load balancing (DLB) block 240 is included in the traffic manager 234. In some embodiments, the switching ASIC 232, traffic manager 234 and ports bank 236 are standard components of conventional egress cards for link bundles, and the DLB block 240 is external to the traffic manager 234.”)
17[b]	at least one of said interface modules being operative to communicate in both an upstream direction and a downstream direction;	Hilla discloses at least one of said interface modules being operative to communicate in both an upstream direction and a downstream direction. For example, Hilla discloses line cards that are capable of forwarding data packets in inbound and outbound directions. Hilla at [0035] (“FIG. 1 is a block diagram that illustrates a network with a link bundle, according to an embodiment. A computer network is a geographically distributed collection of inter-connected sub-networks (e.g, sub-networks 102a, 102b, collectively referenced hereinafter as sub-networks 102) for transporting data between network nodes (e.g., network

No.	’740 Patent Claim 17	Hilla
		nodes 110a, 110b, 110c, 110d, ll0e, 110/, 110g, collectively referenced hereinafter as network nodes 110). A local area network (LAN) is an example of such a sub-network 102. The network's topology is defined by an arrangement of end nodes that communicate with one another, typically through one or more intermediate network nodes such as a router or switch, which facilitates routing data between end nodes. As used herein, an end node is a node that is configured to originate or terminate communications over the network. In contrast, an intermediate network node facilitates the pas-sage of data between end nodes. In FIG. 1, the network nodes 110 include both end nodes (e.g., nodes 110a, 110g) and intermediate nodes (e.g., network node 110c). Each sub-network 102 includes zero or more intermediate net-work nodes.”) Hilla at [0036] (“FIG. 1 illustrates a link bundle 130 that includes five communication links between intermediate network node 110c and network node 110b. Network node 110b may be an intermediate network node or an end node that is connected both to intermediate network node 11 Oc and subnetwork 102a. For purposes of illustration, intermediate network node 110c is connected by a bundle 130 of five communication links to network node 110b and thereby to subnetwork 102a and end node 11 Oa. Similarly, intermediate network node 110c is connected by three unbundled com-munication links to network nodes 110d, ll0e, 110/, respec-tively, and thence to sub-network 102b and end node 110g. In other embodiments an intermediate network node 11 Oc may be connected to more or fewer network nodes with more or fewer links in each of one or more link bundles as part of a network with the same or more end nodes.”) Hilla at [0038] (“FIG. 2 is a block diagram that illustrates a link-bundle switching system 200 in a router, which can be used as intermediate network node 11 Oc, according to an embodi-ment. The switching system 200 includes a switching system component 202 connected to communication links, includ-ing link bundle 130. The switching system component 202 includes multiple ingress line cards, including cards 210a, 210b, 210c, for receiving inbound data packets, switching fabric component 220, and multiple egress line cards for sending outbound data packets, including egress line card 230 for link bundle 130.”)

No.	’740 Patent Claim 17	Hilla
		Hilla at [0039] (“An ingress line card (e.g., 210a) receives a data packet, inspects the appropriate header fields to make a forwarding decision, and forwards the packet toward the egress line card via the switching fabric component 220. The switch fabric component 220 forwards the packet to the appropriate egress line card 230. Multiple ingress line cards can forward traffic toward a single egress line card 230. In this example, the link bundle 130 is capable of transmitting more packets simultaneously because of the increased band-width achieved by bundling. Some ingress line cards (e.g., 210c) are connected to a link bundle with multiple commu-nication links. All communication links in the link bundle are considered a single ingress logical link. In the illustrated embodiment, the five communication links connected to ingress line card 210c are connected to network node 110b. The ingress line card 210c handles incoming traffic on the link bundle 130, while the egress line card 230 handles the outbound traffic on the link bundle 130.”) Hilla at [0040] (“The ingress line cards 210 determine an outbound physical or logical link to use to forward the data packet, for example using data in a routing table stored on the router, and sends information to the egress line card associated with the outbound physical or logical link. All communication links in the link bundle 130 are considered a single logical link. Information about data packets to be output on the link bundle 130 is sent to the egress line card 230.”) Hilla at [0041] (“The egress line card 230 includes a switching application specific integrated circuit (ASIC) 232, a traffic manager block 234, a bank 236 of physical ports, at least some of which are connected to the communication links in the link bundle. In the illustrated embodiment, a dynamic load balancing (DLB) block 240 is included in the traffic manager 234. In some embodiments, the switching ASIC 232, traffic manager 234 and ports bank 236 are standard components of conventional egress cards for link bundles, and the DLB block 240 is external to the traffic manager 234.”) Hilla at [0044] (“The ports bank 236 is a bank of ports which includes multiple ports that are treated by a router as a single logical port for purposes of communicating through link bundle 130 as a single logical link. Each communication link in the link bundle 130 is connected to a different port of the ports bank 236. In the illustrated embodiment, ports bank 236 includes

No.	’740 Patent Claim 17	Hilla
		eight ports 238a, 238b, 238c, 238d, 238e, 238/, 238g, 238h (collectively referenced hereinafter as ports 238). For purposes of illustration, it is assumed that these eight ports are identified by a three bit code that represents the eight values from O through 7 to correspond to ports 238a through 238h, respectively. Each three bit code is called a port number (port #), herein. Only five of the ports (e.g., 238b, 238c, 238d, 238e, 238/, identified by code values 1 through 5, respectively) are connected to communication links of the link bundle 130 in the illustrated embodiment. These are called active ports. The remaining three ports (e.g., 238a, 238g, 238h, identified by code values 0, 6, 7, respectively) are inactive ports. In some embodiments, a port becomes inactive by virtue of a communication link that is attached to the port going down. When a port goes down, it is no longer eligible to send traffic and the link is removed from the route table. In an example embodiment, the hard-ware described in more detail below would detect the link's inability to forward traffic and dynamically shift traffic over to an active link. Such a hardware shift can occur much faster than waiting for a routing table update.”)
17[c]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Hilla discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 1[a].
17[d]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Hilla discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 1[c].
17[e]	a control module, which is arranged to	Hilla discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least

No.	’740 Patent Claim 17	Hilla
	select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame;	one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 1[f].
17[f]	at least one of said first physical links and at least one of said second links being bi-directional links operative to communicate in both said upstream direction and said downstream direction.	Hilla discloses at least one of said first physical links and at least one of said second links being bi-directional links operative to communicate in both said upstream direction and said downstream direction. *See supra at* 1[b], 1[d].

No.	’740 Patent Claim 18	Hilla
18[a]	The apparatus according to claim 17, and comprising a backplane to which the one or more interface modules are coupled,	Hilla discloses the apparatus according to claim 17, and comprising a backplane to which the one or more interface modules are coupled. *See supra at* 3, 17.

No.	’740 Patent Claim 18	Hilla
18[b]	wherein the second physical links comprise backplane traces formed on the backplane.	Hilla discloses wherein the second physical links comprise back plane traces formed on the backplane. *See supra at* 3, 17.

No.	’740 Patent Claim 19	Hilla
19[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Hilla discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
19[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	Hilla discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].
19[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Hilla discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].
19[c]	a second group of second physical links	Hilla discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network.

No.	’740 Patent Claim 19	Hilla
	arranged in parallel so as to couple the one or more interface modules to the communication network; and	*See supra at* 17[d].
19[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,	Hilla discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].
19[e]	at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group.	Hilla discloses at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group. *See supra at* 4[f].

No.	’740 Patent Claim 20	Hilla
20[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Hilla discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
20[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	Hilla discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].
20[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Hilla discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].
20[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Hilla discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].
20[d]	a control module, which is arranged to select for each data	Hilla discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least

No.	’740 Patent Claim 20	Hilla
	frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,	one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].
20[e]	two or more of the first physical links being aggregated into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node.	Hilla discloses two or more of the first physical links being aggregated into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node. *See supra at* 19[e], 5[f].

No.	’740 Patent Claim 21	Hilla
21	The apparatus according to claim 17, and comprising a multiplexer, which is arranged to perform at least one of multiplexing upstream data frames sent from	Hilla discloses the apparatus according to claim 17, and comprising a multiplexer, which is arranged to perform at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node. *See supra at* 6, 17.

No.	’740 Patent Claim 21	Hilla
	the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node.	

No.	’740 Patent Claim 22	Hilla
22	The apparatus according to claim 17, wherein the control module is arranged to balance a frame data rate among at least some of the first and second physical links.	Hilla discloses the apparatus according to claim 17, wherein the control module is arranged to balance a frame data rate among at least some of the first and second physical links. *See supra at* 7, 17.

No.	’740 Patent Claim 23	Hilla
23	The apparatus according to claim 17, wherein the control module is arranged to apply a mapping function to the at least one of the frame attributes so as to select the first and second physical links.	Hilla discloses the apparatus according to claim 17, wherein the control module is arranged to apply a mapping function to the at least one of the frame attributes so as to select the first and second physical links. *See supra at* 8, 17.

No.	’740 Patent Claim 24	Hilla
24	The apparatus according to claim 23, wherein the mapping function comprises a hashing function.	Hilla discloses the apparatus according to claim 23, wherein the mapping function comprises a hashing function. *See supra at* 9, 23.

No.	’740 Patent Claim 25	Hilla
25[a]	The apparatus according to claim 24, wherein the control module is arranged to determine a hashing size responsively to a number of at least some of the first and second physical links,	Hilla discloses the apparatus according to claim 24, wherein the control module is arranged to determine a hashing size responsively to a number of at least some of the first and second physical links. *See supra at* 10[a], 24.
25[b]	to apply the hashing function to the at least one of the frame attributes to produce a hashing key,	Hilla discloses to apply the hashing function to the at least one of the frame attributes to produce a hashing key. *See supra at* 10[b].
25[c]	to calculate a modulo of a division operation of the hashing key by the hashing size, and	Hilla discloses to calculate a modulo of a division operation of the hashing key by the hashing size. *See supra at* 10[c].
25[d]	to select the first and second physical links responsively to the modulo.	Hilla discloses to select the first and second physical links responsively to the modulo. *See supra at* 10[d].

No.	’740 Patent Claim 26	Hilla
26	The apparatus according to claim 25, wherein the control module is arranged to select the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo.	Hilla discloses the apparatus according to claim 25, wherein the control module is arranged to select the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo. *See supra at* 11, 25.

No.	’740 Patent Claim 27	Hilla
27	The apparatus according to claim 17, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port	Hilla discloses the apparatus according to claim 17, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port. *See supra at* 12, 17.

	and a destination TCP port.	

No.	’740 Patent Claim 28	Hilla
28[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Hilla discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
28[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	Hilla discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].
28[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Hilla discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].
28[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Hilla discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].

No.	’740 Patent Claim 28	Hilla
28[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,	Hilla discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].
28[e]	the communication network being arranged to provide a communication service to the network node,	Hilla discloses the communication network being arranged to provide a communication service to the network node. *See supra at* 2[b].
28[f]	the service having specified bandwidth requirements comprising at least one of a committed information rate (CR), a peak information rate (PIR) and an excess information rate (EIR), and	Hilla discloses the service having specified bandwidth requirements comprising at least one of a committed information rate (CR), a peak information rate (PIR) and an excess information rate (EIR). *See supra at* 13[i].

No.	’740 Patent Claim 28	Hilla
28[g]	the first and second groups of physical links being dimensioned to provide an allocated bandwidth for the communication service responsively to the band width requirements.	Hilla discloses the first and second groups of physical links being dimensioned to provide an allocated bandwidth for the communication service responsively to the band width requirements. *See supra at* 13[j].

No.	’740 Patent Claim 29	Hilla
29[preamble]	Apparatus for connecting user ports to a communication network, comprising:	Hilla discloses apparatus for connecting user ports to a communication network. *See supra at* 17[preamble], 14[preamble].
29[a]	one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network,	Hilla discloses one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network. *See supra at* 17[a], 14[a].
29[b]	at least one of said user interface modules being bi-directional and operative to communicate in both	Hilla discloses at least one of said user interface modules being bi-directional and operative to communicate in both an upstream direction and a downstream direction. *See supra at* 17[b], 14[c].

No.	’740 Patent Claim 29	Hilla
	an upstream direction and a downstream direction;	
29[c]	a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network,	Hilla discloses a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network. *See supra at* 14[b]-[e].
29[d]	at least one of said backplane traces being bi-directional and operative to communicate in both said upstream direction and said downstream direction; and	Hilla discloses at least one of said backplane traces being bi-directional and operative to communicate in both said upstream direction and said downstream direction. *See supra at* 14[c], 17[b].
29[e]	a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from	Hilla discloses a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame. *See supra at* 14[e], 17[e].

No.	’740 Patent Claim 29	Hilla
	the plurality of backplane traces over which to send the data frame.	

No.	’740 Patent Claim 30	Hilla
30[preamble]	Apparatus for connecting user ports to a communication network, comprising:	Hilla discloses apparatus for connecting user ports to a communication network. *See supra at* 29[preamble].
30[a]	one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network;	Hilla discloses one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network. *See supra at* 29[a].
30[b]	a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and	Hilla discloses a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network. *See supra at* 29[c].

No.	’740 Patent Claim 30	Hilla
	the communication network;	
30[c]	a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame;	Hilla discloses a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame. *See supra at* 29[e].
30[d]	at least some of the backplane traces are aggregated into an Ethernet link aggregation (LAG) group.	Hilla discloses at least some of the backplane traces are aggregated into an Ethernet link aggregation (LAG) group. *See supra at* 4[f], 15[f].

No.	’740 Patent Claim 31	Hilla
31	The apparatus according to claim 29, wherein the control module is arranged to apply a hashing function to the at least one of the frame attributes so as to select the backplane trace.	Hilla discloses the apparatus according to claim 29, wherein the control module is arranged to apply a hashing function to the at least one of the frame attributes so as to select the backplane trace. *See supra at* 16, 29, 30[c].

EXHIBIT C-2
Defendant’s Preliminary Invalidity Contentions
Orckit Corporation v. Cisco Systems, Inc., 2:22-cv-00276-JRG-RSP

Chart for U.S. Patent 7,545,740 (“the ’740 Patent”)
U.S. Patent Publication No. 2003/0147387 to Devi et al. (“Devi”)

As shown in the chart below, all Asserted Claims of the ’740 Patent are invalid under (1) 35 U.S.C. §§ 102 (a), (b), (e), and (g) because Devi meets each element of those claims, and/or (2) 35 U.S.C. § 103 because Devi renders those claims obvious either alone, or in combination with the knowledge of a person having ordinary skill in the art, and in further combination with the references specifically identified below and in the following claim chart and/or one or more references identified in Defendant’s Preliminary Invalidity Contentions. The following quotations and diagrams come from Devi titled “Forwarding Packets To Aggregated Links Using Distributed Ingress Card Processing”, which was filed on April 24, 2002, and published on August 7, 2003.

Motivations to combine the disclosures in Devi with disclosures in other publications known in the art, as explained in this chart, include at least the similarity in subject matter between the references to the extent they concern methods of data communication systems, and specifically to methods and systems for link aggregation in a data communication network. Insofar as the references cite other patents or publications, or suggest additional changes, one of ordinary skill in the art would look beyond a single reference to other references in the field.

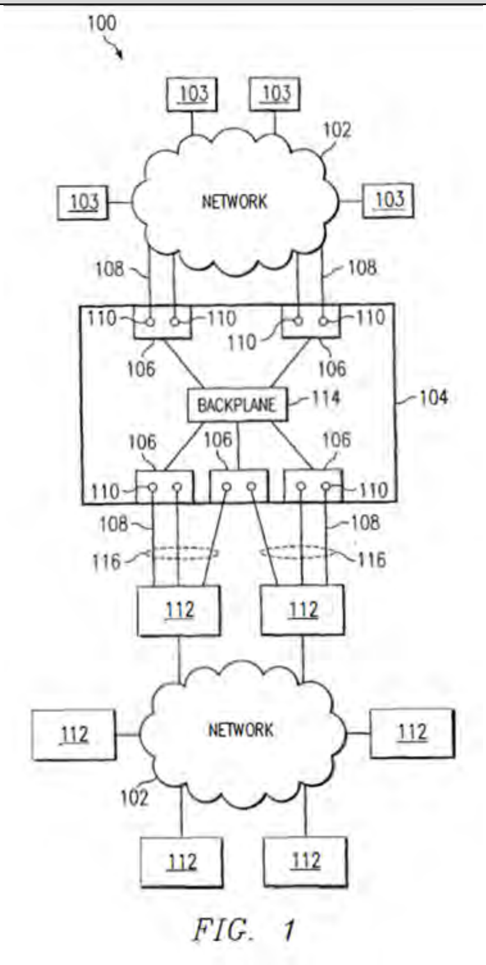
These invalidity contentions are based on Defendant’s present understanding of the asserted claims, and Orckit’s apparent construction of the claims in its November 3, 2022 Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1, and Orckit’s January 19, 2023 First Amended Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1 (Orckit’s “Infringement Disclosures”), which is deficient at least insofar as it fails to cite any documents or identify accused structures, acts, or materials in the Accused Products with particularity. Defendant does not agree with Orckit’s application of the claims, or that the claims satisfy the requirements of 35 U.S.C. § 112. Defendant’s contentions herein are not, and should in no way be seen as, admissions or adoptions as to any particular claim scope or construction, or as any admission that any particular element is met by any accused product in any particular way. Defendant objects to any attempt to imply claim construction from this chart. Defendant’s prior art invalidity contentions are made in a variety of alternatives and do not represent Defendant’s agreement or view as to the meaning, definiteness, written description support for, or enablement of any claim contained therein.

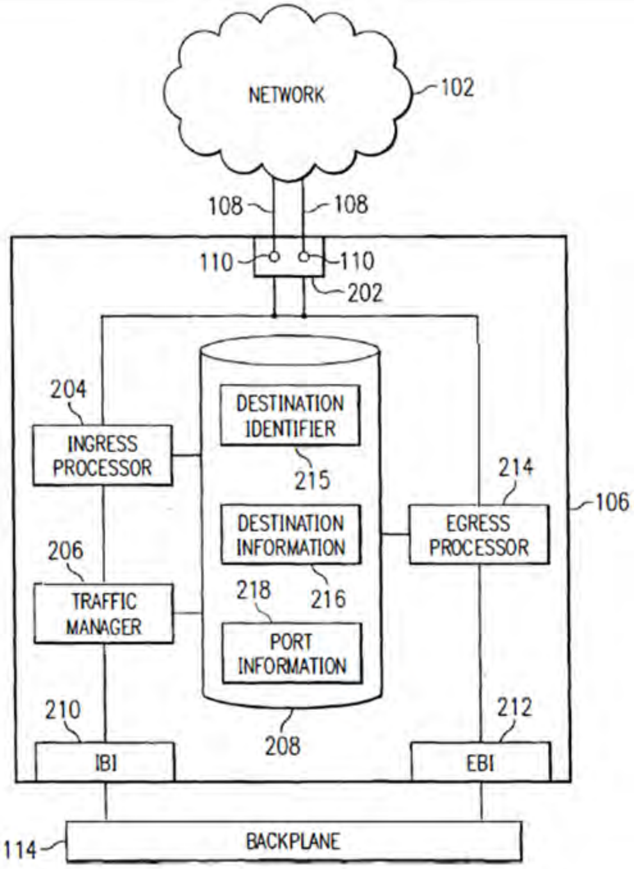
The following contentions are subject to revision and amendment pursuant to Federal Rule of Civil Procedure 26(e), the Local Rules, and the Orders of record in this matter subject to further investigation and discovery regarding the prior art and the Court's construction of the claims at issue.

No.	'740 Patent Claim 1	Devi
1[preamble]	A method for communication, comprising:	Devi discloses a method for communication. For example, Devi discloses a method for communicating between network entities using a port channel. Devi at Abstract ("A method performed by an ingress card includes receiving a packet with a destination identifier, and determining an aggregated link associated with the destination identifier that includes multiple egress ports. The method further includes determining a destination port from the egress ports of the aggregated link, and communicating the packet to the des-tination port.") Devi at [0003] ("As telecommunication networks handle more and more traffic, new methods are constantly being developed that allow the networks to process larger flows of informa-tion. Link aggregation is one example. In link aggregation, several physical links are aggregated to appear as one logical link to a telecommunications system. This aggregation allows all of the links to be used actively, rather than having particular links reserved. Thus, link aggregation provides more efficient use of network resources and better load balancing. The tradeoff, however, is that link aggregation may also require substantial additional processing and/or hardware to implement the aggregated links.") Devi at [0005] ("In accordance with one embodiment of the present invention, a method performed by an ingress card includes receiving a packet comprising a destination identifier. The method also includes determining an aggregated link asso-ciated with the destination identifier, and determining a destination port from the egress ports of the aggregated link. The method further includes communicating the packet to the destination port.") Devi at [0006] ("In accordance with another embodiment of the present invention, a method performed by an ingress card includes receiving a packet and determining egress links for the packet. The method also includes determining whether each link is an aggregated link with

No.	’740 Patent Claim 1	Devi
		multiple egress ports or a non-aggregated link with a single egress port. For each aggregated link, a destination port is determined from the egress ports, and the packet is communicated to the desti-nation port. For each non-aggregated link, a copy of the packet is communicated to the single egress port.”) Devi at [0007] (“Important technical advantages of certain embodi-ments of the present invention include implementing link aggregation using distributed processing among ingress components. For example, individual ingress cards can determine a destination port for a packet, eliminating the need for a separate processing stage implemented in hard-ware and/or software to separately determine the destination port from a packet after the ingress card forwards the packet to a link. This improves the efficiency and speed of packet forwarding in a switch.”) Devi at [0008] (“Yet another important technical advantage of cer-tain embodiments is a distributed architecture that may be used to process a variety of packet traffic. For example, certain embodiments of an ingress card can forward unicast, multicast, and bridging traffic. Particular embodiments may process packets from different protocols as well.”) Devi at [0009] (“Other important technical advantages of certain embodiments of the present invention include load balanc-ing. The tables or other information used by the ingress card to determine a destination port for packets may also include additional usage information that allows the ingress card to determine a destination port from the egress ports in an aggregated link. This means that rather than using one link to the exclusion of others, loads may be distributed among several links in an aggregated link in a more balanced fashion. Particular embodiments of the present invention may have some, all or none of the enumerated technical advantages. Still other important technical advantages will be apparent to one skilled in the art from the following figures, description, and claims.”) Devi at [0015] (“FIG. 1 shows a system 100 that includes a switch 104 coupled to a network 102. Using network 102 and switch 104, packets are communicated from sources 103 to destinations 112. Components of switch 104 are referred to as "ingress" components when

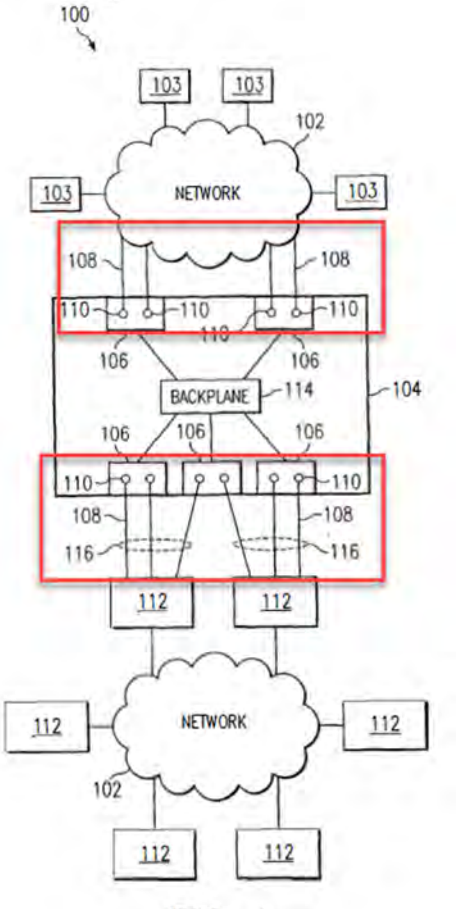
No.	’740 Patent Claim 1	Devi
		receiving packets from sources 103 and "egress" components when sending packets to destinations 112.”) Devi at [0016] (“Network 102 represents any suitable structure for communicating packets, cells, frames, segments, or other portions of data (generally referred to as "packets"). Net-work 102 may represent the Internet, extranet, local area network (LAN), synchronous optical network (SONET), wide area network (WAN), the public switched telephone network (PSTN), or any other suitable network for commu-nicating information. Network 102 may include routers, switches, hubs, endpoints, or any other network device that communicates information. Network 102 contemplates any number or arrangement of components that exchange infor-mation.”) Devi at Figure 1

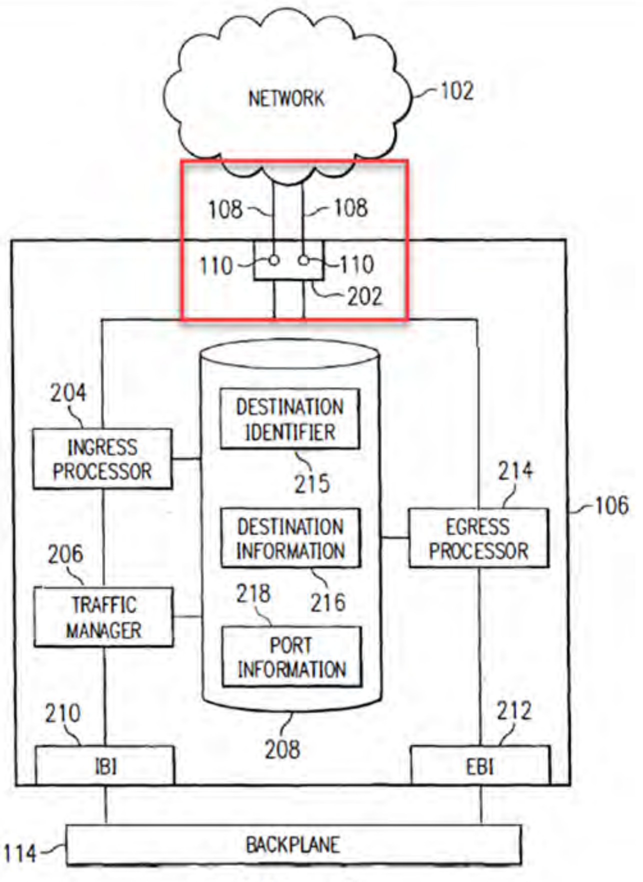
No.	’740 Patent Claim 1	Devi
		 100 103 103 102 103 NETWORK 103 108 108 110 110 110 110 110 106 106 BACKPLANE 114 104 106 106 106 110 110 108 108 116 116 112 112 112 NETWORK 112 102 112 112 FIG. 1 Devi at Figure 2

No.	’740 Patent Claim 1	Devi
		 NETWORK 102 108 108 110 110 202 204 INGRESS PROCESSOR DESTINATION IDENTIFIER 215 214 106 206 DESTINATION INFORMATION EGRESS PROCESSOR 218 216 TRAFFIC MANAGER PORT INFORMATION 210 212 208 IBI EBI 114 BACKPLANE *FIG. 2*
1[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel,	Devi discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. For example, Devi discloses sources 103, which may include endpoints, switches, routers, hubs, or any other suitable network device that are connected to a cards 106 in a switch that has ports 110 coupled to parallel physical 108.

No.	’740 Patent Claim 1	Devi
		Devi at [0005] (“In accordance with one embodiment of the present invention, a method performed by an ingress card includes receiving a packet comprising a destination identifier. The method also includes determining an aggregated link asso-ciated with the destination identifier, and determining a destination port from the egress ports of the aggregated link. The method further includes communicating the packet to the destination port.”) Devi at [0006] (“In accordance with another embodiment of the present invention, a method performed by an ingress card includes receiving a packet and determining egress links for the packet. The method also includes determining whether each link is an aggregated link with multiple egress ports or a non-aggregated link with a single egress port. For each aggregated link, a destination port is determined from the egress ports, and the packet is communicated to the desti-nation port. For each non-aggregated link, a copy of the packet is communicated to the single egress port.”) Devi at [0015] (“FIG. 1 shows a system 100 that includes a switch 104 coupled to a network 102. Using network 102 and switch 104, packets are communicated from sources 103 to destinations 112. Components of switch 104 are referred to as "ingress" components when receiving packets from sources 103 and "egress" components when sending packets to destinations 112.”) Devi at [0016] (“Network 102 represents any suitable structure for communicating packets, cells, frames, segments, or other portions of data (generally referred to as "packets"). Net-work 102 may represent the Internet, extranet, local area network (LAN), synchronous optical network (SONET), wide area network (WAN), the public switched telephone network (PSTN), or any other suitable network for commu-nicating information. Network 102 may include routers, switches, hubs, endpoints, or any other network device that communicates information. Network 102 contemplates any number or arrangement of components that exchange infor-mation.”) Devi at [0017] (“Sources 103 represent any source of information in packet form. Sources 103 need not be the original device that generated the packet, but need only convey a packet

No.	’740 Patent Claim 1	Devi
		to network 102. Sources 103 may be any hardware and/or software configured to communicate information, including endpoints, switches, routers, hubs, or any other suitable network device.”) Devi at [0018] (“Switch 104 sends and receives packets. Switch 104 may represent any suitable device, including an Ethernet switch, router, hub, or any other suitable hardware and/or software configured to receive packets and communicate them to other devices. Switch 104 includes cards 106 coupled to physical links 108 and a backplane 114 that represents hardware and/or software allowing cards 106 in switch 104 to exchange information with one another.”) Devi at [0019] (“Cards 106 represent separate components of hard-ware and/or software in switch 104 that exchange packets with network 102. Cards 106 may include traditional inter-face cards, as well as any other component, module, or part of switch 104 capable of independently receiving packets and communicating those packets to other components of switch 104. Each card 106 has one or more ports 110 coupled to physical links 108. Cards 106 may exchange information and packets with one another using backplane 114. Each card 106 includes sufficient processing capability to identify a port 110 in another card 106 in switch 104 and to communicate a packet to that port 110.”) Devi at [0020] (“Physical links 108 represent physical interfaces between switch 104 and other devices. Links 108 may include fiber optic connections, cables, wireless links, or any other suitable method for communicating information between switch 104 and other devices. Links 108 couple to cards 106 of switch 104 using ports 110. Port 110 represents any suitable physical interface between card 106 and physi-cal link 108 allowing information to be received from link 108 and communicated to link 108. Each physical link 108 is associated with one physical connection in the form of port 110.”) Devi at [0021] (“A card 106, port 110 or link 108 used to receive a packet from a source 103 is referred to as an "ingress" card 106, port 110, or link 108. A card 106, port 110 or link 108 used to communicate a packet to a destination 112 is referred to as an "egress" card 106, port 110, or link 108. Cards 106, ports 110, and links 108 may be bidirectional, so that a particular

No.	’740 Patent Claim 1	Devi
		component may be either an ingress or egress component depending on whether the component sends or receives a packet at a given time.”) Devi at Figure 1 (annotation added)  100 103 103 102 103 NETWORK 103 108 108 110 110 110 110 106 106 BACKPLANE 114 104 106 106 106 110 110 108 108 116 116 112 112 112 NETWORK 112 102 112 112 FIG. 1 Devi at Figure 2 (annotation added)

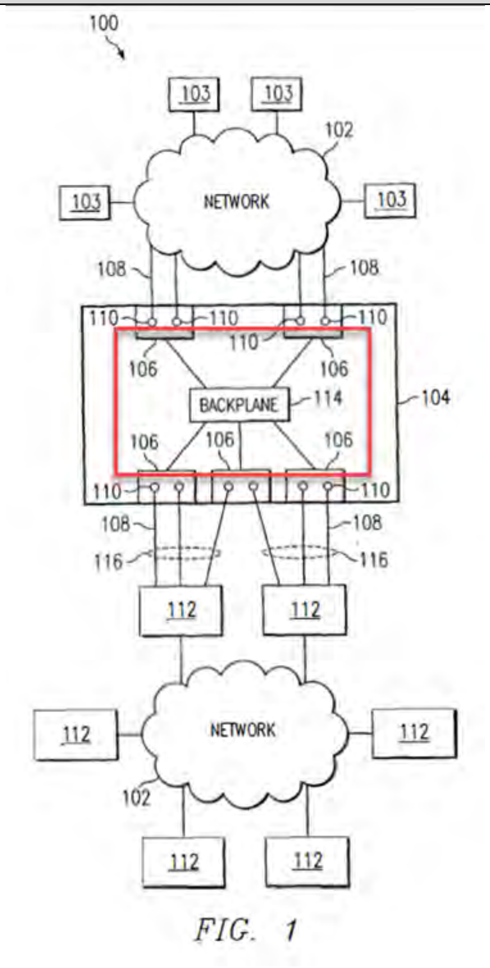
No.	’740 Patent Claim 1	Devi
		 NETWORK 102 108 108 110 110 202 204 INGRESS PROCESSOR DESTINATION IDENTIFIER 215 214 106 DESTINATION INFORMATION EGRESS PROCESSOR 206 TRAFFIC MANAGER 218 216 PORT INFORMATION 210 212 208 IBI EBI 114 BACKPLANE *FIG. 2*
1[b]	at least one of said first physical links being a bi-directional link operative to communicate in both	Devi discloses at least one of said first physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction. For example, Devi discloses both ingress and egress components that are capable of both receiving and sending packets. Devi further discloses cards 106, ports 110, and links 109,

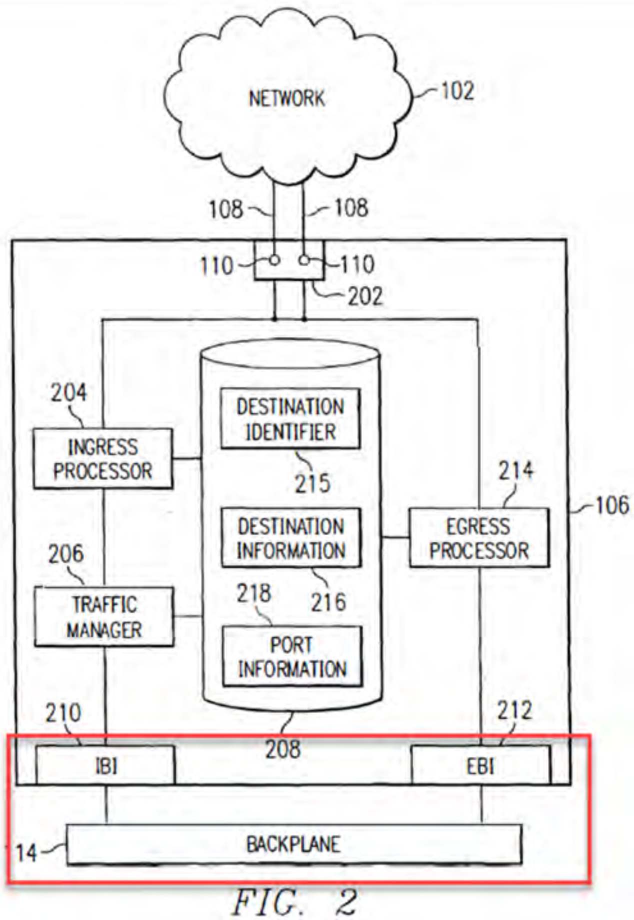
No.	’740 Patent Claim 1	Devi
	an upstream direction and a downstream direction	“may be bidirectional, so that a particular component may be either an ingress or egress component depending on whether the component sends or receives a packet at a given time.”. Devi at [0015] (“FIG. 1 shows a system 100 that includes a switch 104 coupled to a network 102. Using network 102 and switch 104, packets are communicated from sources 103 to destinations 112. Components of switch 104 are referred to as "ingress" components when receiving packets from sources 103 and "egress" components when sending packets to destinations 112.”) Devi at [0020] (“Physical links 108 represent physical interfaces between switch 104 and other devices. Links 108 may include fiber optic connections, cables, wireless links, or any other suitable method for communicating information between switch 104 and other devices. Links 108 couple to cards 106 of switch 104 using ports 110. Port 110 represents any suitable physical interface between card 106 and physi-cal link 108 allowing information to be received from link 108 and communicated to link 108. Each physical link 108 is associated with one physical connection in the form of port 110.”) Devi at [0021] (“A card 106, port 110 or link 108 used to receive a packet from a source 103 is referred to as an "ingress" card 106, port 110, or link 108. A card 106, port 110 or link 108 used to communicate a packet to a destination 112 is referred to as an "egress" card 106, port 110, or link 108. Cards 106, ports 110, and links 108 may be bidirectional, so that a particular component may be either an ingress or egress component depending on whether the component sends or receives a packet at a given time.”) Devi at [0024] (“FIG. 2 shows a card 106 in more detail. Card 106 couples to backplane 114 of switch 104 using ingress backplane interface (IBI) 210 and egress backplane interface (EBI) 212. Card 106 couples to network 102 through physi-cal layer devices 202, including ports 110 coupled to links 108. Card 106 includes an ingress processor 204, an egress processor 214, a traffic manager 206, and a memory 208. Card 106 may exchange packets from network 102 or backplane 114. Packets received from network 102 are considered to be on the "ingress" side, while packets received from the backplane 114 are considered to be on the "egress side" of card 106. Thus, at the level of card 106, the terms "ingress" and "egress"

No.	’740 Patent Claim 1	Devi
		refer to the function of those components with respect to receiving packets from network 102 or backplane 114.”) Devi at [0027] (“IBI 210 and EBI 212 refer to any port or connec-tion, real or virtual, between card 106 and backplane 114 of switch 104. IBI 210 and EBI 212 may represent separate components, or alternatively, may represent the same hard-ware and/or software used to send and receive packets from backplane 114. Functionally, IBI 210 sends packets to backplane 114, while EBI 212 receives packets from back-plane 114.”)
1[c]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel,	Devi discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. For example, Devi discloses card 106 coupled to network 102 via backplane 114 using ingress and egress backplane interfaces, which refer to any port of connection between card 106 and backplane 114. Devi at [0018] (“Switch 104 sends and receives packets. Switch 104 may represent any suitable device, including an Ethernet switch, router, hub, or any other suitable hardware and/or software configured to receive packets and communicate them to other devices. Switch 104 includes cards 106 coupled to physical links 108 and a backplane 114 that represents hardware and/or software allowing cards 106 in switch 104 to exchange information with one another.”) Devi at [0019] (“Cards 106 represent separate components of hard-ware and/or software in switch 104 that exchange packets with network 102. Cards 106 may include traditional inter-face cards, as well as any other component, module, or part of switch 104 capable of independently receiving packets and communicating those packets to other components of switch 104. Each card 106 has one or more ports 110 coupled to physical links 108. Cards 106 may exchange information and packets with one another using backplane 114. Each card 106 includes sufficient processing capability to identify a port 110 in another card 106 in switch 104 and to communicate a packet to that port 110.”)

No.	’740 Patent Claim 1	Devi
		Devi at [0020] (“Physical links 108 represent physical interfaces between switch 104 and other devices. Links 108 may include fiber optic connections, cables, wireless links, or any other suitable method for communicating information between switch 104 and other devices. Links 108 couple to cards 106 of switch 104 using ports 110. Port 110 represents any suitable physical interface between card 106 and physi-cal link 108 allowing information to be received from link 108 and communicated to link 108. Each physical link 108 is associated with one physical connection in the form of port 110.”) Devi at [0021] (“A card 106, port 110 or link 108 used to receive a packet from a source 103 is referred to as an "ingress" card 106, port 110, or link 108. A card 106, port 110 or link 108 used to communicate a packet to a destination 112 is referred to as an "egress" card 106, port 110, or link 108. Cards 106, ports 110, and links 108 may be bidirectional, so that a particular component may be either an ingress or egress component depending on whether the component sends or receives a packet at a given time.”) Devi at [0024] (“FIG. 2 shows a card 106 in more detail. Card 106 couples to backplane 114 of switch 104 using ingress backplane interface (IBI) 210 and egress backplane interface (EBI) 212. Card 106 couples to network 102 through physi-cal layer devices 202, including ports 110 coupled to links 108. Card 106 includes an ingress processor 204, an egress processor 214, a traffic manager 206, and a memory 208. Card 106 may exchange packets from network 102 or backplane 114. Packets received from network 102 are considered to be on the "ingress" side, while packets received from the backplane 114 are considered to be on the "egress side" of card 106. Thus, at the level of card 106, the terms "ingress" and "egress" refer to the function of those components with respect to receiving packets from network 102 or backplane 114.”) Devi at [0025] (“Physical layer devices 202 represent any physical interface between card 106 and network 102. In particular, physical layer devices 202 may represent any combination of ports 110 and links 108 as described in conjunction with FIG. 1. For example, each card 106 may include two ports 110 each coupled to a physical link 108, but any number of ports 110 or links 108 may be used.”)

No.	’740 Patent Claim 1	Devi
		Devi at [0027] (“IBI 210 and EBI 212 refer to any port or connec-tion, real or virtual, between card 106 and backplane 114 of switch 104. IBI 210 and EBI 212 may represent separate components, or alternatively, may represent the same hard-ware and/or software used to send and receive packets from backplane 114. Functionally, IBI 210 sends packets to backplane 114, while EBI 212 receives packets from back-plane 114.”) Devi at Figure 1 (annotation added)

No.	’740 Patent Claim 1	Devi
		 100 103 103 102 103 NETWORK 103 108 108 110 110 110 110 106 106 BACKPLANE 114 104 106 106 106 110 110 108 108 116 116 112 112 112 NETWORK 112 102 112 112 FIG. 1 Devi at Figure 2 (annotation added)

No.	'740 Patent Claim 1	Devi
		 NETWORK 102 108 108 110 110 202 204 INGRESS PROCESSOR DESTINATION IDENTIFIER 215 214 106 DESTINATION INFORMATION EGRESS PROCESSOR 206 TRAFFIC MANAGER 218 216 PORT INFORMATION 210 212 208 IBI EBI 14 BACKPLANE FIG. 2
1[d]	at least one of said second physical links being a bi-directional link operative to communicate in both	Devi discloses at least one of said second physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction.

No.	’740 Patent Claim 1	Devi
	an upstream direction and a downstream direction;	For example, Devi discloses both ingress and egress components that are capable of both receiving and sending packets. Devi further discloses ingress backplane interfaces and egress backplane interfaces that both send and receive packets. Devi at [0015] (“FIG. 1 shows a system 100 that includes a switch 104 coupled to a network 102. Using network 102 and switch 104, packets are communicated from sources 103 to destinations 112. Components of switch 104 are referred to as "ingress" components when receiving packets from sources 103 and "egress" components when sending packets to destinations 112.”) Devi at [0020] (“Physical links 108 represent physical interfaces between switch 104 and other devices. Links 108 may include fiber optic connections, cables, wireless links, or any other suitable method for communicating information between switch 104 and other devices. Links 108 couple to cards 106 of switch 104 using ports 110. Port 110 represents any suitable physical interface between card 106 and physi-cal link 108 allowing information to be received from link 108 and communicated to link 108. Each physical link 108 is associated with one physical connection in the form of port 110.”) Devi at [0021] (“A card 106, port 110 or link 108 used to receive a packet from a source 103 is referred to as an "ingress" card 106, port 110, or link 108. A card 106, port 110 or link 108 used to communicate a packet to a destination 112 is referred to as an "egress" card 106, port 110, or link 108. Cards 106, ports 110, and links 108 may be bidirectional, so that a particular component may be either an ingress or egress component depending on whether the component sends or receives a packet at a given time.”) Devi at [0024] (“FIG. 2 shows a card 106 in more detail. Card 106 couples to backplane 114 of switch 104 using ingress backplane interface (IBI) 210 and egress backplane interface (EBI) 212. Card 106 couples to network 102 through physi-cal layer devices 202, including ports 110 coupled to links 108. Card 106 includes an ingress processor 204, an egress processor 214, a traffic manager 206, and a memory 208. Card 106 may exchange packets from network 102 or backplane 114. Packets received from network 102 are considered to be on the "ingress" side, while packets received from the backplane 114 are considered to be on

No.	’740 Patent Claim 1	Devi
		the "egress side" of card 106. Thus, at the level of card 106, the terms "ingress" and "egress" refer to the function of those components with respect to receiving packets from network 102 or backplane 114.”) Devi at [0027] (“IBI 210 and EBI 212 refer to any port or connec-tion, real or virtual, between card 106 and backplane 114 of switch 104. IBI 210 and EBI 212 may represent separate components, or alternatively, may represent the same hard-ware and/or software used to send and receive packets from backplane 114. Functionally, IBI 210 sends packets to backplane 114, while EBI 212 receives packets from back-plane 114.”)
1[e]	receiving a data frame having frame attributes sent between the communication network and the network node:	Devi discloses receiving a data frame having frame attributes sent between the communication network and the network node. For example, Devi discloses packets and frames with destination identifier 215, which “may include may include a network address for destination 112, a logical address for an egress link, or any other information useful for identifying the destination of a packet. Destination identifier 215 contemplates any information determinable from a packet that allows card 106 to identify one or more destinations 112 for the packet.” Devi at [0005] (“In accordance with one embodiment of the present invention, a method performed by an ingress card includes receiving a packet comprising a destination identifier. The method also includes determining an aggregated link asso-ciated with the destination identifier, and determining a destination port from the egress ports of the aggregated link. The method further includes communicating the packet to the destination port.”) Devi at [0006] (“In accordance with another embodiment of the present invention, a method performed by an ingress card includes receiving a packet and determining egress links for the packet. The method also includes determining whether each link is an aggregated link with multiple egress ports or a non-aggregated link with a single egress port. For each aggregated link, a destination port is determined from the egress ports, and the packet is communicated

No.	’740 Patent Claim 1	Devi
		to the desti-nation port. For each non-aggregated link, a copy of the packet is communicated to the single egress port.”) Devi at [0007] (“Important technical advantages of certain embodi-ments of the present invention include implementing link aggregation using distributed processing among ingress components. For example, individual ingress cards can determine a destination port for a packet, eliminating the need for a separate processing stage implemented in hard-ware and/or software to separately determine the destination port from a packet after the ingress card forwards the packet to a link. This improves the efficiency and speed of packet forwarding in a switch.”) Devi at [0008] (“Yet another important technical advantage of cer-tain embodiments is a distributed architecture that may be used to process a variety of packet traffic. For example, certain embodiments of an ingress card can forward unicast, multicast, and bridging traffic. Particular embodiments may process packets from different protocols as well.”) Devi at [0009] (“Other important technical advantages of certain embodiments of the present invention include load balanc-ing. The tables or other information used by the ingress card to determine a destination port for packets may also include additional usage information that allows the ingress card to determine a destination port from the egress ports in an aggregated link. This means that rather than using one link to the exclusion of others, loads may be distributed among several links in an aggregated link in a more balanced fashion. Particular embodiments of the present invention may have some, all or none of the enumerated technical advantages. Still other important technical advantages will be apparent to one skilled in the art from the following figures, description, and claims.”) Devi at [0015] (“FIG. 1 shows a system 100 that includes a switch 104 coupled to a network 102. Using network 102 and switch 104, packets are communicated from sources 103 to destinations 112. Components of switch 104 are referred to as "ingress" components when receiving packets from sources 103 and "egress" components when sending packets to destinations 112.”)

No.	’740 Patent Claim 1	Devi
		Devi at [0016] (“Network 102 represents any suitable structure for communicating packets, cells, frames, segments, or other portions of data (generally referred to as "packets"). Net-work 102 may represent the Internet, extranet, local area network (LAN), synchronous optical network (SONET), wide area network (WAN), the public switched telephone network (PSTN), or any other suitable network for commu-nicating information. Network 102 may include routers, switches, hubs, endpoints, or any other network device that communicates information. Network 102 contemplates any number or arrangement of components that exchange infor-mation.”) Devi at [0018] (“Switch 104 sends and receives packets. Switch 104 may represent any suitable device, including an Ethernet switch, router, hub, or any other suitable hardware and/or software configured to receive packets and communicate them to other devices. Switch 104 includes cards 106 coupled to physical links 108 and a backplane 114 that represents hardware and/or software allowing cards 106 in switch 104 to exchange information with one another.) Devi at [0030] (“Destination identifier 215 represents information identifying an intermediate or final destination 112 for the packet. Destination identifier 215 may include a network address for destination 112, a logical address for an egress link, or any other information useful for identifying the destination of a packet. Destination identifier 215 contem-plates any information determinable from a packet that allows card 106 to identify one or more destinations 112 for the packet.”) Devi at [0031] (“Destination information 216 associates destination identifiers 215 in packets with egress links. Destination information 216 may be organized in any suitable fashion, including one or more tables, files, databases, or other form of organization. Card 106 uses destination information 216 to determine an egress link or links using destination iden-tifier 215 from a packet. For example, a multicast packet may include a multicast group identifier instead of indi-vidual link addresses. Ingress card 106 receives the packet, looks up the multicast group identifier, and determines the egress links associated with the multicast group. In another example, the packet may include a final destination address (such as an IP address for a receiving device) rather than a link address from switch 104. In that case, ingress link

No.	’740 Patent Claim 1	Devi
		106 uses destination information 216 to determine an egress link to the final destination. Destination information 216 may include one or more layers as well. For example, destination information 216 may include a first table relating a multicast group to destination IP addresses, and a second table relating the destination IP addresses to logical link addresses. Card 106 may also update destination information 216 as new or additional information becomes available, such as when card 106 receives an acknowledgement from a destination 112 indicating that a packet was received.”) Devi at [0033] (“In operation, card 106 receives a packet from network 102. Ingress card 106 determines the destination for the particular packet and consults destination information 216 to see if that destination is associated with a particular egress link. Some packets may have a destination identifier 215 that card 106 has not yet learned, and so destination information 216 does not include an egress link or links associated with destination identifier 215 of the packet. In such cases, card 106 may flood all available egress links so that the packet will arrive at its proper destination, and that destination will send an acknowledgment back to card 106. To flood destinations, card 106 determines logical links coupled to switch 104 using port information 218. Card 106 communicates the packet to these links, excluding the link from which the packet was received. In a particular embodi-ment, flooded cards will replicate the packet to all ports within that card, so card 106 need only communicate the packet once to a particular egress card 106.”) Devi at [0037] (“Card 106 receives a packet at step 302. At step 304, card determines destination identifier 215 for the packet and uses destination information 216 to determine if card 106 has previously learned the destination identifier 215. If the destination has not been previously learned, card 106 floods all egress links at step 306, shown in more detail in FIG. 4 described below.”) Devi at [0038] (“If destination identifier 215 of the packet has been learned, card 106 next determines if destination identifier 215 identifies a multicast or a unicast destination at step 308. If the packet is a multicast packet, card 106 selects one of the destinations for the packet using destination information 216 at step 310. On the other hand, if the packet is a unicast packet, there is only one destination, so there is no need to perform a selection process.”)

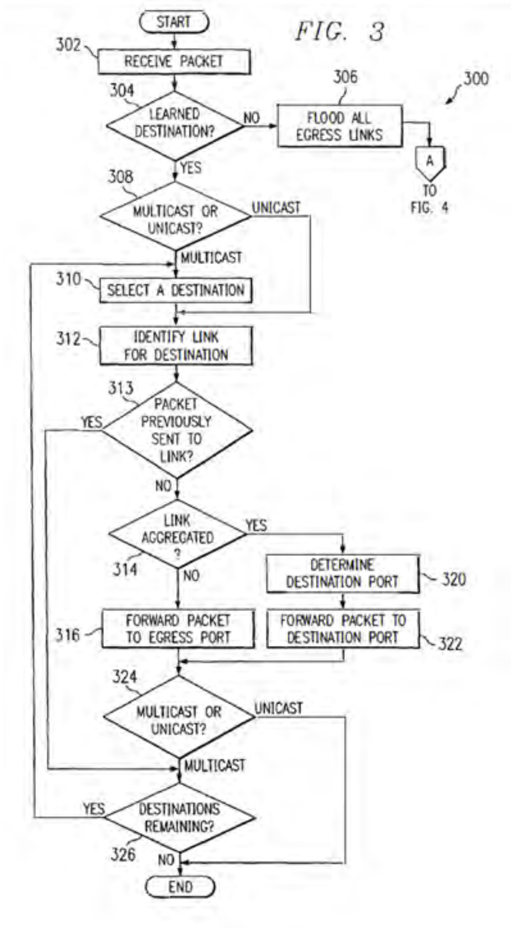
No.	’740 Patent Claim 1	Devi
		Devi at [0048] (“If the egress link is an aggregated isolated link, card 106 communicates the packet to destination port 110 of aggregated link 116. Card 106 determines destination port 110 for the packet at step 416. Card 106 may use any suitable method to determine destination port 110, including any of the methods described in conjunction with FIGS. 2 and 3. Card 106 then communicates the packet to destination port 110 at step 418.”) Devi at [0049] (“Card 106 then determines if there are any remain-ing egress links at step 420. If there are egress links remaining, then card 106 selects a new link to process at step 406. If there are no remaining links, then card 106 waits for an acknowledgement from one of the destinations that the packet was received. Card 106 receives the acknowledge-ment at step 422, and determines the link from which the acknowledgement was received at step 424. Card 106 then updates destination information 216 by associating the source link with destination identifier 215 at step 426. When the packet is a multicast packet, card 106 waits for acknowl-edgements from any remaining destinations at step 428.”)
1[f]:	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	Devi discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. For example, Devi discloses selecting egress links, including egress backplane interfaces and physical link 108 between the switch and destination of a destination port over which to send a packet based on the destination information 216 derived from the destination identifier of the packet. A person of ordinary skill in the art would understand that selecting egress links associated with a destination port would include the egress backplane interfaces. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Devi is found to not meet this limitation, selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group would have been obvious to a person having ordinary skill in the art, as explained below.

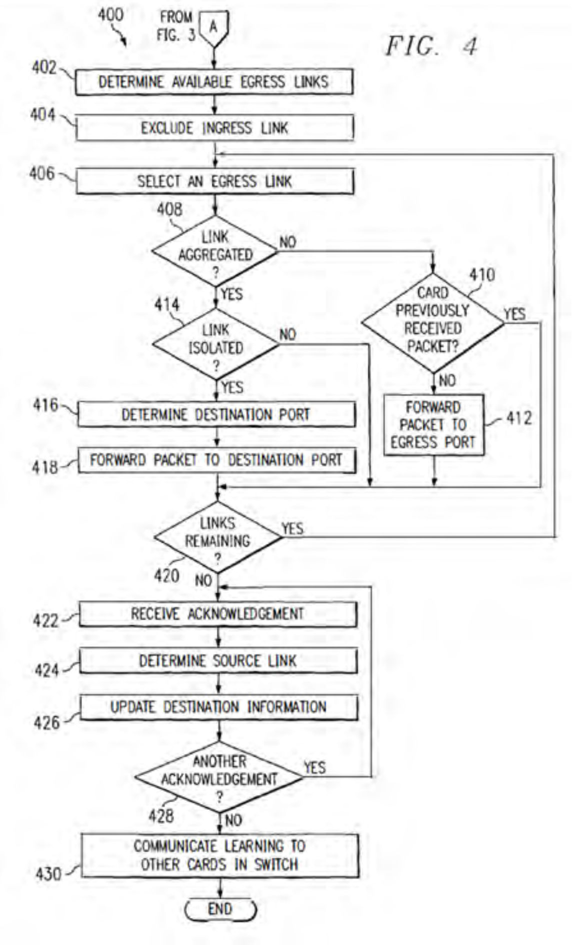
No.	’740 Patent Claim 1	Devi
		Devi at [0005] (“In accordance with one embodiment of the present invention, a method performed by an ingress card includes receiving a packet comprising a destination identifier. The method also includes determining an aggregated link asso-ciated with the destination identifier, and determining a destination port from the egress ports of the aggregated link. The method further includes communicating the packet to the destination port.”) Devi at [0006] (“In accordance with another embodiment of the present invention, a method performed by an ingress card includes receiving a packet and determining egress links for the packet. The method also includes determining whether each link is an aggregated link with multiple egress ports or a non-aggregated link with a single egress port. For each aggregated link, a destination port is determined from the egress ports, and the packet is communicated to the desti-nation port. For each non-aggregated link, a copy of the packet is communicated to the single egress port.”) Devi at [0023] (“Multiple physical links 108 to a particular destina-tion 112 may be aggregated to form an aggregated link 116. Aggregated link 116 represents multiple physical links 108 that are addressed by a single logical address, such as a medium access controller (MAC) address, an Internet pro-tocol (IP) address, or other suitable address or identifier, or otherwise treated as a single logical link between switch 104 and a destination 112. Because aggregated link 116 includes multiple physical links 108 that go to the same destination 112, a packet sent to destination 112 need only be sent to one port 110 corresponding to one physical link 108 of aggre-gated link 116. One technical advantage of certain embodi-ments of the present invention is that ingress cards 106 may communicate packets to ports 110 in some alternating fash-ion, such as round robin, random selection, pseudo-random selection using hash tables, or any other selection technique. This allows the load on any particular physical link 108 to be balanced with the loads on other physical links 108 in aggregated link 116.”) Devi at [0030] (“Destination identifier 215 represents information identifying an intermediate or final destination 112 for the packet. Destination identifier 215 may include a network address for destination 112, a logical address for an egress link, or any other information

No.	’740 Patent Claim 1	Devi
		useful for identifying the destination of a packet. Destination identifier 215 contem-plates any information determinable from a packet that allows card 106 to identify one or more destinations 112 for the packet.”) Devi at [0031] (“Destination information 216 associates destination identifiers 215 in packets with egress links. Destination information 216 may be organized in any suitable fashion, including one or more tables, files, databases, or other form of organization. Card 106 uses destination information 216 to determine an egress link or links using destination iden-tifier 215 from a packet. For example, a multicast packet may include a multicast group identifier instead of indi-vidual link addresses. Ingress card 106 receives the packet, looks up the multicast group identifier, and determines the egress links associated with the multicast group. In another example, the packet may include a final destination address (such as an IP address for a receiving device) rather than a link address from switch 104. In that case, ingress link 106 uses destination information 216 to determine an egress link to the final destination. Destination information 216 may include one or more layers as well. For example, destination information 216 may include a first table relating a multicast group to destination IP addresses, and a second table relating the destination IP addresses to logical link addresses. Card 106 may also update destination information 216 as new or additional information becomes available, such as when card 106 receives an acknowledgement from a destination 112 indicating that a packet was received.”) Devi at [0033] (“In operation, card 106 receives a packet from network 102. Ingress card 106 determines the destination for the particular packet and consults destination information 216 to see if that destination is associated with a particular egress link. Some packets may have a destination identifier 215 that card 106 has not yet learned, and so destination information 216 does not include an egress link or links associated with destination identifier 215 of the packet. In such cases, card 106 may flood all available egress links so that the packet will arrive at its proper destination, and that destination will send an acknowledgment back to card 106. To flood destinations, card 106 determines logical links coupled to switch 104 using port information 218. Card 106 communicates the packet to these links, excluding the link from which the packet was received. In a particular embodi-ment, flooded cards will replicate the

No.	’740 Patent Claim 1	Devi
		packet to all ports within that card, so card 106 need only communicate the packet once to a particular egress card 106.”) Devi at [0037] (“Card 106 receives a packet at step 302. At step 304, card determines destination identifier 215 for the packet and uses destination information 216 to determine if card 106 has previously learned the destination identifier 215. If the destination has not been previously learned, card 106 floods all egress links at step 306, shown in more detail in FIG. 4 described below.”) Devi at [0038] (“If destination identifier 215 of the packet has been learned, card 106 next determines if destination identifier 215 identifies a multicast or a unicast destination at step 308. If the packet is a multicast packet, card 106 selects one of the destinations for the packet using destination information 216 at step 310. On the other hand, if the packet is a unicast packet, there is only one destination, so there is no need to perform a selection process.”) Devi at [0039] (“Card 106 identifies an egress link associated with the destination using destination information 216 at step 312. Card 106 determines whether the packet was previ-ously sent to the selected link at step 313. For example, a multicast packet may have two destination accessible by the same egress link, so that communicating the packet to one of the destinations effectively communicates the packet to both destinations. In such cases, card 106 need not commu-nicate the packet twice, and may move on to another destination, if any, of the multicast packet at step 326.”) Devi at [0041] (“If the link is an aggregated link 116, card 106 determines a destination port 110 of aggregated link 116 using the information in port information 218 at step 320. Card 106 may select a particular destination port 110 from the multiple ports 110 of the aggregated link using random selection, load balancing, pseudo-random selection or hash-ing using part of the packet information, historical tracking, round robin, or any other method of selection. Card 106 then communicates the packet to the destination port 110 at step 322.”) Devi at [0043] (“Although a particular embodiment of the method has been described, numerous variations will be apparent to one skilled in the art. For example, multicast packets

No.	’740 Patent Claim 1	Devi
		may be sent to all destinations at one time rather than selecting individual destinations and sending a packet. Particular steps may be performed concurrently or continuously, and the particular order of steps may be varied as well. Furthermore, particular steps of the method may be omitted and added without changing the overall operation of the method.”) Devi at [0046] (“Card 106 selects one of the egress links at step 406. Card 106 then determines if the link is an aggregated link 116 with multiple egress ports 110 using port information 218 at step 408. If the link is not aggregated, card 106 determines whether the egress card 106 coupled to the egress port 110 of the link has received a copy of the packet at step 410. Card 106 may determine whether a packet has been previously sent to a card 106 by keeping a record of packets sent in memory 208. For example, card 106 may indicate in port information 218 that a particular egress card 106 has received the packet. If the egress card 106 has previously received the packet at step 410, then card 106 does not need to send a copy of the packet to that egress card 106, since egress card 106 already replicates the packet to all available ports 110 of egress card 106. On the other hand, if the egress card 106 has not received the packet, card 106 forwards a copy of the packet to the egress card 106 that includes the egress port 110 of the link at step 412.”) Devi at [0047] (“If the selected link is aggregated, card 106 then determines whether the aggregated link 116 is isolated using port information 218 at step 414. "Isolated" means that aggregated link 116 does not have any ports 110 that share an egress card 106 with non-aggregated link 108. Since card 106 must communicate a copy of the packet to each non-aggregated link 108 to reach the respective destinations 112 of the non-aggregated links 108, any card 106 that includes a port 110 of a non-aggregated link 108 must receive a copy of the packet. Because card 106 replicates packets, any aggregated link 116 that shares a card 106 with a non-aggregated link 108 receives a copy of the packet as well. Therefore, only aggregated links 108 that do not share a card 106 with a non-aggregated link 108 need to receive a copy of the packet.”) Devi at Figure 3

No.	’740 Patent Claim 1	Devi
		 FIG. 3 START 302 RECEIVE PACKET 304 LEARNED DESTINATION? NO 306 FLOOD ALL EGRESS LINKS 300 A TO FIG. 4 YES 308 MULTICAST OR UNICAST? UNICAST MULTICAST 310 SELECT A DESTINATION 312 IDENTIFY LINK FOR DESTINATION 313 PACKET PREVIOUSLY SENT TO LINK? YES NO LINK AGGREGATED ? YES 314 NO DETERMINE DESTINATION PORT 320 316 FORWARD PACKET TO EGRESS PORT FORWARD PACKET TO DESTINATION PORT 322 324 MULTICAST OR UNICAST? UNICAST MULTICAST YES DESTINATIONS REMAINING? 326 NO END Devi at Figure 4

No.	’740 Patent Claim 1	Devi
		 400 FROM FIG. 3 A FIG. 4 402 DETERMINE AVAILABLE EGRESS LINKS 404 EXCLUDE INGRESS LINK 406 SELECT AN EGRESS LINK 408 LINK AGGREGATED ? NO 410 CARD PREVIOUSLY RECEIVED PACKET? YES YES 414 LINK ISOLATED ? NO NO YES 416 DETERMINE DESTINATION PORT FORWARD PACKET TO EGRESS PORT 412 418 FORWARD PACKET TO DESTINATION PORT LINKS REMAINING ? YES 420 NO 422 RECEIVE ACKNOWLEDGEMENT 424 DETERMINE SOURCE LINK 426 UPDATE DESTINATION INFORMATION ANOTHER ACKNOWLEDGEMENT ? YES 428 NO 430 COMMUNICATE LEARNING TO OTHER CARDS IN SWITCH END Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Devi in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 1[f] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references.

No.	’740 Patent Claim 1	Devi
		For example, Basso discloses using a hash function and index table to select for blade/port combinations over which to send the packet over a user port and a switch fabric link. Basso further discloses that this selection is performed based upon packet information. Basso at [0010] (“Upon a network processor receiving a packet of data, the network processor may index into a table, commonly referred to as a forwarding table, to determine the table associated with a particular logical interface as well as the next destination address. The forwarding table may comprise a plurality of entries where each entry may comprise information indicating a particular table associated with a particular logical interface as well as the next destination address. Each logical interface may be associated with a table storing a plurality of entries containing blade/ port combinations as discussed further below. In one embodiment, an entry may be indexed in the forwarding table using a destination address in the received packet header.”) Basso at [0011] (“A hash function may then be performed on the received packet to generate a hash value. In one embodiment, a hash function may be performed on the source and destination address in the packet header to generate a hash value.”) Basso at [0012] (“The hash value generated may be used to index into the table associated with a particular logical interface. Upon indexing into the table associated with the logical interface, an appropriate blade/port combination may be identified to transmit the received packet of data. In one embodiment, a blade/port combination may be selected in the indexed entry of the table associated with the logical interface by using a portion of the bits of the hashed value. The received packet may then be transmitted through the identified blade/port combination to the next destination (next destination previously identified by the next destination address in the forwarding table).”) Basso at [0035] (“By logically grouping a plurality of ports 404 coupled to a particular network device into a logical interface 405, network processor 403 may be configured to transmit processed packets to that particular network device via any blade 402/port 404 combination grouped in that logical interface 405. For example, referring to FIG. 4, ports

No.	’740 Patent Claim 1	Devi
		404A-404I are physically connected to router 104B. If ports 404A-404I were logically grouped into logical interface 405, then a particular network processor 403, e.g., network processor 403A, may be configured to transmit processed packets that are determined to be transmitted to router 104B through any of ports 404A-404I in blades 402A-C, respectively. Network processor 403, e.g., network processor 403A, may be configured to transmit the processed packets to router 104B through ports 404, e.g., ports 404D-I, not in its blade 402, e.g., blade 402A, by forwarding the processed packets to switch fabric 401 which may then direct the processed packets to another appropriate physical blade 402/port 404 combination. Network processor 403, e.g., network processor 403A, may further be configured to transmit the processed packets to router 104B through any ports 404, e.g., ports 404A-C, in its blade 402, e.g., blade402A, instead of just one physical port 404 in its blade 402, e.g., blade 402A. A more detailed description of routing packets implementing logical interface(s) 405 is provided below in FIG. 5.”) Basso at [0040] (“In step 502, network processor 403, e.g., network processor 403A, may receive a packet of data from switch fabric 401. Upon receiving the packet of data, network processor 403, in step 503, may index into a table, commonly referred to as a forwarding table, to determine the table associated with a particular logical interface 405 as well as the next destination address, i.e., the next hop address. The forwarding table may comprise a plurality of entries where each entry may comprise information indicating a particular table associated with a particular logical interface 405 as well as the next destination address. Each logical interface 405 may be associated with a table storing a plurality of entries containing blade 402/port 404 combinations as discussed further below. In one embodiment, an entry may be indexed in the forwarding table using a destination address in the received packet header. It is noted that an entry may be indexed in the forwarding table usingother means and that such means would be recognized by an artisan of ordinary skill in the art. It is further noted that embodiments implementing such means would fall within the scope of the present invention.”) Basso at [0041] (“In step 504, a hash function may be performed on the received packet to generate a hash value. In one embodiment, a hash function may be performed on the source and destination address in the packet header to generate a hash value. It is noted that in other

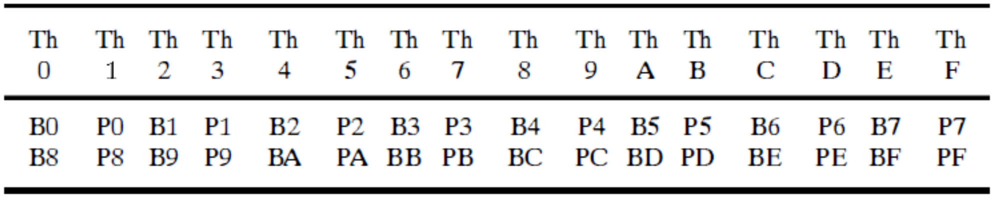
<table>
<tr><th>No.</th><th>’740 Patent Claim 1</th><th>Devi</th></tr>
<tr><td></td><td></td><td>

embodiments a hash function may be performed on different fields, e.g., port, type of service, in the received packet to generate a hash value.”)

Basso at [0042] (“In step 505, the hash value generated in step 504 may be used to index into the table associated with a particular logical interface 405 determined in step 503. Upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be identified in step 506 to transmit the received packet of data as explained below.”)

Basso at [0043] (“As stated above, the table associated with a particular logical interface 405 may comprise a plurality of entries where each entry may comprise a threshold value associated with a particular blade 402/port 404 combination. The threshold value may represent a percentage of the total number of packets received by router 104A that may be transmitted through the blade 402/port 404 combination associated with that threshold value. In one embodiment, the threshold value may be updated periodically by a user, e.g., system administrator, in control of router 104, e.g., router 104A. For example, the threshold value, e.g., twenty percent of the number of packets received by router 104A, associated with a particular blade 402/port 404 combination may be updated by lowering the threshold value by one percent during each update. An example of an entry of the table associated with a particular logical interface 405 is shown in Table 1 below:

TABLE 1



<table>
<tr><th>Th 0</th><th>Th 1</th><th>Th 2</th><th>Th 3</th><th>Th 4</th><th>Th 5</th><th>Th 6</th><th>Th 7</th><th>Th 8</th><th>Th 9</th><th>Th A</th><th>Th B</th><th>Th C</th><th>Th D</th><th>Th E</th><th>Th F</th></tr>
<tr><td>B0</td><td>P0</td><td>B1</td><td>P1</td><td>B2</td><td>P2</td><td>B3</td><td>P3</td><td>B4</td><td>P4</td><td>B5</td><td>P5</td><td>B6</td><td>P6</td><td>B7</td><td>P7</td></tr>
<tr><td>B8</td><td>P8</td><td>B9</td><td>P9</td><td>BA</td><td>PA</td><td>BB</td><td>PB</td><td>BC</td><td>PC</td><td>BD</td><td>PD</td><td>BE</td><td>PE</td><td>BF</td><td>PF</td></tr>
</table>

Basso at [0044] (“Table 1 above illustrates an exemplary entry in the table associated with a particular logical interface 405. Each entry may comprise a plurality of threshold values (16

</td></tr>
</table>

No.	’740 Patent Claim 1	Devi
		threshold values in exemplary Table 1) where each threshold value is associated with a particular blade 402/port 404 combination. For example, threshold value (Th0) is associated with blade B0/port P0 combination where blade BO may refer to a particular blade 402, e.g., blade 402B, and port PO may refer to a particular port 404, e.g., port 404E. Threshold value (Thl) is associated with blade Bl/port Pl combination and so forth. As stated above, each threshold value may represent a percentage of the total number of packets received by router 104A that may be transmitted through the blade 402/port 404 combination associated with that threshold value. For example, threshold value (Th0) may represent a percentage of the total number of packets received by router 104A that may be transmitted through port PO in blade BO. If port PO refers to port 404D and blade BO refers to blade 402B, then if Th0 has a value of twenty percent, a maximum of twenty percent of the total packets received by router 104A may be transmitted through port 404D in blade 402B.”) Basso at [0045] (“As stated above, upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be identified in step 506 to transmit the received packet of data. In one embodiment, the hash value generated in step 504 may be used to select a particular threshold value and hence a blade 402/port 404 combination associated with the selected threshold value. In one embodiment, a portion of the bits of the hash value, e.g., most significant bits, may be used to select a particular threshold value in the entry indexed in step 505. For example, referring to Table 1, since there are 16 different threshold values in each entry of the table associated with logical interface 405, only four bits of the hash value generated in step 504 may be used to select a threshold value. Upon selecting a threshold value, the blade 402/port 404 combination associated with the selected threshold value may be used to transmit the received packet.”) As another example, Wiher discloses using cell header information to at each node to select and route the ATM data cell over a data link and selected backplane. Wiher at 3:43-65 (“In general, in another aspect, the invention features an apparatus for communicating data cells between a data link and a backplane. The apparatus includes

No.	’740 Patent Claim 1	Devi
		transceiver circuitry to transmit and receive data cells over a data link and a plurality of backplane interfaces each including at least one cell signal terminal. Each of the backplane interface is coupled to a backplane interconnection circuit. Each backplane interconnection circuit transmits and receives cells over the cell signal terminals of its associated backplane interface. The apparatus also includes de-multiplexing circuitry coupling the transceiver circuitry to each of the backplane interconnection circuits. The de-multiplexing circuitry receives a data cell from the transceiver circuitry, select a backplane interconnection circuit associated with the data cell, and provide the data cell to the selected backplane interconnection circuit for transmission over the cell signal terminals of the associated backplane interface. The apparatus also includes multiplexing circuitry coupling the plurality of backplane interconnection circuits to the transceiver circuitry. The multiplexing circuitry receives data cells from each of the backplane interconnection circuits and provide the received data cells to the transceiver circuitry.”) Wiher at 3:66-4:22 (“Implementations of the invention may include one or more of the following features. The backplane interconnection circuits may independently receive and transmit data cells over the plurality of backplane interfaces. The de-multiplexing circuitry may select a backplane interface based on data in the header field of the data cell. The apparatus may include header translation circuitry to alter header data in cells sent between the plurality of backplane interfaces and the transceiver circuitry. Each of the plurality of backplane interfaces may include separate terminals to receive cells and separate terminals to transmit cells. The terminals to transmit cells may include a first and second control terminal and at least one outgoing cell data terminal. A backplane interface's backplane interconnection circuitry may accepts a signal on the first control terminal as indicating that a cell may be sent over the interface, asserts a 15 signal on the second control terminal to indicate that a cell is being transmitted, and transmits data bits of the cell on the outgoing cell data terminal. Each backplane interface may include a single outgoing cell data terminal and each bit of the cell may be serially transmitted over the single outgoing cell data terminal. Each backplane interface may include multiple outgoing cell data terminals and bits of the cell may be sent in parallel over the eight outgoing cell data terminals.”)

No.	’740 Patent Claim 1	Devi
1[g]	sending the data frame over the selected first and second physical links,	Devi discloses sending the data frame over the selected first and second physical links. For example, Devi discloses selecting and sending packets over egress links, including selected egress backplane interfaces and the selected physical link 108 between the switch and destination. A person of ordinary skill in the art would understand that selecting egress links associated with a destination port would include the egress backplane interfaces. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. Devi at [0005] (“In accordance with one embodiment of the present invention, a method performed by an ingress card includes receiving a packet comprising a destination identifier. The method also includes determining an aggregated link asso-ciated with the destination identifier, and determining a destination port from the egress ports of the aggregated link. The method further includes communicating the packet to the destination port.”) Devi at [0006] (“In accordance with another embodiment of the present invention, a method performed by an ingress card includes receiving a packet and determining egress links for the packet. The method also includes determining whether each link is an aggregated link with multiple egress ports or a non-aggregated link with a single egress port. For each aggregated link, a destination port is determined from the egress ports, and the packet is communicated to the desti-nation port. For each non-aggregated link, a copy of the packet is communicated to the single egress port.”) Devi at [0023] (“Multiple physical links 108 to a particular destina-tion 112 may be aggregated to form an aggregated link 116. Aggregated link 116 represents multiple physical links 108 that are addressed by a single logical address, such as a medium access controller (MAC) address, an Internet pro-tocol (IP) address, or other suitable address or identifier, or otherwise treated as a single logical link between switch 104 and a destination 112. Because aggregated link 116 includes multiple physical links 108 that go to the same destination 112, a packet sent to destination 112 need only be sent to one port 110 corresponding to one physical link 108 of aggre-gated link 116. One technical advantage of certain embodi-ments

No.	’740 Patent Claim 1	Devi
		of the present invention is that ingress cards 106 may communicate packets to ports 110 in some alternating fash-ion, such as round robin, random selection, pseudo-random selection using hash tables, or any other selection technique. This allows the load on any particular physical link 108 to be balanced with the loads on other physical links 108 in aggregated link 116.”) Devi at [0030] (“Destination identifier 215 represents information identifying an intermediate or final destination 112 for the packet. Destination identifier 215 may include a network address for destination 112, a logical address for an egress link, or any other information useful for identifying the destination of a packet. Destination identifier 215 contem-plates any information determinable from a packet that allows card 106 to identify one or more destinations 112 for the packet.”) Devi at [0031] (“Destination information 216 associates destination identifiers 215 in packets with egress links. Destination information 216 may be organized in any suitable fashion, including one or more tables, files, databases, or other form of organization. Card 106 uses destination information 216 to determine an egress link or links using destination iden-tifier 215 from a packet. For example, a multicast packet may include a multicast group identifier instead of indi-vidual link addresses. Ingress card 106 receives the packet, looks up the multicast group identifier, and determines the egress links associated with the multicast group. In another example, the packet may include a final destination address (such as an IP address for a receiving device) rather than a link address from switch 104. In that case, ingress link 106 uses destination information 216 to determine an egress link to the final destination. Destination information 216 may include one or more layers as well. For example, destination information 216 may include a first table relating a multicast group to destination IP addresses, and a second table relating the destination IP addresses to logical link addresses. Card 106 may also update destination information 216 as new or additional information becomes available, such as when card 106 receives an acknowledgement from a destination 112 indicating that a packet was received.”) Devi at [0033] (“In operation, card 106 receives a packet from network 102. Ingress card 106 determines the destination for the particular packet and consults destination information 216

No.	’740 Patent Claim 1	Devi
		to see if that destination is associated with a particular egress link. Some packets may have a destination identifier 215 that card 106 has not yet learned, and so destination information 216 does not include an egress link or links associated with destination identifier 215 of the packet. In such cases, card 106 may flood all available egress links so that the packet will arrive at its proper destination, and that destination will send an acknowledgment back to card 106. To flood destinations, card 106 determines logical links coupled to switch 104 using port information 218. Card 106 communicates the packet to these links, excluding the link from which the packet was received. In a particular embodi-ment, flooded cards will replicate the packet to all ports within that card, so card 106 need only communicate the packet once to a particular egress card 106.”) Devi at [0037] (“Card 106 receives a packet at step 302. At step 304, card determines destination identifier 215 for the packet and uses destination information 216 to determine if card 106 has previously learned the destination identifier 215. If the destination has not been previously learned, card 106 floods all egress links at step 306, shown in more detail in FIG. 4 described below.”) Devi at [0038] (“If destination identifier 215 of the packet has been learned, card 106 next determines if destination identifier 215 identifies a multicast or a unicast destination at step 308. If the packet is a multicast packet, card 106 selects one of the destinations for the packet using destination information 216 at step 310. On the other hand, if the packet is a unicast packet, there is only one destination, so there is no need to perform a selection process.”) Devi at [0039] (“Card 106 identifies an egress link associated with the destination using destination information 216 at step 312. Card 106 determines whether the packet was previ-ously sent to the selected link at step 313. For example, a multicast packet may have two destination accessible by the same egress link, so that communicating the packet to one of the destinations effectively communicates the packet to both destinations. In such cases, card 106 need not commu-nicate the packet twice, and may move on to another destination, if any, of the multicast packet at step 326.”)

No.	’740 Patent Claim 1	Devi
		Devi at [0041] (“If the link is an aggregated link 116, card 106 determines a destination port 110 of aggregated link 116 using the information in port information 218 at step 320. Card 106 may select a particular destination port 110 from the multiple ports 110 of the aggregated link using random selection, load balancing, pseudo-random selection or hash-ing using part of the packet information, historical tracking, round robin, or any other method of selection. Card 106 then communicates the packet to the destination port 110 at step 322.”) Devi at [0043] (“Although a particular embodiment of the method has been described, numerous variations will be apparent to one skilled in the art. For example, multicast packets may be sent to all destinations at one time rather than selecting individual destinations and sending a packet. Particular steps may be performed concurrently or continuously, and the particular order of steps may be varied as well. Furthermore, particular steps of the method may be omitted and added without changing the overall operation of the method.”) Devi at [0046] (“Card 106 selects one of the egress links at step 406. Card 106 then determines if the link is an aggregated link 116 with multiple egress ports 110 using port information 218 at step 408. If the link is not aggregated, card 106 determines whether the egress card 106 coupled to the egress port 110 of the link has received a copy of the packet at step 410. Card 106 may determine whether a packet has been previously sent to a card 106 by keeping a record of packets sent in memory 208. For example, card 106 may indicate in port information 218 that a particular egress card 106 has received the packet. If the egress card 106 has previously received the packet at step 410, then card 106 does not need to send a copy of the packet to that egress card 106, since egress card 106 already replicates the packet to all available ports 110 of egress card 106. On the other hand, if the egress card 106 has not received the packet, card 106 forwards a copy of the packet to the egress card 106 that includes the egress port 110 of the link at step 412.”) Devi at [0047] (“If the selected link is aggregated, card 106 then determines whether the aggregated link 116 is isolated using port information 218 at step 414. "Isolated" means that aggregated link 116 does not have any ports 110 that share an egress card 106 with non-aggregated link 108. Since card 106 must communicate a copy of the packet to each non-aggregated link 108 to reach the respective destinations 112 of the non-aggregated links

No.	’740 Patent Claim 1	Devi
		108, any card 106 that includes a port 110 of a non-aggregated link 108 must receive a copy of the packet. Because card 106 replicates packets, any aggregated link 116 that shares a card 106 with a non-aggregated link 108 receives a copy of the packet as well. Therefore, only aggregated links 108 that do not share a card 106 with a non-aggregated link 108 need to receive a copy of the packet.”)
1[h]	said sending comprising communicating along at least one of said bi-directional links.	Devi discloses said sending comprising communicating along at least one of said bi-directional links. *See supra at* 1[b], 1[d], 1[g].

No.	’740 Patent Claim 2	Devi
2[a]	The method according to claim 1, wherein the network node comprises a user node, and	Devi discloses the method according to claim 1, wherein the network node comprises a user node. For example, Devi discloses a sources 103, which “may be any hardware and/or software configured to communicate information, including endpoints, switches, routers, hubs, or any other suitable network device,” including a user network device. Devi at [0015] (“FIG. 1 shows a system 100 that includes a switch 104 coupled to a network 102. Using network 102 and switch 104, packets are communicated from sources 103 to destinations 112. Components of switch 104 are referred to as "ingress" components when receiving packets from sources 103 and "egress" components when sending packets to destinations 112.”) Devi at [0016] (“Network 102 represents any suitable structure for communicating packets, cells, frames, segments, or other portions of data (generally referred to as "packets"). Net-work 102 may represent the Internet, extranet, local area network (LAN), synchronous optical network (SONET), wide area network (WAN), the public switched telephone network (PSTN), or any other suitable network for commu-nicating information. Network 102 may

No.	’740 Patent Claim 2	Devi
		include routers, switches, hubs, endpoints, or any other network device that communicates information. Network 102 contemplates any number or arrangement of components that exchange infor-mation.”) Devi at [0017] (“Sources 103 represent any source of information in packet form. Sources 103 need not be the original device that generated the packet, but need only convey a packet to network 102. Sources 103 may be any hardware and/or software configured to communicate information, including endpoints, switches, routers, hubs, or any other suitable network device.”) Devi at [0018] (“Switch 104 sends and receives packets. Switch 104 may represent any suitable device, including an Ethernet switch, router, hub, or any other suitable hardware and/or software configured to receive packets and communicate them to other devices. Switch 104 includes cards 106 coupled to physical links 108 and a backplane 114 that represents hardware and/or software allowing cards 106 in switch 104 to exchange information with one another.”) Devi at [0021] (“A card 106, port 110 or link 108 used to receive a packet from a source 103 is referred to as an "ingress" card 106, port 110, or link 108. A card 106, port 110 or link 108 used to communicate a packet to a destination 112 is referred to as an "egress" card 106, port 110, or link 108. Cards 106, ports 110, and links 108 may be bidirectional, so that a particular component may be either an ingress or egress component depending on whether the component sends or receives a packet at a given time.”) Devi at [0022] (“Destinations 112 represent any hardware and/or software configured to receive packets. Destinations 112 may include routers, switches, endpoints or any other suit-able network device. Destinations 112 may be intermediate destinations for the packet before the packet reaches its final destination. For example, a packet may contain information identifying a final destination, but switch 104 may determine that for the packet to reach its final destination, it must first be communicated to one of the destinations 112 coupled to switch 104. Destinations 112 may also be sources 103 of packets, and sources 103 of packets may be destinations 112 of other packets as well.”)

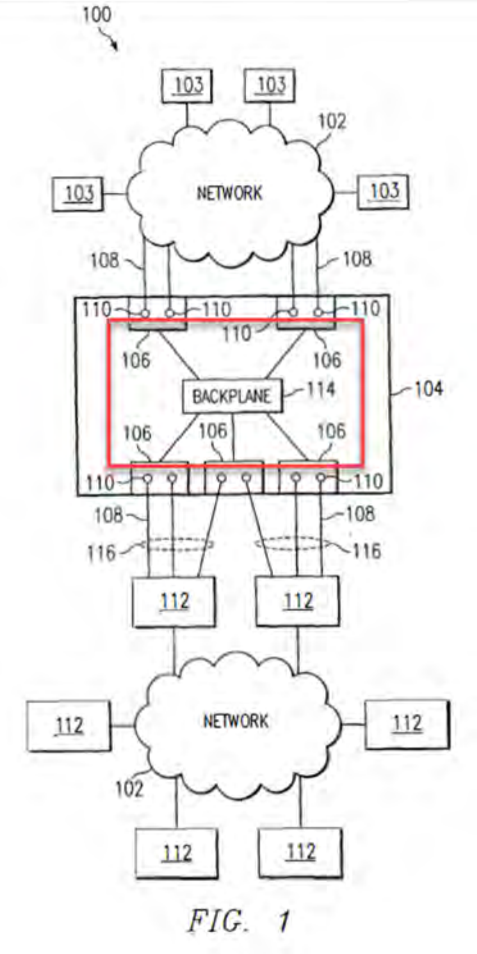
No.	’740 Patent Claim 2	Devi
2[b]	wherein sending the data frame comprises establishing a communication service between the user node and the communication network.	Devi discloses wherein sending the data frame comprises establishing a communication service between the user node and the communication network. For example, Devi discloses a sources 103, including a user network device, communicating information with a network 102. Devi at [0015] (“FIG. 1 shows a system 100 that includes a switch 104 coupled to a network 102. Using network 102 and switch 104, packets are communicated from sources 103 to destinations 112. Components of switch 104 are referred to as "ingress" components when receiving packets from sources 103 and "egress" components when sending packets to destinations 112.”) Devi at [0016] (“Network 102 represents any suitable structure for communicating packets, cells, frames, segments, or other portions of data (generally referred to as "packets"). Net-work 102 may represent the Internet, extranet, local area network (LAN), synchronous optical network (SONET), wide area network (WAN), the public switched telephone network (PSTN), or any other suitable network for commu-nicating information. Network 102 may include routers, switches, hubs, endpoints, or any other network device that communicates information. Network 102 contemplates any number or arrangement of components that exchange infor-mation.”) Devi at [0017] (“Sources 103 represent any source of information in packet form. Sources 103 need not be the original device that generated the packet, but need only convey a packet to network 102. Sources 103 may be any hardware and/or software configured to communicate information, including endpoints, switches, routers, hubs, or any other suitable network device.”) Devi at [0018] (“Switch 104 sends and receives packets. Switch 104 may represent any suitable device, including an Ethernet switch, router, hub, or any other suitable hardware and/or software configured to receive packets and communicate them to other devices. Switch 104 includes cards 106 coupled to physical links 108 and a backplane 114 that represents

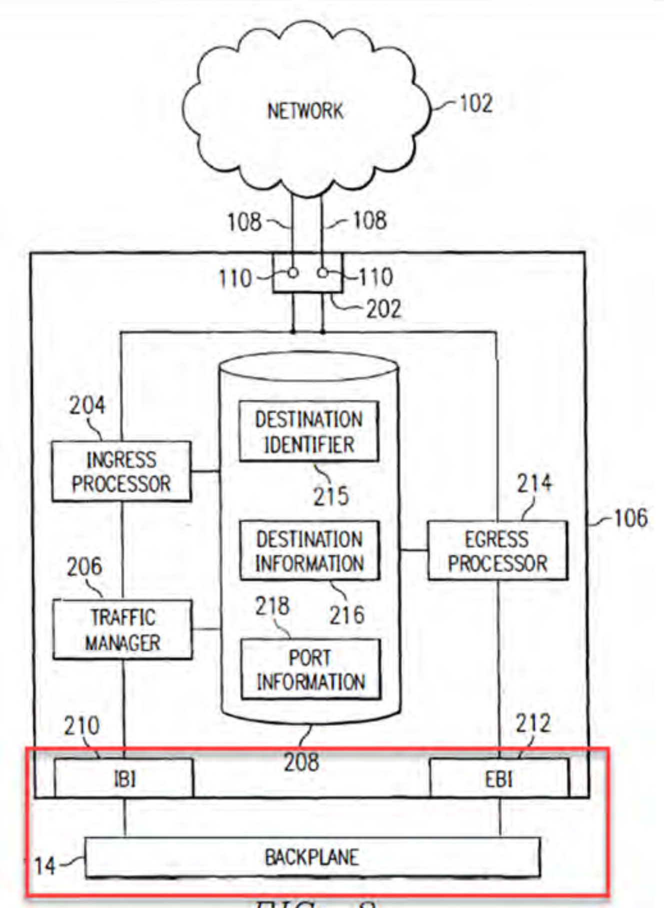
No.	’740 Patent Claim 2	Devi
		hardware and/or software allowing cards 106 in switch 104 to exchange information with one another.”) Devi at [0022] (“Destinations 112 represent any hardware and/or software configured to receive packets. Destinations 112 may include routers, switches, endpoints or any other suit-able network device. Destinations 112 may be intermediate destinations for the packet before the packet reaches its final destination. For example, a packet may contain information identifying a final destination, but switch 104 may determine that for the packet to reach its final destination, it must first be communicated to one of the destinations 112 coupled to switch 104. Destinations 112 may also be sources 103 of packets, and sources 103 of packets may be destinations 112 of other packets as well.”)

No.	’740 Patent Claim 3	Devi
3	The method according to claim 1, wherein the second physical links comprise backplane traces formed on a backplane to which the one or more interface modules are coupled.	Devi discloses the method according to claim 1, wherein the second physical links comprise backplane traces formed on a backplane to which the one or more interface modules are coupled. For example, Devi discloses ingress and egress backplane interfaces on the backplane 114, which refer to any port or connection between card 106 and backplane 114. Devi at [0018] (“Switch 104 sends and receives packets. Switch 104 may represent any suitable device, including an Ethernet switch, router, hub, or any other suitable hardware and/or software configured to receive packets and communicate them to other devices. Switch 104 includes cards 106 coupled to physical links 108 and a backplane 114 that represents hardware and/or software allowing cards 106 in switch 104 to exchange information with one another.”) Devi at [0019] (“Cards 106 represent separate components of hard-ware and/or software in switch 104 that exchange packets with network 102. Cards 106 may include traditional inter-face cards, as well as any other component, module, or part of switch 104 capable of independently receiving packets and communicating those packets to other components of

No.	’740 Patent Claim 3	Devi
		switch 104. Each card 106 has one or more ports 110 coupled to physical links 108. Cards 106 may exchange information and packets with one another using backplane 114. Each card 106 includes sufficient processing capability to identify a port 110 in another card 106 in switch 104 and to communicate a packet to that port 110.”) Devi at [0020] (“Physical links 108 represent physical interfaces between switch 104 and other devices. Links 108 may include fiber optic connections, cables, wireless links, or any other suitable method for communicating information between switch 104 and other devices. Links 108 couple to cards 106 of switch 104 using ports 110. Port 110 represents any suitable physical interface between card 106 and physi-cal link 108 allowing information to be received from link 108 and communicated to link 108. Each physical link 108 is associated with one physical connection in the form of port 110.”) Devi at [0021] (“A card 106, port 110 or link 108 used to receive a packet from a source 103 is referred to as an "ingress" card 106, port 110, or link 108. A card 106, port 110 or link 108 used to communicate a packet to a destination 112 is referred to as an "egress" card 106, port 110, or link 108. Cards 106, ports 110, and links 108 may be bidirectional, so that a particular component may be either an ingress or egress component depending on whether the component sends or receives a packet at a given time.”) Devi at [0024] (“FIG. 2 shows a card 106 in more detail. Card 106 couples to backplane 114 of switch 104 using ingress backplane interface (IBI) 210 and egress backplane interface (EBI) 212. Card 106 couples to network 102 through physi-cal layer devices 202, including ports 110 coupled to links 108. Card 106 includes an ingress processor 204, an egress processor 214, a traffic manager 206, and a memory 208. Card 106 may exchange packets from network 102 or backplane 114. Packets received from network 102 are considered to be on the "ingress" side, while packets received from the backplane 114 are considered to be on the "egress side" of card 106. Thus, at the level of card 106, the terms "ingress" and "egress" refer to the function of those components with respect to receiving packets from network 102 or backplane 114.”)

No.	’740 Patent Claim 3	Devi
		Devi at [0025] (“Physical layer devices 202 represent any physical interface between card 106 and network 102. In particular, physical layer devices 202 may represent any combination of ports 110 and links 108 as described in conjunction with FIG. 1. For example, each card 106 may include two ports 110 each coupled to a physical link 108, but any number of ports 110 or links 108 may be used.”) Devi at [0027] (“IBI 210 and EBI 212 refer to any port or connec-tion, real or virtual, between card 106 and backplane 114 of switch 104. IBI 210 and EBI 212 may represent separate components, or alternatively, may represent the same hard-ware and/or software used to send and receive packets from backplane 114. Functionally, IBI 210 sends packets to backplane 114, while EBI 212 receives packets from back-plane 114.”) Devi at Figure 1 (annotation added)

No.	’740 Patent Claim 3	Devi
		 100 103 103 102 103 NETWORK 103 108 108 110 110 110 110 106 106 BACKPLANE 114 104 106 106 106 110 110 108 108 116 116 112 112 112 NETWORK 112 102 112 112 FIG. 1 Devi at Figure 2 (annotation added)

No.	’740 Patent Claim 3	Devi
		 NETWORK 102 108 108 110 110 202 204 INGRESS PROCESSOR DESTINATION IDENTIFIER 215 214 EGRESS PROCESSOR 106 DESTINATION INFORMATION 206 TRAFFIC MANAGER 218 216 PORT INFORMATION 210 212 208 IBI EBI 14 BACKPLANE *FIG. 2*

No.	’740 Patent Claim 4	Devi
4[preamble]	A method for communication, comprising:	Devi discloses a method for communication. *See supra* at 1[preamble].

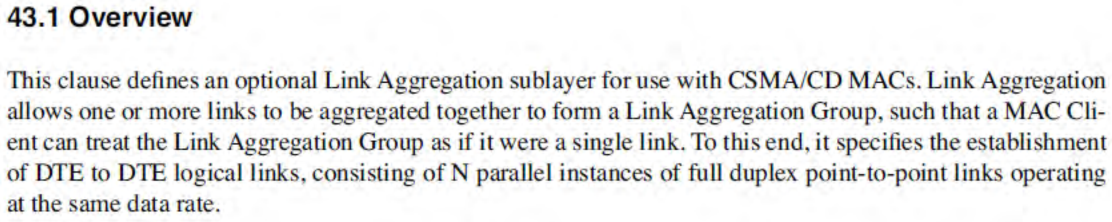
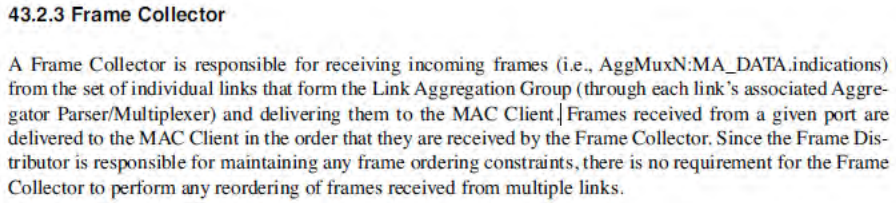
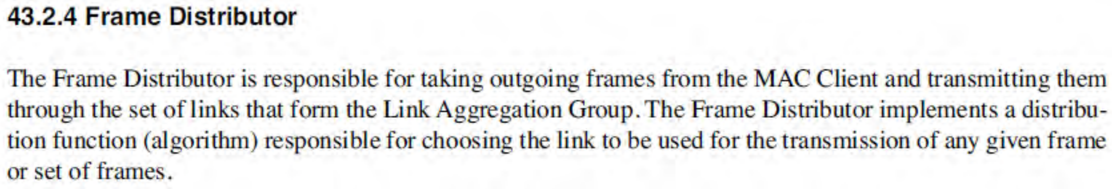
No.	’740 Patent Claim 4	Devi
4[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	Devi discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].
4[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Devi discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra* at 1[c].
4[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	Devi discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra* at 1[e].
4[d]	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	Devi discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. *See supra* at 1[f].
4[e]	sending the data frame over the selected first	Devi discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].

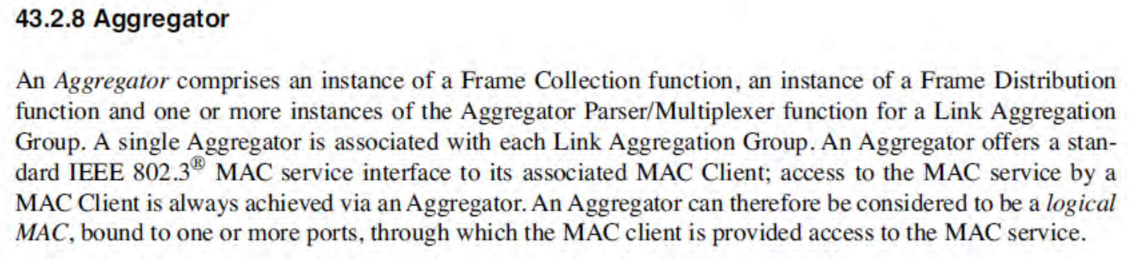
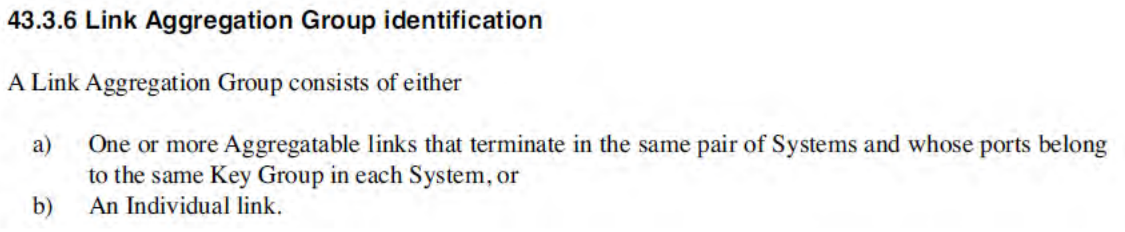
No.	’740 Patent Claim 4	Devi
	and second physical links,	
4[f]	at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group.	Devi discloses at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group. For example, Devi discloses aggregating multiple physical links 108 into an aggregated link 116. A person of ordinary skill in the art would understands that ingress and egress backplane interfaces on the backplane 114 could also be aggregated. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Devi is found to not meet this limitation, at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group would have been obvious to a person having ordinary skill in the art, as explained below. Devi at Abstract (“A method perfomed by an ingress card includes receiving a packet with a destination identifier, and determining an aggregated link associated with the destination identifier that includes multiple egress ports. The method further includes determining a destination port from the egress ports of the aggregated link, and communicating the packet to the des-tination port.”) Devi at [0003] (“As telecommunication networks handle more and more traffic, new methods are constantly being developed that allow the networks to process larger flows of informa-tion. Link aggregation is one example. In link aggregation, several physical links are aggregated to appear as one logical link to a telecommunications system. This aggregation allows all of the links to be used actively, rather than having particular links reserved. Thus, link aggregation provides more efficient use of network resources and better load balancing. The tradeoff, however, is that link aggregation may also require substantial additional processing and/or hardware to implement the aggregated links.”) Devi at [0004] (“In accordance with the present invention, the dis-advantages and problems associated with implementing link aggregation in switches have been substantially reduced or eliminated. In particular, certain embodiments of the present invention provide a method and system for implementing link aggregation using distributed processing. Certain

No.	’740 Patent Claim 4	Devi
		embodiments of the present invention are also compliant with the IEEE standards for link aggregation, set out in IEEE Standard 802.3.”) Devi at [0005] (“In accordance with one embodiment of the present invention, a method performed by an ingress card includes receiving a packet comprising a destination identifier. The method also includes determining an aggregated link asso-ciated with the destination identifier, and determining a destination port from the egress ports of the aggregated link. The method further includes communicating the packet to the destination port.”) Devi at [0006] (“In accordance with another embodiment of the present invention, a method performed by an ingress card includes receiving a packet and determining egress links for the packet. The method also includes determining whether each link is an aggregated link with multiple egress ports or a non-aggregated link with a single egress port. For each aggregated link, a destination port is determined from the egress ports, and the packet is communicated to the desti-nation port. For each non-aggregated link, a copy of the packet is communicated to the single egress port.”) Devi at [0009] (“Other important technical advantages of certain embodiments of the present invention include load balanc-ing. The tables or other information used by the ingress card to determine a destination port for packets may also include additional usage information that allows the ingress card to determine a destination port from the egress ports in an aggregated link. This means that rather than using one link to the exclusion of others, loads may be distributed among several links in an aggregated link in a more balanced fashion. Particular embodiments of the present invention may have some, all or none of the enumerated technical advantages. Still other important technical advantages will be apparent to one skilled in the art from the following figures, description, and claims.”) Devi at [0023] (“Multiple physical links 108 to a particular destina-tion 112 may be aggregated to form an aggregated link 116. Aggregated link 116 represents multiple physical links 108 that are addressed by a single logical address, such as a medium access controller (MAC) address, an Internet pro-tocol (IP) address, or other suitable address or identifier, or otherwise treated as a single logical link between switch 104 and a destination 112. Because

No.	’740 Patent Claim 4	Devi
		aggregated link 116 includes multiple physical links 108 that go to the same destination 112, a packet sent to destination 112 need only be sent to one port 110 corresponding to one physical link 108 of aggre-gated link 116. One technical advantage of certain embodi-ments of the present invention is that ingress cards 106 may communicate packets to ports 110 in some alternating fash-ion, such as round robin, random selection, pseudo-random selection using hash tables, or any other selection technique. This allows the load on any particular physical link 108 to be balanced with the loads on other physical links 108 in aggregated link 116.”) Devi at [0032] (“Port information 218 refers to any form of data that identifies ports 110 associated with a particular link. For example, port information 218 for a non-aggregated link 108 identifies the single egress port 110 for the non-aggregated link 108. Port information 218 for an aggregated link 116 identifies the multiple ports 110 coupled to the physical links 108 of aggregated link 116, which may include ports 110 on a single card 106 or multiple cards 106. Port information 218 may also identify which card 106 is coupled to each port 110. Port information 218 contemplates any suitable arrangement of data, such as a table, database or any other format. Card 106 may update port information 218 as network conditions change, e.g., when a port 110 or physical link 108 fails.”) Devi at [0034] (“If a destination has been previously learned, card 106 determines the appropriate egress link (unicast) or links (multicast) using destination information 216. For each aggregated link, card 106 then uses port information 218 to determine a destination egress port 110 of the aggregrated egress link 116. Card 106 then forwards the packet to the destination egress port 110. For non-aggregated links, card 106 communicates the packet to the single egress port 110 of the non-aggregated link 108.”) Devi at [0041] (“If the link is an aggregated link 116, card 106 determines a destination port 110 of aggregated link 116 using the information in port information 218 at step 320. Card 106 may select a particular destination port 110 from the multiple ports 110 of the aggregated link using random selection, load balancing, pseudo-random selection or hash-ing using part of the packet information, historical tracking, round robin, or any other method of selection. Card 106 then communicates the packet to the destination port 110 at step 322.)

No.	’740 Patent Claim 4	Devi
		Devi at [0048] (“If the selected link is aggregated, card 106 then determines whether the aggregated link 116 is isolated using port information 218 at step 414. "Isolated" means that aggregated link 116 does not have any ports 110 that share an egress card 106 with non-aggregated link 108. Since card 106 must communicate a copy of the packet to each non-aggregated link 108 to reach the respective destinations 112 of the non-aggregated links 108, any card 106 that includes a port 110 of a non-aggregated link 108 must receive a copy of the packet. Because card 106 replicates packets, any aggregated link 116 that shares a card 106 with a non-aggregated link 108 receives a copy of the packet as well. Therefore, only aggregated links 108 that do not share a card 106 with a non-aggregated link 108 need to receive a copy of the packet.”) Devi at [0049] (“If the egress link is an aggregated isolated link, card 106 communicates the packet to destination port 110 of aggregated link 116. Card 106 determines destination port 110 for the packet at step 416. Card 106 may use any suitable method to determine destination port 110, including any of the methods described in conjunction with FIGS. 2 and 3. Card 106 then communicates the packet to destination port 110 at step 418.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Devi in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 4[f] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, IEEE 802.3 discloses the aggregation of one or more links together to form a Link Aggregation Group. IEEE 802.3 at 1465

No.	’740 Patent Claim 4	Devi
		 **43.1 Overview** This clause defines an optional Link Aggregation sublayer for use with CSMA/CD MACs. Link Aggregation allows one or more links to be aggregated together to form a Link Aggregation Group, such that a MAC Client can treat the Link Aggregation Group as if it were a single link. To this end, it specifies the establishment of DTE to DTE logical links, consisting of N parallel instances of full duplex point-to-point links operating at the same data rate. IEEE 802.3 at 1470  **43.2.3 Frame Collector** A Frame Collector is responsible for receiving incoming frames (i.e., AggMuxN:MA_DATA.indications) from the set of individual links that form the Link Aggregation Group (through each link’s associated Aggregator Parser/Multiplexer) and delivering them to the MAC Client. Frames received from a given port are delivered to the MAC Client in the order that they are received by the Frame Collector. Since the Frame Distributor is responsible for maintaining any frame ordering constraints, there is no requirement for the Frame Collector to perform any reordering of frames received from multiple links. IEEE 802.3 at 1471  **43.2.4 Frame Distributor** The Frame Distributor is responsible for taking outgoing frames from the MAC Client and transmitting them through the set of links that form the Link Aggregation Group. The Frame Distributor implements a distribution function (algorithm) responsible for choosing the link to be used for the transmission of any given frame or set of frames. IEEE 802.3 at 1474

No.	’740 Patent Claim 4	Devi
		 **43.2.8 Aggregator** An *Aggregator* comprises an instance of a Frame Collection function, an instance of a Frame Distribution function and one or more instances of the Aggregator Parser/Multiplexer function for a Link Aggregation Group. A single Aggregator is associated with each Link Aggregation Group. An Aggregator offers a standard IEEE 802.3® MAC service interface to its associated MAC Client; access to the MAC service by a MAC Client is always achieved via an Aggregator. An Aggregator can therefore be considered to be a *logical MAC*, bound to one or more ports, through which the MAC client is provided access to the MAC service. IEEE 802.3 at 1481  **43.3.6 Link Aggregation Group identification** A Link Aggregation Group consists of either a) One or more Aggregatable links that terminate in the same pair of Systems and whose ports belong to the same Key Group in each System, or b) An Individual link. For example, Ghosh discloses aggregating physical links, including ports, into aggregate port channels that form a single logical link to increase bandwidth. Ghosh at Abstract (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggregation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent gen-eration of port channel map tables.”) Ghosh at [0004] (“Neighboring nodes in a fibre channel network are typically interconnected through multiple physical links. For example, a local fibre channel switch may be connected to a remote fibre channel switch through four physical links. In many instances, it may be beneficial to aggregate some of the physical links into logical links. That is, multiple physi-cal

No.	’740 Patent Claim 4	Devi
		links can be combined to form a logical interface to provide higher aggregate bandwidth, load balancing, and link redundancy. When a frame is being transmitted over a logical link, it does not matter what particular physical link is being used as long as all the frames of a given flow are transmitted through the same link. If a constituent physical link goes down, the logical link can still remain operational.”) Ghosh at [0007] (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggre-gation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent generation of port channel map tables.”) Ghosh at [0008] (“In one embodiment, a method for aggregating ports in a fibre channel fabric is provided. It is determined that a plurality of local ports at a local fibre channel switch are compatible. Identifiers for the plurality of local ports are sent to a remote fibre channel switch. The remote fibre channel switch determines if a plurality of remote ports are compat-ible, the plurality of remote ports corresponding to the plurality of local ports. An indication that one or more of the remote physical ports are compatible is received. A port channel including one or more of the local ports correspond-ing to the compatible remote ports is created.”) Ghosh at [0010] (“In another embodiment, a fibre channel network is described. The fibre channel network includes a local fibre channel switch and a remote fibre channel switch. The local fibre channel switch aggregates a compatible subset of the plurality of local ports and sends identifiers for the compat-ible subset of the plurality of local ports to the remote fibre channel switch. The remote fibre channel switch determines if a subset of the plurality of remote ports are compatible. The subset of the plurality of remote ports corresponds to the compatible subset of the plurality of local ports.”) Ghosh at [0022] (“Switches in a fibre channel network are typically interconnected using multiple physical links. The physical links connecting a pair of switches allows transmission of data and control signals. In some instances, it is useful to aggregate multiple physical links

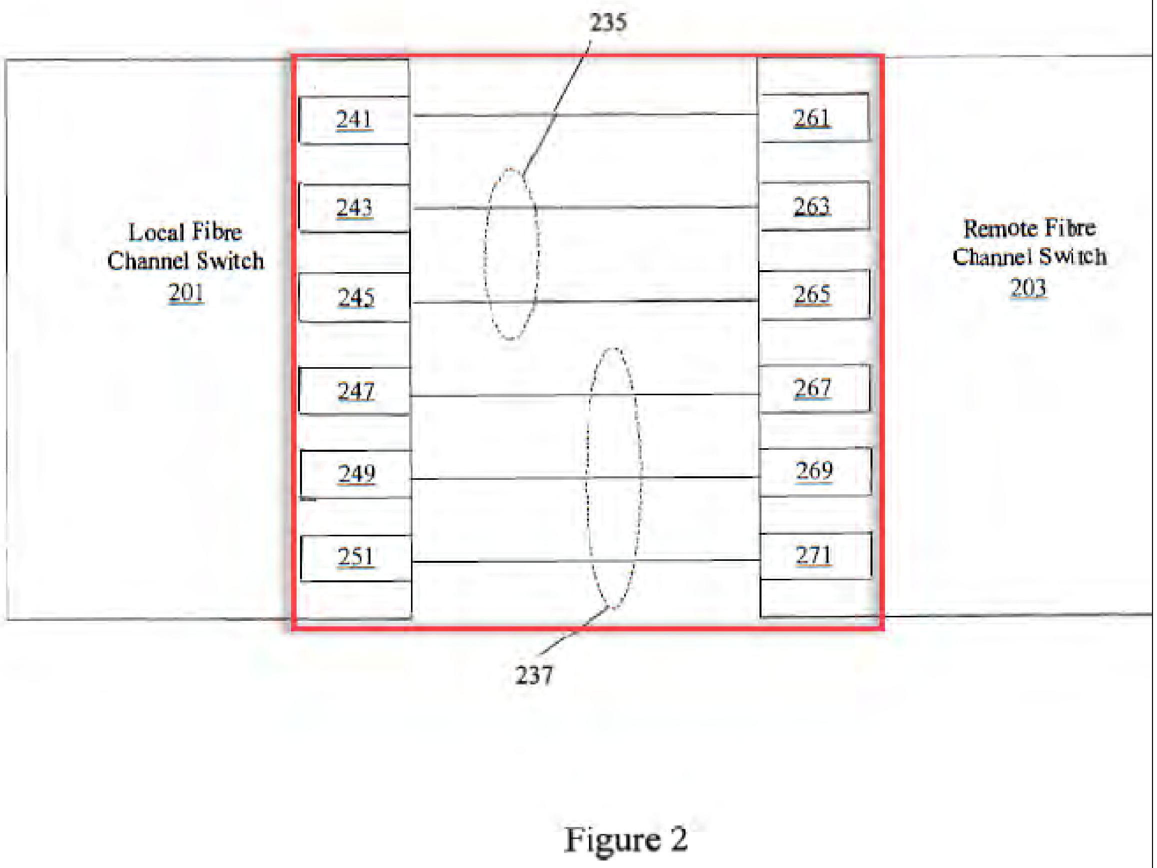
No.	’740 Patent Claim 4	Devi
		into a logical link. Physi-cal links are also referred to herein as physical interfaces and channels while logical links are also referred to herein as logical interfaces and port channels. For example, a local switch may be connected to a remote switch through four physical links. Instead of having to transmit data through a particular physical link, the physical links can be aggregated to form one or more logical links. In one example, all four physical links are aggregated into a single logical link. Instead of having data transmitted through a particular physical link, the data can merely be transmitted over a particular logical link without regard to the particular physi-cal interface used. Aggregating physical links into a logical link allows for higher aggregated bandwidth, load balancing, and link redundancy. For example, if a particular physical link fails or is overloaded, data can still be transmitted over the logical link.”) Ghosh at [0029] (“FIG. 2 is a diagrammatic representation showing links between two switches, such as two fibre channel switches shown in FIG. 1. A local fibre channel switch 201 includes local ports 241, 243, 245, 247, 249, and 251. A remote fibre channel switch 203 includes remote ports 261, 263, 265, 267, 269, and 271. Local port 241 is coupled to remote port 261 through an individual physical link or channel. Connected ports are also referred to herein as peer ports. Local port 243 is coupled to remote port 263 and local port 245 is coupled to remote port 265. The two resulting physical links are aggregated to form port channel 235. Local ports 247, 249, and 251 are coupled to remote ports 267, 269, and 271 respectively. The three resulting physical links are aggregated to form port channel 237.”) Ghosh at [0030] (“According to various embodiments, local fibre channel switch 201 and remote fibre channel switch both have associated identifiers. In some examples, the identifiers are globally unique identifiers such as a global switch world wide names (WWNs). Each local port 241, 243, 245, 247, 249, and 251 and each remote port 261,263,265,267, 269, and 271 can also be associated with identifiers. In some examples, the identifiers are port WWNs. The port WWNs are typically used for debugging or identifying the peer port in alert or warning messages. However, according to various embodiments, the techniques of the present invention use WWNs as globally unique identifiers to aggregate ports instead of using compatibility keys which are only locally unique. Compatibility keys are mechanisms typically used by other protocols such as Ethernet for aggregation.”)

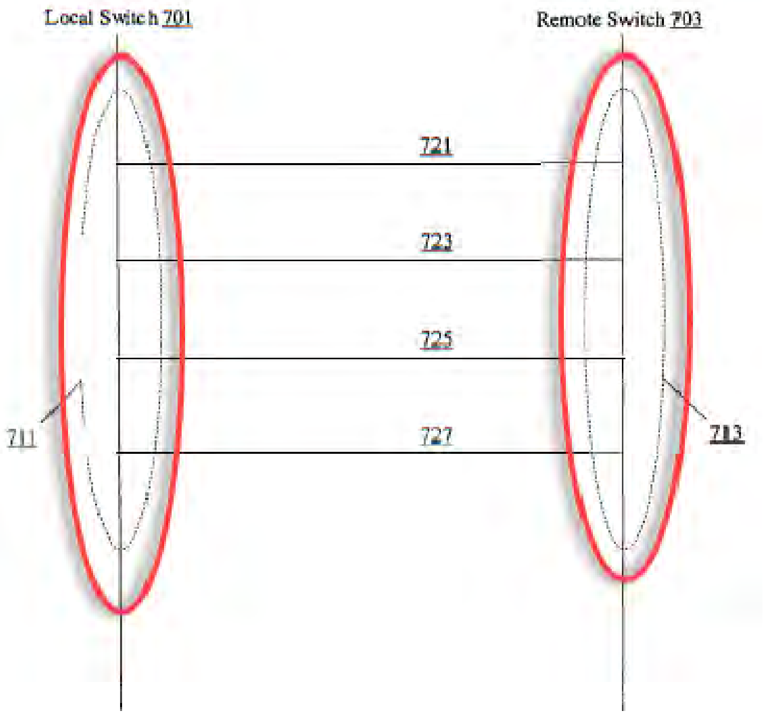
No.	’740 Patent Claim 4	Devi
		Ghosh at [0033] (“A variety of parameters can be used to aggregate physical ports. FIG. 3 is a flow process diagram showing one technique for aggregating physical ports into a logical port. And 301, it is determined if auto create functionality is enabled. According to various embodiments, auto create functionality allows automatic configuration and detection of compatible physical ports as well as aggregation into one or more logical ports. Auto creation does not require user intervention. In other examples, administrators can manu-ally arrange ports for aggregation.”) Ghosh at [0037] (“FIG. 4 is an exchange diagram showing one example of a bring up procedure used for a port creating a new port channel. A local switch 401 is coupled to a remote switch 403. The local switch 401 includes a physical port Al coupled to physical port Bl included in remote switch 403. When two peer ports Al and Bl are being aggregated into a port channel, the peer switches 401 and 403 typically already know the world wide names of the individual physical peer ports. However, the peer switches only know the world wide name of their own logical port or port channel. That is, both switches have the individual physical link configured, but the link is not yet part of a port channel. At 421, a local switch 401 sends a synchronize (sync) message 411 to the remote switch 403 to begin the process of creating a port channel including ports Al and Bl.”) Ghosh at [0038] (“In some examples, the sync message 411 includes a local port channel identifier and a remote port channel identifier. In one particular example, the local port channel identifier is set to the world wide name of the local port channel assigned by the local switch 401. The remote port channel identifier is left blank to indicate that the port Al is being aggregated as part of a new port channel. The sync message 411 can also include other parameters such as channel status, channel model, or channel intent.”) Ghosh at [0042] (“When two peer ports A2 and B2 are aggregated into a port channel Cl, the peer switches 501 and 503 typically already know the world wide names of the indi-vidual physical peer ports A2 and B2 as well as the world wide name information of the port channel Cl. Conse-quently, the port channel is already successfully established. According to various embodiments, local switch 501 and remote switch 503 perform parameter checking to ensure

No.	’740 Patent Claim 4	Devi
		that the new physical port A2 and B2 can be safely added to the existing port channel Cl. At 521, a local switch can check configuration parameters to ensure that physical ports Al and A2 at the local switch 501 are compatible. The compatibility checking can be performed anytime. In some examples, compatibility checking is checked before a local switch 501 sends a synchronize (sync) message 511 to the remote switch 503 to begin the process of aggregating ports A2 and B2 into the port channel.) Ghosh at [0043] (“In some examples, the sync message 511 includes local port channel identifier and a remote port channel identifier. In one particular example, the local port channel identifier is set to the world wide name of the local port channel assigned by the local switch 501. The remote port channel identifier is filled with the existing port channel identifier to indicate that the port A2 is being aggregated into existing port channel C2. The sync message 511 can also include other parameters such as channel status, channel model, or channel intent.”) Ghosh at [0044] (“At 531, remote switch 503 uses the information received from the local switch 501 to verify port B2 is compatible with other port in port channel C2. In one example, configuration parameters associated with B2 are checked against configuration parameters associated with Bl. The remote switch 503 can also check if the port B2 is already assigned to a different port channel. If the port B2 is compatible with port Bl, the remote switch 503 can proceed and send a sync accept message 513 in response to the sync message 511 to indicate that the port B2 can be aggregated into the port channel. The sync accept message indicates that a port channel can now be modified. At 523, local switch 501 uses the information to update its own port channel database. However, the port channel may not yet be fully operational until the hardware configuration is completed. The local switch 501 continues hardware configuration such as line card configuration to make the port A2 part of the port channel Cl. An acknowledgment 527 is sent and received by remote switch 503 at 529. In some examples, the local switch 501 sends a commit signal 515 when hardware configuration is complete.”) Ghosh at [0045] (“The remote switch 503 receives the commit signal at 533 and begins its own hardware configuration. On completion of its hardware configuration, remote switch 503

No.	’740 Patent Claim 4	Devi
		sends out a commit accept signal 517 to indicate to local switch 501 that hardware configuration is completed. According to various embodiments, local switch 501 receives the commit accept signal 517 and notifies relevant applications that the port channel is now fully operational at 525 and that port A2 has been aggregated into port channel Cl. The local switch 501 can also send out an acknowledge message 519. When the remote switch 503 receives the acknowledge, it notifies relevant applications that the port channel is operational at 535 and that port B2 has been aggregated into port channel Cl. In one embodiments, the techniques of the present invention contemplate using a two phase SYNC and COMMIT mechanism similar to the mechanism used in EPP.”) Ghosh at [0046] (“FIGS. 4 and 5 show examples of ports being aggregated into a port channel. At a particular switch, ports can be selected for aggregation into a port channel in a variety of manners. FIG. 6 is an exchange diagram showing automatic selection of ports at a switch for aggregation into a port channel. A local switch 601 is coupled to a remote switch 603. In one example, the local switch 601 includes physical ports Al, A2, A3, and A4 while remote switch 603 includes physical ports Bl, B2, B3, and B4. No port chan-nels have been formed.”) Ghosh at [0049] (“At 631, remote switch 603 uses the information received from the local switch 601 to verify that the peer ports of Al, A2, and A4 are compatible. That is, ports Bl, B2, and B4 are checked for compatibility. In one example, only ports Bl and B2 may be compatible, and consequently only ports Al, A2, Bl, and B2 can be included in the port channel. In another example, ports Bl, B2, and B4 are compatible, so ports Al, A2, A4, Bl, B2, and B4 can be aggregated into port channel Cl. According to various embodiments, if the port B2 is compatible with port Bl, the remote switch 603 can proceed and send a sync accept message 613 in response to the sync message 611 to indicate that the port B2 can be aggregated into the port channel. It should be noted that remote switch 603 can send a list indicating that ports B2 and B4 are compatible with Bl. However, the remote switch 603 sends only one compatible port B2 back for several reasons, and in the process of selection compatible port channels get priority over com-patible individual ports.”) Ghosh at [0050] (“One reason is that aggregation mechanisms and techniques can be implemented more elegantly by handling ports on an individual basis. Any individual port

No.	’740 Patent Claim 4	Devi
		will either start a new port channel, be added to an existing port channel, or operate stand alone. There is no need to keep track of groups of ports to be aggregated. Another reason is that fewer ports need to be locked if only a single port is being aggregated at any one time. The sync accept message indicates that a port channel can now be modified. At 623, local switch 601 receives the information and recognizes that Al and A2 can now be aggregated into port channel Cl. However, the port channel may not yet be fully operational until the hardware configuration is completed. An acknowl-edgment 627 is sent and received by remote switch 603 at 629. In some examples, the local switch 601 sends a commit signal 615 when hardware configuration is complete.”) Ghosh at [0051] (“The remote switch 603 receives the commit signal at 633 to create port channel Cl including ports Bl and B2. Hardware configuration can now be performed. On comple-tion of its hardware configuration, remote switch 603 sends out a commit accept signal 617 to indicate to local switch 601 that hardware configuration is completed. According to various embodiments, local switch 601 receives the commit accept signal 617 and notifies relevant applications that the port channel is now fully operational at 625 and that ports Al and A2 have been aggregated into port channel Cl. The local switch 601 can also send out an acknowledge message 619. When the remote switch 603 receives the acknowledge, it notifies relevant applications that the port channel is fully operational at 635 and that ports Bl and B2 have been aggregated into port channel Cl.”) Ghosh at [0053] (“FIG. 7 is a diagrammatic representation showing synchronous aggregation of ports into a port channel. A local switch 701 is coupled to a remote switch 703 through links 721, 723, 725, and 727. According to various embodiments, the links are being aggregated into port channel 711 at the local switch 701 and port channel 713 at the remote switch 703 in a synchronous manner. That is the peer ports corre-sponding to each link are brought up in the same order at both the local switch 701 and the remote switch 703.”) Ghosh at Figure 2 (annotation added)

No.	’740 Patent Claim 4	Devi
		 235 241 243 245 247 249 251 261 263 265 267 269 271 Local Fibre Channel Switch 201 Remote Fibre Channel Switch 203 237 Figure 2 Ghosh at Figure 7 (annotation added)

No.	’740 Patent Claim 4	Devi
		 Local Switch 701 Remote Switch 703 721 723 725 727 711 713 Figure 7

No.	’740 Patent Claim 5	Devi
5[preamble]	A method for communication, comprising:	Devi discloses a method for communication. *See supra at* 1[preamble].
5[a]	coupling a network node to one or more interface modules	Devi discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel.

No.	’740 Patent Claim 5	Devi
	using a first group of first physical links arranged in parallel;	*See supra at* 1[a].
5[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Devi discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra at* 1[c].
5[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	Devi discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra at* 1[e].
5[d]	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	Devi discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. *See supra at* 1[f].
5[e]	sending the data frame over the selected first and second physical links,	Devi discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].

No.	’740 Patent Claim 5	Devi
5[f]	coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node.	Devi discloses coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node. For example, Devi discloses aggregating multiple physical links 108 into an aggregated link 116, which couples sources and destinations to the cards of a switch that increases all link activity. Devi at Abstract (“A method perfomed by an ingress card includes receiving a packet with a destination identifier, and determining an aggregated link associated with the destination identifier that includes multiple egress ports. The method further includes determining a destination port from the egress ports of the aggregated link, and communicating the packet to the des-tination port.”) Devi at [0003] (“As telecommunication networks handle more and more traffic, new methods are constantly being developed that allow the networks to process larger flows of informa-tion. Link aggregation is one example. In link aggregation, several physical links are aggregated to appear as one logical link to a telecommunications system. This aggregation allows all of the links to be used actively, rather than having particular links reserved. Thus, link aggregation provides more efficient use of network resources and better load balancing. The tradeoff, however, is that link aggregation may also require substantial additional processing and/or hardware to implement the aggregated links.”) Devi at [0004] (“In accordance with the present invention, the dis-advantages and problems associated with implementing link aggregation in switches have been substantially reduced or eliminated. In particular, certain embodiments of the present invention provide a method and system for implementing link aggregation using distributed processing. Certain embodiments of the present invention are also compliant with the IEEE standards for link aggregation, set out in IEEE Standard 802.3.”) Devi at [0005] (“In accordance with one embodiment of the present invention, a method performed by an ingress card includes receiving a packet comprising a destination identifier.

No.	’740 Patent Claim 5	Devi
		The method also includes determining an aggregated link asso-ciated with the destination identifier, and determining a destination port from the egress ports of the aggregated link. The method further includes communicating the packet to the destination port.”) Devi at [0006] (“In accordance with another embodiment of the present invention, a method performed by an ingress card includes receiving a packet and determining egress links for the packet. The method also includes determining whether each link is an aggregated link with multiple egress ports or a non-aggregated link with a single egress port. For each aggregated link, a destination port is determined from the egress ports, and the packet is communicated to the desti-nation port. For each non-aggregated link, a copy of the packet is communicated to the single egress port.”) Devi at [0009] (“Other important technical advantages of certain embodiments of the present invention include load balanc-ing. The tables or other information used by the ingress card to determine a destination port for packets may also include additional usage information that allows the ingress card to determine a destination port from the egress ports in an aggregated link. This means that rather than using one link to the exclusion of others, loads may be distributed among several links in an aggregated link in a more balanced fashion. Particular embodiments of the present invention may have some, all or none of the enumerated technical advantages. Still other important technical advantages will be apparent to one skilled in the art from the following figures, description, and claims.”) Devi at [0023] (“Multiple physical links 108 to a particular destina-tion 112 may be aggregated to form an aggregated link 116. Aggregated link 116 represents multiple physical links 108 that are addressed by a single logical address, such as a medium access controller (MAC) address, an Internet pro-tocol (IP) address, or other suitable address or identifier, or otherwise treated as a single logical link between switch 104 and a destination 112. Because aggregated link 116 includes multiple physical links 108 that go to the same destination 112, a packet sent to destination 112 need only be sent to one port 110 corresponding to one physical link 108 of aggre-gated link 116. One technical advantage of certain embodi-ments of the present invention is that ingress cards 106 may communicate packets to ports 110 in some alternating fash-ion, such as round robin, random selection, pseudo-random selection

No.	’740 Patent Claim 5	Devi
		using hash tables, or any other selection technique. This allows the load on any particular physical link 108 to be balanced with the loads on other physical links 108 in aggregated link 116.”) Devi at [0032] (“Port information 218 refers to any form of data that identifies ports 110 associated with a particular link. For example, port information 218 for a non-aggregated link 108 identifies the single egress port 110 for the non-aggregated link 108. Port information 218 for an aggregated link 116 identifies the multiple ports 110 coupled to the physical links 108 of aggregated link 116, which may include ports 110 on a single card 106 or multiple cards 106. Port information 218 may also identify which card 106 is coupled to each port 110. Port information 218 contemplates any suitable arrangement of data, such as a table, database or any other format. Card 106 may update port information 218 as network conditions change, e.g., when a port 110 or physical link 108 fails.”) Devi at [0034] (“If a destination has been previously learned, card 106 determines the appropriate egress link (unicast) or links (multicast) using destination information 216. For each aggregated link, card 106 then uses port information 218 to determine a destination egress port 110 of the aggregrated egress link 116. Card 106 then forwards the packet to the destination egress port 110. For non-aggregated links, card 106 communicates the packet to the single egress port 110 of the non-aggregated link 108.”) Devi at [0041] (“If the link is an aggregated link 116, card 106 determines a destination port 110 of aggregated link 116 using the information in port information 218 at step 320. Card 106 may select a particular destination port 110 from the multiple ports 110 of the aggregated link using random selection, load balancing, pseudo-random selection or hash-ing using part of the packet information, historical tracking, round robin, or any other method of selection. Card 106 then communicates the packet to the destination port 110 at step 322.) Devi at [0048] (“If the selected link is aggregated, card 106 then determines whether the aggregated link 116 is isolated using port information 218 at step 414. "Isolated" means that aggregated link 116 does not have any ports 110 that share an egress card 106 with non-aggregated link 108. Since card 106 must communicate a copy of the packet to each

No.	’740 Patent Claim 5	Devi
		non-aggregated link 108 to reach the respective destinations 112 of the non-aggregated links 108, any card 106 that includes a port 110 of a non-aggregated link 108 must receive a copy of the packet. Because card 106 replicates packets, any aggregated link 116 that shares a card 106 with a non-aggregated link 108 receives a copy of the packet as well. Therefore, only aggregated links 108 that do not share a card 106 with a non-aggregated link 108 need to receive a copy of the packet.”) Devi at [0049] (“If the egress link is an aggregated isolated link, card 106 communicates the packet to destination port 110 of aggregated link 116. Card 106 determines destination port 110 for the packet at step 416. Card 106 may use any suitable method to determine destination port 110, including any of the methods described in conjunction with FIGS. 2 and 3. Card 106 then communicates the packet to destination port 110 at step 418.”)

No.	’740 Patent Claim 6	Devi
6	The method according to claim 1, wherein coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the	Devi discloses the method according to claim 1, wherein coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node. For example, Devi discloses components of switch 104 and cards 106 that include ingress processor 204 and traffic manager 206 that facilitates efficient routing of packets, which may include multiplexing and demultiplexing network traffic. Devi further discloses that ingress processor 204 and traffic manager 206 may represent any combination of hardware or software for routing packets. A person of ordinary skill in the art would understand that in processing and managing traffic, the ingress processor 104 and traffic manager 206 would multiplex and demultiplex incoming and outgoing network traffic. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Devi is found to not meet this limitation, wherein coupling each of the one or more interface modules to the communication network comprises at least one of

No.	’740 Patent Claim 6	Devi
	communication network to the network node.	multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the *See supra* Claim 1. Devi at [0018] (“Switch 104 sends and receives packets. Switch 104 may represent any suitable device, including an Ethernet switch, router, hub, or any other suitable hardware and/or software configured to receive packets and communicate them to other devices. Switch 104 includes cards 106 coupled to physical links 108 and a backplane 114 that represents hardware and/or software allowing cards 106 in switch 104 to exchange information with one another.”) Devi at [0019] (“Cards 106 represent separate components of hard-ware and/or software in switch 104 that exchange packets with network 102. Cards 106 may include traditional inter-face cards, as well as any other component, module, or part of switch 104 capable of independently receiving packets and communicating those packets to other components of switch 104. Each card 106 has one or more ports 110 coupled to physical links 108. Cards 106 may exchange information and packets with one another using backplane 114. Each card 106 includes sufficient processing capability to identify a port 110 in another card 106 in switch 104 and to communicate a packet to that port 110.”) Devi at [0024] (“FIG. 2 shows a card 106 in more detail. Card 106 couples to backplane 114 of switch 104 using ingress backplane interface (IBI) 210 and egress backplane interface (EBI) 212. Card 106 couples to network 102 through physi-cal layer devices 202, including ports 110 coupled to links 108. Card 106 includes an ingress processor 204, an egress processor 214, a traffic manager 206, and a memory 208. Card 106 may exchange packets from network 102 or backplane 114. Packets received from network 102 are considered to be on the "ingress" side, while packets received from the backplane 114 are considered to be on the "egress side" of card 106. Thus, at the level of card 106, the terms "ingress" and "egress" refer to the function of those components with respect to receiving packets from network 102 or backplane 114.”)

No.	’740 Patent Claim 6	Devi
		Devi at [0026] (“Ingress processor 204 and traffic manager 206 represent any combination of hardware and/or software that allows card 106 to route packets received from physical layer device 202. In a particular embodiment, ingress pro-cessor 204 handles routing for unicast packets, while traffic manager 206 handles routing for multicast packets. "Uni-cast" refers to any packet directed to a single destination, while "multicast" refers to packets that have multiple des-tinations.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Devi in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 6 of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Wiher discloses multiplexing and demultiplexing circuitry to transmit and receive ATM data cells over a data link between the ATM network and network access equipment. Wiher at 3:43-65 (“In general, in another aspect, the invention features an apparatus for communicating data cells between a data link and a backplane. The apparatus includes transceiver circuitry to transmit and receive data cells over a data link and a plurality of backplane interfaces each including at least one cell signal terminal. Each of the backplane interface is coupled to a backplane interconnection circuit. Each backplane interconnection circuit transmits and receives cells over the cell signal terminals of its associated backplane interface. The apparatus also includes de-multiplexing circuitry coupling the transceiver circuitry to each of the backplane interconnection circuits. The de-multiplexing circuitry receives a data cell from the transceiver circuitry, select a backplane interconnection circuit associated with the data cell, and provide the data cell to the selected backplane interconnection circuit for transmission over the cell signal terminals of the associated backplane interface. The apparatus also includes multiplexing circuitry coupling the plurality of backplane interconnection circuits to the transceiver circuitry. The multiplexing circuitry

No.	’740 Patent Claim 6	Devi
		receives data cells from each of the backplane interconnection circuits and provide the received data cells to the transceiver circuitry.”) Wiher at 3:66-4:22 (“Implementations of the invention may include one or more of the following features. The backplane interconnection circuits may independently receive and transmit data cells over the plurality of backplane interfaces. The de-multiplexing circuitry may select a backplane interface based on data in the header field of the data cell. The apparatus may include header translation circuitry to alter header data in cells sent between the plurality of backplane interfaces and the transceiver circuitry. Each of the plurality of backplane interfaces may include separate terminals to receive cells and separate terminals to transmit cells. The terminals to transmit cells may include a first and second control terminal and at least one outgoing cell data terminal. A backplane interface's backplane interconnection circuitry may accepts a signal on the first control terminal as indicating that a cell may be sent over the interface, asserts a 15 signal on the second control terminal to indicate that a cell is being transmitted, and transmits data bits of the cell on the outgoing cell data terminal. Each backplane interface may include a single outgoing cell data terminal and each bit of the cell may be serially transmitted over the single outgoing cell data terminal. Each backplane interface may include multiple outgoing cell data terminals and bits of the cell may be sent in parallel over the eight outgoing cell data terminals.”) As another example, Lebizay discloses an optical add/drop multiplexer that multiplexers and demultiplexes data packets sent between the network boards and network. Lebizay at [0043] (“InfiniBand offers link layer Virtual Lanes (VLs) to support multiple logical channels (i.e. multiplexing) on the same physical link. Infiniband offers up to 16 virtual lanes per link. VLs provide a mechanism to avoid head-of-line blocking and the ability to support Quality of Service (QoS). The difference between a Virtual Lane and a Service Level (SL) is that a Virtual Lane is the actual logical lane (mul-tiplexed) used on a given point-to-point link. The Service Level stays constant as a packet traverses the fabric, and specifies the desired service level within a subnet. The SL (AF, EF or BE) is included in the link header, and each switch maps the SL to a VL supported by the destination link. A switch supporting a limited number of virtual lanes will map the SL field to a VL it supports. Without

No.	’740 Patent Claim 6	Devi
		preserving the SL, the desired SL (AF, EF or BE) would be lost in this mapping, and later in the path, a switch supporting more VLs would be unable to recover finer granularity of SLs between two packets mapped to the same VL.”) Lebizay at [0050] (“The issue with using a ring, however, is how to map the addressing of multiple boards across these fibers. One solution is to employ Wavelength Division Multiplex-ing (WDM). A WDM optical mesh defines a meshed-topology in the wavelength space as opposed to the physical fiber space. By utilizing multiple discrete lambda-waves as optical carriers such that by meshing dedicated optical wavelengths between every two boards, layer 2 protocols are eliminated, thereby creating a dramatic improvement in the efficiency of the transport. Today, every packet transport requires a protocol that allows the end point (and interme-diate points) to decipher the intended path (or consumer) of the packet. This protocol increases the amount of overhead required in the packet bus, allowing less room for actual data to be sent. By moving the protocol into the wavelength of the actual optical signal, the destination is implied by the wavelength and no additional bandwidth needs to be sur-rendered on the signal to provide this information. This makes the efficiency of the transport better and also speeds the routing of the packet through the network. In addition, the use of optical interconnects in a backplane environment greatly increases chassis bandwidth as well as reducing electrical radiation that often accompanies copper intercon-nects. The components involved include an optical back-plane in a physical ring topology, and the necessary trans-mitters and receivers for the size of the installation (i.e., number of slots in the chassis). In addition, optical add/drop multiplexer devices are required.”)

No.	’740 Patent Claim 7	Devi
7	The method according to claim 1, wherein selecting the first and second physical links comprises balancing a frame data rate among	Devi discloses the method according to claim 1, wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the first and second physical links. For example, Devi discloses determining the egress links of a destination port using a load balancing algorithm that includes balancing distribution of packets.

No.	’740 Patent Claim 7	Devi
	at least some of the first and second physical links.	*See supra* Claim 1 Devi at [0003] (“As telecommunication networks handle more and more traffic, new methods are constantly being developed that allow the networks to process larger flows of informa-tion. Link aggregation is one example. In link aggregation, several physical links are aggregated to appear as one logical link to a telecommunications system. This aggregation allows all of the links to be used actively, rather than having particular links reserved. Thus, link aggregation provides more efficient use of network resources and better load balancing. The tradeoff, however, is that link aggregation may also require substantial additional processing and/or hardware to implement the aggregated links.”) Devi at [0009] (“Other important technical advantages of certain embodiments of the present invention include load balanc-ing. The tables or other information used by the ingress card to determine a destination port for packets may also include additional usage information that allows the ingress card to determine a destination port from the egress ports in an aggregated link. This means that rather than using one link to the exclusion of others, loads may be distributed among several links in an aggregated link in a more balanced fashion. Particular embodiments of the present invention may have some, all or none of the enumerated technical advantages. Still other important technical advantages will be apparent to one skilled in the art from the following figures, description, and claims.”) Devi at [0023] (“Multiple physical links 108 to a particular destina-tion 112 may be aggregated to form an aggregated link 116. Aggregated link 116 represents multiple physical links 108 that are addressed by a single logical address, such as a medium access controller (MAC) address, an Internet pro-tocol (IP) address, or other suitable address or identifier, or otherwise treated as a single logical link between switch 104 and a destination 112. Because aggregated link 116 includes multiple physical links 108 that go to the same destination 112, a packet sent to destination 112 need only be sent to one port 110 corresponding to one physical link 108 of aggre-gated link 116. One technical advantage of certain embodi-ments of the present invention is that ingress cards 106 may communicate packets to ports 110 in some alternating fash-ion, such as round robin, random selection, pseudo-random selection

No.	’740 Patent Claim 7	Devi
		using hash tables, or any other selection technique. This allows the load on any particular physical link 108 to be balanced with the loads on other physical links 108 in aggregated link 116.”) Devi at [0041] (“If the link is an aggregated link 116, card 106 determines a destination port 110 of aggregated link 116 using the information in port information 218 at step 320. Card 106 may select a particular destination port 110 from the multiple ports 110 of the aggregated link using random selection, load balancing, pseudo-random selection or hash-ing using part of the packet information, historical tracking, round robin, or any other method of selection. Card 106 then communicates the packet to the destination port 110 at step 322.”) Devi at Claim 4 (“The method of claim 1, wherein the step of determining a destination port comprises determining the destination port according to a load-balancing algorithm that distributes traffic among the egress ports of the aggregated link.”)

No.	’740 Patent Claim 8	Devi
8	The method according to claim 1, wherein selecting the first and second physical links comprises applying a mapping function to the at least one of the frame attributes.	Devi discloses the method according to claim 1, wherein selecting the first and second physical links comprises applying a mapping function to the at least one of the frame attributes. For example, Devi discloses determining egress links associated with a destination port over which to send packets and frames with destination identifier 215, which “may include may include a network address for destination 112, a logical address for an egress link, or any other information useful for identifying the destination of a packet. Destination identifier 215 contemplates any information determinable from a packet that allows card 106 to identify one or more destinations 112 for the packet.” Devi further discloses using hashing or other selection methods, which may include a mapping function, to select the egress links of a particular destination port based on the information in the destination identifier 215 of the packet. A person of ordinary skill in the art would understand that hashing or other selection methods would include a mapping function.

No.	’740 Patent Claim 8	Devi
		See supra Claim 1 Devi at [0005] (“In accordance with one embodiment of the present invention, a method performed by an ingress card includes receiving a packet comprising a destination identifier. The method also includes determining an aggregated link asso-ciated with the destination identifier, and determining a destination port from the egress ports of the aggregated link. The method further includes communicating the packet to the destination port.”) Devi at [0006] (“In accordance with another embodiment of the present invention, a method performed by an ingress card includes receiving a packet and determining egress links for the packet. The method also includes determining whether each link is an aggregated link with multiple egress ports or a non-aggregated link with a single egress port. For each aggregated link, a destination port is determined from the egress ports, and the packet is communicated to the desti-nation port. For each non-aggregated link, a copy of the packet is communicated to the single egress port.”) Devi at [0007] (“Important technical advantages of certain embodi-ments of the present invention include implementing link aggregation using distributed processing among ingress components. For example, individual ingress cards can determine a destination port for a packet, eliminating the need for a separate processing stage implemented in hard-ware and/or software to separately determine the destination port from a packet after the ingress card forwards the packet to a link. This improves the efficiency and speed of packet forwarding in a switch.”) Devi at [0008] (“Yet another important technical advantage of cer-tain embodiments is a distributed architecture that may be used to process a variety of packet traffic. For example, certain embodiments of an ingress card can forward unicast, multicast, and bridging traffic. Particular embodiments may process packets from different protocols as well.”) Devi at [0009] (“Other important technical advantages of certain embodiments of the present invention include load balanc-ing. The tables or other information used by the ingress card to

No.	’740 Patent Claim 8	Devi
		determine a destination port for packets may also include additional usage information that allows the ingress card to determine a destination port from the egress ports in an aggregated link. This means that rather than using one link to the exclusion of others, loads may be distributed among several links in an aggregated link in a more balanced fashion. Particular embodiments of the present invention may have some, all or none of the enumerated technical advantages. Still other important technical advantages will be apparent to one skilled in the art from the following figures, description, and claims.”) Devi at [0015] (“FIG. 1 shows a system 100 that includes a switch 104 coupled to a network 102. Using network 102 and switch 104, packets are communicated from sources 103 to destinations 112. Components of switch 104 are referred to as "ingress" components when receiving packets from sources 103 and "egress" components when sending packets to destinations 112.”) Devi at [0016] (“Network 102 represents any suitable structure for communicating packets, cells, frames, segments, or other portions of data (generally referred to as "packets"). Net-work 102 may represent the Internet, extranet, local area network (LAN), synchronous optical network (SONET), wide area network (WAN), the public switched telephone network (PSTN), or any other suitable network for commu-nicating information. Network 102 may include routers, switches, hubs, endpoints, or any other network device that communicates information. Network 102 contemplates any number or arrangement of components that exchange infor-mation.”) Devi at [0018] (“Switch 104 sends and receives packets. Switch 104 may represent any suitable device, including an Ethernet switch, router, hub, or any other suitable hardware and/or software configured to receive packets and communicate them to other devices. Switch 104 includes cards 106 coupled to physical links 108 and a backplane 114 that represents hardware and/or software allowing cards 106 in switch 104 to exchange information with one another.) Devi at [0023] (“Multiple physical links 108 to a particular destina-tion 112 may be aggregated to form an aggregated link 116. Aggregated link 116 represents multiple physical

No.	’740 Patent Claim 8	Devi
		links 108 that are addressed by a single logical address, such as a medium access controller (MAC) address, an Internet pro-tocol (IP) address, or other suitable address or identifier, or otherwise treated as a single logical link between switch 104 and a destination 112. Because aggregated link 116 includes multiple physical links 108 that go to the same destination 112, a packet sent to destination 112 need only be sent to one port 110 corresponding to one physical link 108 of aggre-gated link 116. One technical advantage of certain embodi-ments of the present invention is that ingress cards 106 may communicate packets to ports 110 in some alternating fash-ion, such as round robin, random selection, pseudo-random selection using hash tables, or any other selection technique. This allows the load on any particular physical link 108 to be balanced with the loads on other physical links 108 in aggregated link 116.”) Devi at [0030] (“Destination identifier 215 represents information identifying an intermediate or final destination 112 for the packet. Destination identifier 215 may include a network address for destination 112, a logical address for an egress link, or any other information useful for identifying the destination of a packet. Destination identifier 215 contem-plates any information determinable from a packet that allows card 106 to identify one or more destinations 112 for the packet.”) Devi at [0031] (“Destination information 216 associates destination identifiers 215 in packets with egress links. Destination information 216 may be organized in any suitable fashion, including one or more tables, files, databases, or other form of organization. Card 106 uses destination information 216 to determine an egress link or links using destination iden-tifier 215 from a packet. For example, a multicast packet may include a multicast group identifier instead of indi-vidual link addresses. Ingress card 106 receives the packet, looks up the multicast group identifier, and determines the egress links associated with the multicast group. In another example, the packet may include a final destination address (such as an IP address for a receiving device) rather than a link address from switch 104. In that case, ingress link 106 uses destination information 216 to determine an egress link to the final destination. Destination information 216 may include one or more layers as well. For example, destination information 216 may include a first table relating a multicast group to destination IP addresses, and a second table relating the destination IP addresses to logical link addresses.

No.	’740 Patent Claim 8	Devi
		Card 106 may also update destination information 216 as new or additional information becomes available, such as when card 106 receives an acknowledgement from a destination 112 indicating that a packet was received.”) Devi at [0033] (“In operation, card 106 receives a packet from network 102. Ingress card 106 determines the destination for the particular packet and consults destination information 216 to see if that destination is associated with a particular egress link. Some packets may have a destination identifier 215 that card 106 has not yet learned, and so destination information 216 does not include an egress link or links associated with destination identifier 215 of the packet. In such cases, card 106 may flood all available egress links so that the packet will arrive at its proper destination, and that destination will send an acknowledgment back to card 106. To flood destinations, card 106 determines logical links coupled to switch 104 using port information 218. Card 106 communicates the packet to these links, excluding the link from which the packet was received. In a particular embodi-ment, flooded cards will replicate the packet to all ports within that card, so card 106 need only communicate the packet once to a particular egress card 106.”) Devi at [0037] (“Card 106 receives a packet at step 302. At step 304, card determines destination identifier 215 for the packet and uses destination information 216 to determine if card 106 has previously learned the destination identifier 215. If the destination has not been previously learned, card 106 floods all egress links at step 306, shown in more detail in FIG. 4 described below.”) Devi at [0038] (“If destination identifier 215 of the packet has been learned, card 106 next determines if destination identifier 215 identifies a multicast or a unicast destination at step 308. If the packet is a multicast packet, card 106 selects one of the destinations for the packet using destination information 216 at step 310. On the other hand, if the packet is a unicast packet, there is only one destination, so there is no need to perform a selection process.”) Devi at [0041] (“If the link is an aggregated link 116, card 106 determines a destination port 110 of aggregated link 116 using the information in port information 218 at step 320. Card 106 may select a particular destination port 110 from the multiple ports 110 of the aggregated

No.	’740 Patent Claim 8	Devi
		link using random selection, load balancing, pseudo-random selection or hash-ing using part of the packet information, historical tracking, round robin, or any other method of selection. Card 106 then communicates the packet to the destination port 110 at step 322.”) Devi at [0048] (“If the egress link is an aggregated isolated link, card 106 communicates the packet to destination port 110 of aggregated link 116. Card 106 determines destination port 110 for the packet at step 416. Card 106 may use any suitable method to determine destination port 110, including any of the methods described in conjunction with FIGS. 2 and 3. Card 106 then communicates the packet to destination port 110 at step 418.”) Devi at [0049] (“Card 106 then determines if there are any remain-ing egress links at step 420. If there are egress links remaining, then card 106 selects a new link to process at step 406. If there are no remaining links, then card 106 waits for an acknowledgement from one of the destinations that the packet was received. Card 106 receives the acknowledge-ment at step 422, and determines the link from which the acknowledgement was received at step 424. Card 106 then updates destination information 216 by associating the source link with destination identifier 215 at step 426. When the packet is a multicast packet, card 106 waits for acknowl-edgements from any remaining destinations at step 428.”)

No.	’740 Patent Claim 9	Devi
9	The method according to claim 8, wherein applying the mapping function comprises applying a hashing function.	Devi discloses the method according to claim 8, wherein applying the mapping function comprises applying a hashing function. For example, Devi discloses using hashing or other selection methods to select the egress links of a particular destination port based on the information in the destination identifier 215 of the packet. *See supra* Claim 8

No.	’740 Patent Claim 9	Devi
		Devi at [0005] (“In accordance with one embodiment of the present invention, a method performed by an ingress card includes receiving a packet comprising a destination identifier. The method also includes determining an aggregated link asso-ciated with the destination identifier, and determining a destination port from the egress ports of the aggregated link. The method further includes communicating the packet to the destination port.”) Devi at [0006] (“In accordance with another embodiment of the present invention, a method performed by an ingress card includes receiving a packet and determining egress links for the packet. The method also includes determining whether each link is an aggregated link with multiple egress ports or a non-aggregated link with a single egress port. For each aggregated link, a destination port is determined from the egress ports, and the packet is communicated to the desti-nation port. For each non-aggregated link, a copy of the packet is communicated to the single egress port.”) Devi at [0007] (“Important technical advantages of certain embodi-ments of the present invention include implementing link aggregation using distributed processing among ingress components. For example, individual ingress cards can determine a destination port for a packet, eliminating the need for a separate processing stage implemented in hard-ware and/or software to separately determine the destination port from a packet after the ingress card forwards the packet to a link. This improves the efficiency and speed of packet forwarding in a switch.”) Devi at [0008] (“Yet another important technical advantage of cer-tain embodiments is a distributed architecture that may be used to process a variety of packet traffic. For example, certain embodiments of an ingress card can forward unicast, multicast, and bridging traffic. Particular embodiments may process packets from different protocols as well.”) Devi at [0009] (“Other important technical advantages of certain embodiments of the present invention include load balanc-ing. The tables or other information used by the ingress card to determine a destination port for packets may also include additional usage information that allows the ingress card to determine a destination port from the egress ports in an aggregated link. This means that rather than using one link to the exclusion of others, loads may be

No.	’740 Patent Claim 9	Devi
		distributed among several links in an aggregated link in a more balanced fashion. Particular embodiments of the present invention may have some, all or none of the enumerated technical advantages. Still other important technical advantages will be apparent to one skilled in the art from the following figures, description, and claims.”) Devi at [0015] (“FIG. 1 shows a system 100 that includes a switch 104 coupled to a network 102. Using network 102 and switch 104, packets are communicated from sources 103 to destinations 112. Components of switch 104 are referred to as "ingress" components when receiving packets from sources 103 and "egress" components when sending packets to destinations 112.”) Devi at [0016] (“Network 102 represents any suitable structure for communicating packets, cells, frames, segments, or other portions of data (generally referred to as "packets"). Net-work 102 may represent the Internet, extranet, local area network (LAN), synchronous optical network (SONET), wide area network (WAN), the public switched telephone network (PSTN), or any other suitable network for commu-nicating information. Network 102 may include routers, switches, hubs, endpoints, or any other network device that communicates information. Network 102 contemplates any number or arrangement of components that exchange infor-mation.”) Devi at [0018] (“Switch 104 sends and receives packets. Switch 104 may represent any suitable device, including an Ethernet switch, router, hub, or any other suitable hardware and/or software configured to receive packets and communicate them to other devices. Switch 104 includes cards 106 coupled to physical links 108 and a backplane 114 that represents hardware and/or software allowing cards 106 in switch 104 to exchange information with one another.) Devi at [0023] (“Multiple physical links 108 to a particular destina-tion 112 may be aggregated to form an aggregated link 116. Aggregated link 116 represents multiple physical links 108 that are addressed by a single logical address, such as a medium access controller (MAC) address, an Internet pro-tocol (IP) address, or other suitable address or identifier, or otherwise treated as a single logical link between switch 104 and a destination 112. Because

No.	’740 Patent Claim 9	Devi
		aggregated link 116 includes multiple physical links 108 that go to the same destination 112, a packet sent to destination 112 need only be sent to one port 110 corresponding to one physical link 108 of aggre-gated link 116. One technical advantage of certain embodi-ments of the present invention is that ingress cards 106 may communicate packets to ports 110 in some alternating fash-ion, such as round robin, random selection, pseudo-random selection using hash tables, or any other selection technique. This allows the load on any particular physical link 108 to be balanced with the loads on other physical links 108 in aggregated link 116.”) Devi at [0030] (“Destination identifier 215 represents information identifying an intermediate or final destination 112 for the packet. Destination identifier 215 may include a network address for destination 112, a logical address for an egress link, or any other information useful for identifying the destination of a packet. Destination identifier 215 contem-plates any information determinable from a packet that allows card 106 to identify one or more destinations 112 for the packet.”) Devi at [0031] (“Destination information 216 associates destination identifiers 215 in packets with egress links. Destination information 216 may be organized in any suitable fashion, including one or more tables, files, databases, or other form of organization. Card 106 uses destination information 216 to determine an egress link or links using destination iden-tifier 215 from a packet. For example, a multicast packet may include a multicast group identifier instead of indi-vidual link addresses. Ingress card 106 receives the packet, looks up the multicast group identifier, and determines the egress links associated with the multicast group. In another example, the packet may include a final destination address (such as an IP address for a receiving device) rather than a link address from switch 104. In that case, ingress link 106 uses destination information 216 to determine an egress link to the final destination. Destination information 216 may include one or more layers as well. For example, destination information 216 may include a first table relating a multicast group to destination IP addresses, and a second table relating the destination IP addresses to logical link addresses. Card 106 may also update destination information 216 as new or additional information becomes available, such as when card 106 receives an acknowledgement from a destination 112 indicating that a packet was received.”)

No.	’740 Patent Claim 9	Devi
		Devi at [0033] (“In operation, card 106 receives a packet from network 102. Ingress card 106 determines the destination for the particular packet and consults destination information 216 to see if that destination is associated with a particular egress link. Some packets may have a destination identifier 215 that card 106 has not yet learned, and so destination information 216 does not include an egress link or links associated with destination identifier 215 of the packet. In such cases, card 106 may flood all available egress links so that the packet will arrive at its proper destination, and that destination will send an acknowledgment back to card 106. To flood destinations, card 106 determines logical links coupled to switch 104 using port information 218. Card 106 communicates the packet to these links, excluding the link from which the packet was received. In a particular embodi-ment, flooded cards will replicate the packet to all ports within that card, so card 106 need only communicate the packet once to a particular egress card 106.”) Devi at [0037] (“Card 106 receives a packet at step 302. At step 304, card determines destination identifier 215 for the packet and uses destination information 216 to determine if card 106 has previously learned the destination identifier 215. If the destination has not been previously learned, card 106 floods all egress links at step 306, shown in more detail in FIG. 4 described below.”) Devi at [0038] (“If destination identifier 215 of the packet has been learned, card 106 next determines if destination identifier 215 identifies a multicast or a unicast destination at step 308. If the packet is a multicast packet, card 106 selects one of the destinations for the packet using destination information 216 at step 310. On the other hand, if the packet is a unicast packet, there is only one destination, so there is no need to perform a selection process.”) Devi at [0041] (“If the link is an aggregated link 116, card 106 determines a destination port 110 of aggregated link 116 using the information in port information 218 at step 320. Card 106 may select a particular destination port 110 from the multiple ports 110 of the aggregated link using random selection, load balancing, pseudo-random selection or hash-ing using part of the packet information, historical tracking, round robin, or any other method of selection. Card 106 then communicates the packet to the destination port 110 at step 322.”)

No.	’740 Patent Claim 9	Devi
		Devi at [0048] (“If the egress link is an aggregated isolated link, card 106 communicates the packet to destination port 110 of aggregated link 116. Card 106 determines destination port 110 for the packet at step 416. Card 106 may use any suitable method to determine destination port 110, including any of the methods described in conjunction with FIGS. 2 and 3. Card 106 then communicates the packet to destination port 110 at step 418.”) Devi at [0049] (“Card 106 then determines if there are any remain-ing egress links at step 420. If there are egress links remaining, then card 106 selects a new link to process at step 406. If there are no remaining links, then card 106 waits for an acknowledgement from one of the destinations that the packet was received. Card 106 receives the acknowledge-ment at step 422, and determines the link from which the acknowledgement was received at step 424. Card 106 then updates destination information 216 by associating the source link with destination identifier 215 at step 426. When the packet is a multicast packet, card 106 waits for acknowl-edgements from any remaining destinations at step 428.”)

No.	’740 Patent Claim 10	Devi
10[a]	The method according to claim 9, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second physical links,	Devi discloses the method according to claim 9, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second physical links. For example, Devi discloses using hashing or other selection methods to select the egress links of a particular destination port based on the information in the destination identifier 215 of the packet. A person of ordinary skill in the art would understand that applying a hash function includes determining parameters responsive to the system and packet features, such as a hashing size based on the number of physical links coupled to ports and ingress and egress backplane interfaces, and generating a result. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Devi is found to not meet this limitation, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second

No.	’740 Patent Claim 10	Devi
		physical links would have been obvious to a person having ordinary skill in the art, as explained below. *See supra* Claim 9 Devi at [0005] (“In accordance with one embodiment of the present invention, a method performed by an ingress card includes receiving a packet comprising a destination identifier. The method also includes determining an aggregated link asso-ciated with the destination identifier, and determining a destination port from the egress ports of the aggregated link. The method further includes communicating the packet to the destination port.”) Devi at [0006] (“In accordance with another embodiment of the present invention, a method performed by an ingress card includes receiving a packet and determining egress links for the packet. The method also includes determining whether each link is an aggregated link with multiple egress ports or a non-aggregated link with a single egress port. For each aggregated link, a destination port is determined from the egress ports, and the packet is communicated to the desti-nation port. For each non-aggregated link, a copy of the packet is communicated to the single egress port.”) Devi at [0007] (“Important technical advantages of certain embodi-ments of the present invention include implementing link aggregation using distributed processing among ingress components. For example, individual ingress cards can determine a destination port for a packet, eliminating the need for a separate processing stage implemented in hard-ware and/or software to separately determine the destination port from a packet after the ingress card forwards the packet to a link. This improves the efficiency and speed of packet forwarding in a switch.”) Devi at [0008] (“Yet another important technical advantage of cer-tain embodiments is a distributed architecture that may be used to process a variety of packet traffic. For example, certain embodiments of an ingress card can forward unicast, multicast, and bridging traffic. Particular embodiments may process packets from different protocols as well.”)

No.	’740 Patent Claim 10	Devi
		Devi at [0009] (“Other important technical advantages of certain embodiments of the present invention include load balanc-ing. The tables or other information used by the ingress card to determine a destination port for packets may also include additional usage information that allows the ingress card to determine a destination port from the egress ports in an aggregated link. This means that rather than using one link to the exclusion of others, loads may be distributed among several links in an aggregated link in a more balanced fashion. Particular embodiments of the present invention may have some, all or none of the enumerated technical advantages. Still other important technical advantages will be apparent to one skilled in the art from the following figures, description, and claims.”) Devi at [0015] (“FIG. 1 shows a system 100 that includes a switch 104 coupled to a network 102. Using network 102 and switch 104, packets are communicated from sources 103 to destinations 112. Components of switch 104 are referred to as "ingress" components when receiving packets from sources 103 and "egress" components when sending packets to destinations 112.”) Devi at [0016] (“Network 102 represents any suitable structure for communicating packets, cells, frames, segments, or other portions of data (generally referred to as "packets"). Net-work 102 may represent the Internet, extranet, local area network (LAN), synchronous optical network (SONET), wide area network (WAN), the public switched telephone network (PSTN), or any other suitable network for commu-nicating information. Network 102 may include routers, switches, hubs, endpoints, or any other network device that communicates information. Network 102 contemplates any number or arrangement of components that exchange infor-mation.”) Devi at [0018] (“Switch 104 sends and receives packets. Switch 104 may represent any suitable device, including an Ethernet switch, router, hub, or any other suitable hardware and/or software configured to receive packets and communicate them to other devices. Switch 104 includes cards 106 coupled to physical links 108 and a backplane 114 that represents hardware and/or software allowing cards 106 in switch 104 to exchange information with one another.)

No.	’740 Patent Claim 10	Devi
		Devi at [0023] (“Multiple physical links 108 to a particular destina-tion 112 may be aggregated to form an aggregated link 116. Aggregated link 116 represents multiple physical links 108 that are addressed by a single logical address, such as a medium access controller (MAC) address, an Internet pro-tocol (IP) address, or other suitable address or identifier, or otherwise treated as a single logical link between switch 104 and a destination 112. Because aggregated link 116 includes multiple physical links 108 that go to the same destination 112, a packet sent to destination 112 need only be sent to one port 110 corresponding to one physical link 108 of aggre-gated link 116. One technical advantage of certain embodi-ments of the present invention is that ingress cards 106 may communicate packets to ports 110 in some alternating fash-ion, such as round robin, random selection, pseudo-random selection using hash tables, or any other selection technique. This allows the load on any particular physical link 108 to be balanced with the loads on other physical links 108 in aggregated link 116.”) Devi at [0030] (“Destination identifier 215 represents information identifying an intermediate or final destination 112 for the packet. Destination identifier 215 may include a network address for destination 112, a logical address for an egress link, or any other information useful for identifying the destination of a packet. Destination identifier 215 contem-plates any information determinable from a packet that allows card 106 to identify one or more destinations 112 for the packet.”) Devi at [0031] (“Destination information 216 associates destination identifiers 215 in packets with egress links. Destination information 216 may be organized in any suitable fashion, including one or more tables, files, databases, or other form of organization. Card 106 uses destination information 216 to determine an egress link or links using destination iden-tifier 215 from a packet. For example, a multicast packet may include a multicast group identifier instead of indi-vidual link addresses. Ingress card 106 receives the packet, looks up the multicast group identifier, and determines the egress links associated with the multicast group. In another example, the packet may include a final destination address (such as an IP address for a receiving device) rather than a link address from switch 104. In that case, ingress link 106 uses destination information 216 to determine an egress link to the final destination. Destination information 216 may include one or more layers as well. For example, destination

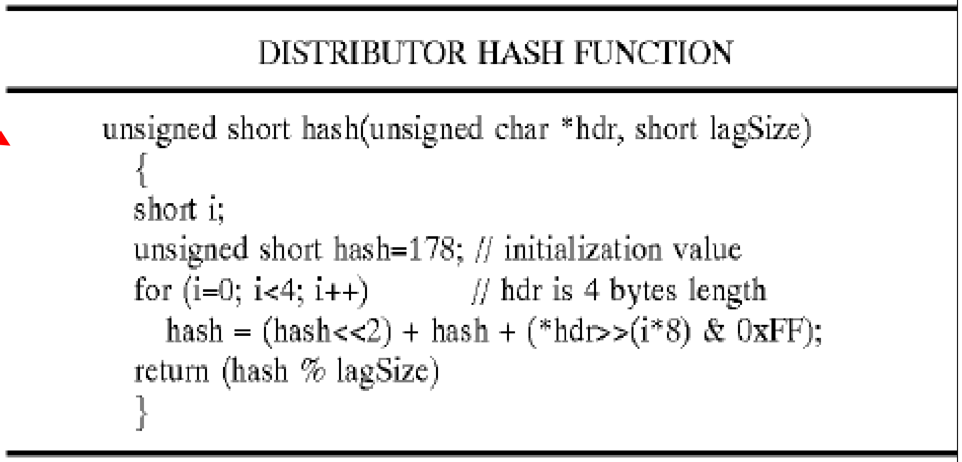
No.	’740 Patent Claim 10	Devi
		information 216 may include a first table relating a multicast group to destination IP addresses, and a second table relating the destination IP addresses to logical link addresses. Card 106 may also update destination information 216 as new or additional information becomes available, such as when card 106 receives an acknowledgement from a destination 112 indicating that a packet was received.”) Devi at [0033] (“In operation, card 106 receives a packet from network 102. Ingress card 106 determines the destination for the particular packet and consults destination information 216 to see if that destination is associated with a particular egress link. Some packets may have a destination identifier 215 that card 106 has not yet learned, and so destination information 216 does not include an egress link or links associated with destination identifier 215 of the packet. In such cases, card 106 may flood all available egress links so that the packet will arrive at its proper destination, and that destination will send an acknowledgment back to card 106. To flood destinations, card 106 determines logical links coupled to switch 104 using port information 218. Card 106 communicates the packet to these links, excluding the link from which the packet was received. In a particular embodi-ment, flooded cards will replicate the packet to all ports within that card, so card 106 need only communicate the packet once to a particular egress card 106.”) Devi at [0037] (“Card 106 receives a packet at step 302. At step 304, card determines destination identifier 215 for the packet and uses destination information 216 to determine if card 106 has previously learned the destination identifier 215. If the destination has not been previously learned, card 106 floods all egress links at step 306, shown in more detail in FIG. 4 described below.”) Devi at [0038] (“If destination identifier 215 of the packet has been learned, card 106 next determines if destination identifier 215 identifies a multicast or a unicast destination at step 308. If the packet is a multicast packet, card 106 selects one of the destinations for the packet using destination information 216 at step 310. On the other hand, if the packet is a unicast packet, there is only one destination, so there is no need to perform a selection process.”)

No.	’740 Patent Claim 10	Devi
		Devi at [0041] (“If the link is an aggregated link 116, card 106 determines a destination port 110 of aggregated link 116 using the information in port information 218 at step 320. Card 106 may select a particular destination port 110 from the multiple ports 110 of the aggregated link using random selection, load balancing, pseudo-random selection or hash-ing using part of the packet information, historical tracking, round robin, or any other method of selection. Card 106 then communicates the packet to the destination port 110 at step 322.”) Devi at [0048] (“If the egress link is an aggregated isolated link, card 106 communicates the packet to destination port 110 of aggregated link 116. Card 106 determines destination port 110 for the packet at step 416. Card 106 may use any suitable method to determine destination port 110, including any of the methods described in conjunction with FIGS. 2 and 3. Card 106 then communicates the packet to destination port 110 at step 418.”) Devi at [0049] (“Card 106 then determines if there are any remain-ing egress links at step 420. If there are egress links remaining, then card 106 selects a new link to process at step 406. If there are no remaining links, then card 106 waits for an acknowledgement from one of the destinations that the packet was received. Card 106 receives the acknowledge-ment at step 422, and determines the link from which the acknowledgement was received at step 424. Card 106 then updates destination information 216 by associating the source link with destination identifier 215 at step 426. When the packet is a multicast packet, card 106 waits for acknowl-edgements from any remaining destinations at step 428.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Devi in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[a] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses applying a distributor hash function to the frame information which includes determining a number of the plurality of physical links. Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of

No.	’740 Patent Claim 10	Devi
		the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of

No.	’740 Patent Claim 10	Devi
		an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”) Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I: Bruckman at Table 1 (annotated)

No.	’740 Patent Claim 10	Devi

		hashing function *“mapping function”*
		
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		
		For example, Solomon discloses applying a distributor hash function to the frame information which includes determining a number of the plurality of physical links.
		
		Solomon at [0024] (“In another embodiment, switching the data packets includes mapping the data packets to the selected port responsively to the label. Additionally or alternatively, map-ping the data packets includes applying a hashing function to the label so as to determine a number of the selected port, and choosing the label includes applying an inverse of the hashing function to the number of the selected port.”)
		
		Solomon at [0048] (“The mapping function typically uses MPLS label 52 for mapping, since the MPLS label uniquely identifies MPLS tunnel 28, and it is required that all MPLS packets

No.	’740 Patent Claim 10	Devi
		belonging to the same tunnel be switched through the same physical port 24. Additionally or alternatively, the mapping function uses a "PW" label (pseudo wire label, formerly known as a virtual connection, or VC label), which is optionally added to MPLS header 50. The PW label com-prises information that the egress node requires for deliver-ing the packet to its destination, and is optionally added during the encapsulation of MPLS packets. Additional details regarding the VC label can be found in an IETF draft by Martini et al. entitled "Encapsulation Methods for Trans-port of Ethernet Frames Over IP/MPLS Networks" (IETF draft-ietf-pwe3-ethernet-encap-07.txt, May, 2004), which is incorporated herein by reference. In some embodiments, mapper 34 applies a hashing function to the MPLS and/or PW label, as will be described below.”) Solomon at [0059] (“In this method, the mapping function used by mapper 34 of switch A is a hashing function. Various hashing functions are known in the art, and any suitable hashing function may be used in mapper 34. Since the hashing operation is performed for each packet, it is desirable to have a hashing function that is computationally simple.”) Solomon at [0060] (“As mentioned above, the hashing function typically hashes the value of MPLS label 52 to determine the selected physical port, as the MPLS label uniquely identifies tunnel 28. For example, the following hashing function may be used by mapper 34: Selected port number=l+((MPLS label) mod N), wherein N denotes the number of physical Ethernet ports in LAG group 25, and "mod" denotes the modulus operator. Assuming the values of MPLS labels are distrib-uted uniformly over a certain range, this function achieves a uniform distribution of port allocations for the different MPLS labels. It can also be seen that all packets carrying the same MPLS label (in other words-belonging to the same MPLS tunnel) will be mapped to the same physical port.”) Solomon at [0065] (“Mapper 34 of switch A maps each received packet to the selected physical port of LAG group 25 using the hashing function, at a hashing step 90. Mapper 34 extracts the MPLS label from each received packet and uses the hashing function to calculate the serial number of the selected physical port, which was selected by the CAC processor at step 82. Following the numerical example given above, the mapper extracts MPLS label=65647 from the packet. Substituting this value and N=3 into the hashing function gives:

No.	’740 Patent Claim 10	Devi
		Selected port number=1+(65647 mod 3)=2, which is indeed the port number selected in the example above.”)
10[b]	applying the hashing function to the at least one of the frame attributes to produce a hashing key,	Devi discloses applying the hashing function to the at least one of the frame attributes to produce a hashing key. For example, Devi discloses using hashing or other selection methods to select the egress links of a particular destination port based on the information in the destination identifier 215 of the packet. A person of ordinary skill in the art would understand that applying a hash function includes determining parameters responsive to the system and packet features, such as a hashing key based on the packet destination identifier 215, and generating a result. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Devi is found to not meet this limitation, applying the hashing function to the at least one of the frame attributes to produce a hashing key would have been obvious to a person having ordinary skill in the art, as explained below. Devi at [0005] (“In accordance with one embodiment of the present invention, a method performed by an ingress card includes receiving a packet comprising a destination identifier. The method also includes determining an aggregated link asso-ciated with the destination identifier, and determining a destination port from the egress ports of the aggregated link. The method further includes communicating the packet to the destination port.”) Devi at [0006] (“In accordance with another embodiment of the present invention, a method performed by an ingress card includes receiving a packet and determining egress links for the packet. The method also includes determining whether each link is an aggregated link with multiple egress ports or a non-aggregated link with a single egress port. For each aggregated link, a destination port is determined from the egress ports, and the packet is communicated to the desti-nation port. For each non-aggregated link, a copy of the packet is communicated to the single egress port.”)

No.	’740 Patent Claim 10	Devi
		Devi at [0007] (“Important technical advantages of certain embodi-ments of the present invention include implementing link aggregation using distributed processing among ingress components. For example, individual ingress cards can determine a destination port for a packet, eliminating the need for a separate processing stage implemented in hard-ware and/or software to separately determine the destination port from a packet after the ingress card forwards the packet to a link. This improves the efficiency and speed of packet forwarding in a switch.”) Devi at [0008] (“Yet another important technical advantage of cer-tain embodiments is a distributed architecture that may be used to process a variety of packet traffic. For example, certain embodiments of an ingress card can forward unicast, multicast, and bridging traffic. Particular embodiments may process packets from different protocols as well.”) Devi at [0009] (“Other important technical advantages of certain embodiments of the present invention include load balanc-ing. The tables or other information used by the ingress card to determine a destination port for packets may also include additional usage information that allows the ingress card to determine a destination port from the egress ports in an aggregated link. This means that rather than using one link to the exclusion of others, loads may be distributed among several links in an aggregated link in a more balanced fashion. Particular embodiments of the present invention may have some, all or none of the enumerated technical advantages. Still other important technical advantages will be apparent to one skilled in the art from the following figures, description, and claims.”) Devi at [0015] (“FIG. 1 shows a system 100 that includes a switch 104 coupled to a network 102. Using network 102 and switch 104, packets are communicated from sources 103 to destinations 112. Components of switch 104 are referred to as "ingress" components when receiving packets from sources 103 and "egress" components when sending packets to destinations 112.”) Devi at [0016] (“Network 102 represents any suitable structure for communicating packets, cells, frames, segments, or other portions of data (generally referred to as "packets"). Net-work 102 may represent the Internet, extranet, local area network (LAN), synchronous

No.	’740 Patent Claim 10	Devi
		optical network (SONET), wide area network (WAN), the public switched telephone network (PSTN), or any other suitable network for commu-nicating information. Network 102 may include routers, switches, hubs, endpoints, or any other network device that communicates information. Network 102 contemplates any number or arrangement of components that exchange infor-mation.”) Devi at [0018] (“Switch 104 sends and receives packets. Switch 104 may represent any suitable device, including an Ethernet switch, router, hub, or any other suitable hardware and/or software configured to receive packets and communicate them to other devices. Switch 104 includes cards 106 coupled to physical links 108 and a backplane 114 that represents hardware and/or software allowing cards 106 in switch 104 to exchange information with one another.) Devi at [0023] (“Multiple physical links 108 to a particular destina-tion 112 may be aggregated to form an aggregated link 116. Aggregated link 116 represents multiple physical links 108 that are addressed by a single logical address, such as a medium access controller (MAC) address, an Internet pro-tocol (IP) address, or other suitable address or identifier, or otherwise treated as a single logical link between switch 104 and a destination 112. Because aggregated link 116 includes multiple physical links 108 that go to the same destination 112, a packet sent to destination 112 need only be sent to one port 110 corresponding to one physical link 108 of aggre-gated link 116. One technical advantage of certain embodi-ments of the present invention is that ingress cards 106 may communicate packets to ports 110 in some alternating fash-ion, such as round robin, random selection, pseudo-random selection using hash tables, or any other selection technique. This allows the load on any particular physical link 108 to be balanced with the loads on other physical links 108 in aggregated link 116.”) Devi at [0030] (“Destination identifier 215 represents information identifying an intermediate or final destination 112 for the packet. Destination identifier 215 may include a network address for destination 112, a logical address for an egress link, or any other information useful for identifying the destination of a packet. Destination identifier 215 contem-plates any

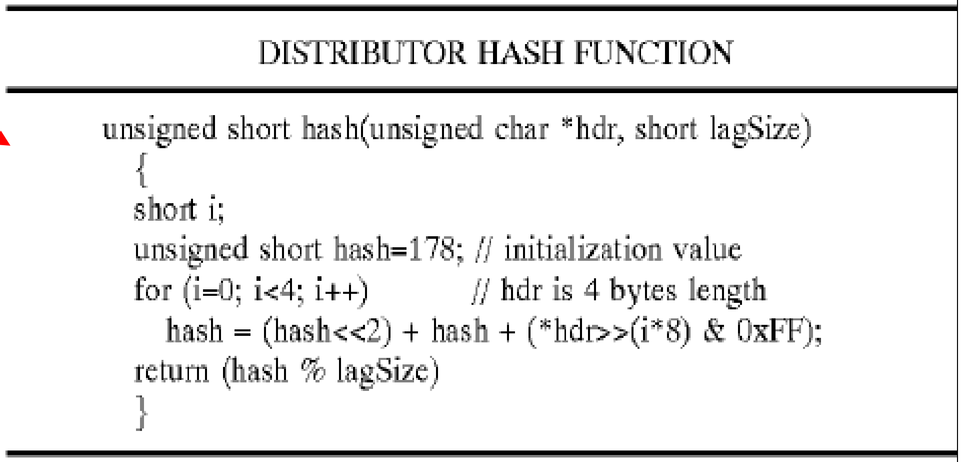
No.	’740 Patent Claim 10	Devi
		information determinable from a packet that allows card 106 to identify one or more destinations 112 for the packet.”) Devi at [0031] (“Destination information 216 associates destination identifiers 215 in packets with egress links. Destination information 216 may be organized in any suitable fashion, including one or more tables, files, databases, or other form of organization. Card 106 uses destination information 216 to determine an egress link or links using destination iden-tifier 215 from a packet. For example, a multicast packet may include a multicast group identifier instead of indi-vidual link addresses. Ingress card 106 receives the packet, looks up the multicast group identifier, and determines the egress links associated with the multicast group. In another example, the packet may include a final destination address (such as an IP address for a receiving device) rather than a link address from switch 104. In that case, ingress link 106 uses destination information 216 to determine an egress link to the final destination. Destination information 216 may include one or more layers as well. For example, destination information 216 may include a first table relating a multicast group to destination IP addresses, and a second table relating the destination IP addresses to logical link addresses. Card 106 may also update destination information 216 as new or additional information becomes available, such as when card 106 receives an acknowledgement from a destination 112 indicating that a packet was received.”) Devi at [0033] (“In operation, card 106 receives a packet from network 102. Ingress card 106 determines the destination for the particular packet and consults destination information 216 to see if that destination is associated with a particular egress link. Some packets may have a destination identifier 215 that card 106 has not yet learned, and so destination information 216 does not include an egress link or links associated with destination identifier 215 of the packet. In such cases, card 106 may flood all available egress links so that the packet will arrive at its proper destination, and that destination will send an acknowledgment back to card 106. To flood destinations, card 106 determines logical links coupled to switch 104 using port information 218. Card 106 communicates the packet to these links, excluding the link from which the packet was received. In a particular embodi-ment, flooded cards will replicate the packet to all ports within that card, so card 106 need only communicate the packet once to a particular egress card 106.”)

No.	’740 Patent Claim 10	Devi
		Devi at [0037] (“Card 106 receives a packet at step 302. At step 304, card determines destination identifier 215 for the packet and uses destination information 216 to determine if card 106 has previously learned the destination identifier 215. If the destination has not been previously learned, card 106 floods all egress links at step 306, shown in more detail in FIG. 4 described below.”) Devi at [0038] (“If destination identifier 215 of the packet has been learned, card 106 next determines if destination identifier 215 identifies a multicast or a unicast destination at step 308. If the packet is a multicast packet, card 106 selects one of the destinations for the packet using destination information 216 at step 310. On the other hand, if the packet is a unicast packet, there is only one destination, so there is no need to perform a selection process.”) Devi at [0041] (“If the link is an aggregated link 116, card 106 determines a destination port 110 of aggregated link 116 using the information in port information 218 at step 320. Card 106 may select a particular destination port 110 from the multiple ports 110 of the aggregated link using random selection, load balancing, pseudo-random selection or hash-ing using part of the packet information, historical tracking, round robin, or any other method of selection. Card 106 then communicates the packet to the destination port 110 at step 322.”) Devi at [0048] (“If the egress link is an aggregated isolated link, card 106 communicates the packet to destination port 110 of aggregated link 116. Card 106 determines destination port 110 for the packet at step 416. Card 106 may use any suitable method to determine destination port 110, including any of the methods described in conjunction with FIGS. 2 and 3. Card 106 then communicates the packet to destination port 110 at step 418.”) Devi at [0049] (“Card 106 then determines if there are any remain-ing egress links at step 420. If there are egress links remaining, then card 106 selects a new link to process at step 406. If there are no remaining links, then card 106 waits for an acknowledgement from one of the destinations that the packet was received. Card 106 receives the acknowledge-ment at step 422, and determines the link from which the acknowledgement was received at step 424. Card 106 then updates destination information 216 by associating the source link with

No.	’740 Patent Claim 10	Devi
		destination identifier 215 at step 426. When the packet is a multicast packet, card 106 waits for acknowl-edgements from any remaining destinations at step 428.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Devi in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[b] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses applying a distributor hash function to the frame information which includes determining a number of the plurality of physical links. Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”)

No.	’740 Patent Claim 10	Devi
		Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”)

No.	’740 Patent Claim 10	Devi

		Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:
		Bruckman at Table 1 (annotated)
		hashing function *“mapping function”*
		
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		    hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		For example, Alexander discloses applying a distributor hash function to the frame information which includes determining a hash key based on packet information.
		Alexander at 3:1-40 (“The hash function is preferably selected such that suc-cessive application of the hash function to all source and destination addresses expected to be seen by the Ethernet switch will produce a lowest value hash key, a highest value hash key, and a

No.	’740 Patent Claim 10	Devi
		group of hash keys having intermediate values distributed evenly between the lowest and highest values. The distribution table contains a separate port identifier look-up table for each aggregated grouping of outgoing ports. Advantageously, the hash key is an N bit hash key; and, each port identifier look-up table contains 2^N entries occupying 2^N consecutive locations, with each entry being an identifier of a particular one of the physical outgoing ports. Identifiers for particular outgoing ports are retrieved from the distribution table by extracting first and second N bit hash keys which form part of the retrieved destination and source address contexts respectively. The hash keys are combined to form an N bit connection identifier. The port identifier look-up table corresponding to the aggregated grouping represented by the retrieved destination address is selected, and the entry at the table location corresponding to the value of the N bit connection identifier is retrieved. If the address look-up table does not contain a destination address corresponding to the extracted destination address then first and second hash keys are produced by applying a hash function to the extracted source and destination addresses respectively. The hash keys are combined to form an N bit connection identifier. The incoming port on which the packet containing the extracted source address was received is identified. All of the aggregated groupings are scanned to identify all outgoing ports to which packets may be directed from the incoming port on which the packet was received. For each one of those outgoing ports, the port identifier look-up table corresponding to the aggregated grouping containing that outgoing port is selected, the entry at the table location corresponding to the value of the N bit connection identifier is retrieved, and the received packet is queued for outgoing transmission on the outgoing port corresponding to the retrieved entry.”) Alexander at 5:10-35 (“If a packet arrives bearing a source Ethernet MAC address that was not found in look-up table 12 by address resolution unit 10, learning function 16 is invoked to update look-up table 12 with the new address (i.e. processing branches along the "No" exit from FIG. 2, block 36). Learning function 16 first computes a hash function on the source Ethernet MAC address, generating an N-bit hash key ("partial connection identifier") from

No.	’740 Patent Claim 10	Devi
		the 48-bit MAC address, where N is some small integer in the range of 3 to 8 (FIG. 2, block 38). The physical port on which the packet arrived is then determined. If the physical port is found to be associated with an aggregate group (i.e., it is one of a set of ports that have been bound into a single logical port), then the logical identifier assigned to the aggregate group is also determined. The hash key is then stored into address look-up table 12 in conjunction with the actual Ethernet MAC address and the port identifier (FIG. 2, block 40). The physical port identifier is used if the port is not part of an aggregate group (i.e. if processing branched along the "No" exit from block 30 and through block 32), while the logical identifier is used for ports that have been aggregated (i.e. if processing branched along the "Yes" exit from block 30 and through block 34). The hash key and port identifier are considered to form the "context" for the given MAC address.”) Alexander at 5:36-46 (“The hash function should be selected to ensure an even distribution of hash key values over the range of MAC addresses that are expected to be seen by the Ethernet switch. As a specific example, the EXACT™ Ethernet switch system employs an exclusive-OR based hash function, wherein the 48-bit MAC address is divided into 16-bit blocks, which are then exclusive-ORed together to form a single 16-bit number; the 3 least significant bits (LSBs) of this number are taken to produce a 3-bit hash key. Other schemes such as CRC-based or checksum-based hashes may also be used.”) Alexander at 6:49-65 (“If the context information for the destination address indicates, however, that the target is an aggregate group (i.e. if processing branches along the "Yes" exit from FIG. 2, block 42) then the logical identifier assigned to the aggregate group is retrieved and is used to select the proper look-up table contained within the distribution table data structure. The hash keys (partial connection identifiers) stored into the contexts for the source and destination MAC addresses are obtained from address resolution unit 10 and combined to generate a "connection identifier" with the same number of bits (FIG. 2, block 44). (In the EXACT™ Ethernet switch, a Boolean exclusive-OR operation is used to combine the hash keys without increasing the number of bits.) This connection identifier is then used to index into the selected look-up table, and finally retrieve an actual physical port index on which the packet must be transmitted (FIG. 2, block 46).”)

No.	’740 Patent Claim 10	Devi
10[c]	calculating a modulo of a division operation of the hashing key by the hashing size, and	Devi discloses calculating a modulo of a division operation of the hashing key by the hashing size. For example, Devi discloses using hashing or other selection methods to select the egress links of a particular destination port based on the information in the destination identifier 215 of the packet. A person of ordinary skill in the art would understand that applying a hash function includes determining parameters responsive to the system and packet features, such as a performing a division operation to calculate a modulo, and generating a result. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Devi is found to not meet this limitation, calculating a modulo of a division operation of the hashing key by the hashing size would have been obvious to a person having ordinary skill in the art, as explained below. Devi at [0005] (“In accordance with one embodiment of the present invention, a method performed by an ingress card includes receiving a packet comprising a destination identifier. The method also includes determining an aggregated link asso-ciated with the destination identifier, and determining a destination port from the egress ports of the aggregated link. The method further includes communicating the packet to the destination port.”) Devi at [0006] (“In accordance with another embodiment of the present invention, a method performed by an ingress card includes receiving a packet and determining egress links for the packet. The method also includes determining whether each link is an aggregated link with multiple egress ports or a non-aggregated link with a single egress port. For each aggregated link, a destination port is determined from the egress ports, and the packet is communicated to the desti-nation port. For each non-aggregated link, a copy of the packet is communicated to the single egress port.”) Devi at [0007] (“Important technical advantages of certain embodi-ments of the present invention include implementing link aggregation using distributed processing among ingress components. For example, individual ingress cards can determine a destination port for a packet, eliminating the need for a separate processing stage implemented in hard-ware and/or

No.	’740 Patent Claim 10	Devi
		software to separately determine the destination port from a packet after the ingress card forwards the packet to a link. This improves the efficiency and speed of packet forwarding in a switch.”) Devi at [0008] (“Yet another important technical advantage of cer-tain embodiments is a distributed architecture that may be used to process a variety of packet traffic. For example, certain embodiments of an ingress card can forward unicast, multicast, and bridging traffic. Particular embodiments may process packets from different protocols as well.”) Devi at [0009] (“Other important technical advantages of certain embodiments of the present invention include load balanc-ing. The tables or other information used by the ingress card to determine a destination port for packets may also include additional usage information that allows the ingress card to determine a destination port from the egress ports in an aggregated link. This means that rather than using one link to the exclusion of others, loads may be distributed among several links in an aggregated link in a more balanced fashion. Particular embodiments of the present invention may have some, all or none of the enumerated technical advantages. Still other important technical advantages will be apparent to one skilled in the art from the following figures, description, and claims.”) Devi at [0015] (“FIG. 1 shows a system 100 that includes a switch 104 coupled to a network 102. Using network 102 and switch 104, packets are communicated from sources 103 to destinations 112. Components of switch 104 are referred to as "ingress" components when receiving packets from sources 103 and "egress" components when sending packets to destinations 112.”) Devi at [0016] (“Network 102 represents any suitable structure for communicating packets, cells, frames, segments, or other portions of data (generally referred to as "packets"). Net-work 102 may represent the Internet, extranet, local area network (LAN), synchronous optical network (SONET), wide area network (WAN), the public switched telephone network (PSTN), or any other suitable network for commu-nicating information. Network 102 may include routers, switches, hubs, endpoints, or any other network device that communicates

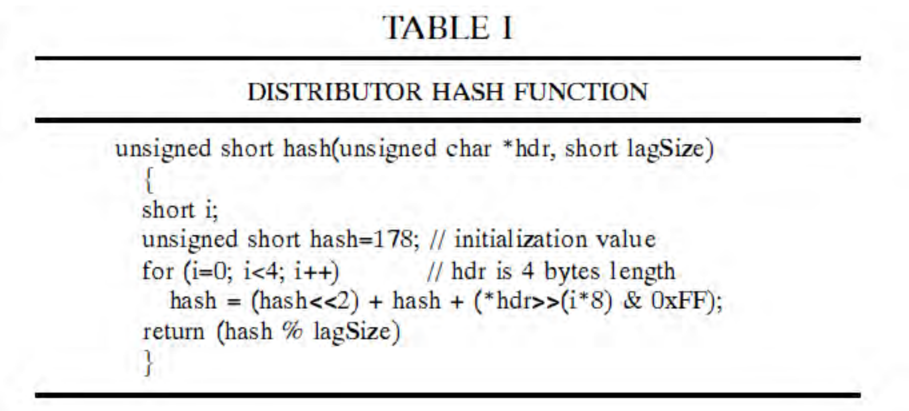
No.	’740 Patent Claim 10	Devi
		information. Network 102 contemplates any number or arrangement of components that exchange infor-mation.”) Devi at [0018] (“Switch 104 sends and receives packets. Switch 104 may represent any suitable device, including an Ethernet switch, router, hub, or any other suitable hardware and/or software configured to receive packets and communicate them to other devices. Switch 104 includes cards 106 coupled to physical links 108 and a backplane 114 that represents hardware and/or software allowing cards 106 in switch 104 to exchange information with one another.) Devi at [0023] (“Multiple physical links 108 to a particular destina-tion 112 may be aggregated to form an aggregated link 116. Aggregated link 116 represents multiple physical links 108 that are addressed by a single logical address, such as a medium access controller (MAC) address, an Internet pro-tocol (IP) address, or other suitable address or identifier, or otherwise treated as a single logical link between switch 104 and a destination 112. Because aggregated link 116 includes multiple physical links 108 that go to the same destination 112, a packet sent to destination 112 need only be sent to one port 110 corresponding to one physical link 108 of aggre-gated link 116. One technical advantage of certain embodi-ments of the present invention is that ingress cards 106 may communicate packets to ports 110 in some alternating fash-ion, such as round robin, random selection, pseudo-random selection using hash tables, or any other selection technique. This allows the load on any particular physical link 108 to be balanced with the loads on other physical links 108 in aggregated link 116.”) Devi at [0030] (“Destination identifier 215 represents information identifying an intermediate or final destination 112 for the packet. Destination identifier 215 may include a network address for destination 112, a logical address for an egress link, or any other information useful for identifying the destination of a packet. Destination identifier 215 contem-plates any information determinable from a packet that allows card 106 to identify one or more destinations 112 for the packet.”)

No.	’740 Patent Claim 10	Devi
		Devi at [0031] (“Destination information 216 associates destination identifiers 215 in packets with egress links. Destination information 216 may be organized in any suitable fashion, including one or more tables, files, databases, or other form of organization. Card 106 uses destination information 216 to determine an egress link or links using destination iden-tifier 215 from a packet. For example, a multicast packet may include a multicast group identifier instead of indi-vidual link addresses. Ingress card 106 receives the packet, looks up the multicast group identifier, and determines the egress links associated with the multicast group. In another example, the packet may include a final destination address (such as an IP address for a receiving device) rather than a link address from switch 104. In that case, ingress link 106 uses destination information 216 to determine an egress link to the final destination. Destination information 216 may include one or more layers as well. For example, destination information 216 may include a first table relating a multicast group to destination IP addresses, and a second table relating the destination IP addresses to logical link addresses. Card 106 may also update destination information 216 as new or additional information becomes available, such as when card 106 receives an acknowledgement from a destination 112 indicating that a packet was received.”) Devi at [0033] (“In operation, card 106 receives a packet from network 102. Ingress card 106 determines the destination for the particular packet and consults destination information 216 to see if that destination is associated with a particular egress link. Some packets may have a destination identifier 215 that card 106 has not yet learned, and so destination information 216 does not include an egress link or links associated with destination identifier 215 of the packet. In such cases, card 106 may flood all available egress links so that the packet will arrive at its proper destination, and that destination will send an acknowledgment back to card 106. To flood destinations, card 106 determines logical links coupled to switch 104 using port information 218. Card 106 communicates the packet to these links, excluding the link from which the packet was received. In a particular embodi-ment, flooded cards will replicate the packet to all ports within that card, so card 106 need only communicate the packet once to a particular egress card 106.”) Devi at [0037] (“Card 106 receives a packet at step 302. At step 304, card determines destination identifier 215 for the packet and uses destination information 216 to determine if

No.	’740 Patent Claim 10	Devi
		card 106 has previously learned the destination identifier 215. If the destination has not been previously learned, card 106 floods all egress links at step 306, shown in more detail in FIG. 4 described below.”) Devi at [0038] (“If destination identifier 215 of the packet has been learned, card 106 next determines if destination identifier 215 identifies a multicast or a unicast destination at step 308. If the packet is a multicast packet, card 106 selects one of the destinations for the packet using destination information 216 at step 310. On the other hand, if the packet is a unicast packet, there is only one destination, so there is no need to perform a selection process.”) Devi at [0041] (“If the link is an aggregated link 116, card 106 determines a destination port 110 of aggregated link 116 using the information in port information 218 at step 320. Card 106 may select a particular destination port 110 from the multiple ports 110 of the aggregated link using random selection, load balancing, pseudo-random selection or hash-ing using part of the packet information, historical tracking, round robin, or any other method of selection. Card 106 then communicates the packet to the destination port 110 at step 322.”) Devi at [0048] (“If the egress link is an aggregated isolated link, card 106 communicates the packet to destination port 110 of aggregated link 116. Card 106 determines destination port 110 for the packet at step 416. Card 106 may use any suitable method to determine destination port 110, including any of the methods described in conjunction with FIGS. 2 and 3. Card 106 then communicates the packet to destination port 110 at step 418.”) Devi at [0049] (“Card 106 then determines if there are any remain-ing egress links at step 420. If there are egress links remaining, then card 106 selects a new link to process at step 406. If there are no remaining links, then card 106 waits for an acknowledgement from one of the destinations that the packet was received. Card 106 receives the acknowledge-ment at step 422, and determines the link from which the acknowledgement was received at step 424. Card 106 then updates destination information 216 by associating the source link with destination identifier 215 at step 426. When the packet is a multicast packet, card 106 waits for acknowl-edgements from any remaining destinations at step 428.”)

No.	’740 Patent Claim 10	Devi
		Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Devi in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[c] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses distributing data frames over physical links and traces based on a hash function involving a division operation (%). Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0025] (“Typically, setting the protection policy includes determining a maximum number of the physical links that may fail while the logical link continues to provide at least the guaranteed bandwidth for the connection. In one embodiment, the guaranteed bandwidth is a bandwidth B, and the plurality of physical links consists of N links, and the maximum number is an integer P, and the link bandwidth allocated to each of the links is no less than B/(N-P). Conveying the data may further include managing the transmission of the data responsively to an actual number X of the physical links that have failed so that the guaranteed bandwidth on each of the links is limited to B/(N-X), X P, and an excess bandwidth on the physical links over the guaranteed bandwidth is available for other connections.”) Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the

No.	’740 Patent Claim 10	Devi

		guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”)
		Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:
		
		TABLE I
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		”)
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		Bruckman at [0067] (“A similar problem may arise if there is a failure in a link in an aggregation group or in one of a number of line cards serving the aggregation group. In this case, to maintain the bandwidth allocation B made by CAC 44, each of the remaining links in the group must now carry, on average, B/(N-M) traffic, wherein M is the number of links in the group that are out of service. If only BIN has been allocated to each link, the remaining active links may not have sufficient bandwidth to continue to provide the bandwidth that has been guaranteed to the connections that they are required to carry. A similar problem arises

No.	’740 Patent Claim 10	Devi
		with respect to loading of traces 52. For example, if there is a failure in LC2 or in one of links 30 in group 36 that connect to LC2, the trace connecting the multiplexer 50 in LCl will have to carry a substantially larger share of the bandwidth, or even all of the bandwidth, that is allocated to the connection in ques-tion.”) Bruckman at [0068] (“FIG. 3 is a flow chart that schematically illustrates a method for dealing with these problems of fluctuating bandwidth requirements, in accordance with an embodiment of the present invention. In order to provide sufficient bandwidth for failure protection, CAC 44 uses a safety margin based on a protection parameter P, which is assigned at a protection setting step 60. P represents the maximum number of links in the group that can be out of service while still permitting the aggregation group to provide a given connection with the bandwidth that has been guaranteed to the connection. CAC 44 will then allocate at least B/(N-P) bandwidth to each link in the group, so that if P links fail, the group still provides total bandwidth of (N-P)*B/(N-P)= B. Setting P=l is equivalent to 1:N protection, so that the group will be unaffected by failure of a single link. In the example of group 36, shown in FIG. 2, setting P=2 will give both facility and equipment protection, i.e., the group will be unaffected not only by failure of a link, but also by failure of one of line cards 34. In the extreme case, in which P=N-1, CAC 44 will allocate the full bandwidth B on each link in the group.”) As another example, Singh discloses determining a ratio between the number of ingress and egress links and the number of links carrying data to the backplane and using a modulo to correspond to the channel’s link number. Singh at 9:30-43 (“The ratio between the number of line ingress links and the number of links carrying data to the backplane gives the backplane speedup for the system. In this example, there are 10 ingress links into the MS and 20 links (2 backplane channels) carrying that data to the backplane. This gives a backplane speedup of 2x. As another example, with 8 ingress links and 12 backplane links, there is a speedup of 1.5x. It should be noted that in addition to the backplane speedup, there is also an ingress/egress speedup. With 10 ingress links capable of carrying 2 Gbps each of raw data, this presents a 20 Gbps interface to the MS. An OC-192

No.	’740 Patent Claim 10	Devi
		only has approximately 10 Gbps worth of data. Taking into account cell overhead and cell quantization inefficiencies, there still remains excess capacity in the links.”) Singh at 11:29-38 (“FIG. 9 is a diagram illustrating link to channel assignments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a channel, whether it is a backplane channel or an ingress/egress channel. There is no logical relationship between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.”) Singh at 15:15-39 (“The number of crossbars that are required in a system is dependent on how many links are being used to create the backplane channels. There should be an even number of crossbars and they would be divided evenly across the switch cards. The following equation, for most cases, pro-vides the correct number of crossbars: # of Crossbars=(# links per ingress channelx# of ingress channels per portx# of port cardsx speedup)/32. For the 8x8 configuration, the # of crossbars should be multiplied by (4x# of iMS)/(# backplane channels per port card). The number of port cards should be rounded up to the nearest supported configuration, i.e. 8, 16, or 32. The speedup in the case of crossbars should be the fractional speedup that is desired. Example to determine the number of arbiters and cross-bars for the following system: 4 channel port cards (40 Gbps) 8 links per channel 16 port cards Speedup=l.5 # of arbiters=(4x2x2)/2=8 # of crossbars=(8x4x16xl.5)/32=24. This would give 3crossbars per arbiter.”)

No.	’740 Patent Claim 10	Devi
		Singh at 16:28-44 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 16x16 and 32x32 is the organization of the switchplane. The port card remains the same. Backplane channels 1 and 2 are used for the backplane connectivity. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 16, correspond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”) Singh at 17:31-49 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 8x8 and 16x16 is the organi-zation of the switchplane. The port card remains the same. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Links 0-7 and 24-31 on the arbiters would not be used and would be powered off. Links 0-7 and 24-31 on the crossbars would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Backplane channels 1 and 2 are used for the backplane connectivity. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 8, corre-spond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”)
10[d]	selecting the first and second physical links responsively to the modulo.	Devi discloses selecting the first and second physical links responsively to the modulo. For example, Devi discloses using hashing or other selection methods to select the egress links of a particular destination port based on the information in the destination identifier 215 of the packet. A person of ordinary skill in the art would understand that applying a hash

No.	’740 Patent Claim 10	Devi
		function includes determining parameters responsive to the system and packet features, such as using the modulo to determine the egress links associated with a destination port, and generating a result. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Devi is found to not meet this limitation, selecting the first and second physical links responsively to the modulo would have been obvious to a person having ordinary skill in the art, as explained below. *See supra* Claim 8 Devi at [0005] (“In accordance with one embodiment of the present invention, a method performed by an ingress card includes receiving a packet comprising a destination identifier. The method also includes determining an aggregated link asso-ciated with the destination identifier, and determining a destination port from the egress ports of the aggregated link. The method further includes communicating the packet to the destination port.”) Devi at [0006] (“In accordance with another embodiment of the present invention, a method performed by an ingress card includes receiving a packet and determining egress links for the packet. The method also includes determining whether each link is an aggregated link with multiple egress ports or a non-aggregated link with a single egress port. For each aggregated link, a destination port is determined from the egress ports, and the packet is communicated to the desti-nation port. For each non-aggregated link, a copy of the packet is communicated to the single egress port.”) Devi at [0007] (“Important technical advantages of certain embodi-ments of the present invention include implementing link aggregation using distributed processing among ingress components. For example, individual ingress cards can determine a destination port for a packet, eliminating the need for a separate processing stage implemented in hard-ware and/or software to separately determine the destination port from a packet after the ingress card forwards the packet to a link. This improves the efficiency and speed of packet forwarding in a switch.”)

No.	’740 Patent Claim 10	Devi
		Devi at [0008] (“Yet another important technical advantage of cer-tain embodiments is a distributed architecture that may be used to process a variety of packet traffic. For example, certain embodiments of an ingress card can forward unicast, multicast, and bridging traffic. Particular embodiments may process packets from different protocols as well.”) Devi at [0009] (“Other important technical advantages of certain embodiments of the present invention include load balanc-ing. The tables or other information used by the ingress card to determine a destination port for packets may also include additional usage information that allows the ingress card to determine a destination port from the egress ports in an aggregated link. This means that rather than using one link to the exclusion of others, loads may be distributed among several links in an aggregated link in a more balanced fashion. Particular embodiments of the present invention may have some, all or none of the enumerated technical advantages. Still other important technical advantages will be apparent to one skilled in the art from the following figures, description, and claims.”) Devi at [0015] (“FIG. 1 shows a system 100 that includes a switch 104 coupled to a network 102. Using network 102 and switch 104, packets are communicated from sources 103 to destinations 112. Components of switch 104 are referred to as "ingress" components when receiving packets from sources 103 and "egress" components when sending packets to destinations 112.”) Devi at [0016] (“Network 102 represents any suitable structure for communicating packets, cells, frames, segments, or other portions of data (generally referred to as "packets"). Net-work 102 may represent the Internet, extranet, local area network (LAN), synchronous optical network (SONET), wide area network (WAN), the public switched telephone network (PSTN), or any other suitable network for commu-nicating information. Network 102 may include routers, switches, hubs, endpoints, or any other network device that communicates information. Network 102 contemplates any number or arrangement of components that exchange infor-mation.”) Devi at [0018] (“Switch 104 sends and receives packets. Switch 104 may represent any suitable device, including an Ethernet switch, router, hub, or any other suitable hardware

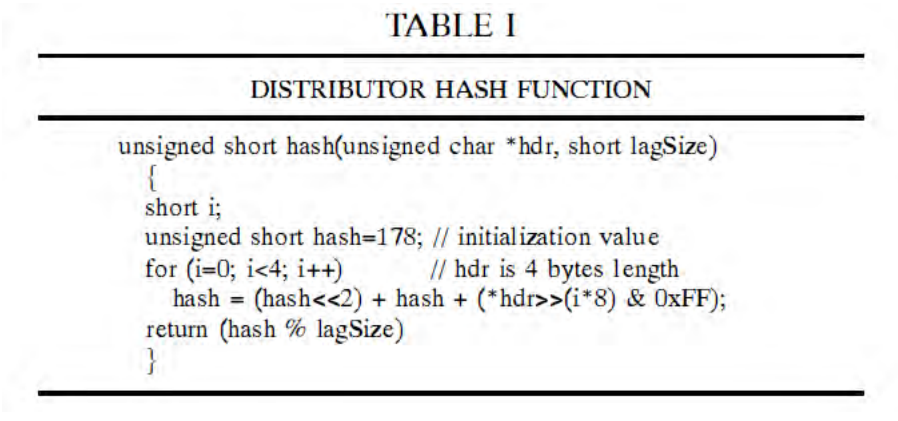
No.	’740 Patent Claim 10	Devi
		and/or software configured to receive packets and communicate them to other devices. Switch 104 includes cards 106 coupled to physical links 108 and a backplane 114 that represents hardware and/or software allowing cards 106 in switch 104 to exchange information with one another.) Devi at [0023] (“Multiple physical links 108 to a particular destina-tion 112 may be aggregated to form an aggregated link 116. Aggregated link 116 represents multiple physical links 108 that are addressed by a single logical address, such as a medium access controller (MAC) address, an Internet pro-tocol (IP) address, or other suitable address or identifier, or otherwise treated as a single logical link between switch 104 and a destination 112. Because aggregated link 116 includes multiple physical links 108 that go to the same destination 112, a packet sent to destination 112 need only be sent to one port 110 corresponding to one physical link 108 of aggre-gated link 116. One technical advantage of certain embodi-ments of the present invention is that ingress cards 106 may communicate packets to ports 110 in some alternating fash-ion, such as round robin, random selection, pseudo-random selection using hash tables, or any other selection technique. This allows the load on any particular physical link 108 to be balanced with the loads on other physical links 108 in aggregated link 116.”) Devi at [0030] (“Destination identifier 215 represents information identifying an intermediate or final destination 112 for the packet. Destination identifier 215 may include a network address for destination 112, a logical address for an egress link, or any other information useful for identifying the destination of a packet. Destination identifier 215 contem-plates any information determinable from a packet that allows card 106 to identify one or more destinations 112 for the packet.”) Devi at [0031] (“Destination information 216 associates destination identifiers 215 in packets with egress links. Destination information 216 may be organized in any suitable fashion, including one or more tables, files, databases, or other form of organization. Card 106 uses destination information 216 to determine an egress link or links using destination iden-tifier 215 from a packet. For example, a multicast packet may include a multicast group identifier instead of indi-vidual link addresses. Ingress card 106 receives the packet, looks up the

No.	’740 Patent Claim 10	Devi
		multicast group identifier, and determines the egress links associated with the multicast group. In another example, the packet may include a final destination address (such as an IP address for a receiving device) rather than a link address from switch 104. In that case, ingress link 106 uses destination information 216 to determine an egress link to the final destination. Destination information 216 may include one or more layers as well. For example, destination information 216 may include a first table relating a multicast group to destination IP addresses, and a second table relating the destination IP addresses to logical link addresses. Card 106 may also update destination information 216 as new or additional information becomes available, such as when card 106 receives an acknowledgement from a destination 112 indicating that a packet was received.”) Devi at [0033] (“In operation, card 106 receives a packet from network 102. Ingress card 106 determines the destination for the particular packet and consults destination information 216 to see if that destination is associated with a particular egress link. Some packets may have a destination identifier 215 that card 106 has not yet learned, and so destination information 216 does not include an egress link or links associated with destination identifier 215 of the packet. In such cases, card 106 may flood all available egress links so that the packet will arrive at its proper destination, and that destination will send an acknowledgment back to card 106. To flood destinations, card 106 determines logical links coupled to switch 104 using port information 218. Card 106 communicates the packet to these links, excluding the link from which the packet was received. In a particular embodi-ment, flooded cards will replicate the packet to all ports within that card, so card 106 need only communicate the packet once to a particular egress card 106.”) Devi at [0037] (“Card 106 receives a packet at step 302. At step 304, card determines destination identifier 215 for the packet and uses destination information 216 to determine if card 106 has previously learned the destination identifier 215. If the destination has not been previously learned, card 106 floods all egress links at step 306, shown in more detail in FIG. 4 described below.”) Devi at [0038] (“If destination identifier 215 of the packet has been learned, card 106 next determines if destination identifier 215 identifies a multicast or a unicast destination at step

No.	’740 Patent Claim 10	Devi
		308. If the packet is a multicast packet, card 106 selects one of the destinations for the packet using destination information 216 at step 310. On the other hand, if the packet is a unicast packet, there is only one destination, so there is no need to perform a selection process.”) Devi at [0041] (“If the link is an aggregated link 116, card 106 determines a destination port 110 of aggregated link 116 using the information in port information 218 at step 320. Card 106 may select a particular destination port 110 from the multiple ports 110 of the aggregated link using random selection, load balancing, pseudo-random selection or hash-ing using part of the packet information, historical tracking, round robin, or any other method of selection. Card 106 then communicates the packet to the destination port 110 at step 322.”) Devi at [0048] (“If the egress link is an aggregated isolated link, card 106 communicates the packet to destination port 110 of aggregated link 116. Card 106 determines destination port 110 for the packet at step 416. Card 106 may use any suitable method to determine destination port 110, including any of the methods described in conjunction with FIGS. 2 and 3. Card 106 then communicates the packet to destination port 110 at step 418.”) Devi at [0049] (“Card 106 then determines if there are any remain-ing egress links at step 420. If there are egress links remaining, then card 106 selects a new link to process at step 406. If there are no remaining links, then card 106 waits for an acknowledgement from one of the destinations that the packet was received. Card 106 receives the acknowledge-ment at step 422, and determines the link from which the acknowledgement was received at step 424. Card 106 then updates destination information 216 by associating the source link with destination identifier 215 at step 426. When the packet is a multicast packet, card 106 waits for acknowl-edgements from any remaining destinations at step 428.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Devi in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[d] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references.

No.	’740 Patent Claim 10	Devi
		For example, Bruckman discloses distributing data frames over physical links and traces based on a hash function involving a division operation (%). Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0025] (“Typically, setting the protection policy includes determining a maximum number of the physical links that may fail while the logical link continues to provide at least the guaranteed bandwidth for the connection. In one embodiment, the guaranteed bandwidth is a bandwidth B, and the plurality of physical links consists of N links, and the maximum number is an integer P, and the link bandwidth allocated to each of the links is no less than B/(N-P). Conveying the data may further include managing the transmission of the data responsively to an actual number X of the physical links that have failed so that the guaranteed bandwidth on each of the links is limited to B/(N-X), X P, and an excess bandwidth on the physical links over the guaranteed bandwidth is available for other connections.”) Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”)

No.	’740 Patent Claim 10	Devi

		Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:
		
		TABLE I
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		”)
		
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		
		Bruckman at [0067] (“A similar problem may arise if there is a failure in a link in an aggregation group or in one of a number of line cards serving the aggregation group. In this case, to maintain the bandwidth allocation B made by CAC 44, each of the remaining links in the group must now carry, on average, B/(N-M) traffic, wherein M is the number of links in the group that are out of service. If only BIN has been allocated to each link, the remaining active links may not have sufficient bandwidth to continue to provide the bandwidth that has been guaranteed to the connections that they are required to carry. A similar problem arises with respect to loading of traces 52. For example, if there is a failure in LC2 or in one of links 30 in group 36 that connect to LC2, the trace connecting the multiplexer 50 in LCl will have to carry a substantially larger share of the bandwidth, or even all of the bandwidth, that is allocated to the connection in ques-tion.”)

No.	’740 Patent Claim 10	Devi
		Bruckman at [0068] (“FIG. 3 is a flow chart that schematically illustrates a method for dealing with these problems of fluctuating bandwidth requirements, in accordance with an embodiment of the present invention. In order to provide sufficient bandwidth for failure protection, CAC 44 uses a safety margin based on a protection parameter P, which is assigned at a protection setting step 60. P represents the maximum number of links in the group that can be out of service while still permitting the aggregation group to provide a given connection with the bandwidth that has been guaranteed to the connection. CAC 44 will then allocate at least B/(N-P) bandwidth to each link in the group, so that if P links fail, the group still provides total bandwidth of (N-P)*B/(N-P)= B. Setting P=l is equivalent to 1:N protection, so that the group will be unaffected by failure of a single link. In the example of group 36, shown in FIG. 2, setting P=2 will give both facility and equipment protection, i.e., the group will be unaffected not only by failure of a link, but also by failure of one of line cards 34. In the extreme case, in which P=N-1, CAC 44 will allocate the full bandwidth B on each link in the group.”) As another example, Singh discloses determining a ratio between the number of ingress and egress links and the number of links carrying data to the backplane and using a modulo to correspond to the channel’s link number. Singh at 9:30-43 (“The ratio between the number of line ingress links and the number of links carrying data to the backplane gives the backplane speedup for the system. In this example, there are 10 ingress links into the MS and 20 links (2 backplane channels) carrying that data to the backplane. This gives a backplane speedup of 2x. As another example, with 8 ingress links and 12 backplane links, there is a speedup of 1.5x. It should be noted that in addition to the backplane speedup, there is also an ingress/egress speedup. With 10 ingress links capable of carrying 2 Gbps each of raw data, this presents a 20 Gbps interface to the MS. An OC-192 only has approximately 10 Gbps worth of data. Taking into account cell overhead and cell quantization inefficiencies, there still remains excess capacity in the links.”) Singh at 11:29-38 (“FIG. 9 is a diagram illustrating link to channel assignments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio

No.	’740 Patent Claim 10	Devi
		between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a channel, whether it is a backplane channel or an ingress/egress channel. There is no logical relationship between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.”) Singh at 15:15-39 (“The number of crossbars that are required in a system is dependent on how many links are being used to create the backplane channels. There should be an even number of crossbars and they would be divided evenly across the switch cards. The following equation, for most cases, pro-vides the correct number of crossbars: # of Crossbars=(# links per ingress channelx# of ingress channels per portx# of port cardsx speedup)/32. For the 8x8 configuration, the # of crossbars should be multiplied by (4x# of iMS)/(# backplane channels per port card). The number of port cards should be rounded up to the nearest supported configuration, i.e. 8, 16, or 32. The speedup in the case of crossbars should be the fractional speedup that is desired. Example to determine the number of arbiters and cross-bars for the following system: 4 channel port cards (40 Gbps) 8 links per channel 16 port cards Speedup=l.5 # of arbiters=(4x2x2)/2=8 # of crossbars=(8x4x16xl.5)/32=24. This would give 3crossbars per arbiter.”) Singh at 16:28-44 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 16x16 and 32x32 is the organization of the switchplane. The port card remains the same. Backplane channels 1 and 2 are used for the backplane connectivity. Ingress and egress links 30-39 on the MS would

No.	’740 Patent Claim 10	Devi
		not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 16, correspond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”) Singh at 17:31-49 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 8x8 and 16x16 is the organi-zation of the switchplane. The port card remains the same. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Links 0-7 and 24-31 on the arbiters would not be used and would be powered off. Links 0-7 and 24-31 on the crossbars would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Backplane channels 1 and 2 are used for the backplane connectivity. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 8, corre-spond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”)

No.	’740 Patent Claim 11	Devi
11	The method according to claim 10, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical links	Devi discloses the method according to claim 10, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo. For example, Devi discloses using hashing or other selection methods to select the egress links of a particular destination port based on the information in the destination identifier 215 of the packet. A person of ordinary skill in the art would understand that applying a hash

No.	’740 Patent Claim 11	Devi
	responsively to respective first and second subsets of bits in a binary representation of the modulo.	function includes determining parameters responsive to the system and packet features, determining the egress links associated with a destination port based on subsets of bits of the binary representation of the modulo, and generating a result. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Devi is found to not meet this limitation, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo would have been obvious to a person having ordinary skill in the art, as explained below. *See supra* Claim 10 Devi at [0005] (“In accordance with one embodiment of the present invention, a method performed by an ingress card includes receiving a packet comprising a destination identifier. The method also includes determining an aggregated link asso-ciated with the destination identifier, and determining a destination port from the egress ports of the aggregated link. The method further includes communicating the packet to the destination port.”) Devi at [0006] (“In accordance with another embodiment of the present invention, a method performed by an ingress card includes receiving a packet and determining egress links for the packet. The method also includes determining whether each link is an aggregated link with multiple egress ports or a non-aggregated link with a single egress port. For each aggregated link, a destination port is determined from the egress ports, and the packet is communicated to the desti-nation port. For each non-aggregated link, a copy of the packet is communicated to the single egress port.”) Devi at [0007] (“Important technical advantages of certain embodi-ments of the present invention include implementing link aggregation using distributed processing among ingress components. For example, individual ingress cards can determine a destination port for a packet, eliminating the need for a separate processing stage implemented in hard-ware and/or software to separately determine the destination port from a packet after the ingress card

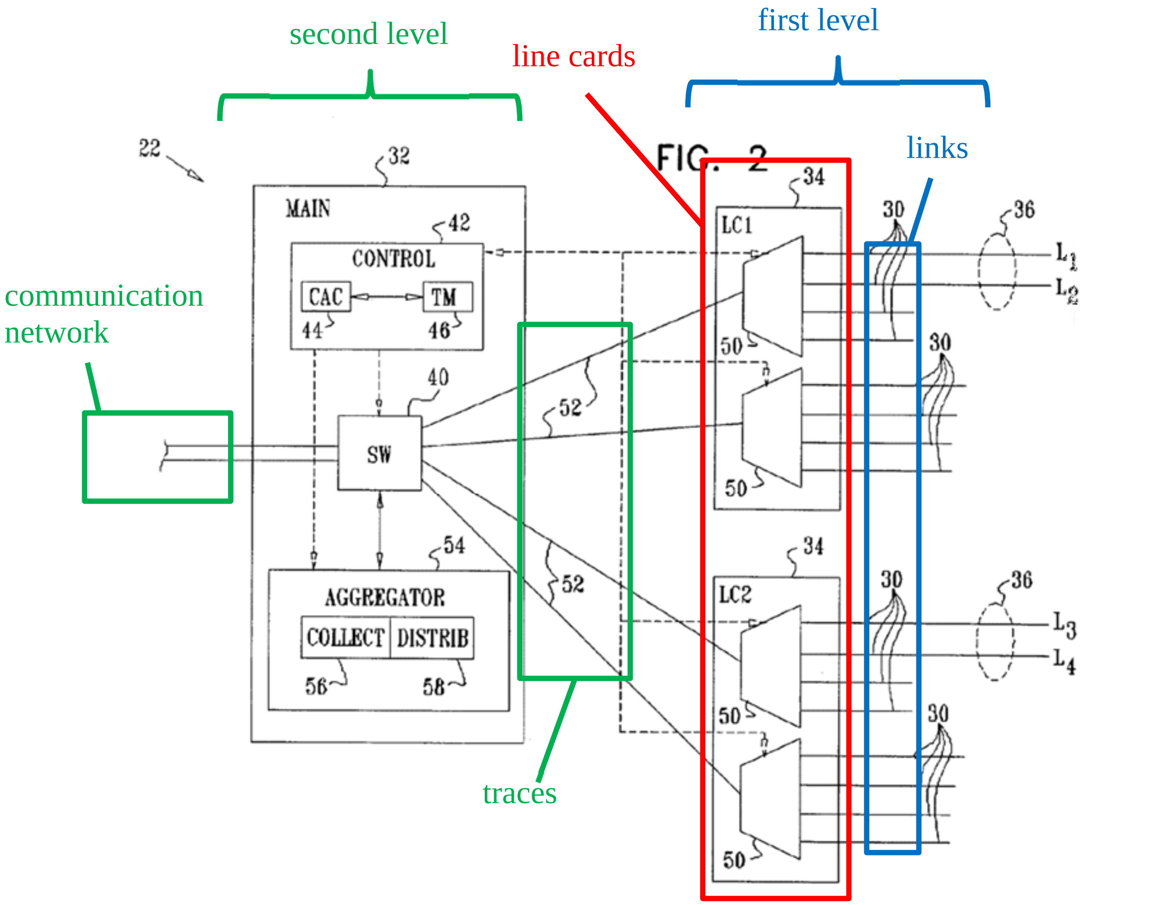
No.	’740 Patent Claim 11	Devi
		forwards the packet to a link. This improves the efficiency and speed of packet forwarding in a switch.”) Devi at [0008] (“Yet another important technical advantage of cer-tain embodiments is a distributed architecture that may be used to process a variety of packet traffic. For example, certain embodiments of an ingress card can forward unicast, multicast, and bridging traffic. Particular embodiments may process packets from different protocols as well.”) Devi at [0009] (“Other important technical advantages of certain embodiments of the present invention include load balanc-ing. The tables or other information used by the ingress card to determine a destination port for packets may also include additional usage information that allows the ingress card to determine a destination port from the egress ports in an aggregated link. This means that rather than using one link to the exclusion of others, loads may be distributed among several links in an aggregated link in a more balanced fashion. Particular embodiments of the present invention may have some, all or none of the enumerated technical advantages. Still other important technical advantages will be apparent to one skilled in the art from the following figures, description, and claims.”) Devi at [0015] (“FIG. 1 shows a system 100 that includes a switch 104 coupled to a network 102. Using network 102 and switch 104, packets are communicated from sources 103 to destinations 112. Components of switch 104 are referred to as "ingress" components when receiving packets from sources 103 and "egress" components when sending packets to destinations 112.”) Devi at [0016] (“Network 102 represents any suitable structure for communicating packets, cells, frames, segments, or other portions of data (generally referred to as "packets"). Net-work 102 may represent the Internet, extranet, local area network (LAN), synchronous optical network (SONET), wide area network (WAN), the public switched telephone network (PSTN), or any other suitable network for commu-nicating information. Network 102 may include routers, switches, hubs, endpoints, or any other network device that communicates information. Network 102 contemplates any number or arrangement of components that exchange infor-mation.”)

No.	’740 Patent Claim 11	Devi
		Devi at [0018] (“Switch 104 sends and receives packets. Switch 104 may represent any suitable device, including an Ethernet switch, router, hub, or any other suitable hardware and/or software configured to receive packets and communicate them to other devices. Switch 104 includes cards 106 coupled to physical links 108 and a backplane 114 that represents hardware and/or software allowing cards 106 in switch 104 to exchange information with one another.) Devi at [0023] (“Multiple physical links 108 to a particular destina-tion 112 may be aggregated to form an aggregated link 116. Aggregated link 116 represents multiple physical links 108 that are addressed by a single logical address, such as a medium access controller (MAC) address, an Internet pro-tocol (IP) address, or other suitable address or identifier, or otherwise treated as a single logical link between switch 104 and a destination 112. Because aggregated link 116 includes multiple physical links 108 that go to the same destination 112, a packet sent to destination 112 need only be sent to one port 110 corresponding to one physical link 108 of aggre-gated link 116. One technical advantage of certain embodi-ments of the present invention is that ingress cards 106 may communicate packets to ports 110 in some alternating fash-ion, such as round robin, random selection, pseudo-random selection using hash tables, or any other selection technique. This allows the load on any particular physical link 108 to be balanced with the loads on other physical links 108 in aggregated link 116.”) Devi at [0030] (“Destination identifier 215 represents information identifying an intermediate or final destination 112 for the packet. Destination identifier 215 may include a network address for destination 112, a logical address for an egress link, or any other information useful for identifying the destination of a packet. Destination identifier 215 contem-plates any information determinable from a packet that allows card 106 to identify one or more destinations 112 for the packet.”) Devi at [0031] (“Destination information 216 associates destination identifiers 215 in packets with egress links. Destination information 216 may be organized in any suitable fashion, including one or more tables, files, databases, or other form of organization. Card 106 uses

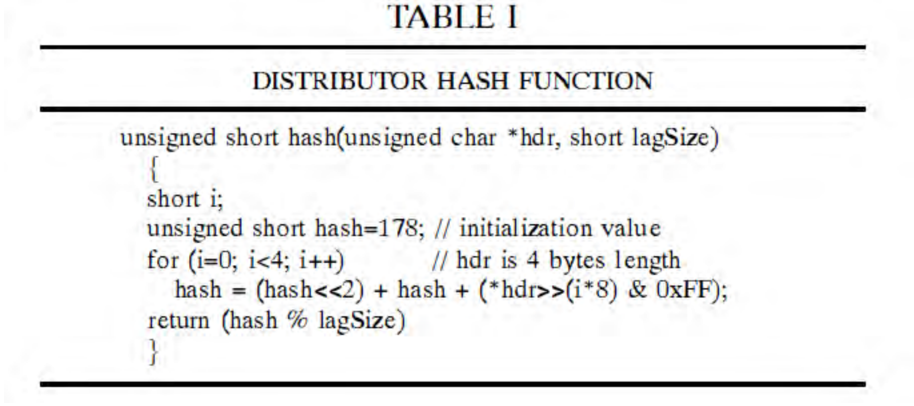
No.	’740 Patent Claim 11	Devi
		destination information 216 to determine an egress link or links using destination iden-tifier 215 from a packet. For example, a multicast packet may include a multicast group identifier instead of indi-vidual link addresses. Ingress card 106 receives the packet, looks up the multicast group identifier, and determines the egress links associated with the multicast group. In another example, the packet may include a final destination address (such as an IP address for a receiving device) rather than a link address from switch 104. In that case, ingress link 106 uses destination information 216 to determine an egress link to the final destination. Destination information 216 may include one or more layers as well. For example, destination information 216 may include a first table relating a multicast group to destination IP addresses, and a second table relating the destination IP addresses to logical link addresses. Card 106 may also update destination information 216 as new or additional information becomes available, such as when card 106 receives an acknowledgement from a destination 112 indicating that a packet was received.”) Devi at [0033] (“In operation, card 106 receives a packet from network 102. Ingress card 106 determines the destination for the particular packet and consults destination information 216 to see if that destination is associated with a particular egress link. Some packets may have a destination identifier 215 that card 106 has not yet learned, and so destination information 216 does not include an egress link or links associated with destination identifier 215 of the packet. In such cases, card 106 may flood all available egress links so that the packet will arrive at its proper destination, and that destination will send an acknowledgment back to card 106. To flood destinations, card 106 determines logical links coupled to switch 104 using port information 218. Card 106 communicates the packet to these links, excluding the link from which the packet was received. In a particular embodi-ment, flooded cards will replicate the packet to all ports within that card, so card 106 need only communicate the packet once to a particular egress card 106.”) Devi at [0037] (“Card 106 receives a packet at step 302. At step 304, card determines destination identifier 215 for the packet and uses destination information 216 to determine if card 106 has previously learned the destination identifier 215. If the destination has not been previously learned, card 106 floods all egress links at step 306, shown in more detail in FIG. 4 described below.”)

No.	’740 Patent Claim 11	Devi
		Devi at [0038] (“If destination identifier 215 of the packet has been learned, card 106 next determines if destination identifier 215 identifies a multicast or a unicast destination at step 308. If the packet is a multicast packet, card 106 selects one of the destinations for the packet using destination information 216 at step 310. On the other hand, if the packet is a unicast packet, there is only one destination, so there is no need to perform a selection process.”) Devi at [0041] (“If the link is an aggregated link 116, card 106 determines a destination port 110 of aggregated link 116 using the information in port information 218 at step 320. Card 106 may select a particular destination port 110 from the multiple ports 110 of the aggregated link using random selection, load balancing, pseudo-random selection or hash-ing using part of the packet information, historical tracking, round robin, or any other method of selection. Card 106 then communicates the packet to the destination port 110 at step 322.”) Devi at [0048] (“If the egress link is an aggregated isolated link, card 106 communicates the packet to destination port 110 of aggregated link 116. Card 106 determines destination port 110 for the packet at step 416. Card 106 may use any suitable method to determine destination port 110, including any of the methods described in conjunction with FIGS. 2 and 3. Card 106 then communicates the packet to destination port 110 at step 418.”) Devi at [0049] (“Card 106 then determines if there are any remain-ing egress links at step 420. If there are egress links remaining, then card 106 selects a new link to process at step 406. If there are no remaining links, then card 106 waits for an acknowledgement from one of the destinations that the packet was received. Card 106 receives the acknowledge-ment at step 422, and determines the link from which the acknowledgement was received at step 424. Card 106 then updates destination information 216 by associating the source link with destination identifier 215 at step 426. When the packet is a multicast packet, card 106 waits for acknowl-edgements from any remaining destinations at step 428.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Devi in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references

No.	’740 Patent Claim 11	Devi
		identified in element 11 of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses distributing data frames over physical links and traces based on a division operation in the hash function involving specific byte lengths of the frame information. Bruckman at Figure 2 (annotated)

No.	’740 Patent Claim 11	Devi
		 FIG. 2 second level line cards first level links communication network traces 22 32 MAIN 42 CONTROL CAC TM 44 46 40 SW 54 AGGREGATOR COLLECT DISTRIB 56 58 52 52 34 LC1 50 50 30 36 L1 L2 30 34 LC2 50 50 30 36 L3 L4 30 Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:

No.	’740 Patent Claim 11	Devi

		
		TABLE I
		DISTRIBUTOR HASH FUNCTION
		`unsigned short hash(unsigned char *hdr, short lagSize)`
		`  {`
		`  short i;`
		`  unsigned short hash=178; // initialization value`
		`  for (i=0; i<4; i++)          // hdr is 4 bytes length`
		`    hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);`
		`  return (hash % lagSize)`
		`  }`
		”)
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		As another example, Solomon discloses using a subset of bits to encode for the selected physical port involving specific byte lengths of the frame information.
		Solomon at [0054] (“Having selected a physical port, RSVP-TE proces-sor 30 of switch A now generates a suitable MPLS label, at a label generation step 64. The preceding node upstream of switch A will subsequently attach this MPLS label to all MPLS packets transmitted through tunnel 28 to switch A. The label is assigned, in conjunction with the mapping function of mapper 34, so as to ensure that all MPLS packets carrying this label are switched through the physical port that was selected for this tunnel at step 62. For this purpose, RSVP-TE processor 30 of switch A dedicates a sub-set of the bits of MPLS label 52 to encode the serial number of the selected physical port. For example, the four least-signifi-cant bits of MPLS label 52 may be used for encoding the selected port number. This configuration is suitable for representing LAG groups having up to 16 physical ports (N<16). The remaining

No.	’740 Patent Claim 11	Devi
		bits of MPLS label 52 may be chosen at random or using any suitable method known in the art.”) Solomon at [0056] (“Mapper 34 of switch A maps the received packets belonging to tunnel 28 to the selected physical Ethernet port at a mapping step 70. For this purpose, mapper 34 extracts the MPLS label from each received packet and decodes the selected physical port number from the dedicated sub-set of bits, such as the four LSB, as described in step 64 above. The decoded value is used for mapping the packet to the selected physical port, which was allocated by the CAC processor at step 62 above. In the four-bit example described above, the mapping function may be written explicitly as: Selected port number=((MPLS label) and (0x0000F)), wherein "and" denotes the "bitwise and" operator.”)

No.	’740 Patent Claim 12	Devi
12	The method according to claim 1, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port	Devi discloses the method according to claim 1, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port. For example, Devi discloses packer destination identifier 215 which may include “a network address for destination 112, a logical address for an egress link, or any other information useful for identifying the destination of a packet” “such as a medium access controller (MAC) address, an Internet protocol (IP) address, or other suitable address or identifier.” *See supra* Claim 1 Devi at [0005] (“In accordance with one embodiment of the present invention, a method performed by an ingress card includes receiving a packet comprising a destination identifier. The method also includes determining an aggregated link asso-ciated with the destination identifier, and determining a destination port from the egress ports of the aggregated link. The method further includes communicating the packet to the destination port.”)

No.	’740 Patent Claim 12	Devi
	and a destination TCP port.	Devi at [0006] (“In accordance with another embodiment of the present invention, a method performed by an ingress card includes receiving a packet and determining egress links for the packet. The method also includes determining whether each link is an aggregated link with multiple egress ports or a non-aggregated link with a single egress port. For each aggregated link, a destination port is determined from the egress ports, and the packet is communicated to the desti-nation port. For each non-aggregated link, a copy of the packet is communicated to the single egress port.”) Devi at [0009] (“Other important technical advantages of certain embodiments of the present invention include load balanc-ing. The tables or other information used by the ingress card to determine a destination port for packets may also include additional usage information that allows the ingress card to determine a destination port from the egress ports in an aggregated link. This means that rather than using one link to the exclusion of others, loads may be distributed among several links in an aggregated link in a more balanced fashion. Particular embodiments of the present invention may have some, all or none of the enumerated technical advantages. Still other important technical advantages will be apparent to one skilled in the art from the following figures, description, and claims.”) Devi at [0023] (“Multiple physical links 108 to a particular destina-tion 112 may be aggregated to form an aggregated link 116. Aggregated link 116 represents multiple physical links 108 that are addressed by a single logical address, such as a medium access controller (MAC) address, an Internet pro-tocol (IP) address, or other suitable address or identifier, or otherwise treated as a single logical link between switch 104 and a destination 112. Because aggregated link 116 includes multiple physical links 108 that go to the same destination 112, a packet sent to destination 112 need only be sent to one port 110 corresponding to one physical link 108 of aggre-gated link 116. One technical advantage of certain embodi-ments of the present invention is that ingress cards 106 may communicate packets to ports 110 in some alternating fash-ion, such as round robin, random selection, pseudo-random selection using hash tables, or any other selection technique. This allows the load on any particular physical link 108 to be balanced with the loads on other physical links 108 in aggregated link 116.”)

No.	’740 Patent Claim 12	Devi
		Devi at [0030] (“Destination identifier 215 represents information identifying an intermediate or final destination 112 for the packet. Destination identifier 215 may include a network address for destination 112, a logical address for an egress link, or any other information useful for identifying the destination of a packet. Destination identifier 215 contem-plates any information determinable from a packet that allows card 106 to identify one or more destinations 112 for the packet.”) Devi at [0031] (“Destination information 216 associates destination identifiers 215 in packets with egress links. Destination information 216 may be organized in any suitable fashion, including one or more tables, files, databases, or other form of organization. Card 106 uses destination information 216 to determine an egress link or links using destination iden-tifier 215 from a packet. For example, a multicast packet may include a multicast group identifier instead of indi-vidual link addresses. Ingress card 106 receives the packet, looks up the multicast group identifier, and determines the egress links associated with the multicast group. In another example, the packet may include a final destination address (such as an IP address for a receiving device) rather than a link address from switch 104. In that case, ingress link 106 uses destination information 216 to determine an egress link to the final destination. Destination information 216 may include one or more layers as well. For example, destination information 216 may include a first table relating a multicast group to destination IP addresses, and a second table relating the destination IP addresses to logical link addresses. Card 106 may also update destination information 216 as new or additional information becomes available, such as when card 106 receives an acknowledgement from a destination 112 indicating that a packet was received.”) Devi at [0032] (“Port information 218 refers to any form of data that identifies ports 110 associated with a particular link. For example, port information 218 for a non-aggregated link 108 identifies the single egress port 110 for the non-aggregated link 108. Port information 218 for an aggregated link 116 identifies the multiple ports 110 coupled to the physical links 108 of aggregated link 116, which may include ports 110 on a single card 106 or multiple cards 106. Port information 218 may also identify which card 106 is coupled to each port 110.

No.	’740 Patent Claim 12	Devi
		Port information 218 contemplates any suitable arrangement of data, such as a table, database or any other format. Card 106 may update port information 218 as network conditions change, e.g., when a port 110 or physical link 108 fails.”)

No.	’740 Patent Claim 13	Devi
13[preamble]	A method for communication, comprising:	Devi discloses a method for communication. *See supra at* 1[preamble].
13[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	Devi discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].
13[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Devi discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra at* 1[c].
13[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	Devi discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra at* 1[e].

No.	’740 Patent Claim 13	Devi
13[d]	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	Devi discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. *See supra at* 1[f].
13[e]	sending the data frame over the selected first and second physical links,	Devi discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].
13[f]	coupling the network node to the one or more interface modules and	Devi discloses coupling the network node to the one or more interface modules. *See supra at* 1[a].
13[g]	coupling each of the one or more interface modules to the communication network comprising	Devi discloses coupling each of the one or more interface modules to the communication network. *See supra at* 1[c].
13[h]	specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication	Devi discloses specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node. For example, Devi discloses a communication network with specific processing capabilities of network service. A person of ordinary skill in the art would understand that these processing capabilities can include one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Devi is found to not meet this limitation, coupling

No.	’740 Patent Claim 13	Devi
	service provided by the communication network to the network node, and	each of the one or more interface modules to the communication network comprising specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node would have been obvious to a person having ordinary skill in the art, as explained below. Devi at [0003] (“As telecommunication networks handle more and more traffic, new methods are constantly being developed that allow the networks to process larger flows of informa-tion. Link aggregation is one example. In link aggregation, several physical links are aggregated to appear as one logical link to a telecommunications system. This aggregation allows all of the links to be used actively, rather than having particular links reserved. Thus, link aggregation provides more efficient use of network resources and better load balancing. The tradeoff, however, is that link aggregation may also require substantial additional processing and/or hardware to implement the aggregated links.”) Devi at [0009] (“Other important technical advantages of certain embodiments of the present invention include load balanc-ing. The tables or other information used by the ingress card to determine a destination port for packets may also include additional usage information that allows the ingress card to determine a destination port from the egress ports in an aggregated link. This means that rather than using one link to the exclusion of others, loads may be distributed among several links in an aggregated link in a more balanced fashion. Particular embodiments of the present invention may have some, all or none of the enumerated technical advantages. Still other important technical advantages will be apparent to one skilled in the art from the following figures, description, and claims.”) Devi at [0015] (“FIG. 1 shows a system 100 that includes a switch 104 coupled to a network 102. Using network 102 and switch 104, packets are communicated from sources 103 to destinations 112. Components of switch 104 are referred to as "ingress" components when receiving packets from sources 103 and "egress" components when sending packets to destinations 112.”)

No.	’740 Patent Claim 13	Devi
		Devi at [0016] (“Network 102 represents any suitable structure for communicating packets, cells, frames, segments, or other portions of data (generally referred to as "packets"). Net-work 102 may represent the Internet, extranet, local area network (LAN), synchronous optical network (SONET), wide area network (WAN), the public switched telephone network (PSTN), or any other suitable network for commu-nicating information. Network 102 may include routers, switches, hubs, endpoints, or any other network device that communicates information. Network 102 contemplates any number or arrangement of components that exchange infor-mation.”) Devi at [0019] (“Cards 106 represent separate components of hard-ware and/or software in switch 104 that exchange packets with network 102. Cards 106 may include traditional inter-face cards, as well as any other component, module, or part of switch 104 capable of independently receiving packets and communicating those packets to other components of switch 104. Each card 106 has one or more ports 110 coupled to physical links 108. Cards 106 may exchange information and packets with one another using backplane 114. Each card 106 includes sufficient processing capability to identify a port 110 in another card 106 in switch 104 and to communicate a packet to that port 110.”) Devi at [0023] (“Multiple physical links 108 to a particular destina-tion 112 may be aggregated to form an aggregated link 116. Aggregated link 116 represents multiple physical links 108 that are addressed by a single logical address, such as a medium access controller (MAC) address, an Internet pro-tocol (IP) address, or other suitable address or identifier, or otherwise treated as a single logical link between switch 104 and a destination 112. Because aggregated link 116 includes multiple physical links 108 that go to the same destination 112, a packet sent to destination 112 need only be sent to one port 110 corresponding to one physical link 108 of aggre-gated link 116. One technical advantage of certain embodi-ments of the present invention is that ingress cards 106 may communicate packets to ports 110 in some alternating fash-ion, such as round robin, random selection, pseudo-random selection using hash tables, or any other selection technique. This allows the load on any particular physical link 108 to be balanced with the loads on other physical links 108 in aggregated link 116.”)

No.	’740 Patent Claim 13	Devi
		Devi at [0041] (“If the link is an aggregated link 116, card 106 determines a destination port 110 of aggregated link 116 using the information in port information 218 at step 320. Card 106 may select a particular destination port 110 from the multiple ports 110 of the aggregated link using random selection, load balancing, pseudo-random selection or hash-ing using part of the packet information, historical tracking, round robin, or any other method of selection. Card 106 then communicates the packet to the destination port 110 at step 322.”) Devi at Claim 4 (“The method of claim 1, wherein the step of determining a destination port comprises determining the destination port according to a load-balancing algorithm that distributes traffic among the egress ports of the aggregated link.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Devi in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 13[h] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses specifying certain committed, excess, and guaranteed bandwidths, including CIR, EIR, and PIR, respectively. Bruckman at [0013] (“Service level agreements between network service providers and customers commonly specify a certain com-mitted bandwidth, or committed information rate (CIR), which the service provider guarantees to provide to the customer at all times, regardless of bandwidth stress on the network. Additionally or alternatively, the agreement may specify an excess bandwidth, which is available to the customer when network traffic permits. The excess band-width is typically used by customers for lower-priority services, which do not require committed bandwidth. The network service provider may guarantee the customer a certain minimum excess bandwidth, or excess information rate (EIR), in order to avoid starvation of such services in case of bandwidth stress. In general, the bandwidth guaran-teed by a service provider, referred to as the peak informa-tion rate (PIR), may include either CIR, or EIR, or both CIR and EIR (in which case PIR=CIR+EIR). The term "guaran-teed bandwidth," as used in the context of the present patent application and in the claims, includes all these types of guaranteed bandwidth.”)

No.	’740 Patent Claim 13	Devi
		As another example, Solomon discloses a service property of a guaranteed bandwidth, sometimes denoted as CIR-Committed Information Rate and PIR-Peak Information Rate. Solomon at [0023] (“In another embodiment, establishing the path includes receiving an indication of a requested service property of the flow, and selecting the port includes assign-ing the port to the flow so as to comply with the requested service property. In a disclosed embodiment, the requested service property includes at least one of a guaranteed bandwidth, a peak bandwidth and a class-of-service. Addi-tionally or alternatively, assigning the port includes selecting the port having a maximum available bandwidth out of the plurality of aggregated ports. Further additionally or alter-natively, assigning the port includes selecting the port hav-ing a minimum available bandwidth out of the plurality of aggregated ports, which is still greater than or equal to the guaranteed bandwidth.”) Solomon at [0050] (“The method of FIG. 3 begins when the preceding node asks to establish a part of tunnel 28 (comprising one or more hops) for sending MPLS packets to MPLS/LAG switch 26 A. The preceding node requests and then receives the MPLS label, which it will subsequently attach to all packets that are sent to MPLS/LAG switch 26 labeledA. The preceding node sends downstream an RSVP-TE PATH mes-sage augmented with a LABEL_REQUEST object, as defined by RSVP-TE, to MPLS/LAG switch A, at a label requesting step 60. The PATH message typically comprises information regarding service properties that are requested for tunnel 28. The service properties may comprise a guar-anteed bandwidth (sometimes denoted CIR-Committed Information Rate) and a peak bandwidth (sometimes denoted PIR-Peak Information Rate), as well as a requested CoS (Class of Service-a measure of packet priority).”)
13[i]	allocating a bandwidth for the communication service over the first and second physical links responsively to	Devi discloses allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements. For example, Devi discloses using a load balancing algorithm that includes balancing distribution of packets according to the capabilities of the communication service. A person

No.	’740 Patent Claim 13	Devi
	the bandwidth requirements.	of ordinary skill in the art would understand that load-balancing involves distributing packet according to the bandwidth of the communication service. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Devi is found to not meet this limitation, allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements would have been obvious to a person having ordinary skill in the art, as explained below. Devi at [0003] (“As telecommunication networks handle more and more traffic, new methods are constantly being developed that allow the networks to process larger flows of informa-tion. Link aggregation is one example. In link aggregation, several physical links are aggregated to appear as one logical link to a telecommunications system. This aggregation allows all of the links to be used actively, rather than having particular links reserved. Thus, link aggregation provides more efficient use of network resources and better load balancing. The tradeoff, however, is that link aggregation may also require substantial additional processing and/or hardware to implement the aggregated links.”) Devi at [0009] (“Other important technical advantages of certain embodiments of the present invention include load balanc-ing. The tables or other information used by the ingress card to determine a destination port for packets may also include additional usage information that allows the ingress card to determine a destination port from the egress ports in an aggregated link. This means that rather than using one link to the exclusion of others, loads may be distributed among several links in an aggregated link in a more balanced fashion. Particular embodiments of the present invention may have some, all or none of the enumerated technical advantages. Still other important technical advantages will be apparent to one skilled in the art from the following figures, description, and claims.”) Devi at [0015] (“FIG. 1 shows a system 100 that includes a switch 104 coupled to a network 102. Using network 102 and switch 104, packets are communicated from sources 103 to destinations 112. Components of switch 104 are referred to as "ingress" components when

No.	’740 Patent Claim 13	Devi
		receiving packets from sources 103 and "egress" components when sending packets to destinations 112.”) Devi at [0016] (“Network 102 represents any suitable structure for communicating packets, cells, frames, segments, or other portions of data (generally referred to as "packets"). Net-work 102 may represent the Internet, extranet, local area network (LAN), synchronous optical network (SONET), wide area network (WAN), the public switched telephone network (PSTN), or any other suitable network for commu-nicating information. Network 102 may include routers, switches, hubs, endpoints, or any other network device that communicates information. Network 102 contemplates any number or arrangement of components that exchange infor-mation.”) Devi at [0019] (“Cards 106 represent separate components of hard-ware and/or software in switch 104 that exchange packets with network 102. Cards 106 may include traditional inter-face cards, as well as any other component, module, or part of switch 104 capable of independently receiving packets and communicating those packets to other components of switch 104. Each card 106 has one or more ports 110 coupled to physical links 108. Cards 106 may exchange information and packets with one another using backplane 114. Each card 106 includes sufficient processing capability to identify a port 110 in another card 106 in switch 104 and to communicate a packet to that port 110.”) Devi at [0023] (“Multiple physical links 108 to a particular destina-tion 112 may be aggregated to form an aggregated link 116. Aggregated link 116 represents multiple physical links 108 that are addressed by a single logical address, such as a medium access controller (MAC) address, an Internet pro-tocol (IP) address, or other suitable address or identifier, or otherwise treated as a single logical link between switch 104 and a destination 112. Because aggregated link 116 includes multiple physical links 108 that go to the same destination 112, a packet sent to destination 112 need only be sent to one port 110 corresponding to one physical link 108 of aggre-gated link 116. One technical advantage of certain embodi-ments of the present invention is that ingress cards 106 may communicate packets to ports 110 in some alternating fash-ion, such as round robin, random selection, pseudo-random selection using hash tables, or any other selection technique. This allows the load on any particular

No.	’740 Patent Claim 13	Devi
		physical link 108 to be balanced with the loads on other physical links 108 in aggregated link 116.”) Devi at [0041] (“If the link is an aggregated link 116, card 106 determines a destination port 110 of aggregated link 116 using the information in port information 218 at step 320. Card 106 may select a particular destination port 110 from the multiple ports 110 of the aggregated link using random selection, load balancing, pseudo-random selection or hash-ing using part of the packet information, historical tracking, round robin, or any other method of selection. Card 106 then communicates the packet to the destination port 110 at step 322.”) Devi at Claim 4 (“The method of claim 1, wherein the step of determining a destination port comprises determining the destination port according to a load-balancing algorithm that distributes traffic among the egress ports of the aggregated link.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Devi in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 13[h] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses specifying certain committed, excess, and guaranteed bandwidths, including CIR, EIR, and PIR, respectively. Bruckman at [0013] (“Service level agreements between network service providers and customers commonly specify a certain com-mitted bandwidth, or committed information rate (CIR), which the service provider guarantees to provide to the customer at all times, regardless of bandwidth stress on the network. Additionally or alternatively, the agreement may specify an excess bandwidth, which is available to the customer when network traffic permits. The excess band-width is typically used by customers for lower-priority services, which do not require committed bandwidth. The network service provider may guarantee the customer a certain minimum excess bandwidth, or excess information rate (EIR), in order to avoid starvation of such services in case of bandwidth stress. In general, the bandwidth guaran-teed by a service provider, referred to as the peak informa-tion rate (PIR), may include either CIR,

No.	’740 Patent Claim 13	Devi
		or EIR, or both CIR and EIR (in which case PIR=CIR+EIR). The term "guaran-teed bandwidth," as used in the context of the present patent application and in the claims, includes all these types of guaranteed bandwidth.”) As another example, Solomon discloses a service property of a guaranteed bandwidth, sometimes denoted as CIR-Committed Information Rate and PIR-Peak Information Rate. Solomon at [0023] (“In another embodiment, establishing the path includes receiving an indication of a requested service property of the flow, and selecting the port includes assign-ing the port to the flow so as to comply with the requested service property. In a disclosed embodiment, the requested service property includes at least one of a guaranteed bandwidth, a peak bandwidth and a class-of-service. Addi-tionally or alternatively, assigning the port includes selecting the port having a maximum available bandwidth out of the plurality of aggregated ports. Further additionally or alter-natively, assigning the port includes selecting the port hav-ing a minimum available bandwidth out of the plurality of aggregated ports, which is still greater than or equal to the guaranteed bandwidth.”) Solomon at [0050] (“The method of FIG. 3 begins when the preceding node asks to establish a part of tunnel 28 (comprising one or more hops) for sending MPLS packets to MPLS/LAG switch 26 A. The preceding node requests and then receives the MPLS label, which it will subsequently attach to all packets that are sent to MPLS/LAG switch 26 labeledA. The preceding node sends downstream an RSVP-TE PATH mes-sage augmented with a LABEL_REQUEST object, as defined by RSVP-TE, to MPLS/LAG switch A, at a label requesting step 60. The PATH message typically comprises information regarding service properties that are requested for tunnel 28. The service properties may comprise a guar-anteed bandwidth (sometimes denoted CIR-Committed Information Rate) and a peak bandwidth (sometimes denoted PIR-Peak Information Rate), as well as a requested CoS (Class of Service-a measure of packet priority).”)

No.	’740 Patent Claim 14	Devi
14[preamble]	A method for connecting user ports to a communication network, comprising:	Devi discloses a method for connecting user ports to a communication network. For example, Devi discloses a method for coupling ingress and egress ports to a network using physical links. Devi at Abstract (“A method performed by an ingress card includes receiving a packet with a destination identifier, and determining an aggregated link associated with the destination identifier that includes multiple egress ports. The method further includes determining a destination port from the egress ports of the aggregated link, and communicating the packet to the des-tination port.”) Devi at [0003] (“As telecommunication networks handle more and more traffic, new methods are constantly being developed that allow the networks to process larger flows of informa-tion. Link aggregation is one example. In link aggregation, several physical links are aggregated to appear as one logical link to a telecommunications system. This aggregation allows all of the links to be used actively, rather than having particular links reserved. Thus, link aggregation provides more efficient use of network resources and better load balancing. The tradeoff, however, is that link aggregation may also require substantial additional processing and/or hardware to implement the aggregated links.”) Devi at [0005] (“In accordance with one embodiment of the present invention, a method performed by an ingress card includes receiving a packet comprising a destination identifier. The method also includes determining an aggregated link asso-ciated with the destination identifier, and determining a destination port from the egress ports of the aggregated link. The method further includes communicating the packet to the destination port.”) Devi at [0006] (“In accordance with another embodiment of the present invention, a method performed by an ingress card includes receiving a packet and determining egress links for the packet. The method also includes determining whether each link is an aggregated link with multiple egress ports or a non-aggregated link with a single egress port. For each aggregated link, a destination port is determined from the egress ports, and the packet is communicated

No.	’740 Patent Claim 14	Devi
		to the desti-nation port. For each non-aggregated link, a copy of the packet is communicated to the single egress port.”) Devi at [0007] (“Important technical advantages of certain embodi-ments of the present invention include implementing link aggregation using distributed processing among ingress components. For example, individual ingress cards can determine a destination port for a packet, eliminating the need for a separate processing stage implemented in hard-ware and/or software to separately determine the destination port from a packet after the ingress card forwards the packet to a link. This improves the efficiency and speed of packet forwarding in a switch.”) Devi at [0008] (“Yet another important technical advantage of cer-tain embodiments is a distributed architecture that may be used to process a variety of packet traffic. For example, certain embodiments of an ingress card can forward unicast, multicast, and bridging traffic. Particular embodiments may process packets from different protocols as well.”) Devi at [0009] (“Other important technical advantages of certain embodiments of the present invention include load balanc-ing. The tables or other information used by the ingress card to determine a destination port for packets may also include additional usage information that allows the ingress card to determine a destination port from the egress ports in an aggregated link. This means that rather than using one link to the exclusion of others, loads may be distributed among several links in an aggregated link in a more balanced fashion. Particular embodiments of the present invention may have some, all or none of the enumerated technical advantages. Still other important technical advantages will be apparent to one skilled in the art from the following figures, description, and claims.”) Devi at [0015] (“FIG. 1 shows a system 100 that includes a switch 104 coupled to a network 102. Using network 102 and switch 104, packets are communicated from sources 103 to destinations 112. Components of switch 104 are referred to as "ingress" components when receiving packets from sources 103 and "egress" components when sending packets to destinations 112.”)

No.	’740 Patent Claim 14	Devi
		Devi at [0016] (“Network 102 represents any suitable structure for communicating packets, cells, frames, segments, or other portions of data (generally referred to as "packets"). Net-work 102 may represent the Internet, extranet, local area network (LAN), synchronous optical network (SONET), wide area network (WAN), the public switched telephone network (PSTN), or any other suitable network for commu-nicating information. Network 102 may include routers, switches, hubs, endpoints, or any other network device that communicates information. Network 102 contemplates any number or arrangement of components that exchange infor-mation.”)
14[a]	coupling the user ports to one or more user interface modules;	Devi discloses coupling the user ports to one or more user interface modules. *See supra at* 1[a].
14[b]	coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel,	Devi discloses coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel. *See supra at* 1[c], 3.
14[c]	at least one of said backplane traces being bi-directional and operative to communicate in both an upstream direction and a downstream direction;	Devi discloses at least one of said backplane traces being bi-directional and operative to communicate in both an upstream direction and a downstream direction. *See supra at* 14[b], 1[d].
14[d]	receiving data frames sent between the user ports and the communication	Devi discloses receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes. *See supra at* 14[a], 1[e].

No.	’740 Patent Claim 14	Devi
	network, the data frames having respective frame attributes;	
14[e]	for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces; and	Devi discloses for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces. *See supra at* 14[b], 1[f].
14[f]	sending the data frame over the selected backplane trace;	Devi discloses sending the data frame over the selected backplane trace. *See supra at* 14[e], 1[g].
14[g]	said sending comprising communicating along said at least one of said backplane traces.	Devi discloses said sending comprising communicating along said at least one of said backplane traces. *See supra at* 14[f], 1[h].

No.	’740 Patent Claim 15	Devi
15[preamble]	A method for connecting user ports to a communication network, comprising:	Devi discloses a method for connecting user ports to a communication network. *See supra at* 14[preamble].
15[a]	coupling the user ports to one or more user interface modules;	Devi discloses coupling the user ports to one or more user interface modules. *See supra at* 14[a].
15[b]	coupling each user interface module to the communication	Devi discloses coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel.

No.	’740 Patent Claim 15	Devi
	network via a backplane using two or more backplane traces arranged in parallel;	*See supra at* 14[b].
15[c]	receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes;	Devi discloses receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes. *See supra at* 14[d].
15[d]	for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces; and	Devi discloses for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces. *See supra at* 14[e].
15[e]	sending the data frame over the selected backplane trace,	Devi discloses sending the data frame over the selected backplane trace. *See supra at* 14[f].
15[f]	at least some of the backplane traces being aggregated into an Ethernet link aggregation (LAG) group.	Devi discloses at least some of the backplane traces being aggregated into an Ethernet link aggregation (LAG) group. *See supra at* 15[e], 4[f], 3.

No.	’740 Patent Claim 16	Devi
16	The method according to claim 14, wherein selecting the backplane trace comprises applying a hashing function to the at least one of the frame attributes.	Devi discloses the method according to claim 14, wherein selecting the backplane trace comprises applying a hashing function to the at least one of the frame attributes. *See supra* at 14, 9, 8.

No.	’740 Patent Claim 17	Devi
17[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Devi discloses apparatus for connecting a network node with a communication network. For example, Devi discloses a switch 104 used to connect sources and destinations with a network. Devi at Abstract (“A method performed by an ingress card includes receiving a packet with a destination identifier, and determining an aggregated link associated with the destination identifier that includes multiple egress ports. The method further includes determining a destination port from the egress ports of the aggregated link, and communicating the packet to the des-tination port.”) Devi at [0003] (“As telecommunication networks handle more and more traffic, new methods are constantly being developed that allow the networks to process larger flows of informa-tion. Link aggregation is one example. In link aggregation, several physical links are aggregated to appear as one logical link to a telecommunications system. This aggregation allows all of the links to be used actively, rather than having particular links reserved. Thus, link aggregation provides more efficient use of network resources and better load balancing. The tradeoff, however, is that link aggregation may also require substantial additional processing and/or hardware to implement the aggregated links.”) Devi at [0005] (“In accordance with one embodiment of the present invention, a method performed by an ingress card includes receiving a packet comprising a destination identifier.

No.	’740 Patent Claim 17	Devi
		The method also includes determining an aggregated link asso-ciated with the destination identifier, and determining a destination port from the egress ports of the aggregated link. The method further includes communicating the packet to the destination port.”) Devi at [0006] (“In accordance with another embodiment of the present invention, a method performed by an ingress card includes receiving a packet and determining egress links for the packet. The method also includes determining whether each link is an aggregated link with multiple egress ports or a non-aggregated link with a single egress port. For each aggregated link, a destination port is determined from the egress ports, and the packet is communicated to the desti-nation port. For each non-aggregated link, a copy of the packet is communicated to the single egress port.”) Devi at [0007] (“Important technical advantages of certain embodi-ments of the present invention include implementing link aggregation using distributed processing among ingress components. For example, individual ingress cards can determine a destination port for a packet, eliminating the need for a separate processing stage implemented in hard-ware and/or software to separately determine the destination port from a packet after the ingress card forwards the packet to a link. This improves the efficiency and speed of packet forwarding in a switch.”) Devi at [0008] (“Yet another important technical advantage of cer-tain embodiments is a distributed architecture that may be used to process a variety of packet traffic. For example, certain embodiments of an ingress card can forward unicast, multicast, and bridging traffic. Particular embodiments may process packets from different protocols as well.”) Devi at [0009] (“Other important technical advantages of certain embodiments of the present invention include load balanc-ing. The tables or other information used by the ingress card to determine a destination port for packets may also include additional usage information that allows the ingress card to determine a destination port from the egress ports in an aggregated link. This means that rather than using one link to the exclusion of others, loads may be distributed among several links in an aggregated link in a more balanced fashion. Particular

No.	’740 Patent Claim 17	Devi
		embodiments of the present invention may have some, all or none of the enumerated technical advantages. Still other important technical advantages will be apparent to one skilled in the art from the following figures, description, and claims.”) Devi at [0015] (“FIG. 1 shows a system 100 that includes a switch 104 coupled to a network 102. Using network 102 and switch 104, packets are communicated from sources 103 to destinations 112. Components of switch 104 are referred to as "ingress" components when receiving packets from sources 103 and "egress" components when sending packets to destinations 112.”) Devi at [0016] (“Network 102 represents any suitable structure for communicating packets, cells, frames, segments, or other portions of data (generally referred to as "packets"). Net-work 102 may represent the Internet, extranet, local area network (LAN), synchronous optical network (SONET), wide area network (WAN), the public switched telephone network (PSTN), or any other suitable network for commu-nicating information. Network 102 may include routers, switches, hubs, endpoints, or any other network device that communicates information. Network 102 contemplates any number or arrangement of components that exchange infor-mation.”) Devi at [0017] (“Sources 103 represent any source of information in packet form. Sources 103 need not be the original device that generated the packet, but need only convey a packet to network 102. Sources 103 may be any hardware and/or software configured to communicate information, including endpoints, switches, routers, hubs, or any other suitable network device.”) Devi at [0018] (“Switch 104 sends and receives packets. Switch 104 may represent any suitable device, including an Ethernet switch, router, hub, or any other suitable hardware and/or software configured to receive packets and communicate them to other devices. Switch 104 includes cards 106 coupled to physical links 108 and a backplane 114 that represents hardware and/or software allowing cards 106 in switch 104 to exchange information with one another.”)

No.	’740 Patent Claim 17	Devi
		Devi at [0019] (“Cards 106 represent separate components of hard-ware and/or software in switch 104 that exchange packets with network 102. Cards 106 may include traditional inter-face cards, as well as any other component, module, or part of switch 104 capable of independently receiving packets and communicating those packets to other components of switch 104. Each card 106 has one or more ports 110 coupled to physical links 108. Cards 106 may exchange information and packets with one another using backplane 114. Each card 106 includes sufficient processing capability to identify a port 110 in another card 106 in switch 104 and to communicate a packet to that port 110.”) Devi at [0020] (“Physical links 108 represent physical interfaces between switch 104 and other devices. Links 108 may include fiber optic connections, cables, wireless links, or any other suitable method for communicating information between switch 104 and other devices. Links 108 couple to cards 106 of switch 104 using ports 110. Port 110 represents any suitable physical interface between card 106 and physi-cal link 108 allowing information to be received from link 108 and communicated to link 108. Each physical link 108 is associated with one physical connection in the form of port 110.”) Devi at [0021] (“A card 106, port 110 or link 108 used to receive a packet from a source 103 is referred to as an "ingress" card 106, port 110, or link 108. A card 106, port 110 or link 108 used to communicate a packet to a destination 112 is referred to as an "egress" card 106, port 110, or link 108. Cards 106, ports 110, and links 108 may be bidirectional, so that a particular component may be either an ingress or egress component depending on whether the component sends or receives a packet at a given time.”)
17[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the	Devi discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. For example, Devi discloses cards 106, i.e., interface modules that are capable of processing packets with destination identifiers 215 that are exchanged between sources/destinations and a network.

No.	’740 Patent Claim 17	Devi
	communication network,	Devi at Abstract (“A method performed by an ingress card includes receiving a packet with a destination identifier, and determining an aggregated link associated with the destination identifier that includes multiple egress ports. The method further includes determining a destination port from the egress ports of the aggregated link, and communicating the packet to the des-tination port.”) Devi at [0003] (“As telecommunication networks handle more and more traffic, new methods are constantly being developed that allow the networks to process larger flows of informa-tion. Link aggregation is one example. In link aggregation, several physical links are aggregated to appear as one logical link to a telecommunications system. This aggregation allows all of the links to be used actively, rather than having particular links reserved. Thus, link aggregation provides more efficient use of network resources and better load balancing. The tradeoff, however, is that link aggregation may also require substantial additional processing and/or hardware to implement the aggregated links.”) Devi at [0005] (“In accordance with one embodiment of the present invention, a method performed by an ingress card includes receiving a packet comprising a destination identifier. The method also includes determining an aggregated link asso-ciated with the destination identifier, and determining a destination port from the egress ports of the aggregated link. The method further includes communicating the packet to the destination port.”) Devi at [0006] (“In accordance with another embodiment of the present invention, a method performed by an ingress card includes receiving a packet and determining egress links for the packet. The method also includes determining whether each link is an aggregated link with multiple egress ports or a non-aggregated link with a single egress port. For each aggregated link, a destination port is determined from the egress ports, and the packet is communicated to the desti-nation port. For each non-aggregated link, a copy of the packet is communicated to the single egress port.”) Devi at [0007] (“Important technical advantages of certain embodi-ments of the present invention include implementing link aggregation using distributed processing among ingress

No.	’740 Patent Claim 17	Devi
		components. For example, individual ingress cards can determine a destination port for a packet, eliminating the need for a separate processing stage implemented in hard-ware and/or software to separately determine the destination port from a packet after the ingress card forwards the packet to a link. This improves the efficiency and speed of packet forwarding in a switch.”) Devi at [0008] (“Yet another important technical advantage of cer-tain embodiments is a distributed architecture that may be used to process a variety of packet traffic. For example, certain embodiments of an ingress card can forward unicast, multicast, and bridging traffic. Particular embodiments may process packets from different protocols as well.”) Devi at [0009] (“Other important technical advantages of certain embodiments of the present invention include load balanc-ing. The tables or other information used by the ingress card to determine a destination port for packets may also include additional usage information that allows the ingress card to determine a destination port from the egress ports in an aggregated link. This means that rather than using one link to the exclusion of others, loads may be distributed among several links in an aggregated link in a more balanced fashion. Particular embodiments of the present invention may have some, all or none of the enumerated technical advantages. Still other important technical advantages will be apparent to one skilled in the art from the following figures, description, and claims.”) Devi at [0015] (“FIG. 1 shows a system 100 that includes a switch 104 coupled to a network 102. Using network 102 and switch 104, packets are communicated from sources 103 to destinations 112. Components of switch 104 are referred to as "ingress" components when receiving packets from sources 103 and "egress" components when sending packets to destinations 112.”) Devi at [0016] (“Network 102 represents any suitable structure for communicating packets, cells, frames, segments, or other portions of data (generally referred to as "packets"). Net-work 102 may represent the Internet, extranet, local area network (LAN), synchronous optical network (SONET), wide area network (WAN), the public switched telephone network (PSTN), or any other suitable network for commu-nicating information. Network 102 may

No.	’740 Patent Claim 17	Devi
		include routers, switches, hubs, endpoints, or any other network device that communicates information. Network 102 contemplates any number or arrangement of components that exchange infor-mation.”) Devi at [0017] (“Sources 103 represent any source of information in packet form. Sources 103 need not be the original device that generated the packet, but need only convey a packet to network 102. Sources 103 may be any hardware and/or software configured to communicate information, including endpoints, switches, routers, hubs, or any other suitable network device.”) Devi at [0018] (“Switch 104 sends and receives packets. Switch 104 may represent any suitable device, including an Ethernet switch, router, hub, or any other suitable hardware and/or software configured to receive packets and communicate them to other devices. Switch 104 includes cards 106 coupled to physical links 108 and a backplane 114 that represents hardware and/or software allowing cards 106 in switch 104 to exchange information with one another.”) Devi at [0019] (“Cards 106 represent separate components of hard-ware and/or software in switch 104 that exchange packets with network 102. Cards 106 may include traditional inter-face cards, as well as any other component, module, or part of switch 104 capable of independently receiving packets and communicating those packets to other components of switch 104. Each card 106 has one or more ports 110 coupled to physical links 108. Cards 106 may exchange information and packets with one another using backplane 114. Each card 106 includes sufficient processing capability to identify a port 110 in another card 106 in switch 104 and to communicate a packet to that port 110.”) Devi at [0020] (“Physical links 108 represent physical interfaces between switch 104 and other devices. Links 108 may include fiber optic connections, cables, wireless links, or any other suitable method for communicating information between switch 104 and other devices. Links 108 couple to cards 106 of switch 104 using ports 110. Port 110 represents any suitable physical interface between card 106 and physi-cal link 108 allowing information to be

No.	’740 Patent Claim 17	Devi
		received from link 108 and communicated to link 108. Each physical link 108 is associated with one physical connection in the form of port 110.”) Devi at [0021] (“A card 106, port 110 or link 108 used to receive a packet from a source 103 is referred to as an "ingress" card 106, port 110, or link 108. A card 106, port 110 or link 108 used to communicate a packet to a destination 112 is referred to as an "egress" card 106, port 110, or link 108. Cards 106, ports 110, and links 108 may be bidirectional, so that a particular component may be either an ingress or egress component depending on whether the component sends or receives a packet at a given time.”)
17[b]	at least one of said interface modules being operative to communicate in both an upstream direction and a downstream direction;	Devi discloses at least one of said interface modules being operative to communicate in both an upstream direction and a downstream direction. For example, Devi discloses both ingress and egress components that are capable of both receiving and sending packets. Devi further discloses cards 106, ports 110, and links 109, “may be bidirectional, so that a particular component may be either an ingress or egress component depending on whether the component sends or receives a packet at a given time.” Devi at [0015] (“FIG. 1 shows a system 100 that includes a switch 104 coupled to a network 102. Using network 102 and switch 104, packets are communicated from sources 103 to destinations 112. Components of switch 104 are referred to as "ingress" components when receiving packets from sources 103 and "egress" components when sending packets to destinations 112.”) Devi at [0020] (“Physical links 108 represent physical interfaces between switch 104 and other devices. Links 108 may include fiber optic connections, cables, wireless links, or any other suitable method for communicating information between switch 104 and other devices. Links 108 couple to cards 106 of switch 104 using ports 110. Port 110 represents any suitable physical interface between card 106 and physi-cal link 108 allowing information to be received from link 108 and communicated to link 108. Each physical link 108 is associated with one physical connection in the form of port 110.”)

No.	’740 Patent Claim 17	Devi
		Devi at [0021] (“A card 106, port 110 or link 108 used to receive a packet from a source 103 is referred to as an "ingress" card 106, port 110, or link 108. A card 106, port 110 or link 108 used to communicate a packet to a destination 112 is referred to as an "egress" card 106, port 110, or link 108. Cards 106, ports 110, and links 108 may be bidirectional, so that a particular component may be either an ingress or egress component depending on whether the component sends or receives a packet at a given time.”) Devi at [0024] (“FIG. 2 shows a card 106 in more detail. Card 106 couples to backplane 114 of switch 104 using ingress backplane interface (IBI) 210 and egress backplane interface (EBI) 212. Card 106 couples to network 102 through physi-cal layer devices 202, including ports 110 coupled to links 108. Card 106 includes an ingress processor 204, an egress processor 214, a traffic manager 206, and a memory 208. Card 106 may exchange packets from network 102 or backplane 114. Packets received from network 102 are considered to be on the "ingress" side, while packets received from the backplane 114 are considered to be on the "egress side" of card 106. Thus, at the level of card 106, the terms "ingress" and "egress" refer to the function of those components with respect to receiving packets from network 102 or backplane 114.”) Devi at [0027] (“IBI 210 and EBI 212 refer to any port or connec-tion, real or virtual, between card 106 and backplane 114 of switch 104. IBI 210 and EBI 212 may represent separate components, or alternatively, may represent the same hard-ware and/or software used to send and receive packets from backplane 114. Functionally, IBI 210 sends packets to backplane 114, while EBI 212 receives packets from back-plane 114.”)
17[c]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Devi discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 1[a].

No.	’740 Patent Claim 17	Devi
17[d]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Devi discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 1[c].
17[e]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame;	Devi discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 1[f].
17[f]	at least one of said first physical links and at least one of said second links being bi-directional links operative to communicate in both said upstream	Devi discloses at least one of said first physical links and at least one of said second links being bi-directional links operative to communicate in both said upstream direction and said downstream direction. *See supra at* 1[b], 1[d].

No.	’740 Patent Claim 17	Devi
	direction and said downstream direction.	

No.	’740 Patent Claim 18	Devi
18[a]	The apparatus according to claim 17, and comprising a backplane to which the one or more interface modules are coupled,	Devi discloses the apparatus according to claim 17, and comprising a backplane to which the one or more interface modules are coupled. *See supra at* 3, 17.
18[b]	wherein the second physical links comprise backplane traces formed on the backplane.	Devi discloses wherein the second physical links comprise backplane traces formed on the backplane. *See supra at* 3, 17.

No.	’740 Patent Claim 19	Devi
19[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Devi discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
19[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the	Devi discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].

No.	’740 Patent Claim 19	Devi
	communication network;	
19[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Devi discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].
19[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Devi discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].
19[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second	Devi discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].

No.	’740 Patent Claim 19	Devi
	group over which to send the data frame,	
19[e]	at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group.	Devi discloses at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group. *See supra at* 4[f].

No.	’740 Patent Claim 20	Devi
20[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Devi discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
20[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	Devi discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].
20[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Devi discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].

No.	’740 Patent Claim 20	Devi
20[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Devi discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].
20[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,	Devi discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].
20[e]	two or more of the first physical links being aggregated into an external Ethernet link aggregation (LAG) group so as to increase a data	Devi discloses two or more of the first physical links being aggregated into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node. *See supra at* 19[e], 5[f].

No.	’740 Patent Claim 20	Devi
	bandwidth provided to the network node.	

No.	’740 Patent Claim 21	Devi
21	The apparatus according to claim 17, and comprising a multiplexer, which is arranged to perform at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node.	Devi discloses the apparatus according to claim 17, and comprising a multiplexer, which is arranged to perform at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node. *See supra at* 6, 17.

No.	’740 Patent Claim 22	Devi
22	The apparatus according to claim 17, wherein the control module is arranged to balance a frame data rate among at least some of the first and second physical links.	Devi discloses the apparatus according to claim 17, wherein the control module is arranged to balance a frame data rate among at least some of the first and second physical links. *See supra at* 7, 17.

No.	’740 Patent Claim 23	Devi
23	The apparatus according to claim 17, wherein the control module is arranged to apply a mapping function to the at least one of the frame attributes so as to select the first and second physical links.	Devi discloses the apparatus according to claim 17, wherein the control module is arranged to apply a mapping function to the at least one of the frame attributes so as to select the first and second physical links. *See supra at* 8, 17.

No.	’740 Patent Claim 24	Devi
24	The apparatus according to claim 23, wherein the mapping function comprises a hashing function.	Devi discloses the apparatus according to claim 23, wherein the mapping function comprises a hashing function. *See supra at* 9, 23.

No.	’740 Patent Claim 25	Devi
25[a]	The apparatus according to claim 24, wherein the control module is arranged to determine a hashing size responsively to a number of at least some of the first and second physical links,	Devi discloses the apparatus according to claim 24, wherein the control module is arranged to determine a hashing size responsively to a number of at least some of the first and second physical links. *See supra at* 10[a], 24.
25[b]	to apply the hashing function to the at least one of the frame	Devi discloses to apply the hashing function to the at least one of the frame attributes to produce a hashing key. *See supra at* 10[b].

No.	’740 Patent Claim 25	Devi
	attributes to produce a hashing key,	
25[c]	to calculate a modulo of a division operation of the hashing key by the hashing size, and	Devi discloses to calculate a modulo of a division operation of the hashing key by the hashing size. *See supra at* 10[c].
25[d]	to select the first and second physical links responsively to the modulo.	Devi discloses to select the first and second physical links responsively to the modulo. *See supra at* 10[d].

No.	’740 Patent Claim 26	Devi
26	The apparatus according to claim 25, wherein the control module is arranged to select the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo.	Devi discloses the apparatus according to claim 25, wherein the control module is arranged to select the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo. *See supra at* 11, 25.

No.	’740 Patent Claim 27	Devi
27	The apparatus according to claim 17, wherein the at least one of the frame attributes comprises at	Devi discloses the apparatus according to claim 17, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port.

No.	’740 Patent Claim 27	Devi
	least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port.	*See supra at* 12, 17.

No.	’740 Patent Claim 28	Devi
28[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Devi discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
28[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	Devi discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].

No.	’740 Patent Claim 28	Devi
28[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Devi discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].
28[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Devi discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].
28[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,	Devi discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].

No.	’740 Patent Claim 28	Devi
28[e]	the communication network being arranged to provide a communication service to the network node,	Devi discloses the communication network being arranged to provide a communication service to the network node. *See supra at* 2[b].
28[f]	the service having specified bandwidth requirements comprising at least one of a committed information rate (CR), a peak information rate (PIR) and an excess information rate (EIR), and	Devi discloses the service having specified bandwidth requirements comprising at least one of a committed information rate (CR), a peak information rate (PIR) and an excess information rate (EIR). *See supra at* 13[i].
28[g]	the first and second groups of physical links being dimensioned to provide an allocated bandwidth for the communication service responsively to the bandwidth requirements.	Devi discloses the first and second groups of physical links being dimensioned to provide an allocated bandwidth for the communication service responsively to the bandwidth requirements. *See supra at* 13[j].

No.	’740 Patent Claim 29	Devi
29[preamble]	Apparatus for connecting user ports to a communication network, comprising:	Devi discloses apparatus for connecting user ports to a communication network. *See supra at* 17[preamble], 14[preamble].

No.	’740 Patent Claim 29	Devi
29[a]	one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network,	Devi discloses one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network. *See supra at* 17[a], 14[a].
29[b]	at least one of said user interface modules being bi-directional and operative to communicate in both an upstream direction and a downstream direction;	Devi discloses at least one of said user interface modules being bi-directional and operative to communicate in both an upstream direction and a downstream direction. *See supra at* 17[b], 14[c].
29[c]	a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network,	Devi discloses a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network. *See supra at* 14[b]-[e].

No.	’740 Patent Claim 29	Devi
29[d]	at least one of said backplane traces being bi-directional and operative to communicate in both said upstream direction and said downstream direction; and	Devi discloses at least one of said backplane traces being bi-directional and operative to communicate in both said upstream direction and said downstream direction. *See supra at* 14[c], 17[b].
29[e]	a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame.	Devi discloses a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame. *See supra at* 14[e], 17[e].

No.	’740 Patent Claim 30	Devi
30[preamble]	Apparatus for connecting user ports to a communication network, comprising:	Devi discloses apparatus for connecting user ports to a communication network. *See supra at* 29[preamble].
30[a]	one or more user interface modules coupled to the user ports, which are arranged to process data frames having	Devi discloses one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network. *See supra at* 29[a].

No.	’740 Patent Claim 30	Devi
	frame attributes sent between the user ports and the communication network;	
30[b]	a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network;	Devi discloses a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network. *See supra at* 29[c].
30[c]	a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame;	Devi discloses a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame. *See supra at* 29[e].
30[d]	at least some of the backplane traces are aggregated into an	Devi discloses at least some of the backplane traces are aggregated into an Ethernet link aggregation (LAG) group.

No.	’740 Patent Claim 30	Devi
	Ethernet link aggregation (LAG) group.	*See supra at* 4[f], 15[f].

No.	’740 Patent Claim 31	Devi
31	The apparatus according to claim 29, wherein the control module is arranged to apply a hashing function to the at least one of the frame attributes so as to select the backplane trace.	Devi discloses the apparatus according to claim 29, wherein the control module is arranged to apply a hashing function to the at least one of the frame attributes so as to select the backplane trace. *See supra at* 16, 29, 30[c].

EXHIBIT C-3
Defendant’s Preliminary Invalidity Contentions
Orckit Corporation v. Cisco Systems, Inc., 2:22-cv-00276-JRG-RSP

Chart for U.S. Patent 7,545,740 (“the ’740 Patent”)
Cisco EtherChannel Implementation In Cisco Products (“Cisco EtherChannel System”)

Cisco products implementing EtherChannel with other standard Cisco technology, such as Cisco Express Forwarding, (“Cisco EtherChannel System”), which was conceived and reduced to practice no later than April 7, 2006, constitute prior art to the ’740 Patent. Such Cisco EtherChannel System includes, without limitation, the Cisco Catalyst 6500 series, Cisco Catalyst 6000 series, Cisco Catalyst 3500 series, Cisco Catalyst 2900 series, Cisco Catalyst 4500 series, Cisco Catalyst 4000 series, Cisco Catalyst 5000 series, and Cisco 7500 router series. As shown in the chart below, all Asserted Claims of the ’740 Patent are invalid under (1) 35 U.S.C. §§ 102 (a) and (g) because Cisco EtherChannel System meets each element of those claims, (2) 35 U.S.C. §§ (a) and (b) because the references describing the Cisco EtherChannel System disclose every limitation of every Asserted Claim, and/or (3) 35 U.S.C. § 103 because Cisco EtherChannel System renders those claims obvious either alone, or in combination with the knowledge of a person having ordinary skill in the art, and in further combination with the references specifically identified below and in the following claim chart and/or one or more references identified in Defendant’s Preliminary Invalidity Contentions. The following quotations and diagrams come from documentation describing Cisco EtherChannel System and its functionalities that were published prior to April 7, 2006.

Cisco EtherChannel System is described at least in the following documents and other materials published before April 7, 2006 and cited in this chart:

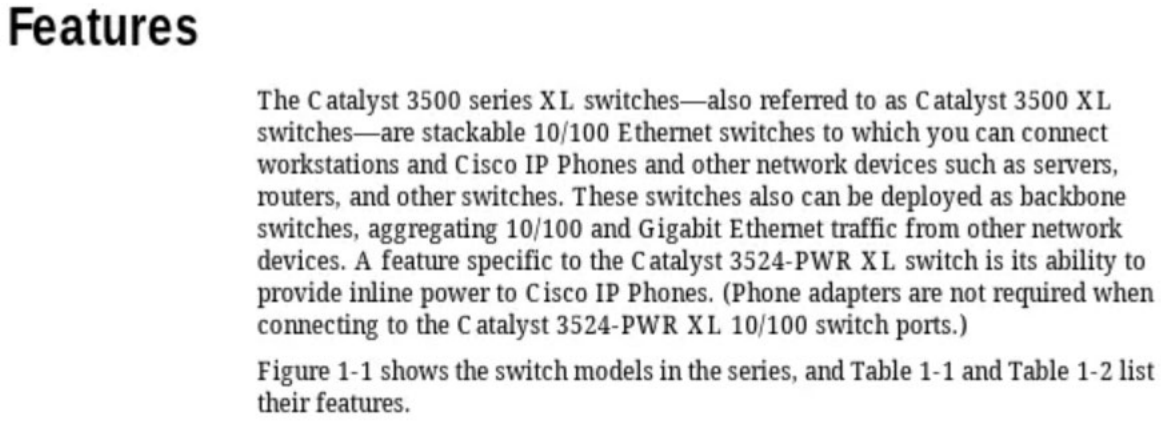
- https://www.mtmnet.com/PDF_FILES/WS-C6513_DataSheet.pdf (“Cisco Catalyst 6500 Data Sheet”)
- https://www.cisco.com/c/en/us/support/docs/switches/catalyst-6500-series-switches/29423-162.html (“Catalyst 6500/6000 Module”)
- https://www.cisco.com/c/en/us/support/docs/lan-switching/etherchannel/22422-144.html (“Layer 2 EtherChannel”)
- https://www.cisco.com/c/en/us/support/docs/lan-switching/etherchannel/21066-135.html (“Configuring EtherChannel”)
- https://www.cisco.com/c/en/us/support/docs/ip/express-forwarding-cef/18285-loadbal-cef.html#hardware (“Load Balancing”)
- https://www.cisco.com/c/en/us/support/docs/lan-switching/etherchannel/19642-126.html (“Configuring LACP”)

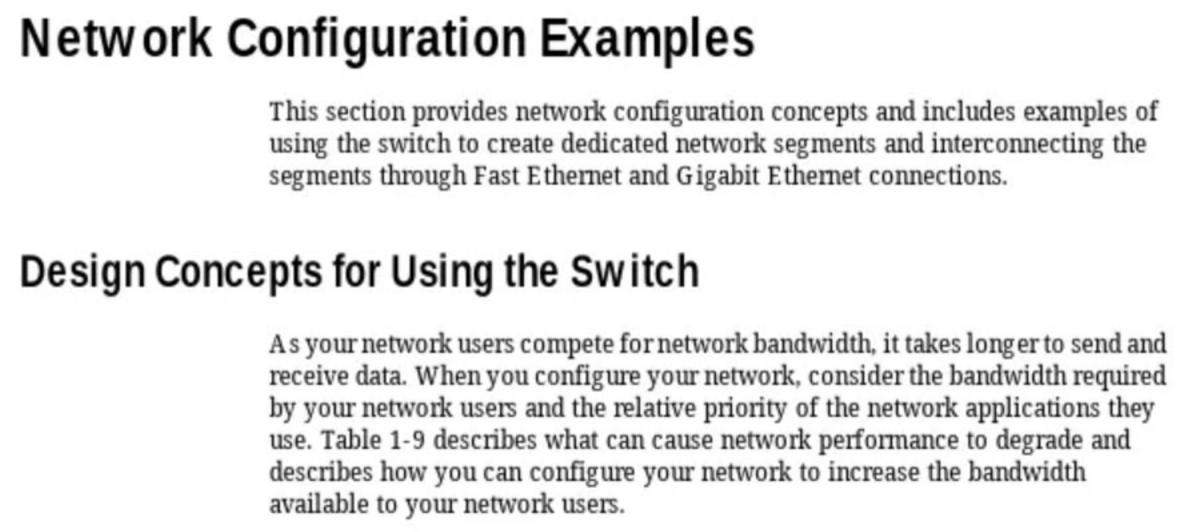
- https://safe.rbi-umbrella.com/doc/docview/viewer/docNAB51938228E821b6319ed81369856bf59b4b1697ce730a97d44367a672a587563f0f068d0915 ("Catalyst 3500 Installation Guide")
- https://teamkci.com/wp-content/uploads/WS-C3550.pdf ("Catalyst 3500 End of Sale")
- https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3560/software/release/12-2_25_sec/configuration/guide/3560scg.pdf ("Catalyst 3560 Configuration Guide")
- https://newsroom.cisco.com/c/r/newsroom/en/us/a/y1997/m03/cisco-s-fast-etherchannel-provides-smooth-migration-to-gigabit-ethernet.html ("Cisco's Fast EtherChannel")
- https://www.cisco.com/c/en/us/support/docs/switches/catalyst-2950-series-switches/24042-158.html ("Configuring EtherChannel and 802.1Q Trunking")

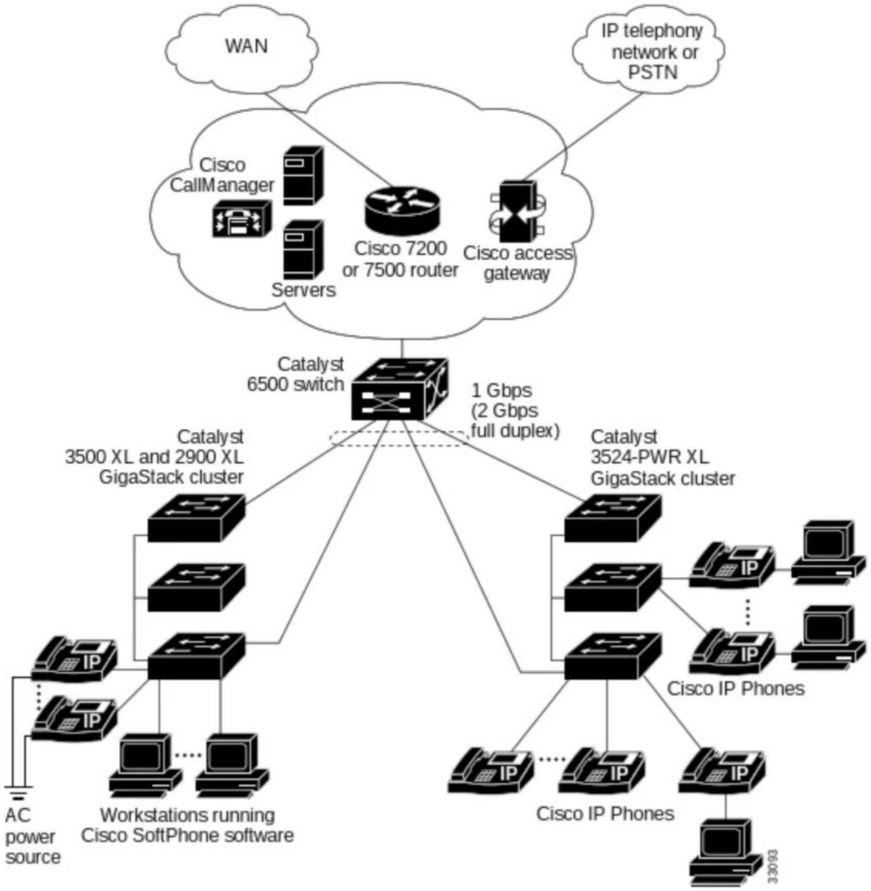
Motivations to combine the disclosures in Cisco EtherChannel System publications with disclosures in other publications known in the art, as explained in this chart, include at least the similarity in subject matter between the to the extent they concern methods of data communication systems, and specifically to methods and systems for link aggregation in a data communication network. Insofar as the references cite other patents or publications, or suggest additional changes, one of ordinary skill in the art would look beyond a single reference to other references in the field.

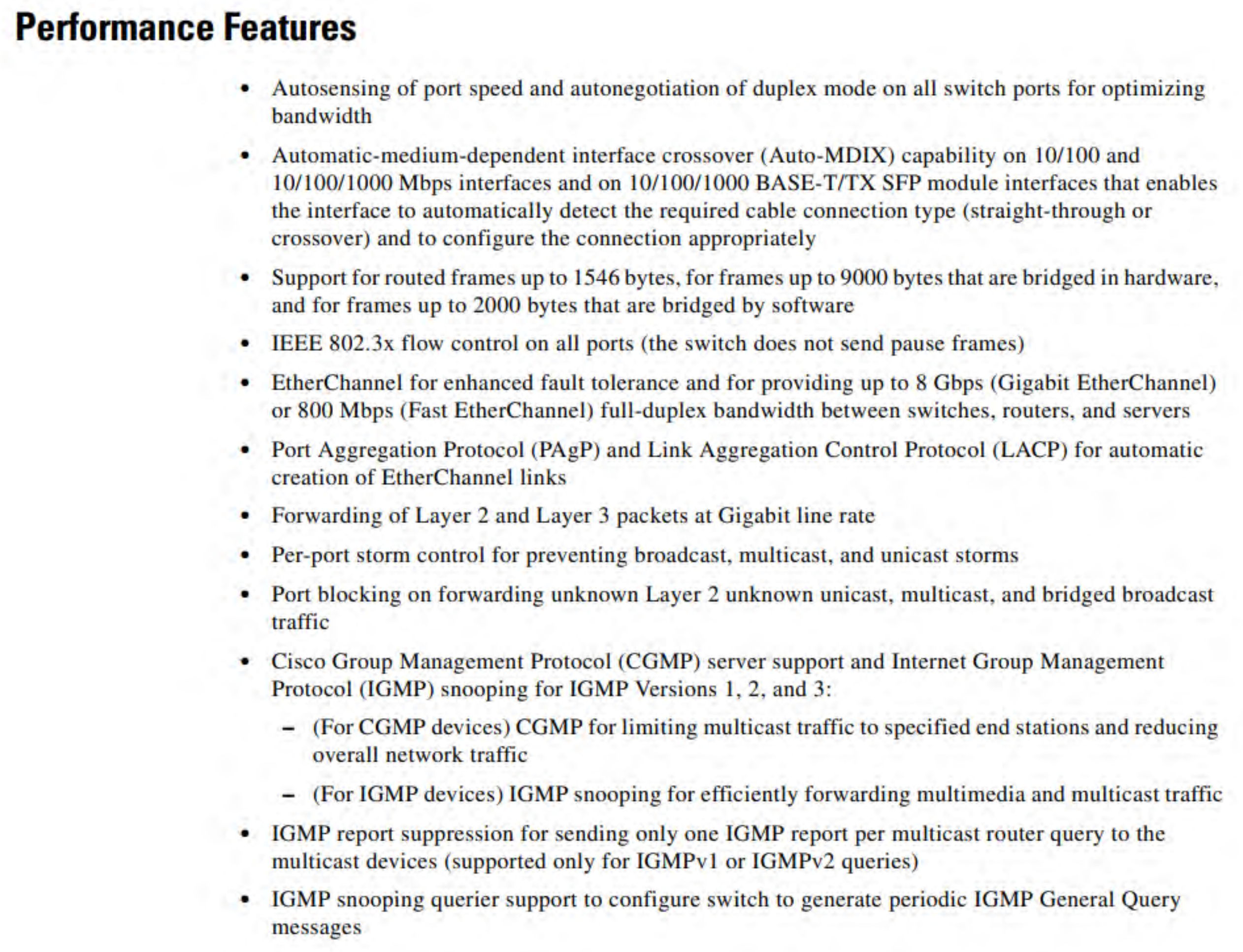
These invalidity contentions are based on Defendant's present understanding of the asserted claims, and Orckit's apparent construction of the claims in its November 3, 2022 Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1, and Orckit's January 19, 2023 First Amended Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1 (Orckit's "Infringement Disclosures"), which is deficient at least insofar as it fails to cite any documents or identify accused structures, acts, or materials in the Accused Products with particularity. Defendant does not agree with Orckit's application of the claims, or that the claims satisfy the requirements of 35 U.S.C. § 112. Defendant's contentions herein are not, and should in no way be seen as, admissions or adoptions as to any particular claim scope or construction, or as any admission that any particular element is met by any accused product in any particular way. Defendant objects to any attempt to imply claim construction from this chart. Defendant's prior art invalidity contentions are made in a variety of alternatives and do not represent Defendant's agreement or view as to the meaning, definiteness, written description support for, or enablement of any claim contained therein.

The following contentions are subject to revision and amendment pursuant to Federal Rule of Civil Procedure 26(e), the Local Rules, and the Orders of record in this matter subject to further investigation and discovery regarding the prior art and the Court's construction of the claims at issue.

No.	’740 Patent Claim 1	Cisco EtherChannel
1[preamble]	A method for communication, comprising:	Cisco EtherChannel System discloses a method for communication. For example, Cisco EtherChannel System discloses a communication method to deliver data using aggregation and load balancing techniques. Cisco Catalyst 6500 Data Cisco Catalyst 6500 Data Sheet at 1 (“The Cisco® Catalyst® 6500 and 6500-E Series sets the new standard for IP Communications and application delivery in enterprise campus and service provider networks by maximizing user productivity and enhancing operational control. As the premier intelligent, multilayer modular Cisco switch, the Catalyst 6500 Series delivers secure, converged, end-to-end services, from the wiring closet to the core network, the data center, and the WAN edge. The 6500-E series switches offer enhancements to scale beyond the 4000W power supply. Today’s 6500 series switch cannot scale beyond 4000W of total system capacity except for 6513. The 6513 can support up to 6000W of power.”) Catalyst 3500 Installation Guide at 1-1  Features The Catalyst 3500 series XL switches—also referred to as Catalyst 3500 XL switches—are stackable 10/100 Ethernet switches to which you can connect workstations and Cisco IP Phones and other network devices such as servers, routers, and other switches. These switches also can be deployed as backbone switches, aggregating 10/100 and Gigabit Ethernet traffic from other network devices. A feature specific to the Catalyst 3524-PWR XL switch is its ability to provide inline power to Cisco IP Phones. (Phone adapters are not required when connecting to the Catalyst 3524-PWR XL 10/100 switch ports.) Figure 1-1 shows the switch models in the series, and Table 1-1 and Table 1-2 list their features. Catalyst 3500 Installation Guide at 1-28

No.	’740 Patent Claim 1	Cisco EtherChannel
		 **Network Configuration Examples** This section provides network configuration concepts and includes examples of using the switch to create dedicated network segments and interconnecting the segments through Fast Ethernet and Gigabit Ethernet connections. **Design Concepts for Using the Switch** As your network users compete for network bandwidth, it takes longer to send and receive data. When you configure your network, consider the bandwidth required by your network users and the relative priority of the network applications they use. Table 1-9 describes what can cause network performance to degrade and describes how you can configure your network to increase the bandwidth available to your network users. Catalyst 3500 Installation Guide at Figure 1-24

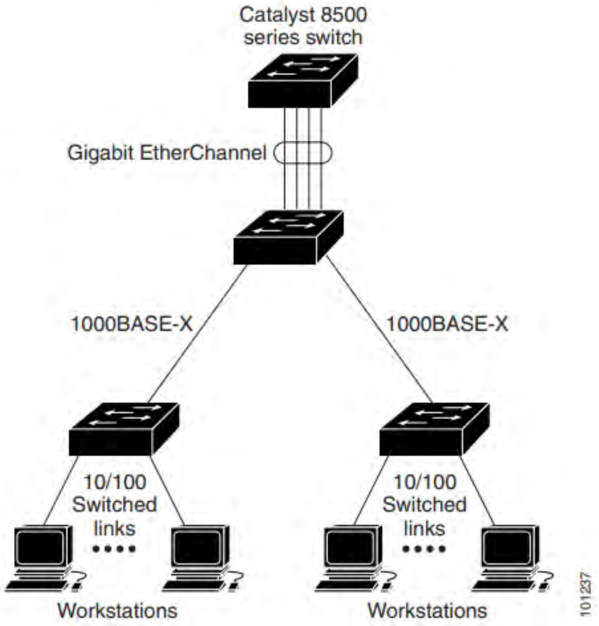
No.	’740 Patent Claim 1	Cisco EtherChannel
		Figure 1-24 Large Campus Configuration  WAN IP telephony network or PSTN Cisco CallManager Servers Cisco 7200 or 7500 router Cisco access gateway Catalyst 6500 switch 1 Gbps (2 Gbps full duplex) Catalyst 3500 XL and 2900 XL GigaStack cluster Catalyst 3524-PWR XL GigaStack cluster IP Cisco IP Phones AC power source Workstations running Cisco SoftPhone software Cisco IP Phones 33093 Catalyst 3560 Configuration Guide at 1-3

No.	’740 Patent Claim 1	Cisco EtherChannel
		 **Performance Features** • Autosensing of port speed and autonegotiation of duplex mode on all switch ports for optimizing bandwidth • Automatic-medium-dependent interface crossover (Auto-MDIX) capability on 10/100 and 10/100/1000 Mbps interfaces and on 10/100/1000 BASE-T/TX SFP module interfaces that enables the interface to automatically detect the required cable connection type (straight-through or crossover) and to configure the connection appropriately • Support for routed frames up to 1546 bytes, for frames up to 9000 bytes that are bridged in hardware, and for frames up to 2000 bytes that are bridged by software • IEEE 802.3x flow control on all ports (the switch does not send pause frames) • EtherChannel for enhanced fault tolerance and for providing up to 8 Gbps (Gigabit EtherChannel) or 800 Mbps (Fast EtherChannel) full-duplex bandwidth between switches, routers, and servers • Port Aggregation Protocol (PAgP) and Link Aggregation Control Protocol (LACP) for automatic creation of EtherChannel links • Forwarding of Layer 2 and Layer 3 packets at Gigabit line rate • Per-port storm control for preventing broadcast, multicast, and unicast storms • Port blocking on forwarding unknown Layer 2 unknown unicast, multicast, and bridged broadcast traffic • Cisco Group Management Protocol (CGMP) server support and Internet Group Management Protocol (IGMP) snooping for IGMP Versions 1, 2, and 3: – (For CGMP devices) CGMP for limiting multicast traffic to specified end stations and reducing overall network traffic – (For IGMP devices) IGMP snooping for efficiently forwarding multimedia and multicast traffic • IGMP report suppression for sending only one IGMP report per multicast router query to the multicast devices (supported only for IGMPv1 or IGMPv2 queries) • IGMP snooping querier support to configure switch to generate periodic IGMP General Query messages Catalyst 3560 Configuration Guide at 33-2

EtherChannel Overview

An EtherChannel consists of individual Fast Ethernet or Gigabit Ethernet links bundled into a single logical link as shown in Figure 33-1.

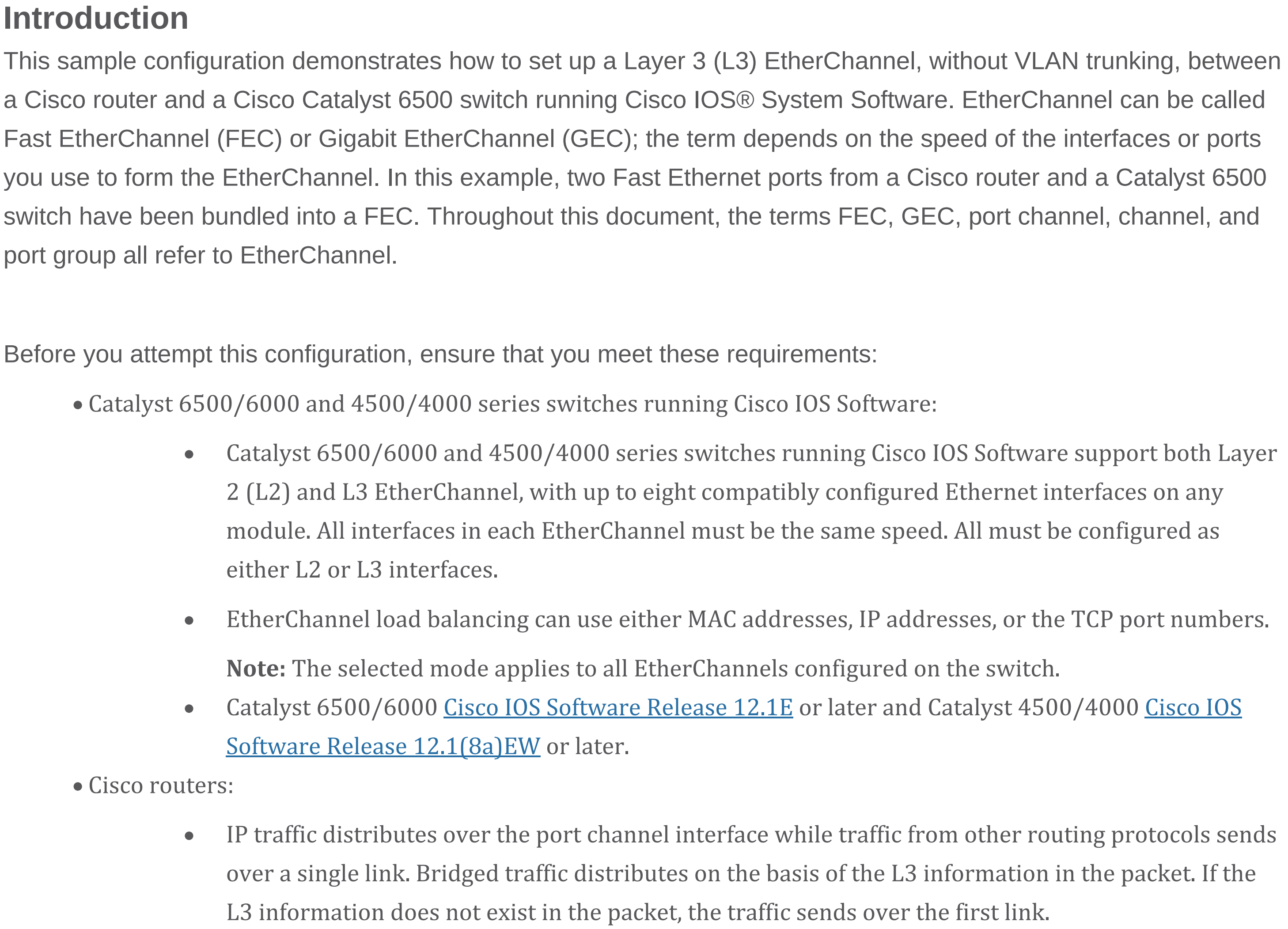
Figure 33-1 Typical EtherChannel Configuration

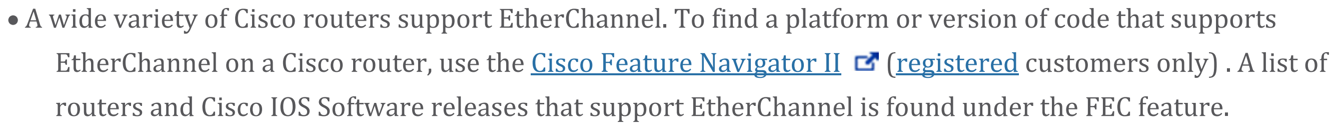


The EtherChannel provides full-duplex bandwidth up to 800 Mbps (Fast EtherChannel) or 8 Gbps (Gigabit EtherChannel) between your switch and another switch or host.

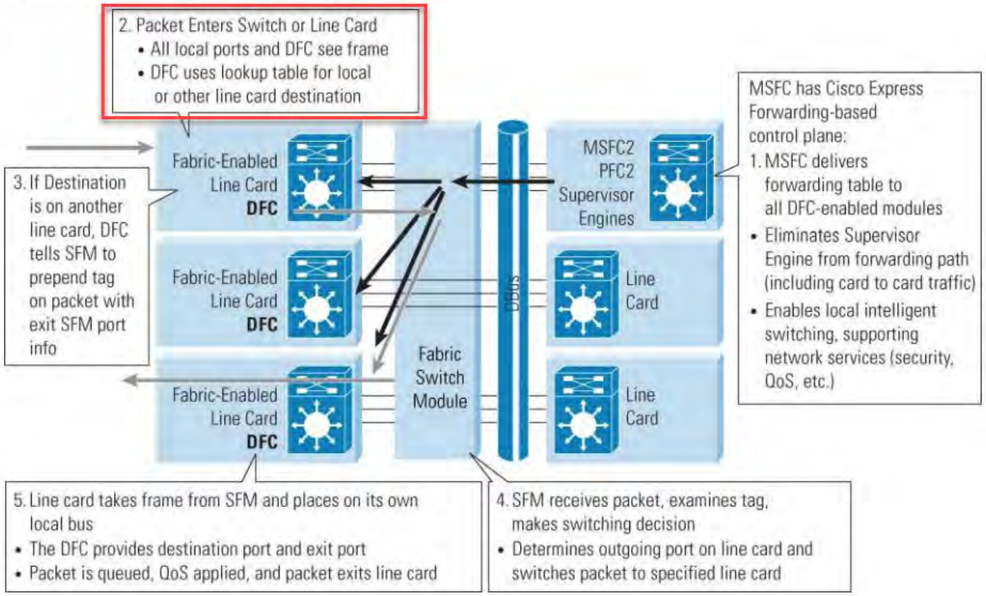
Each EtherChannel can consist of up to eight compatibly configured Ethernet ports. All ports in each EtherChannel must be configured as either Layer 2 or Layer 3 ports. The number of EtherChannels is limited to 48. For more information, see the "EtherChannel Configuration Guidelines" section on page 33-9. The EtherChannel Layer 3 ports are made up of routed ports. Routed ports are physical ports configured to be in Layer 3 mode by using the **no switchport** interface configuration command. For more information, see the Chapter 10, "Configuring Interface Characteristics."

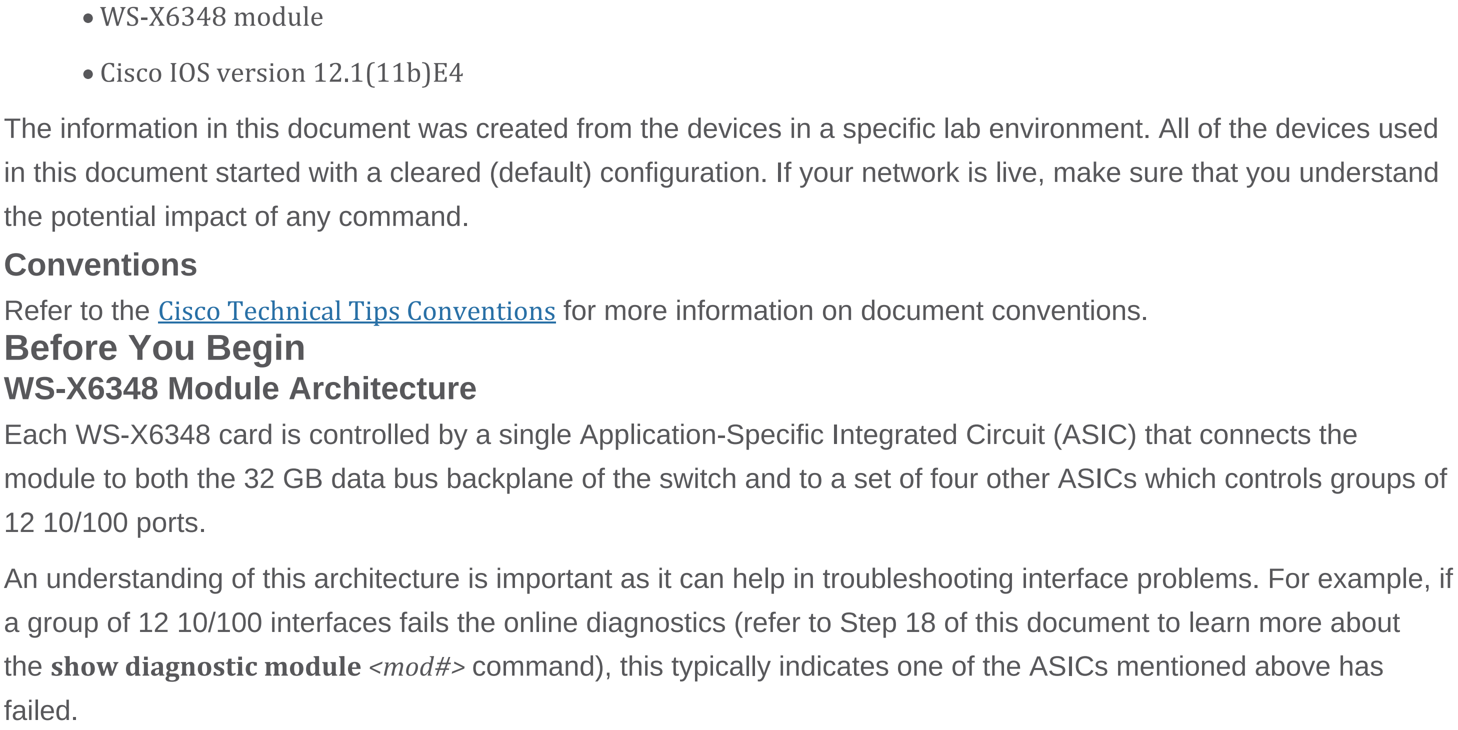
If a link within an EtherChannel fails, traffic previously carried over that failed link changes to the remaining links within the EtherChannel. A trap is sent for a failure, identifying the switch, the EtherChannel, and the failed link. Inbound broadcast and multicast packets on one link in an EtherChannel are blocked from returning on any other link of the EtherChannel.

No.	’740 Patent Claim 1	Cisco EtherChannel
		Configuring EtherChannel at 1 -2  **Introduction** This sample configuration demonstrates how to set up a Layer 3 (L3) EtherChannel, without VLAN trunking, between a Cisco router and a Cisco Catalyst 6500 switch running Cisco IOS® System Software. EtherChannel can be called Fast EtherChannel (FEC) or Gigabit EtherChannel (GEC); the term depends on the speed of the interfaces or ports you use to form the EtherChannel. In this example, two Fast Ethernet ports from a Cisco router and a Catalyst 6500 switch have been bundled into a FEC. Throughout this document, the terms FEC, GEC, port channel, channel, and port group all refer to EtherChannel. Before you attempt this configuration, ensure that you meet these requirements: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software support both Layer 2 (L2) and L3 EtherChannel, with up to eight compatibly configured Ethernet interfaces on any module. All interfaces in each EtherChannel must be the same speed. All must be configured as either L2 or L3 interfaces. • EtherChannel load balancing can use either MAC addresses, IP addresses, or the TCP port numbers. **Note:** The selected mode applies to all EtherChannels configured on the switch. • Catalyst 6500/6000 Cisco IOS Software Release 12.1E or later and Catalyst 4500/4000 Cisco IOS Software Release 12.1(8a)EW or later. • Cisco routers: • IP traffic distributes over the port channel interface while traffic from other routing protocols sends over a single link. Bridged traffic distributes on the basis of the L3 information in the packet. If the L3 information does not exist in the packet, the traffic sends over the first link.

No.	’740 Patent Claim 1	Cisco EtherChannel
		 • A wide variety of Cisco routers support EtherChannel. To find a platform or version of code that supports EtherChannel on a Cisco router, use the Cisco Feature Navigator II (registered customers only) . A list of routers and Cisco IOS Software releases that support EtherChannel is found under the FEC feature.
1[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel,	Cisco EtherChannel System discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. For example, Cisco EtherChannel System discloses connecting a phone, router, switch or server to interface modules or line cards using multiple parallel ports. Cisco Catalyst 6500 Data Sheet at 3 (“Highest Level of Interface Flexibility, Scalability, and Density • Provides the port densities and interface choices that large, mission-critical wiring closets, core enterprise networks, and distribution networks require • Supports up to 576 10/100/1000 gigabit-over-copper ports or 1152 10/100 Ethernet ports • Features the industry’s first 96-port 10/100 RJ-45 module, with optional, field-upgradable support for 802.3af PoE • Provides up to 192 Gigabit Ethernet ports • Features the industry’s first 10 Gigabit Ethernet, Channelized OC-48 dense OC-3 packet over SONET (POS) • Provides investment protection by using Cisco 7xxx Series port adapters on the Cisco Catalyst 6500 Series FlexWAN Line Card, supporting T1/E1 through OC-48 WAN interfaces • Chassis sizes range from 3-slot (Cisco Catalyst 6503 Switch) to 13-slot (Cisco Catalyst 6513 Switch)”) Cisco Catalyst 6500 Data Sheet at 11 (“Cisco Catalyst 6500 Series Ethernet interface modules, designed for wiring closet, distribution and core network, and data center applications, as well as service provider and Metro Ethernet environments, use one of the following types of Ethernet interfaces. • 10/100 Mbps over Copper—For wiring closets providing 10/100-Mbps performance with autonegotiation and support for IEEE 802.3af PoE (inline power); up to 96 ports per module; includes Classic and CEF256 interface modules.

No.	’740 Patent Claim 1	Cisco EtherChannel
		• 10/100/1000 Mbps Gigabit over Copper—For wiring closets and data centers providing 10/100/1000-Mbps performance with autonegotiation and support for IEEE 802.3af PoE (inline power); up to 48 ports per module; includes Classic, CEF256, and CEF720 interface modules. • 100 Mbps over Fiber—For secure wiring closets and long-haul router and switch interconnects; up to 24 ports per module; includes Classic and CEF256 interface modules. • 1 Gbps—For distribution and core layers and for data centers providing 1-Gbps performance; up to 48 ports per module; includes Classic, CEF256, dCEF256, and CEF720 interface modules. • 10 Gbps—For distribution and core layers providing 10-Gbps performance in 2-port or 4-port modules; includes CEF256 and dCEF720 interface modules.”) Cisco Catalyst 6500 Data Sheet at 11 (“Optical Services Module (OSM)—A dedicated line card that provides several interfaces, including OC-3/STM-1, OC-12/STM-4, OC-48/STM-16, Channelized T3, Channelized OC-12/STM-4 POS, Gigabit Ethernet, OC-12/STM-4 ATM, and OC-48/STM-16 Dynamic Packet Transport (DPT)”) Cisco Catalyst 6500 Data Sheet at 11 (“Optical services modules (OSMs) are line cards that provide high-speed WAN connectivity with onboard network processors for distributed-line-rate IP services applications. For more information about OSMs, see the following data sheets.”) Cisco Catalyst 6500 Data Sheet at 14 (“Using the same ASIC engine design as the central PFCx, distributed forwarding cards (DFCs) located on the interface modules forward packets between two ports, directly or across the switch fabric, without involving the supervisor engine. With the DFC, each interface module has a dedicated forwarding engine complete with the full forwarding tables. Distributed Cisco Express Forwarding (Figure 6) works like this: • As in standard Cisco Express Forwarding, the central PFCx located on the supervisor engine and the DFC engines located on the interface modules are loaded with the same Cisco Express Forwarding information derived from the forwarding table before any user traffic arrives at the switch. • As a packet arrives at an interface module, its DFC engine inspects the packet and uses the information in the Cisco Express Forwarding table (including Layer 2, Layer 3, ACLs, and QoS) to make a completely hardware-based forwarding decision for that packet.

No.	’740 Patent Claim 1	Cisco EtherChannel
		• The Distributed Cisco Express Forwarding engine manages all hardware-based forwarding for traffic on that module, including Layer 2 and Layer 3 forwarding, ACLs, QoS policing and marking, and NetFlow. • Because the DFCs make all the switching decisions locally, the central PFCx can dedicate more hardware-forwarding resources to any modules not equipped with a DFC.”) Cisco Catalyst 6500 Data Sheet at Figure 6 (annotation added) **Figure 6.** Distributed Cisco Express Forwarding Packet Flow  2. Packet Enters Switch or Line Card • All local ports and DFC see frame • DFC uses lookup table for local or other line card destination 3. If Destination is on another line card, DFC tells SFM to prepend tag on packet with exit SFM port info MSFC has Cisco Express Forwarding-based control plane: 1. MSFC delivers forwarding table to all DFC-enabled modules • Eliminates Supervisor Engine from forwarding path (including card to card traffic) • Enables local intelligent switching, supporting network services (security, QoS, etc.) 5. Line card takes frame from SFM and places on its own local bus • The DFC provides destination port and exit port • Packet is queued, QoS applied, and packet exits line card 4. SFM receives packet, examines tag, makes switching decision • Determines outgoing port on line card and switches packet to specified line card Fabric-Enabled Line Card DFC Fabric-Enabled Line Card DFC Fabric-Enabled Line Card DFC Fabric Switch Module DBus MSFC2 PFC2 Supervisor Engines Line Card Line Card Catalyst 6500/6000 Module **Components Used** The information in this document is based on these software and hardware versions: • Catalyst 6500 with Supervisor II with Multilayer Switch Feature Card 2 (MSFC2)

No.	’740 Patent Claim 1	Cisco EtherChannel
		 - WS-X6348 module - Cisco IOS version 12.1(11b)E4 The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command. **Conventions** Refer to the Cisco Technical Tips Conventions for more information on document conventions. **Before You Begin** **WS-X6348 Module Architecture** Each WS-X6348 card is controlled by a single Application-Specific Integrated Circuit (ASIC) that connects the module to both the 32 GB data bus backplane of the switch and to a set of four other ASICs which controls groups of 12 10/100 ports. An understanding of this architecture is important as it can help in troubleshooting interface problems. For example, if a group of 12 10/100 interfaces fails the online diagnostics (refer to Step 18 of this document to learn more about the **show diagnostic module** *<mod#>* command), this typically indicates one of the ASICs mentioned above has failed. Catalyst 3500 Installation Guide at Table 1-1

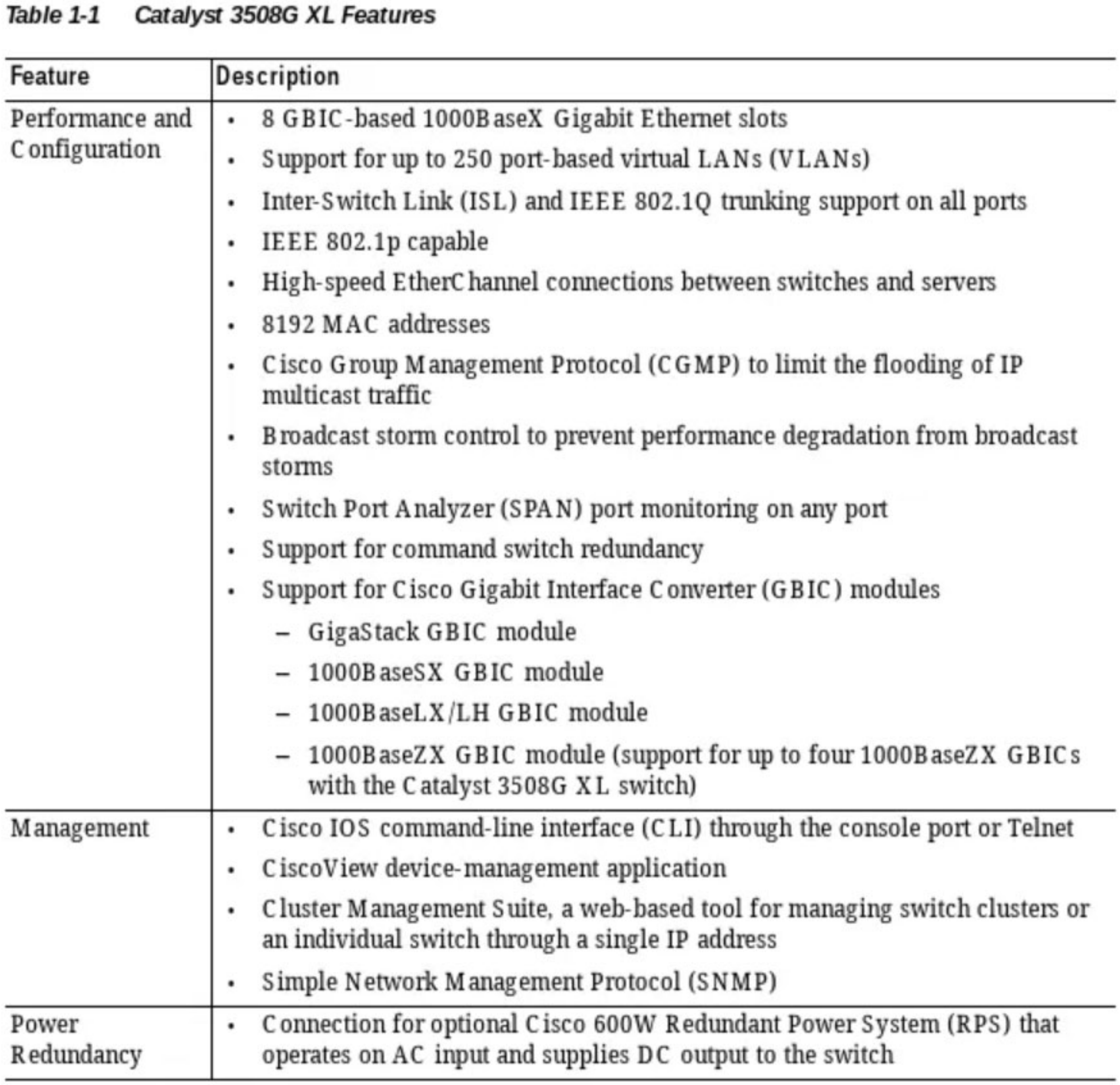
No.	’740 Patent Claim 1	Cisco EtherChannel
		

Table 1-1 Catalyst 3508G XL Features

Feature	Description
Performance and Configuration	• 8 GBIC-based 1000BaseX Gigabit Ethernet slots • Support for up to 250 port-based virtual LANs (VLANs) • Inter-Switch Link (ISL) and IEEE 802.1Q trunking support on all ports • IEEE 802.1p capable • High-speed EtherChannel connections between switches and servers • 8192 MAC addresses • Cisco Group Management Protocol (CGMP) to limit the flooding of IP multicast traffic • Broadcast storm control to prevent performance degradation from broadcast storms • Switch Port Analyzer (SPAN) port monitoring on any port • Support for command switch redundancy • Support for Cisco Gigabit Interface Converter (GBIC) modules – GigaStack GBIC module – 1000BaseSX GBIC module – 1000BaseLX/LH GBIC module – 1000BaseZX GBIC module (support for up to four 1000BaseZX GBICs with the Catalyst 3508G XL switch)
Management	• Cisco IOS command-line interface (CLI) through the console port or Telnet • CiscoView device-management application • Cluster Management Suite, a web-based tool for managing switch clusters or an individual switch through a single IP address • Simple Network Management Protocol (SNMP)
Power Redundancy	• Connection for optional Cisco 600W Redundant Power System (RPS) that operates on AC input and supplies DC output to the switch

<table>
<tr><th>No.</th><th>’740 Patent Claim 1</th><th>Cisco EtherChannel</th></tr>
<tr><td></td><td></td><td>
Catalyst 3500 Installation Guide at 1-2

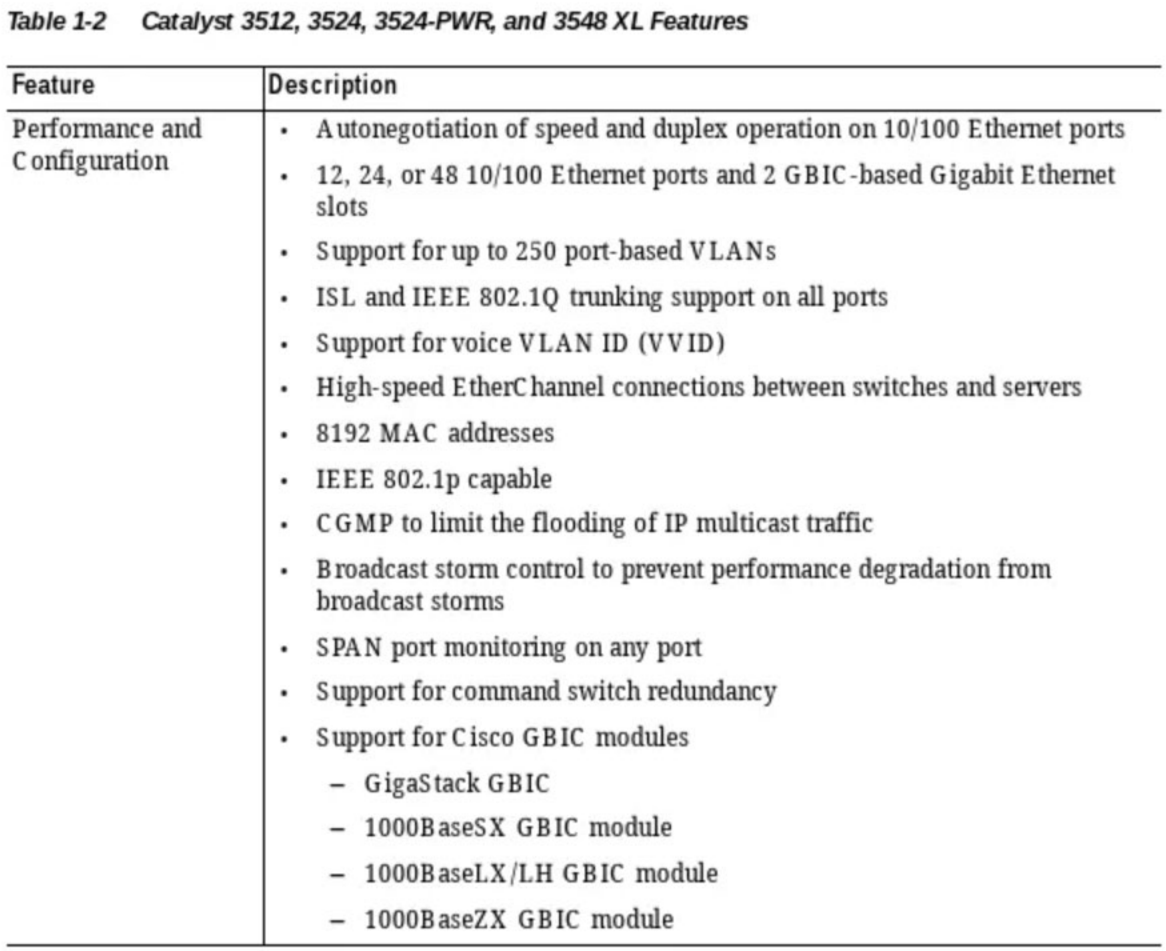


Table 1-2 Catalyst 3512, 3524, 3524-PWR, and 3548 XL Features

<table>
<tr><th>Feature</th><th>Description</th></tr>
<tr><td>Performance and Configuration</td><td>• Autonegotiation of speed and duplex operation on 10/100 Ethernet ports
• 12, 24, or 48 10/100 Ethernet ports and 2 GBIC-based Gigabit Ethernet slots
• Support for up to 250 port-based VLANs
• ISL and IEEE 802.1Q trunking support on all ports
• Support for voice VLAN ID (VVID)
• High-speed EtherChannel connections between switches and servers
• 8192 MAC addresses
• IEEE 802.1p capable
• CGMP to limit the flooding of IP multicast traffic
• Broadcast storm control to prevent performance degradation from broadcast storms
• SPAN port monitoring on any port
• Support for command switch redundancy
• Support for Cisco GBIC modules
– GigaStack GBIC
– 1000BaseSX GBIC module
– 1000BaseLX/LH GBIC module
– 1000BaseZX GBIC module</td></tr>
</table>

Catalyst 3500 Installation Guide at Table 1-9
</td></tr>
</table>

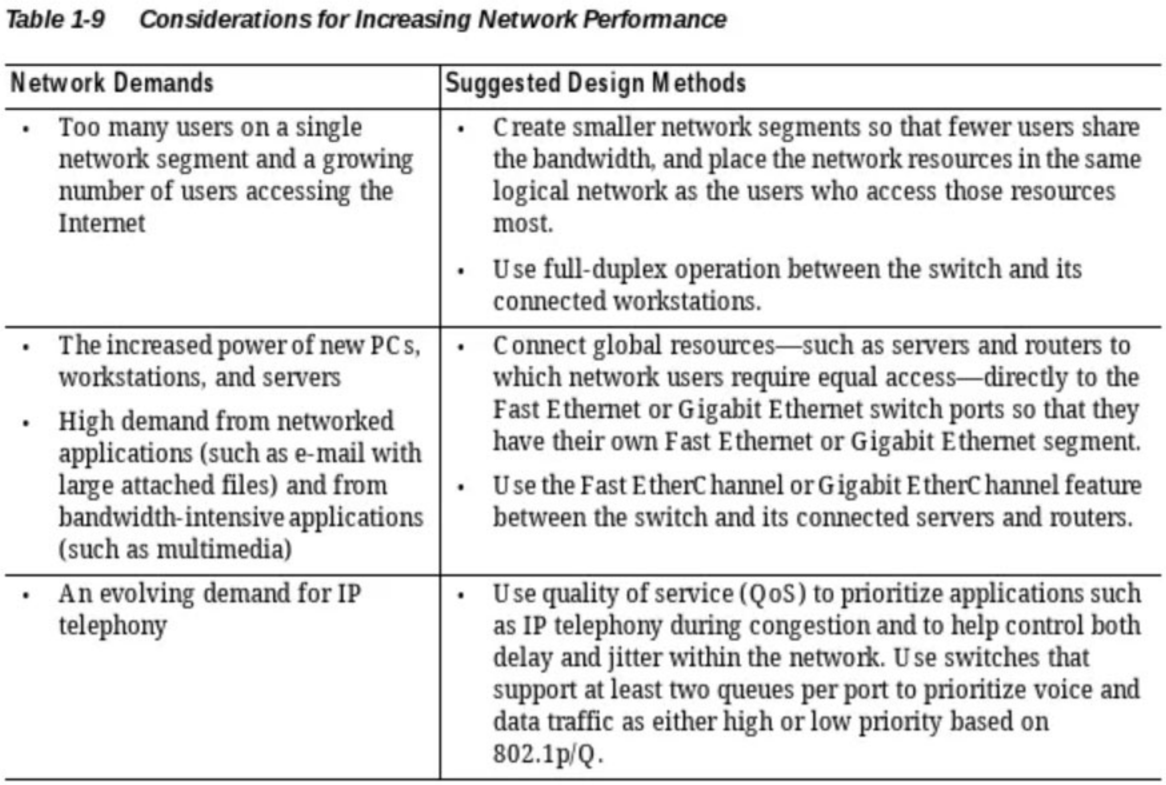
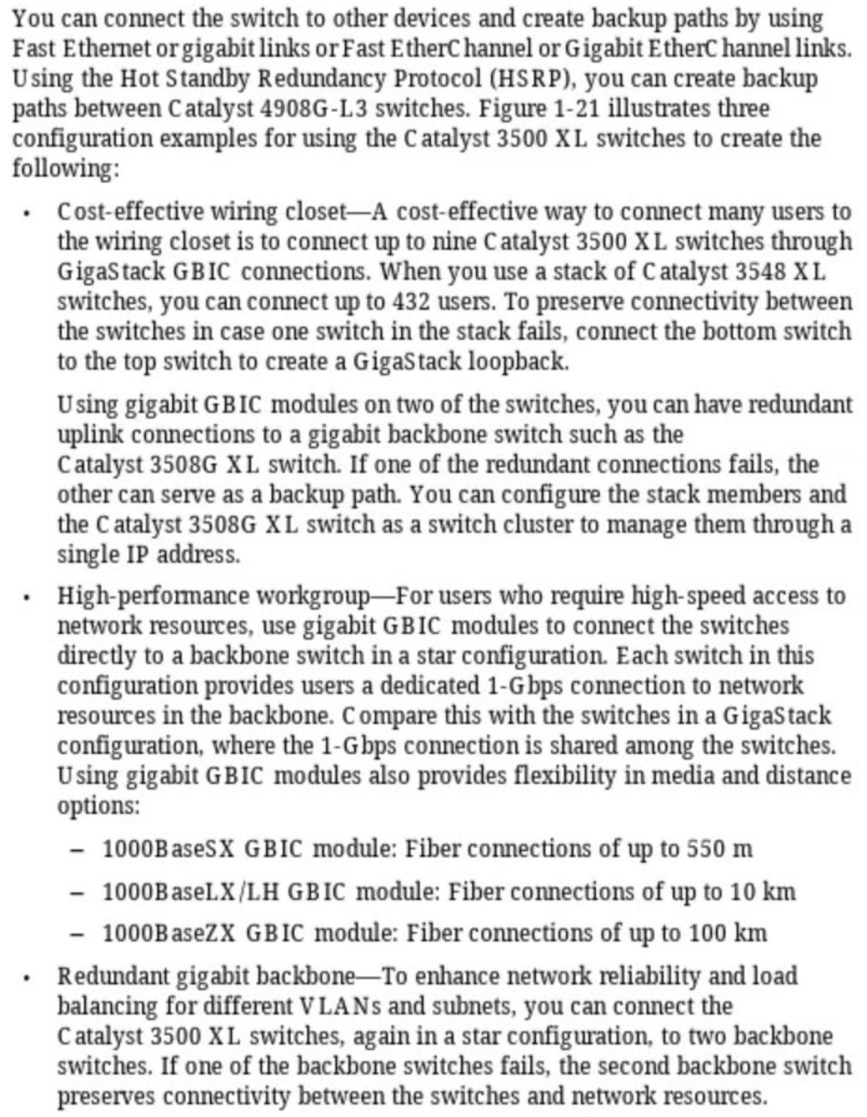
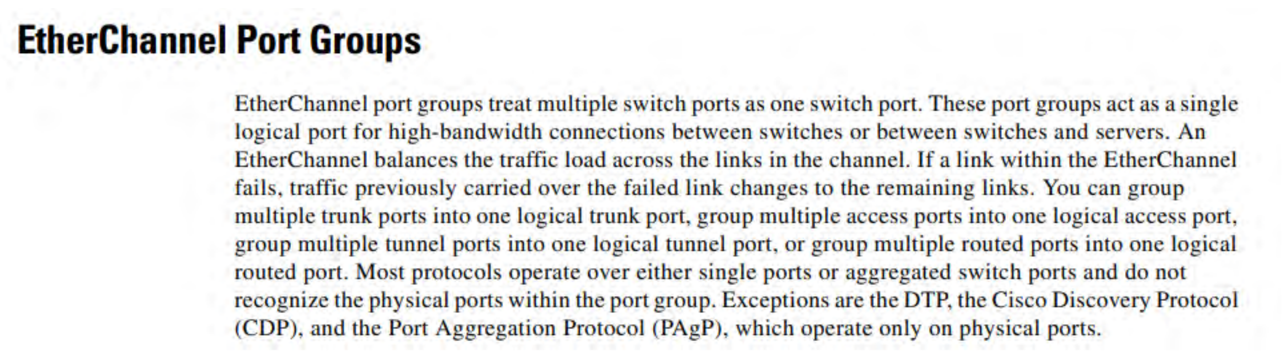
<table>
<tr><th>No.</th><th>’740 Patent Claim 1</th><th>Cisco EtherChannel</th></tr>
<tr><td></td><td></td><td>


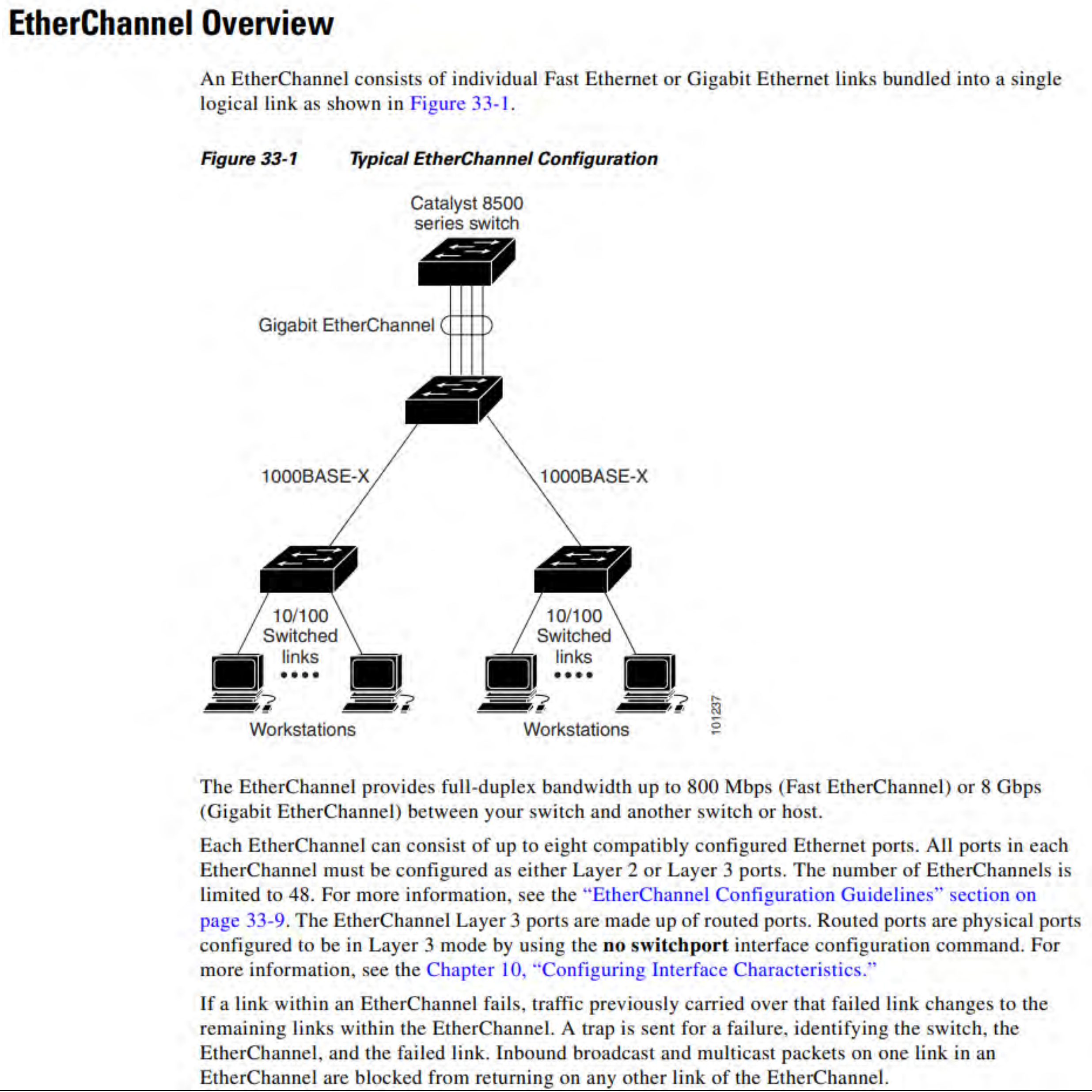
Table 1-9 Considerations for Increasing Network Performance

<table>
<tr><th>Network Demands</th><th>Suggested Design Methods</th></tr>
<tr><td>• Too many users on a single network segment and a growing number of users accessing the Internet</td><td>• Create smaller network segments so that fewer users share the bandwidth, and place the network resources in the same logical network as the users who access those resources most.
• Use full-duplex operation between the switch and its connected workstations.</td></tr>
<tr><td>• The increased power of new PCs, workstations, and servers
• High demand from networked applications (such as e-mail with large attached files) and from bandwidth-intensive applications (such as multimedia)</td><td>• Connect global resources—such as servers and routers to which network users require equal access—directly to the Fast Ethernet or Gigabit Ethernet switch ports so that they have their own Fast Ethernet or Gigabit Ethernet segment.
• Use the Fast EtherChannel or Gigabit EtherChannel feature between the switch and its connected servers and routers.</td></tr>
<tr><td>• An evolving demand for IP telephony</td><td>• Use quality of service (QoS) to prioritize applications such as IP telephony during congestion and to help control both delay and jitter within the network. Use switches that support at least two queues per port to prioritize voice and data traffic as either high or low priority based on 802.1p/Q.</td></tr>
</table>

Catalyst 3500 Installation Guide at 1-30
</td></tr>
</table>

No.	’740 Patent Claim 1	Cisco EtherChannel
		 You can connect the switch to other devices and create backup paths by using Fast Ethernet or gigabit links or Fast EtherChannel or Gigabit EtherChannel links. Using the Hot Standby Redundancy Protocol (HSRP), you can create backup paths between Catalyst 4908G-L3 switches. Figure 1-21 illustrates three configuration examples for using the Catalyst 3500 XL switches to create the following: • Cost-effective wiring closet—A cost-effective way to connect many users to the wiring closet is to connect up to nine Catalyst 3500 XL switches through GigaStack GBIC connections. When you use a stack of Catalyst 3548 XL switches, you can connect up to 432 users. To preserve connectivity between the switches in case one switch in the stack fails, connect the bottom switch to the top switch to create a GigaStack loopback. Using gigabit GBIC modules on two of the switches, you can have redundant uplink connections to a gigabit backbone switch such as the Catalyst 3508G XL switch. If one of the redundant connections fails, the other can serve as a backup path. You can configure the stack members and the Catalyst 3508G XL switch as a switch cluster to manage them through a single IP address. • High-performance workgroup—For users who require high-speed access to network resources, use gigabit GBIC modules to connect the switches directly to a backbone switch in a star configuration. Each switch in this configuration provides users a dedicated 1-Gbps connection to network resources in the backbone. Compare this with the switches in a GigaStack configuration, where the 1-Gbps connection is shared among the switches. Using gigabit GBIC modules also provides flexibility in media and distance options: – 1000BaseSX GBIC module: Fiber connections of up to 550 m – 1000BaseLX/LH GBIC module: Fiber connections of up to 10 km – 1000BaseZX GBIC module: Fiber connections of up to 100 km • Redundant gigabit backbone—To enhance network reliability and load balancing for different VLANs and subnets, you can connect the Catalyst 3500 XL switches, again in a star configuration, to two backbone switches. If one of the backbone switches fails, the second backbone switch preserves connectivity between the switches and network resources.

No.	’740 Patent Claim 1	Cisco EtherChannel
		Catalyst 3560 Configuration Guide at 10-5  EtherChannel Port Groups EtherChannel port groups treat multiple switch ports as one switch port. These port groups act as a single logical port for high-bandwidth connections between switches or between switches and servers. An EtherChannel balances the traffic load across the links in the channel. If a link within the EtherChannel fails, traffic previously carried over the failed link changes to the remaining links. You can group multiple trunk ports into one logical trunk port, group multiple access ports into one logical access port, group multiple tunnel ports into one logical tunnel port, or group multiple routed ports into one logical routed port. Most protocols operate over either single ports or aggregated switch ports and do not recognize the physical ports within the port group. Exceptions are the DTP, the Cisco Discovery Protocol (CDP), and the Port Aggregation Protocol (PAgP), which operate only on physical ports. Catalyst 3560 Configuration Guide at 33-2



EtherChannel Overview

An EtherChannel consists of individual Fast Ethernet or Gigabit Ethernet links bundled into a single logical link as shown in Figure 33-1.

Figure 33-1 Typical EtherChannel Configuration

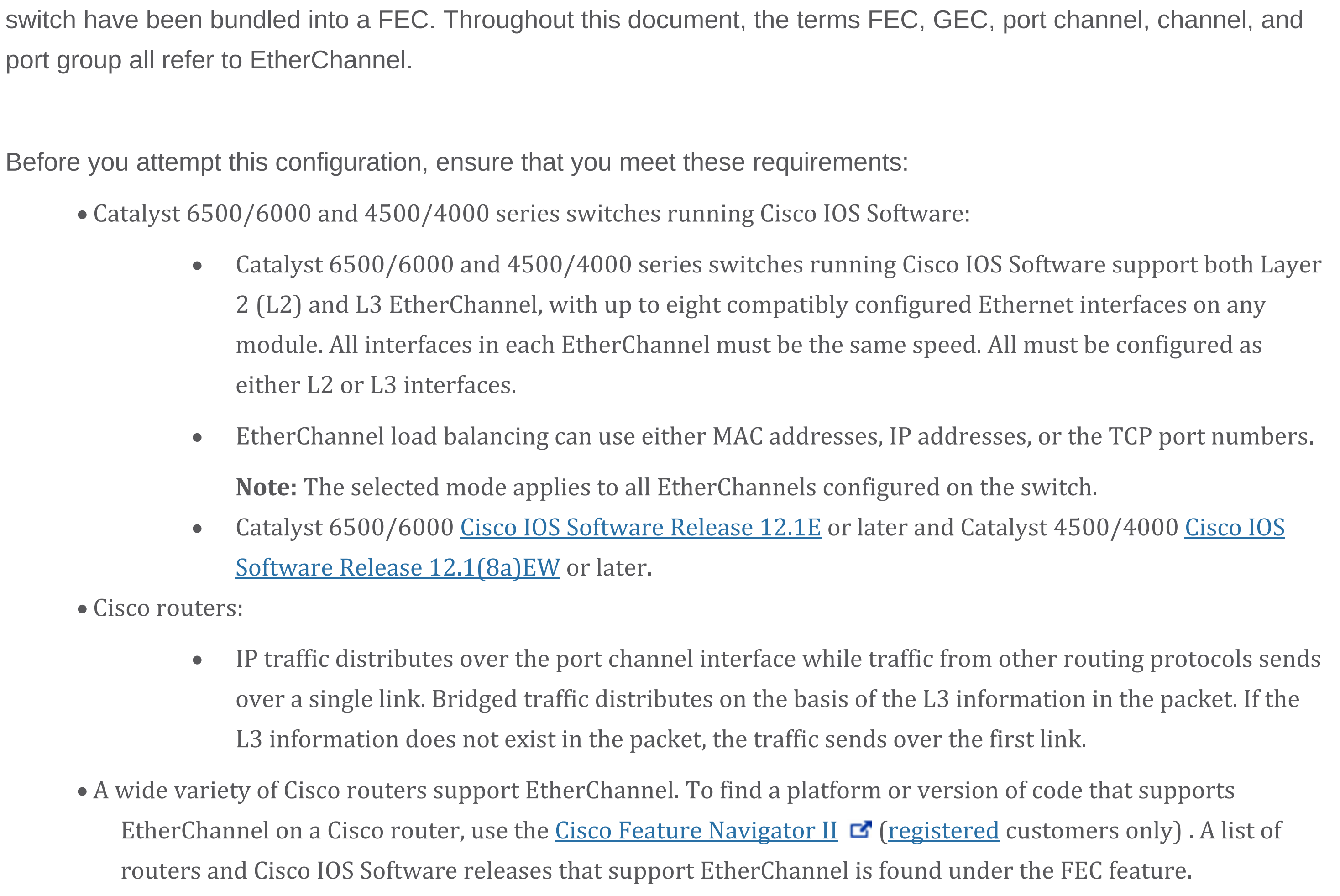
Catalyst 8500
series switch
Gigabit EtherChannel
1000BASE-X
1000BASE-X
10/100
Switched
links
10/100
Switched
links
Workstations
Workstations
101237

The EtherChannel provides full-duplex bandwidth up to 800 Mbps (Fast EtherChannel) or 8 Gbps (Gigabit EtherChannel) between your switch and another switch or host.

Each EtherChannel can consist of up to eight compatibly configured Ethernet ports. All ports in each EtherChannel must be configured as either Layer 2 or Layer 3 ports. The number of EtherChannels is limited to 48. For more information, see the "EtherChannel Configuration Guidelines" section on page 33-9. The EtherChannel Layer 3 ports are made up of routed ports. Routed ports are physical ports configured to be in Layer 3 mode by using the **no switchport** interface configuration command. For more information, see the Chapter 10, "Configuring Interface Characteristics."

If a link within an EtherChannel fails, traffic previously carried over that failed link changes to the remaining links within the EtherChannel. A trap is sent for a failure, identifying the switch, the EtherChannel, and the failed link. Inbound broadcast and multicast packets on one link in an EtherChannel are blocked from returning on any other link of the EtherChannel.

No.	’740 Patent Claim 1	Cisco EtherChannel
		Layer 2 EtherChannel **Catalyst 6500 That Runs Cisco IOS Software** Catalyst 6500 switches that run Cisco IOS Software support L2 (switchport) and Layer 3 (L3) (routed port) EtherChannel configurations. A Catalyst 6500/6000 series switch supports a maximum of 64 EtherChannels (256 with Cisco IOS Software Release 12.1(2)E and earlier). You can form an EtherChannel with up to eight compatibly configured LAN ports on any module in a Catalyst 6000 series switch, with the exception of Digital Feature Card (DFC)-equipped modules (such as WS-X6816 and so on) which currently allow an L2 channel only using ports on the same DFC module. However, an L3 channel can be configured across different DFC-equipped modules. This limitation has been removed in Catalyst 6500/6000 Cisco IOS Software Release 12.1(11b)EX and later. This document configures an L2 EtherChannel. The Catalyst 6500/6000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP addresses, or Layer 4 (L4) port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. Catalyst 6500/6000 switches support both ISL and 802.1Q trunking encapsulations and DTP. Detailed information on port capabilities is available by issuing the **show interface** *interface_id* **capabilities** command. Configuring EtherChannel at 1 -2 **Introduction** This sample configuration demonstrates how to set up a Layer 3 (L3) EtherChannel, without VLAN trunking, between a Cisco router and a Cisco Catalyst 6500 switch running Cisco IOS® System Software. EtherChannel can be called Fast EtherChannel (FEC) or Gigabit EtherChannel (GEC); the term depends on the speed of the interfaces or ports you use to form the EtherChannel. In this example, two Fast Ethernet ports from a Cisco router and a Catalyst 6500

No.	’740 Patent Claim 1	Cisco EtherChannel
		 switch have been bundled into a FEC. Throughout this document, the terms FEC, GEC, port channel, channel, and port group all refer to EtherChannel. Before you attempt this configuration, ensure that you meet these requirements: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software support both Layer 2 (L2) and L3 EtherChannel, with up to eight compatibly configured Ethernet interfaces on any module. All interfaces in each EtherChannel must be the same speed. All must be configured as either L2 or L3 interfaces. • EtherChannel load balancing can use either MAC addresses, IP addresses, or the TCP port numbers. **Note:** The selected mode applies to all EtherChannels configured on the switch. • Catalyst 6500/6000 Cisco IOS Software Release 12.1E or later and Catalyst 4500/4000 Cisco IOS Software Release 12.1(8a)EW or later. • Cisco routers: • IP traffic distributes over the port channel interface while traffic from other routing protocols sends over a single link. Bridged traffic distributes on the basis of the L3 information in the packet. If the L3 information does not exist in the packet, the traffic sends over the first link. • A wide variety of Cisco routers support EtherChannel. To find a platform or version of code that supports EtherChannel on a Cisco router, use the Cisco Feature Navigator II (registered customers only) . A list of routers and Cisco IOS Software releases that support EtherChannel is found under the FEC feature.
1[b]	at least one of said first physical links being a bi-	Cisco EtherChannel System discloses at least one of said first physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction.

No.	’740 Patent Claim 1	Cisco EtherChannel
	directional link operative to communicate in both an upstream direction and a downstream direction	For example, Cisco EtherChannel System discloses full duplex connections operating over Ethernet ports. Fully duplexed connections mean that such connections operate both upstream and downstream traffic. Cisco Catalyst 6500 Data Sheet at 9 (“Cisco Catalyst 6500 Series interface modules support the following forwarding technology and switch-fabric combinations. • Classic Interface Modules—Use the centralized Cisco Express Forwarding engine located on the supervisor engine’s PFC, connect to the 32-Gbps switching bus only, and forward packets at up to 15 mpps • CEF256 Interface Modules—Use the centralized Cisco Express Forwarding engine located on the supervisor engine’s PFC, connect to both the 256-Gbps fabric located on the supervisor engine with a single 8-Gbps full-duplex fabric connection and the 32-Gbps switching bus, and forward packets at up to 30 mpps • dCEF256 Interface Modules—Use the distributed Cisco Express Forwarding engine on the DFC (located on the interface module), connect to a 256-Gbps fabric located on the supervisor engine or a switch fabric module with 16-Gbps full-duplex fabric connections, and forward packets at up to 210 mpps • dCEF720 Interface Modules—Use the distributed Cisco Express Forwarding engine on the DFC3 (located on the interface module), connect to the 720-Gbps fabric located on the supervisor engine with dual 20-Gbps full-duplex fabric connections, and forward packets at up to 400 mpps of sustained performance”) Catalyst 3500 Installation Guide at 1-2

No.	’740 Patent Claim 1	Cisco EtherChannel
		Catalyst 3500 Installation Guide at 1-8

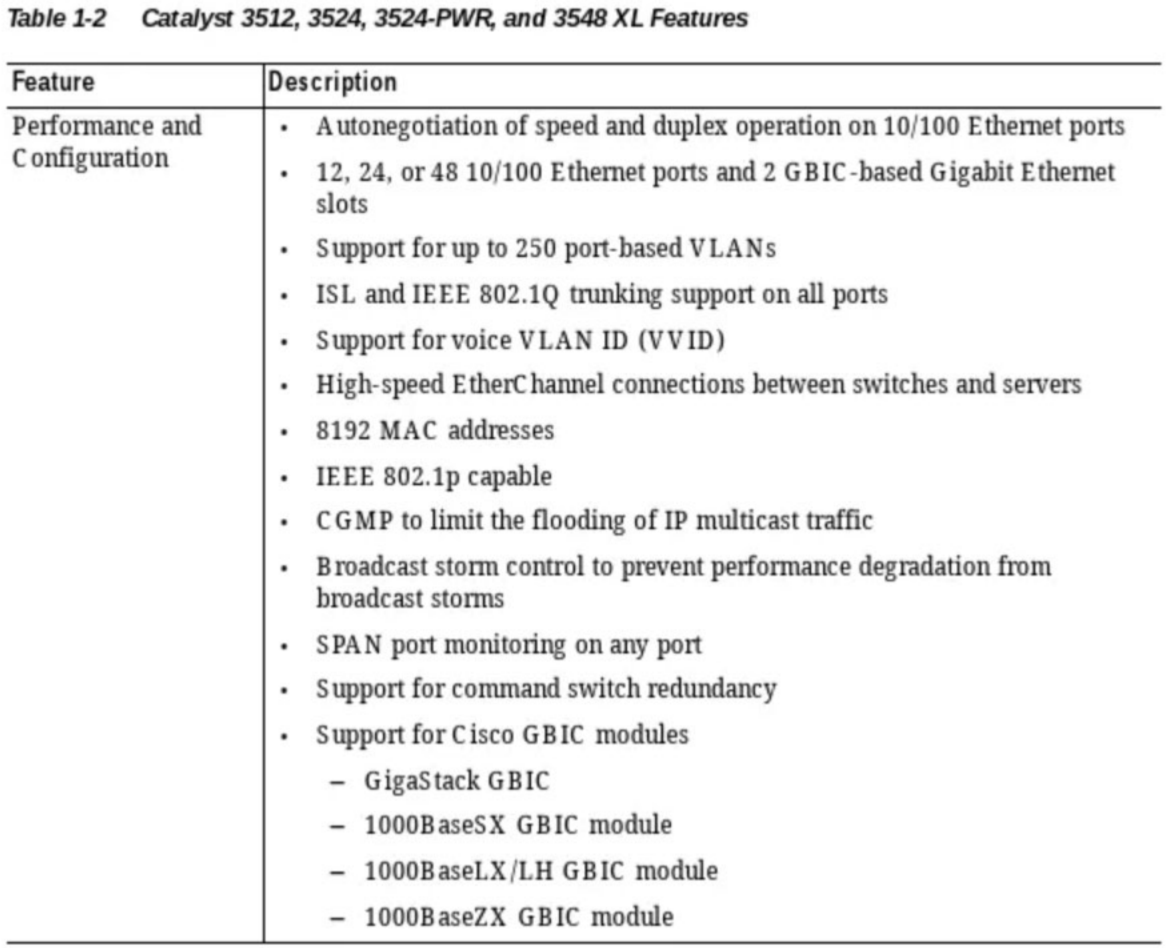
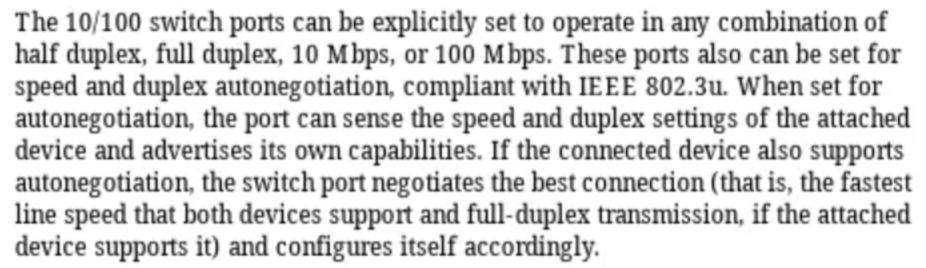
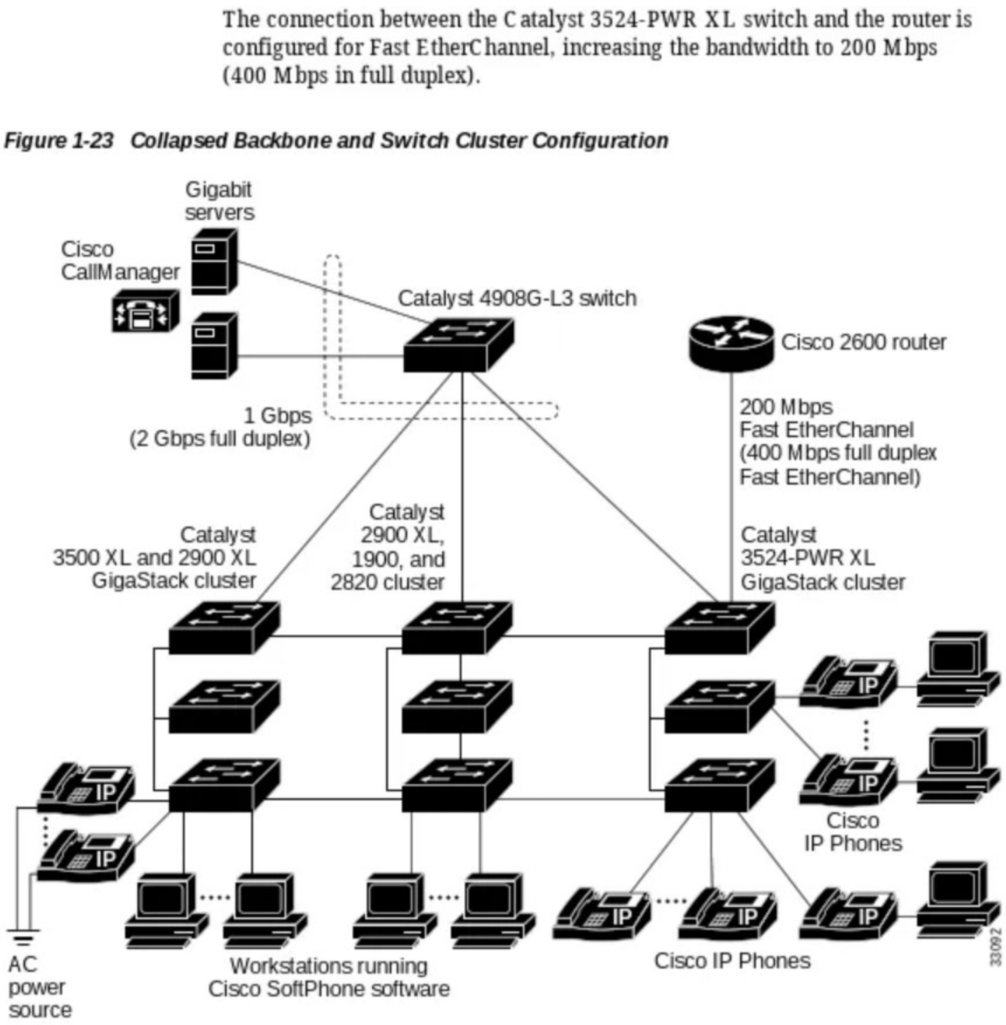


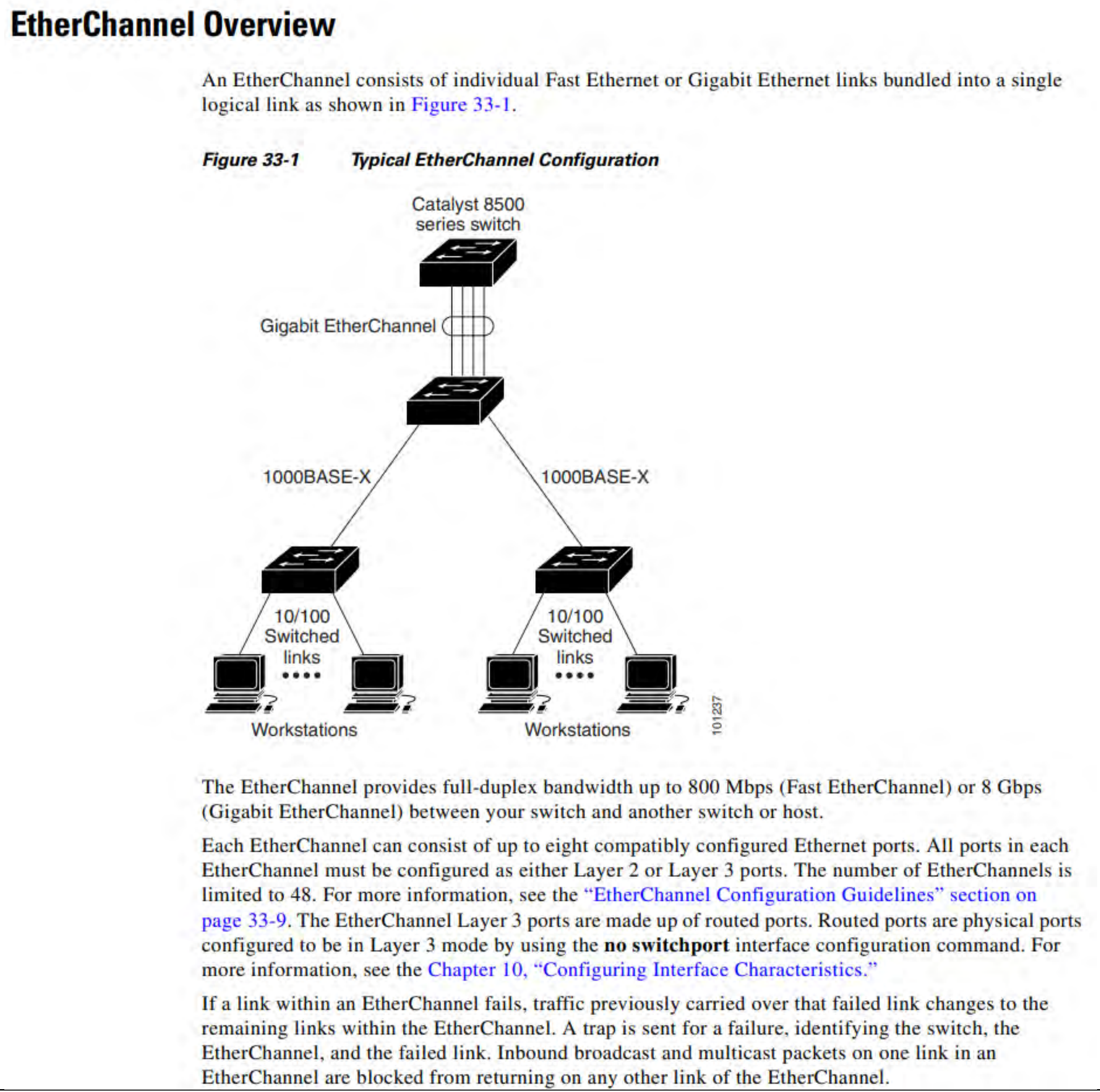
Table 1-2 Catalyst 3512, 3524, 3524-PWR, and 3548 XL Features

Feature	Description
Performance and Configuration	• Autonegotiation of speed and duplex operation on 10/100 Ethernet ports • 12, 24, or 48 10/100 Ethernet ports and 2 GBIC-based Gigabit Ethernet slots • Support for up to 250 port-based VLANs • ISL and IEEE 802.1Q trunking support on all ports • Support for voice VLAN ID (VVID) • High-speed EtherChannel connections between switches and servers • 8192 MAC addresses • IEEE 802.1p capable • CGMP to limit the flooding of IP multicast traffic • Broadcast storm control to prevent performance degradation from broadcast storms • SPAN port monitoring on any port • Support for command switch redundancy • Support for Cisco GBIC modules – GigaStack GBIC – 1000BaseSX GBIC module – 1000BaseLX/LH GBIC module – 1000BaseZX GBIC module

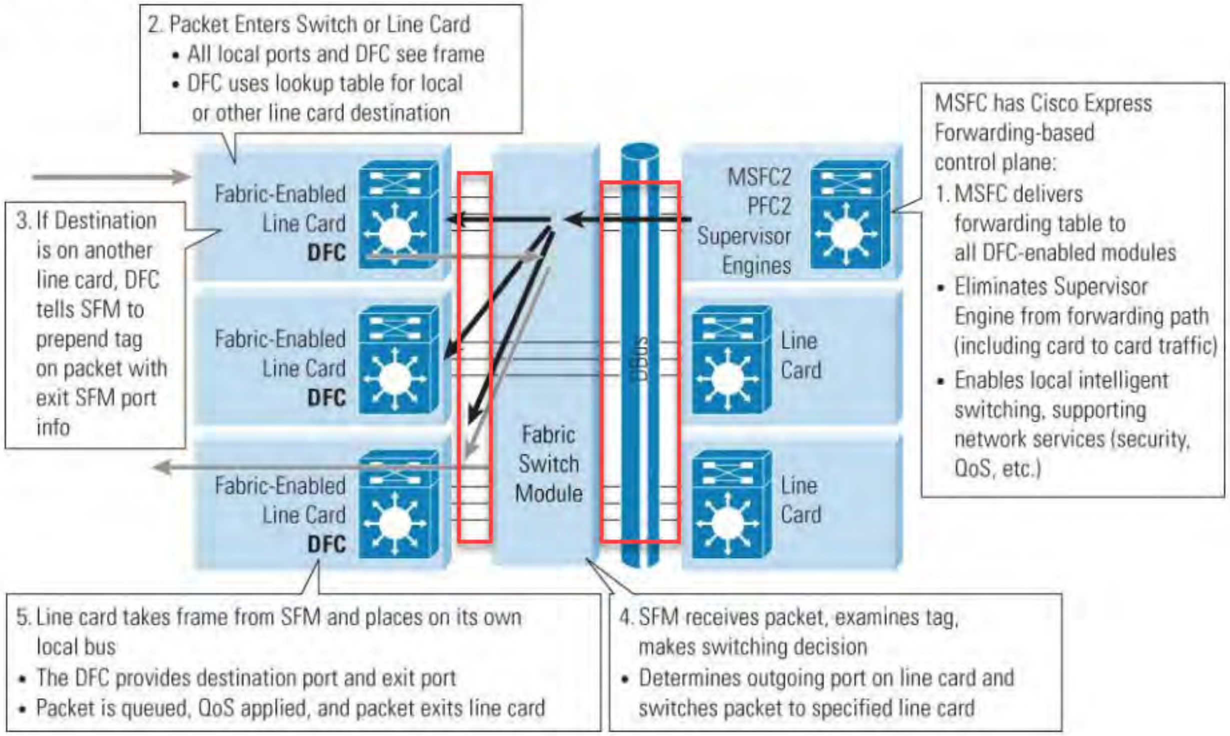
Catalyst 3500 Installation Guide at 1-8

No.	’740 Patent Claim 1	Cisco EtherChannel
		 The 10/100 switch ports can be explicitly set to operate in any combination of half duplex, full duplex, 10 Mbps, or 100 Mbps. These ports also can be set for speed and duplex autonegotiation, compliant with IEEE 802.3u. When set for autonegotiation, the port can sense the speed and duplex settings of the attached device and advertises its own capabilities. If the connected device also supports autonegotiation, the switch port negotiates the best connection (that is, the fastest line speed that both devices support and full-duplex transmission, if the attached device supports it) and configures itself accordingly. Catalyst 3500 Installation Guide at Figure 1-23

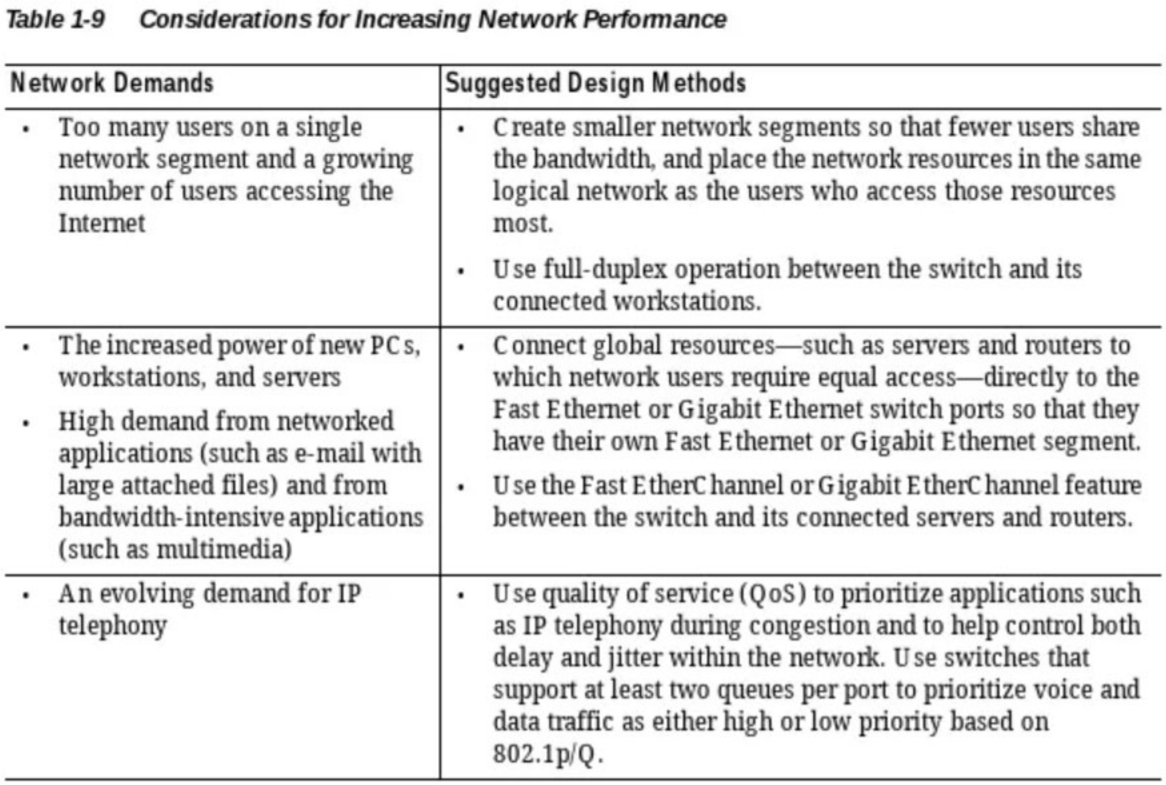
No.	’740 Patent Claim 1	Cisco EtherChannel
		 The connection between the Catalyst 3524-PWR XL switch and the router is configured for Fast EtherChannel, increasing the bandwidth to 200 Mbps (400 Mbps in full duplex). *Figure 1-23 Collapsed Backbone and Switch Cluster Configuration* Gigabit servers Cisco CallManager Catalyst 4908G-L3 switch Cisco 2600 router 1 Gbps (2 Gbps full duplex) 200 Mbps Fast EtherChannel (400 Mbps full duplex Fast EtherChannel) Catalyst 3500 XL and 2900 XL GigaStack cluster Catalyst 2900 XL, 1900, and 2820 cluster Catalyst 3524-PWR XL GigaStack cluster IP Cisco IP Phones AC power source Workstations running Cisco SoftPhone software Cisco IP Phones 33092 Catalyst 3560 Configuration Guide at 33-2

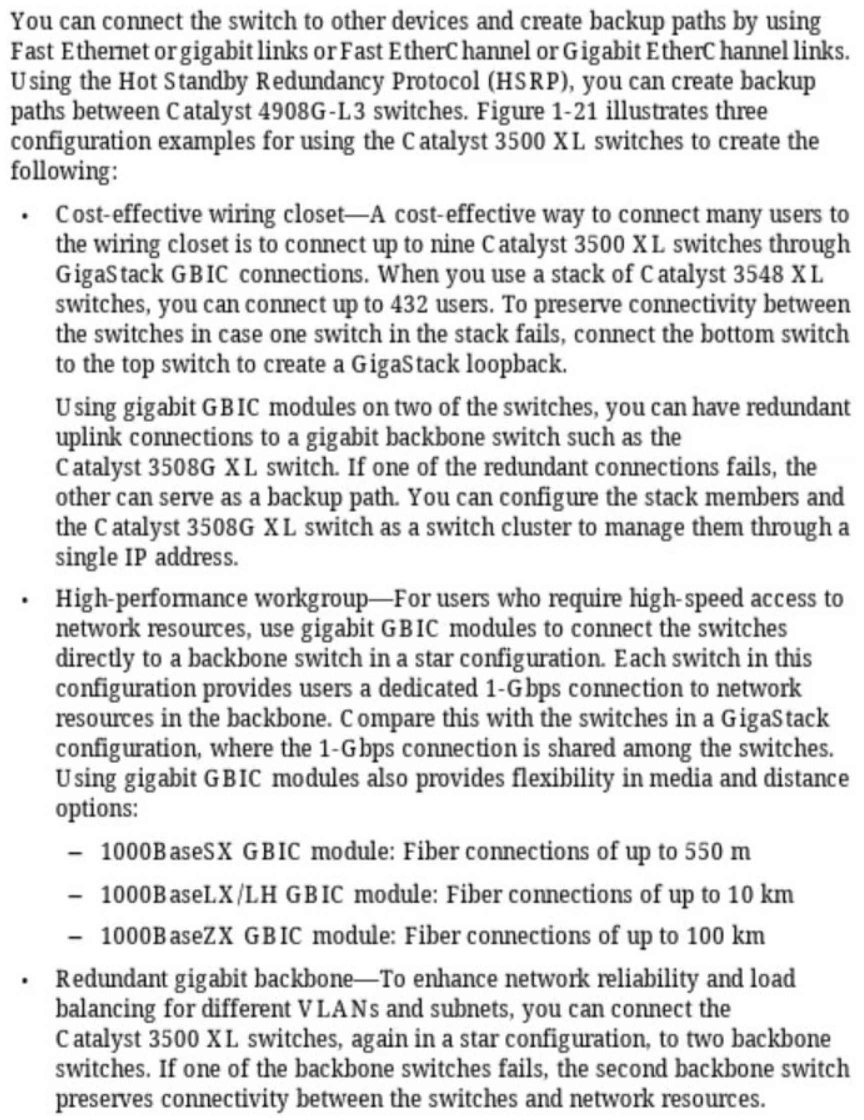
No.	’740 Patent Claim 1	Cisco EtherChannel
		 **EtherChannel Overview** An EtherChannel consists of individual Fast Ethernet or Gigabit Ethernet links bundled into a single logical link as shown in Figure 33-1. *Figure 33-1 Typical EtherChannel Configuration* Catalyst 8500 series switch Gigabit EtherChannel 1000BASE-X 1000BASE-X 10/100 Switched links 10/100 Switched links Workstations Workstations 101237 The EtherChannel provides full-duplex bandwidth up to 800 Mbps (Fast EtherChannel) or 8 Gbps (Gigabit EtherChannel) between your switch and another switch or host. Each EtherChannel can consist of up to eight compatibly configured Ethernet ports. All ports in each EtherChannel must be configured as either Layer 2 or Layer 3 ports. The number of EtherChannels is limited to 48. For more information, see the “EtherChannel Configuration Guidelines” section on page 33-9. The EtherChannel Layer 3 ports are made up of routed ports. Routed ports are physical ports configured to be in Layer 3 mode by using the **no switchport** interface configuration command. For more information, see the Chapter 10, “Configuring Interface Characteristics.” If a link within an EtherChannel fails, traffic previously carried over that failed link changes to the remaining links within the EtherChannel. A trap is sent for a failure, identifying the switch, the EtherChannel, and the failed link. Inbound broadcast and multicast packets on one link in an EtherChannel are blocked from returning on any other link of the EtherChannel.

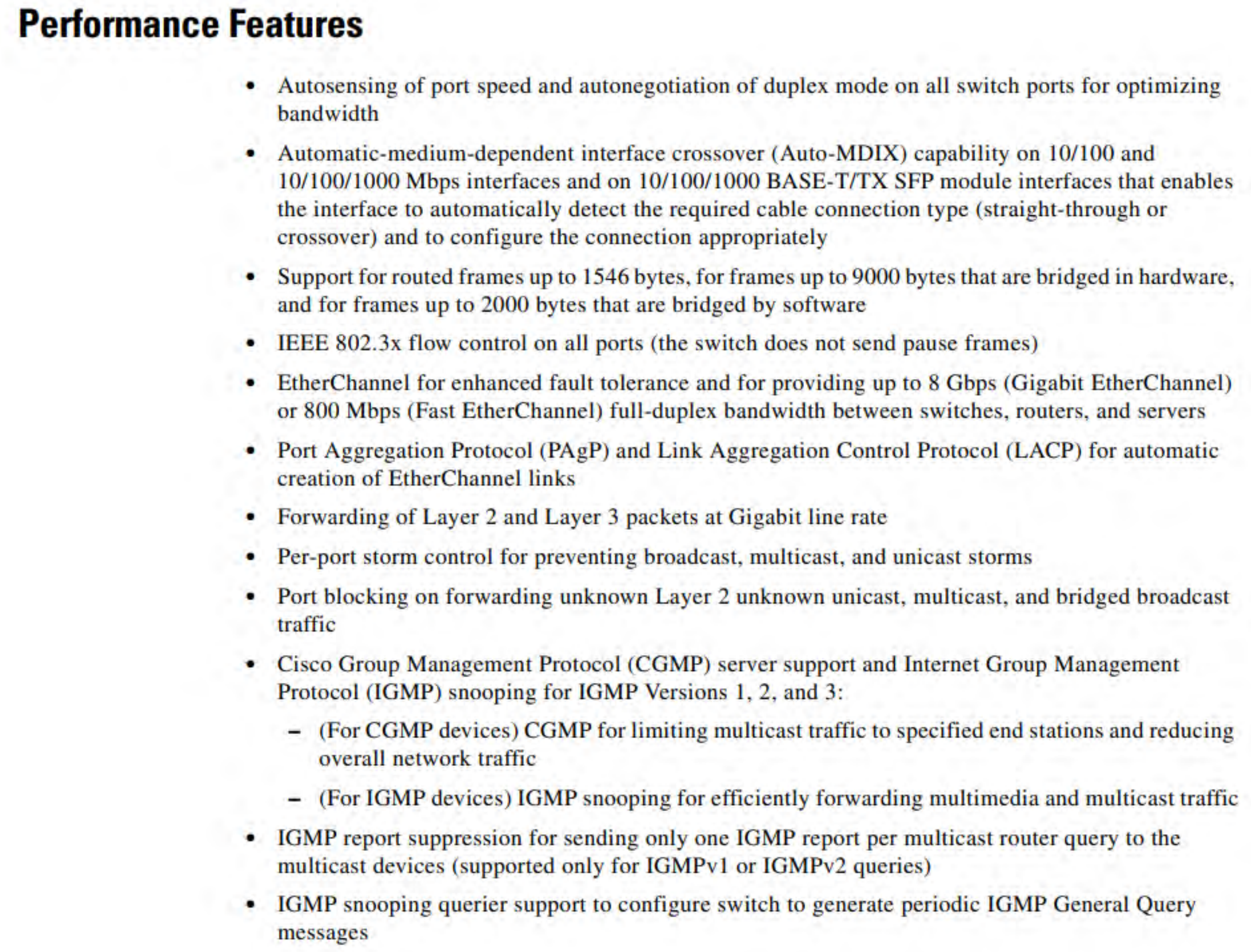
No.	’740 Patent Claim 1	Cisco EtherChannel
1[c]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel,	Cisco EtherChannel System discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. For example, Cisco EtherChannel System discloses connecting the interface modules or line cards to a network via a switch fabric with parallel switch fabric interconnections. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Cisco EtherChannel System is found to not meet this limitation, coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel would have been obvious to a person having ordinary skill in the art, as explained below. Cisco Catalyst 6500 Data Sheet at 2 (“Offers optional, redundant, high-performance Cisco Catalyst 6500 Series Supervisor Engine 720, passive backplane, multimodule Cisco EtherChannel® technology, IEEE 802.3ad link aggregation, IEEE 802.1s only, and Hot Standby Router Protocol/Virtual Router Redundancy Protocol (HSRP/VRRP) high-availability features”) Cisco Catalyst 6500 Data Sheet at 13 (“**Switch Fabric Modules** Designed to support distributed forwarding for interface modules with that capability, the Cisco Catalyst 6500 Series Switch Fabric Module (SFM or SFM2), in combination with the Cisco Catalyst 6000 Multilayer Switch Feature Card (MSFC2) and Cisco distributed forwarding cards (DFCs) on interface modules, increases available system bandwidth from 32 to 256 Gbps. The SFM or SFM2 supports the Cisco Catalyst 6500 CEF256 and dCEF256 interface modules. Designed to support new interface modules with 720-Gbps forwarding capabilities, the switch fabric onboard the Cisco Catalyst 6500 Series Supervisor Engine 720 increases available bandwidth to 720 Gbps and provides packet-forwarding rates up to 400 mpps. By using automatic sensing and negotiation, the switch fabric is fully interoperable with the 8- and 16-Gbps switch-fabric interconnections used by the CEF256 and dCEF256 interface modules. When a CEF256 or dCEF256 interface module is detected, the switch fabric will automatically connect those modules by offering 8 to 16 Gbps of bandwidth to each module, as applicable.”) Cisco Catalyst 6500 Data Sheet at Figure 6

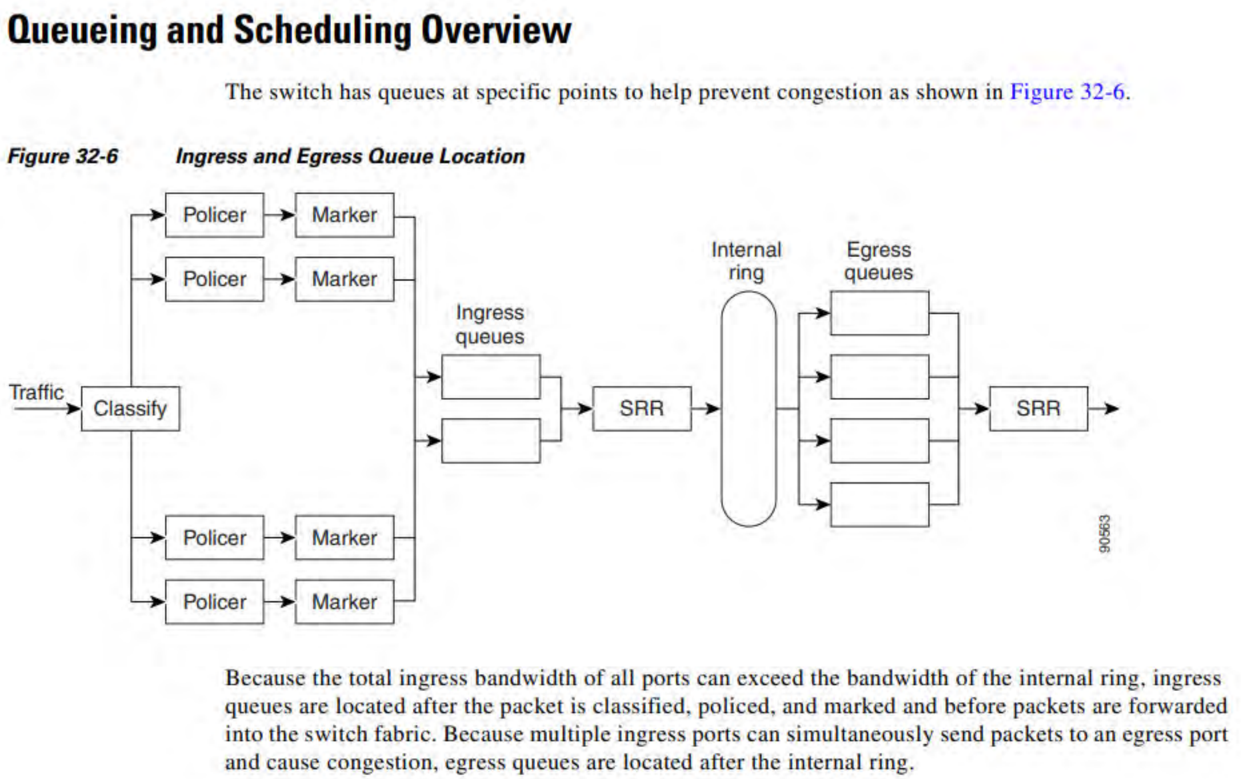
No.	’740 Patent Claim 1	Cisco EtherChannel
		Figure 6. Distributed Cisco Express Forwarding Packet Flow  Fabric-Enabled Line Card DFC Fabric-Enabled Line Card DFC Fabric-Enabled Line Card DFC Fabric Switch Module DBus MSFC2 PFC2 Supervisor Engines Line Card Line Card 2. Packet Enters Switch or Line Card • All local ports and DFC see frame • DFC uses lookup table for local or other line card destination 3. If Destination is on another line card, DFC tells SFM to prepend tag on packet with exit SFM port info MSFC has Cisco Express Forwarding-based control plane: 1. MSFC delivers forwarding table to all DFC-enabled modules • Eliminates Supervisor Engine from forwarding path (including card to card traffic) • Enables local intelligent switching, supporting network services (security, QoS, etc.) 5. Line card takes frame from SFM and places on its own local bus • The DFC provides destination port and exit port • Packet is queued, QoS applied, and packet exits line card 4. SFM receives packet, examines tag, makes switching decision • Determines outgoing port on line card and switches packet to specified line card Catalyst 6500/6000 Module **Components Used** The information in this document is based on these software and hardware versions: • Catalyst 6500 with Supervisor II with Multilayer Switch Feature Card 2 (MSFC2) • WS-X6348 module • Cisco IOS version 12.1(11b)E4

No.	’740 Patent Claim 1	Cisco EtherChannel
		The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command. **Conventions** Refer to the Cisco Technical Tips Conventions for more information on document conventions. **Before You Begin** **WS-X6348 Module Architecture** Each WS-X6348 card is controlled by a single Application-Specific Integrated Circuit (ASIC) that connects the module to both the 32 GB data bus backplane of the switch and to a set of four other ASICs which controls groups of 12 10/100 ports. An understanding of this architecture is important as it can help in troubleshooting interface problems. For example, if a group of 12 10/100 interfaces fails the online diagnostics (refer to Step 18 of this document to learn more about the **show diagnostic module** *<mod#>* command), this typically indicates one of the ASICs mentioned above has failed. Catalyst 3500 Installation Guide at Table 1-9

<table>
<tr><th>No.</th><th>’740 Patent Claim 1</th><th>Cisco EtherChannel</th></tr>
<tr><td></td><td></td><td>

<i>Table 1-9 Considerations for Increasing Network Performance</i>
<table>
<tr><th>Network Demands</th><th>Suggested Design Methods</th></tr>
<tr><td>• Too many users on a single network segment and a growing number of users accessing the Internet</td><td>• Create smaller network segments so that fewer users share the bandwidth, and place the network resources in the same logical network as the users who access those resources most.
• Use full-duplex operation between the switch and its connected workstations.</td></tr>
<tr><td>• The increased power of new PCs, workstations, and servers
• High demand from networked applications (such as e-mail with large attached files) and from bandwidth-intensive applications (such as multimedia)</td><td>• Connect global resources—such as servers and routers to which network users require equal access—directly to the Fast Ethernet or Gigabit Ethernet switch ports so that they have their own Fast Ethernet or Gigabit Ethernet segment.
• Use the Fast EtherChannel or Gigabit EtherChannel feature between the switch and its connected servers and routers.</td></tr>
<tr><td>• An evolving demand for IP telephony</td><td>• Use quality of service (QoS) to prioritize applications such as IP telephony during congestion and to help control both delay and jitter within the network. Use switches that support at least two queues per port to prioritize voice and data traffic as either high or low priority based on 802.1p/Q.</td></tr>
</table>
Catalyst 3500 Installation Guide at 1-30</td></tr>
</table>

No.	’740 Patent Claim 1	Cisco EtherChannel
		 You can connect the switch to other devices and create backup paths by using Fast Ethernet or gigabit links or Fast EtherChannel or Gigabit EtherChannel links. Using the Hot Standby Redundancy Protocol (HSRP), you can create backup paths between Catalyst 4908G-L3 switches. Figure 1-21 illustrates three configuration examples for using the Catalyst 3500 XL switches to create the following: • Cost-effective wiring closet—A cost-effective way to connect many users to the wiring closet is to connect up to nine Catalyst 3500 XL switches through GigaStack GBIC connections. When you use a stack of Catalyst 3548 XL switches, you can connect up to 432 users. To preserve connectivity between the switches in case one switch in the stack fails, connect the bottom switch to the top switch to create a GigaStack loopback. Using gigabit GBIC modules on two of the switches, you can have redundant uplink connections to a gigabit backbone switch such as the Catalyst 3508G XL switch. If one of the redundant connections fails, the other can serve as a backup path. You can configure the stack members and the Catalyst 3508G XL switch as a switch cluster to manage them through a single IP address. • High-performance workgroup—For users who require high-speed access to network resources, use gigabit GBIC modules to connect the switches directly to a backbone switch in a star configuration. Each switch in this configuration provides users a dedicated 1-Gbps connection to network resources in the backbone. Compare this with the switches in a GigaStack configuration, where the 1-Gbps connection is shared among the switches. Using gigabit GBIC modules also provides flexibility in media and distance options: – 1000BaseSX GBIC module: Fiber connections of up to 550 m – 1000BaseLX/LH GBIC module: Fiber connections of up to 10 km – 1000BaseZX GBIC module: Fiber connections of up to 100 km • Redundant gigabit backbone—To enhance network reliability and load balancing for different VLANs and subnets, you can connect the Catalyst 3500 XL switches, again in a star configuration, to two backbone switches. If one of the backbone switches fails, the second backbone switch preserves connectivity between the switches and network resources.

No.	’740 Patent Claim 1	Cisco EtherChannel
		Catalyst 3560 Configuration Guide at 1-3  **Performance Features** • Autosensing of port speed and autonegotiation of duplex mode on all switch ports for optimizing bandwidth • Automatic-medium-dependent interface crossover (Auto-MDIX) capability on 10/100 and 10/100/1000 Mbps interfaces and on 10/100/1000 BASE-T/TX SFP module interfaces that enables the interface to automatically detect the required cable connection type (straight-through or crossover) and to configure the connection appropriately • Support for routed frames up to 1546 bytes, for frames up to 9000 bytes that are bridged in hardware, and for frames up to 2000 bytes that are bridged by software • IEEE 802.3x flow control on all ports (the switch does not send pause frames) • EtherChannel for enhanced fault tolerance and for providing up to 8 Gbps (Gigabit EtherChannel) or 800 Mbps (Fast EtherChannel) full-duplex bandwidth between switches, routers, and servers • Port Aggregation Protocol (PAgP) and Link Aggregation Control Protocol (LACP) for automatic creation of EtherChannel links • Forwarding of Layer 2 and Layer 3 packets at Gigabit line rate • Per-port storm control for preventing broadcast, multicast, and unicast storms • Port blocking on forwarding unknown Layer 2 unknown unicast, multicast, and bridged broadcast traffic • Cisco Group Management Protocol (CGMP) server support and Internet Group Management Protocol (IGMP) snooping for IGMP Versions 1, 2, and 3: – (For CGMP devices) CGMP for limiting multicast traffic to specified end stations and reducing overall network traffic – (For IGMP devices) IGMP snooping for efficiently forwarding multimedia and multicast traffic • IGMP report suppression for sending only one IGMP report per multicast router query to the multicast devices (supported only for IGMPv1 or IGMPv2 queries) • IGMP snooping querier support to configure switch to generate periodic IGMP General Query messages Catalyst 3560 Configuration Guide at Figure 32-6

No.	’740 Patent Claim 1	Cisco EtherChannel
		 **Queueing and Scheduling Overview** The switch has queues at specific points to help prevent congestion as shown in Figure 32-6. *Figure 32-6 Ingress and Egress Queue Location* Traffic Classify Policer Marker Policer Marker Policer Marker Policer Marker Ingress queues SRR Internal ring Egress queues SRR 90563 Because the total ingress bandwidth of all ports can exceed the bandwidth of the internal ring, ingress queues are located after the packet is classified, policed, and marked and before packets are forwarded into the switch fabric. Because multiple ingress ports can simultaneously send packets to an egress port and cause congestion, egress queues are located after the internal ring. Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Cisco EtherChannel System in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 1[c] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of three such references. For example, Ghosh discloses connecting the line cards to the active supervisor via the backplane using parallel interface circuitry.

No.	’740 Patent Claim 1	Cisco EtherChannel
		Ghosh at [0059] (“Line cards 803, 805, and 807 can communicate with an active supervisor 811 through interface circuitry 883, 885, and 887 and the backplane 815. According to various embodiments, each line card includes a plurality of ports that can act as either input ports or output ports for communication with external fibre channel network entities 851 and 853. The backplane 815 can provide a communi-cations channel for all traffic between line cards and super- visors. Individual line cards 803 and 807 can also be coupled to external fibre channel network entities 851 and 853 through fibre channel ports 843 and 847.”) As another example, Bruckman discloses connecting the line cards to the network using traces comprising a backplane. Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”) For example, Basso discloses coupling the blades to the communication network via link connections to the switch fabric. Basso at [0009] (“A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more ports. These ports may be connected to another one or more network devices. The switch fabric may be configured to direct incoming packets of data to particular blades where one or more of the network proces-sors in the recipient blade may be configured to process the received packets.”)

No.	’740 Patent Claim 1	Cisco EtherChannel
1[d]	at least one of said second physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction;	Cisco EtherChannel System discloses at least one of said second physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction. For example, Cisco EtherChannel System discloses full duplex fabric connection. Fully duplexed fabric connections mean that such switch fabric interconnections operate both upstream and downstream traffic. Cisco Catalyst 6500 Data Sheet at 9 (“Cisco Catalyst 6500 Series Modules Cisco Catalyst 6500 Series interface modules support the following forwarding technology and switch-fabric combinations. • Classic Interface Modules—Use the centralized Cisco Express Forwarding engine located on the supervisor engine’s PFC, connect to the 32-Gbps switching bus only, and forward packets at up to 15 mpps • CEF256 Interface Modules—Use the centralized Cisco Express Forwarding engine located on the supervisor engine’s PFC, connect to both the 256-Gbps fabric located on the supervisor engine with a single 8-Gbps full-duplex fabric connection and the 32-Gbps switching bus, and forward packets at up to 30 mpps • dCEF256 Interface Modules—Use the distributed Cisco Express Forwarding engine on the DFC (located on the interface module), connect to a 256-Gbps fabric located on the supervisor engine or a switch fabric module with 16-Gbps full-duplex fabric connections, and forward packets at up to 210 mpps • dCEF720 Interface Modules—Use the distributed Cisco Express Forwarding engine on the DFC3 (located on the interface module), connect to the 720-Gbps fabric located on the supervisor engine with dual 20-Gbps full-duplex fabric connections, and forward packets at up to 400 mpps of sustained performance”) Catalyst 3500 Installation Guide at 1-2

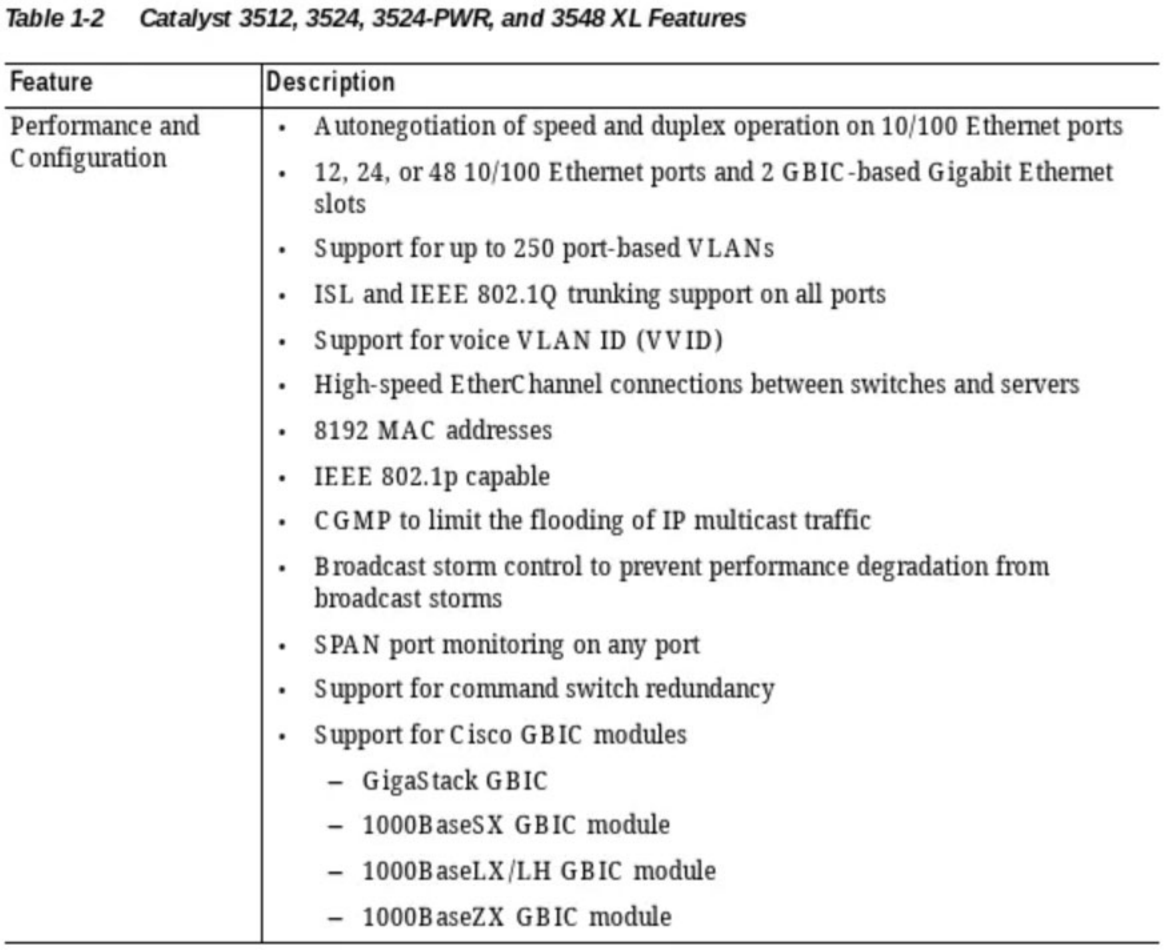
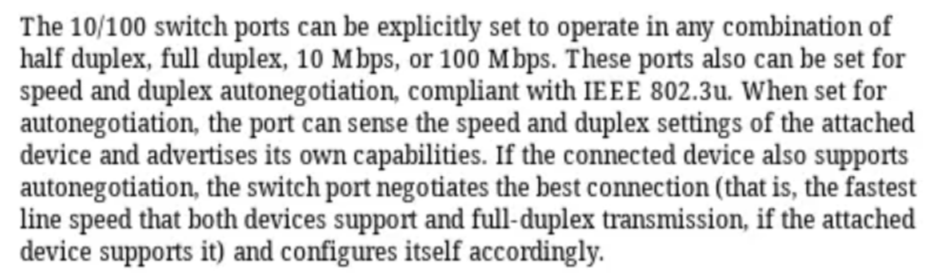
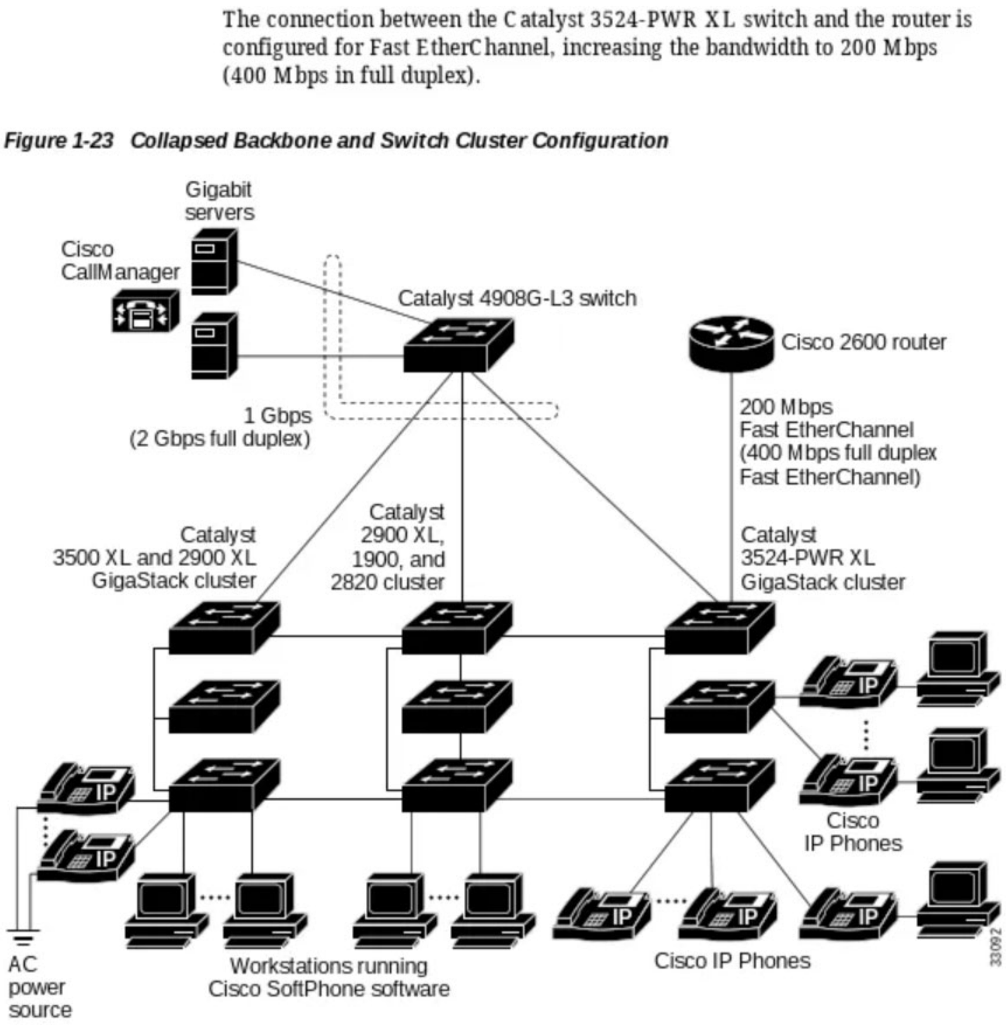
<table>
<tr><th>No.</th><th>’740 Patent Claim 1</th><th>Cisco EtherChannel</th></tr>
<tr><td></td><td></td><td>


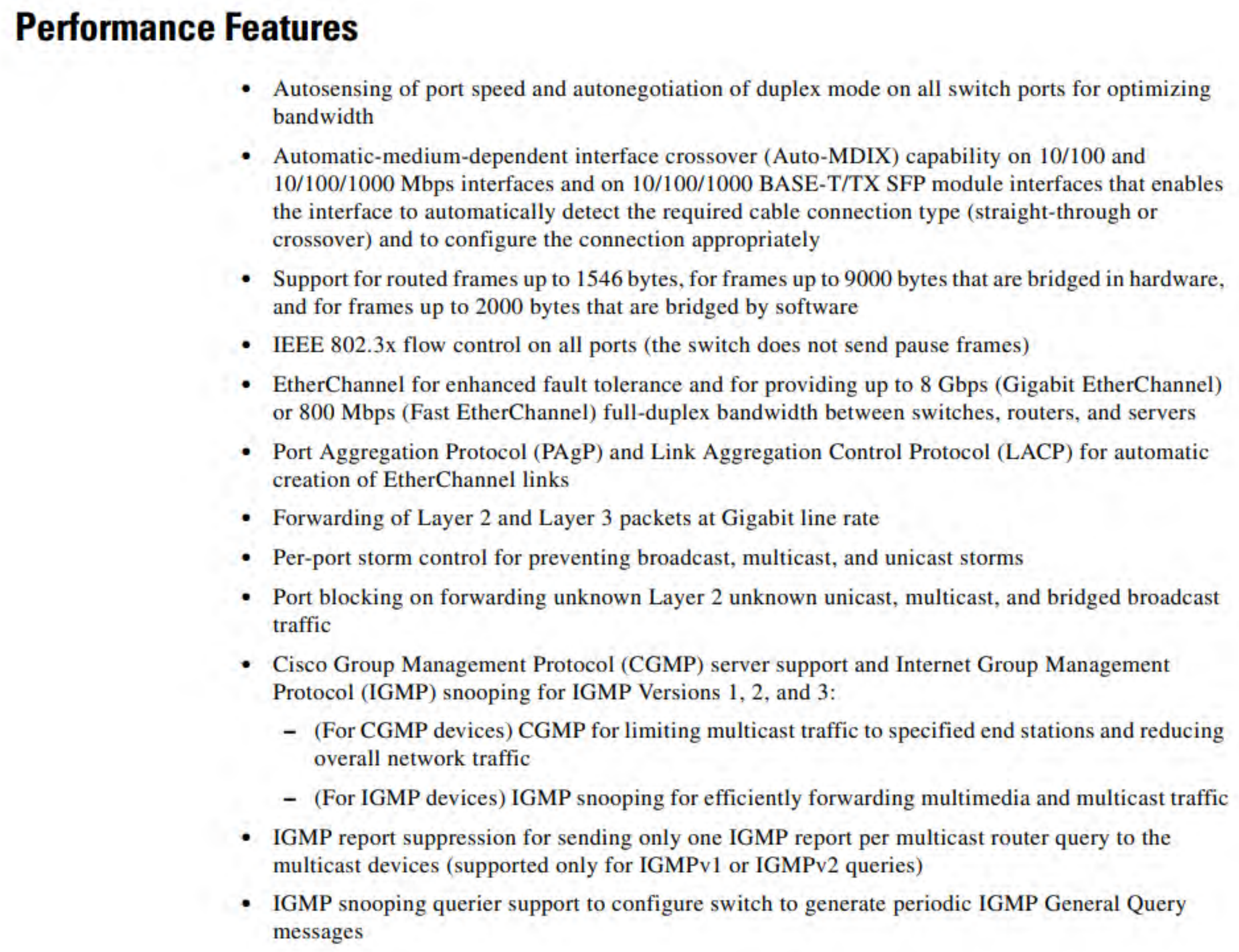
Table 1-2 Catalyst 3512, 3524, 3524-PWR, and 3548 XL Features

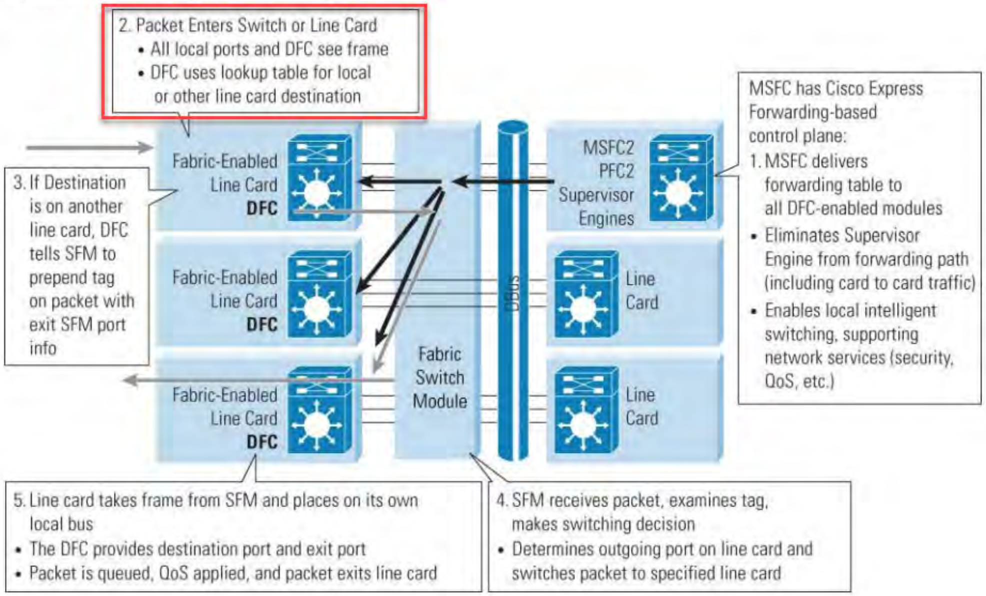
<table>
<tr><th>Feature</th><th>Description</th></tr>
<tr><td>Performance and Configuration</td><td>• Autonegotiation of speed and duplex operation on 10/100 Ethernet ports
• 12, 24, or 48 10/100 Ethernet ports and 2 GBIC-based Gigabit Ethernet slots
• Support for up to 250 port-based VLANs
• ISL and IEEE 802.1Q trunking support on all ports
• Support for voice VLAN ID (VVID)
• High-speed EtherChannel connections between switches and servers
• 8192 MAC addresses
• IEEE 802.1p capable
• CGMP to limit the flooding of IP multicast traffic
• Broadcast storm control to prevent performance degradation from broadcast storms
• SPAN port monitoring on any port
• Support for command switch redundancy
• Support for Cisco GBIC modules
– GigaStack GBIC
– 1000BaseSX GBIC module
– 1000BaseLX/LH GBIC module
– 1000BaseZX GBIC module</td></tr>
</table>

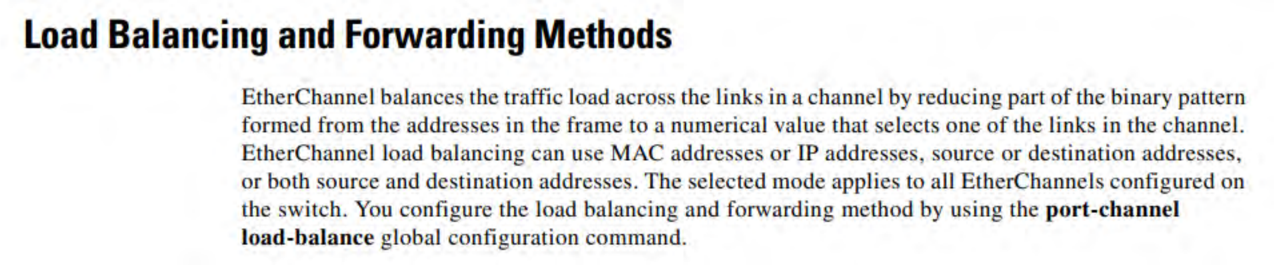
Catalyst 3500 Installation Guide at 1-8
</td></tr>
</table>

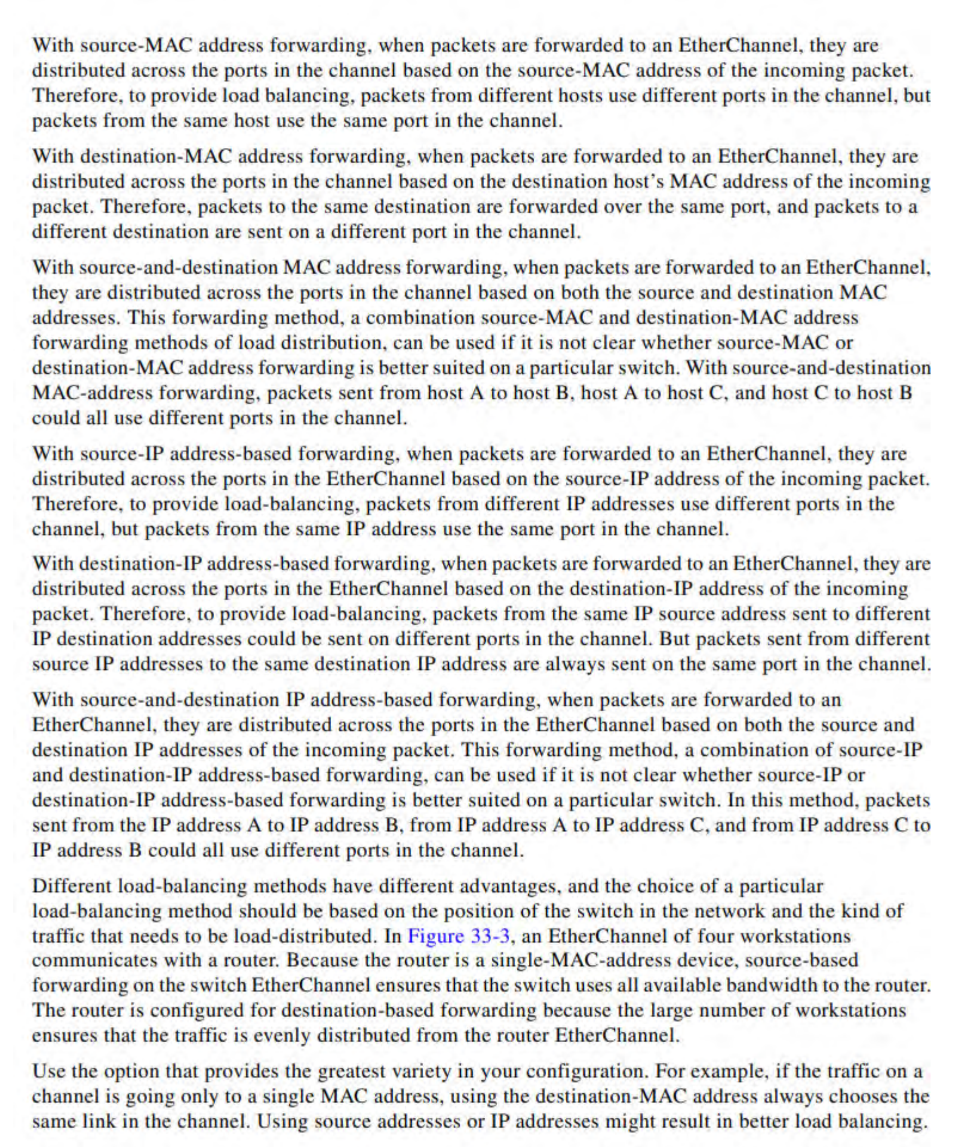
No.	’740 Patent Claim 1	Cisco EtherChannel
		 The 10/100 switch ports can be explicitly set to operate in any combination of half duplex, full duplex, 10 Mbps, or 100 Mbps. These ports also can be set for speed and duplex autonegotiation, compliant with IEEE 802.3u. When set for autonegotiation, the port can sense the speed and duplex settings of the attached device and advertises its own capabilities. If the connected device also supports autonegotiation, the switch port negotiates the best connection (that is, the fastest line speed that both devices support and full-duplex transmission, if the attached device supports it) and configures itself accordingly. Catalyst 3500 Installation Guide at Figure 1-23

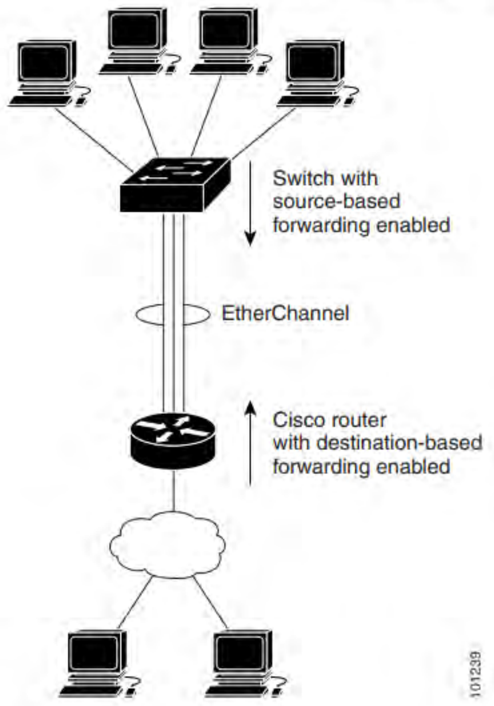
No.	’740 Patent Claim 1	Cisco EtherChannel
		 The connection between the Catalyst 3524-PWR XL switch and the router is configured for Fast EtherChannel, increasing the bandwidth to 200 Mbps (400 Mbps in full duplex). ***Figure 1-23 Collapsed Backbone and Switch Cluster Configuration*** Gigabit servers Cisco CallManager Catalyst 4908G-L3 switch Cisco 2600 router 1 Gbps (2 Gbps full duplex) 200 Mbps Fast EtherChannel (400 Mbps full duplex Fast EtherChannel) Catalyst 3500 XL and 2900 XL GigaStack cluster Catalyst 2900 XL, 1900, and 2820 cluster Catalyst 3524-PWR XL GigaStack cluster IP Cisco IP Phones AC power source Workstations running Cisco SoftPhone software Cisco IP Phones 33092 Catalyst 3560 Configuration Guide at 1-3

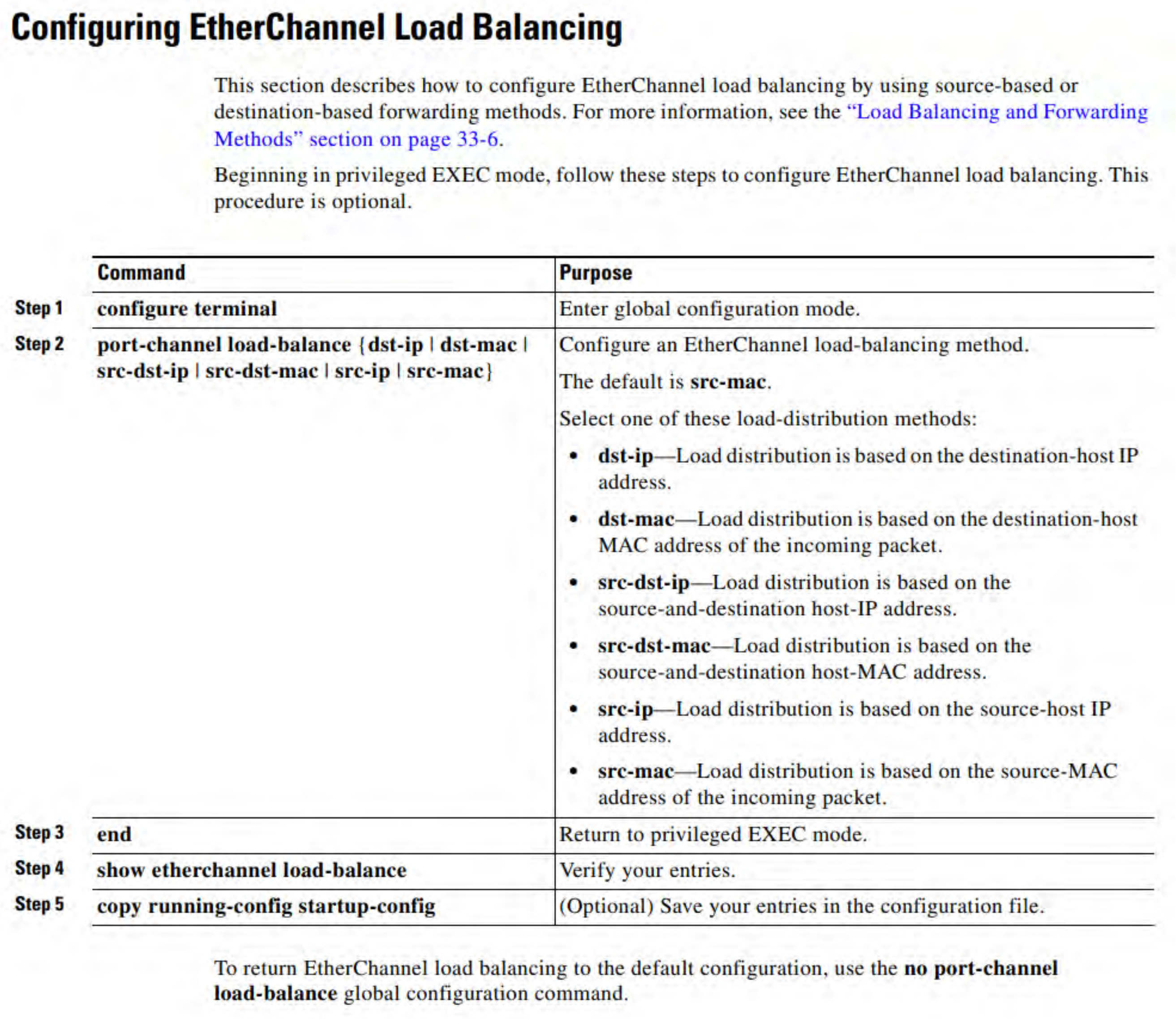
No.	’740 Patent Claim 1	Cisco EtherChannel
		 **Performance Features** • Autosensing of port speed and autonegotiation of duplex mode on all switch ports for optimizing bandwidth • Automatic-medium-dependent interface crossover (Auto-MDIX) capability on 10/100 and 10/100/1000 Mbps interfaces and on 10/100/1000 BASE-T/TX SFP module interfaces that enables the interface to automatically detect the required cable connection type (straight-through or crossover) and to configure the connection appropriately • Support for routed frames up to 1546 bytes, for frames up to 9000 bytes that are bridged in hardware, and for frames up to 2000 bytes that are bridged by software • IEEE 802.3x flow control on all ports (the switch does not send pause frames) • EtherChannel for enhanced fault tolerance and for providing up to 8 Gbps (Gigabit EtherChannel) or 800 Mbps (Fast EtherChannel) full-duplex bandwidth between switches, routers, and servers • Port Aggregation Protocol (PAgP) and Link Aggregation Control Protocol (LACP) for automatic creation of EtherChannel links • Forwarding of Layer 2 and Layer 3 packets at Gigabit line rate • Per-port storm control for preventing broadcast, multicast, and unicast storms • Port blocking on forwarding unknown Layer 2 unknown unicast, multicast, and bridged broadcast traffic • Cisco Group Management Protocol (CGMP) server support and Internet Group Management Protocol (IGMP) snooping for IGMP Versions 1, 2, and 3: – (For CGMP devices) CGMP for limiting multicast traffic to specified end stations and reducing overall network traffic – (For IGMP devices) IGMP snooping for efficiently forwarding multimedia and multicast traffic • IGMP report suppression for sending only one IGMP report per multicast router query to the multicast devices (supported only for IGMPv1 or IGMPv2 queries) • IGMP snooping querier support to configure switch to generate periodic IGMP General Query messages
1[e]	receiving a data frame having frame attributes sent between the	Cisco EtherChannel System discloses receiving a data frame having frame attributes sent between the communication network and the network node. For example, Cisco EtherChannel System discloses the router or switch routing data frames and packets with specific information including header and payload information such as source and destination addresses.

No.	’740 Patent Claim 1	Cisco EtherChannel
	communication network and the network node:	Cisco Catalyst 6500 Data Sheet at Figure 6 (annotation added) **Figure 6.** Distributed Cisco Express Forwarding Packet Flow  2. Packet Enters Switch or Line Card • All local ports and DFC see frame • DFC uses lookup table for local or other line card destination MSFC has Cisco Express Forwarding-based control plane: 1. MSFC delivers forwarding table to all DFC-enabled modules • Eliminates Supervisor Engine from forwarding path (including card to card traffic) • Enables local intelligent switching, supporting network services (security, QoS, etc.) 3. If Destination is on another line card, DFC tells SFM to prepend tag on packet with exit SFM port info Fabric-Enabled Line Card DFC Fabric-Enabled Line Card DFC Fabric-Enabled Line Card DFC Fabric Switch Module DBus MSFC2 PFC2 Supervisor Engines Line Card Line Card 5. Line card takes frame from SFM and places on its own local bus • The DFC provides destination port and exit port • Packet is queued, QoS applied, and packet exits line card 4. SFM receives packet, examines tag, makes switching decision • Determines outgoing port on line card and switches packet to specified line card Cisco Catalyst 6500 Data Sheet at 13 (“How Cisco Express Forwarding Works Cisco Express Forwarding is a Layer 3 technology that provides increased forwarding scalability and performance to manage the many short-duration traffic flows common in today’s enterprise and service provider networks. To meet the needs of environments managing large amounts of short-flow, Web-based, or highly interactive types of traffic, Cisco Express Forwarding forwards all packets in hardware, and maintains its forwarding rate independent of the number of flows going though the switch. On the Cisco Catalyst 6500 Series, the Cisco Express Forwarding Layer 3 forwarding engine is located centrally on the supervisor engine’s policy feature card (PFC2 or PFC3)—the same device that performs hardware-based Layer 2 and Layer 3 forwarding, access control list (ACL) checking, QoS policing and marking, and NetFlow statistics gathering.

No.	’740 Patent Claim 1	Cisco EtherChannel
		Using the routing table that Cisco IOS Software builds to define configured interfaces and routing protocols, the Cisco Express Forwarding architecture creates Cisco Express Forwarding tables and downloads them into the hardware forwarding engine before any user traffic is sent through the switch. The Cisco Express Forwarding architecture places only the routing prefixes in its Cisco Express Forwarding tables—the only information it requires to make the Layer 3 forwarding decisions—relying on the routing protocols to do route selection. By performing a simple Cisco Express Forwarding table lookup, the switch forwards packets at wire rate, independent of the number of flows transiting the switch.”) Catalyst 3560 Configuration Guide at 33-6 – 33-8  **Load Balancing and Forwarding Methods** EtherChannel balances the traffic load across the links in a channel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel. EtherChannel load balancing can use MAC addresses or IP addresses, source or destination addresses, or both source and destination addresses. The selected mode applies to all EtherChannels configured on the switch. You configure the load balancing and forwarding method by using the **port-channel load-balance** global configuration command.

No.	’740 Patent Claim 1	Cisco EtherChannel
		 With source-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the source-MAC address of the incoming packet. Therefore, to provide load balancing, packets from different hosts use different ports in the channel, but packets from the same host use the same port in the channel. With destination-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the destination host’s MAC address of the incoming packet. Therefore, packets to the same destination are forwarded over the same port, and packets to a different destination are sent on a different port in the channel. With source-and-destination MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on both the source and destination MAC addresses. This forwarding method, a combination source-MAC and destination-MAC address forwarding methods of load distribution, can be used if it is not clear whether source-MAC or destination-MAC address forwarding is better suited on a particular switch. With source-and-destination MAC-address forwarding, packets sent from host A to host B, host A to host C, and host C to host B could all use different ports in the channel. With source-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the source-IP address of the incoming packet. Therefore, to provide load-balancing, packets from different IP addresses use different ports in the channel, but packets from the same IP address use the same port in the channel. With destination-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the destination-IP address of the incoming packet. Therefore, to provide load-balancing, packets from the same IP source address sent to different IP destination addresses could be sent on different ports in the channel. But packets sent from different source IP addresses to the same destination IP address are always sent on the same port in the channel. With source-and-destination IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on both the source and destination IP addresses of the incoming packet. This forwarding method, a combination of source-IP and destination-IP address-based forwarding, can be used if it is not clear whether source-IP or destination-IP address-based forwarding is better suited on a particular switch. In this method, packets sent from the IP address A to IP address B, from IP address A to IP address C, and from IP address C to IP address B could all use different ports in the channel. Different load-balancing methods have different advantages, and the choice of a particular load-balancing method should be based on the position of the switch in the network and the kind of traffic that needs to be load-distributed. In Figure 33-3, an EtherChannel of four workstations communicates with a router. Because the router is a single-MAC-address device, source-based forwarding on the switch EtherChannel ensures that the switch uses all available bandwidth to the router. The router is configured for destination-based forwarding because the large number of workstations ensures that the traffic is evenly distributed from the router EtherChannel. Use the option that provides the greatest variety in your configuration. For example, if the traffic on a channel is going only to a single MAC address, using the destination-MAC address always chooses the same link in the channel. Using source addresses or IP addresses might result in better load balancing.

No.	’740 Patent Claim 1	Cisco EtherChannel
		Figure 33-3 Load Distribution and Forwarding Methods  Switch with source-based forwarding enabled EtherChannel Cisco router with destination-based forwarding enabled 101239 Catalyst 3560 Configuration Guide at 33-16

No.	’740 Patent Claim 1	Cisco EtherChannel
		

Configuring EtherChannel Load Balancing

This section describes how to configure EtherChannel load balancing by using source-based or destination-based forwarding methods. For more information, see the “Load Balancing and Forwarding Methods” section on page 33-6.

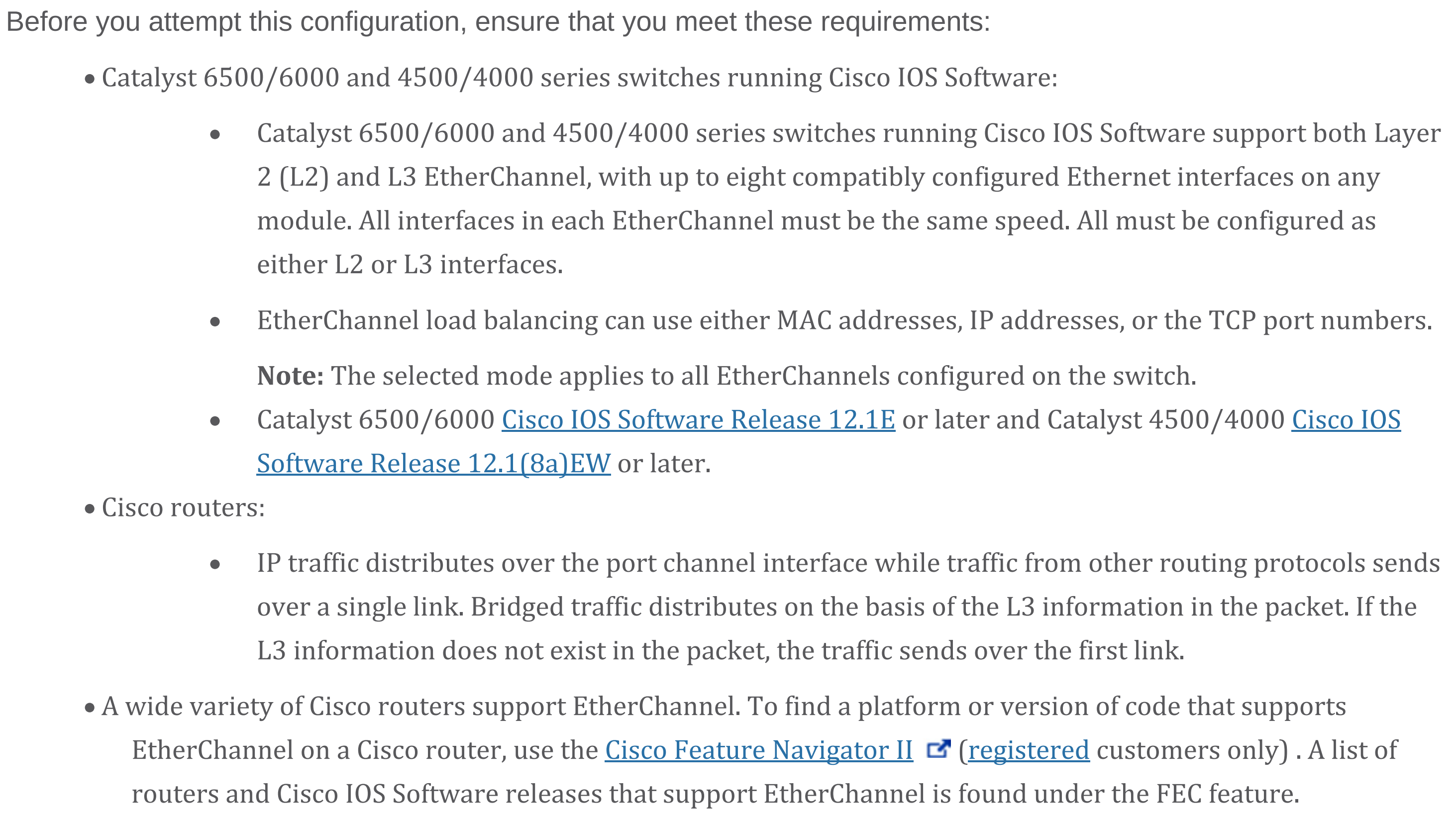
Beginning in privileged EXEC mode, follow these steps to configure EtherChannel load balancing. This procedure is optional.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**port-channel load-balance** {**dst-ip** | **dst-mac** | **src-dst-ip** | **src-dst-mac** | **src-ip** | **src-mac**}	Configure an EtherChannel load-balancing method. The default is **src-mac**. Select one of these load-distribution methods: • **dst-ip**—Load distribution is based on the destination-host IP address. • **dst-mac**—Load distribution is based on the destination-host MAC address of the incoming packet. • **src-dst-ip**—Load distribution is based on the source-and-destination host-IP address. • **src-dst-mac**—Load distribution is based on the source-and-destination host-MAC address. • **src-ip**—Load distribution is based on the source-host IP address. • **src-mac**—Load distribution is based on the source-MAC address of the incoming packet.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show etherchannel load-balance**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

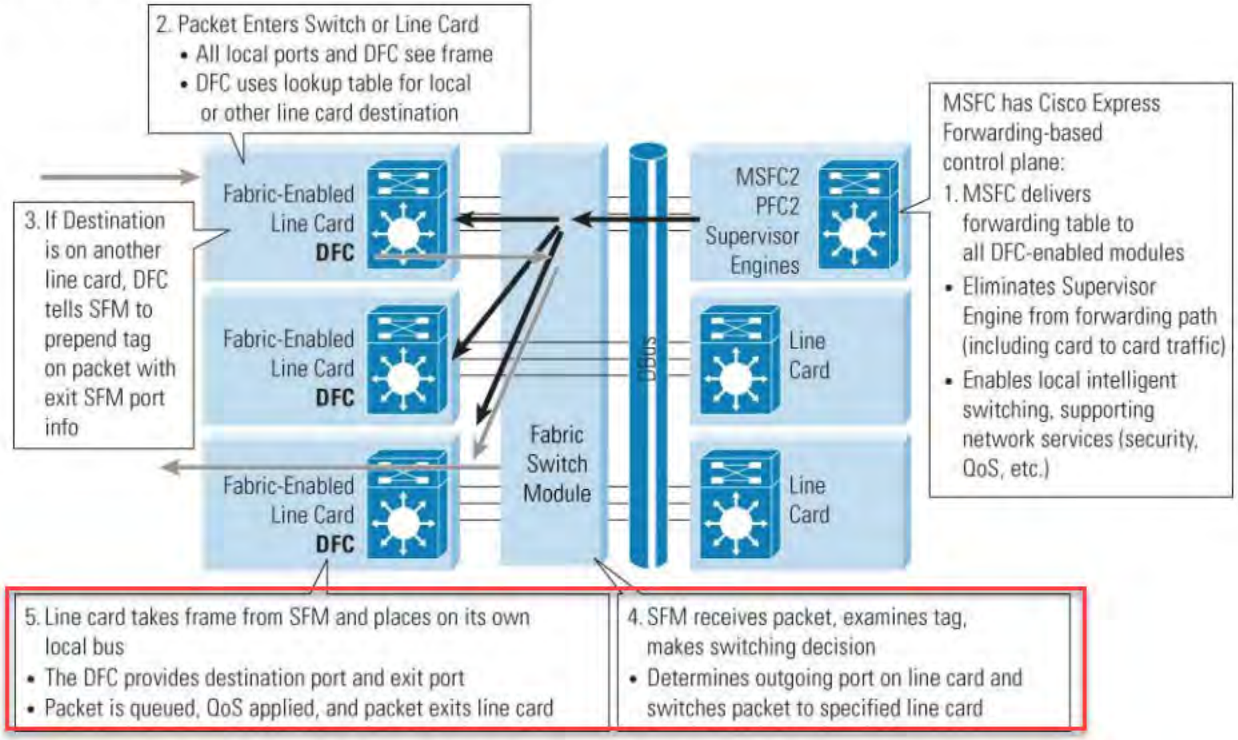
To return EtherChannel load balancing to the default configuration, use the **no port-channel load-balance** global configuration command.

No.	’740 Patent Claim 1	Cisco EtherChannel
		Layer 2 EtherChannel **Catalyst 2950** Catalyst 2950 switches only support 802.1Q trunking and do not support ISL trunking. Catalyst 2950 switches support DTP and PAgP dynamic trunking and channel negotiation with Cisco IOS Software Release 12.1 releases and static modes only with Cisco IOS Software Release 12.0 releases. EtherChannel load balancing can use either source-MAC or destination-MAC address forwarding. You can configure the load balancing method by issuing the **port-channel load-balance** global configuration command. These switches support up to eight switch ports per channel. **Catalyst 6500 That Runs Cisco IOS Software** Catalyst 6500 switches that run Cisco IOS Software support L2 (switchport) and Layer 3 (L3) (routed port) EtherChannel configurations. A Catalyst 6500/6000 series switch supports a maximum of 64 EtherChannels (256 with Cisco IOS Software Release 12.1(2)E and earlier). You can form an EtherChannel with up to eight compatibly configured LAN ports on any module in a Catalyst 6000 series switch, with the exception of Digital Feature Card (DFC)-equipped modules (such as WS-X6816 and so on) which currently allow an L2 channel only using ports on the same DFC module. However, an L3 channel can be configured across different DFC-equipped modules. This limitation has been removed in Catalyst 6500/6000 Cisco IOS Software Release 12.1(11b)EX and later. This document configures an L2 EtherChannel. The Catalyst 6500/6000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP addresses, or Layer 4 (L4) port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. Catalyst 6500/6000 switches support both ISL and 802.1Q trunking encapsulations and DTP. Detailed information on port capabilities is available by issuing the **show interface** *interface_id* **capabilities** command.

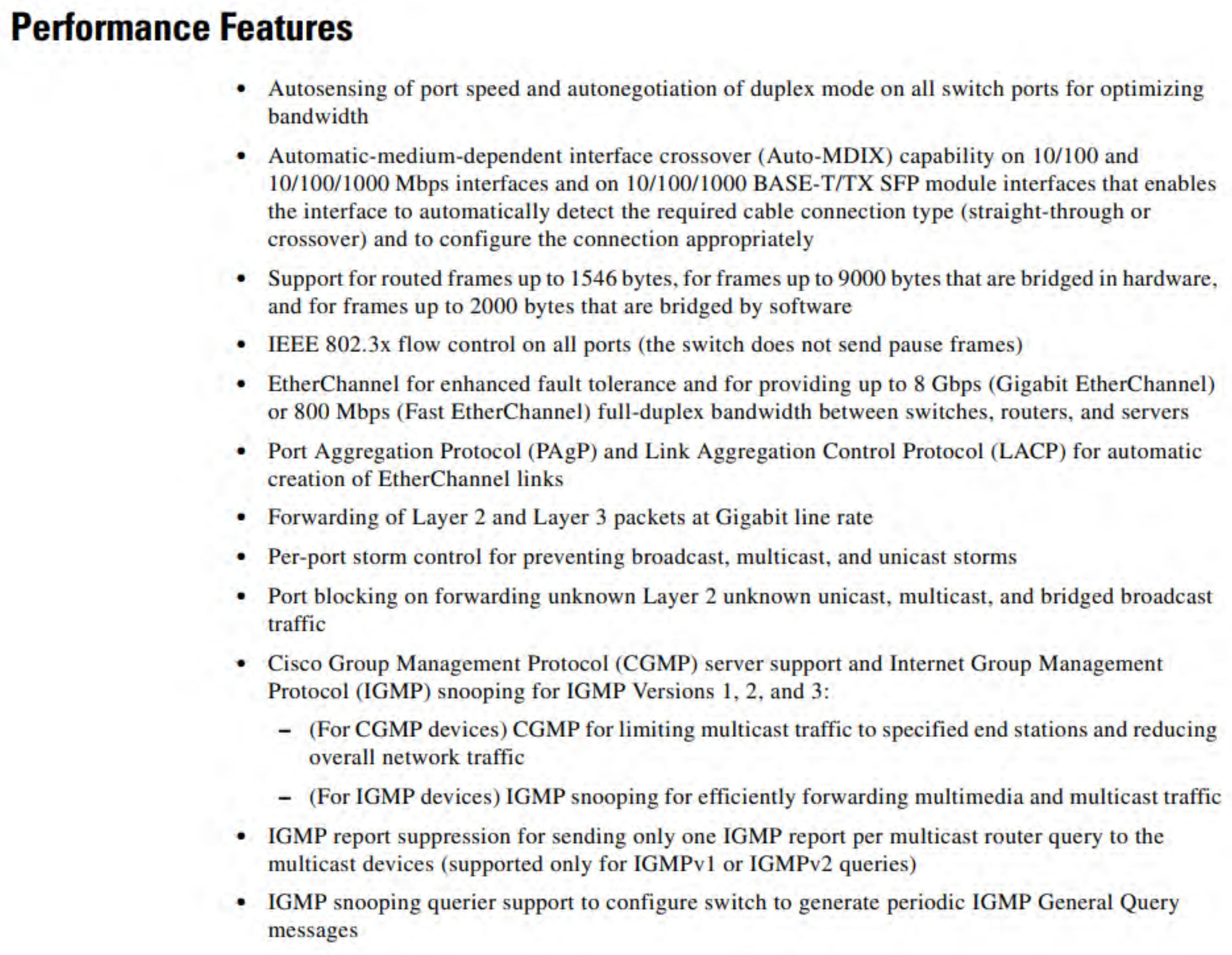
No.	’740 Patent Claim 1	Cisco EtherChannel
		Catalyst 4000 That Runs Cisco IOS Software Catalyst 4000 switches that run Cisco IOS Software (with Supervisor Engine III and IV) support L2 (switchport) and L3 (routed port) EtherChannel configurations. A Catalyst 4000 series switch supports a maximum of 64 EtherChannels. You can form an EtherChannel with up to eight compatibly configured Ethernet interfaces on any module, and across modules in a Catalyst 4000 series switch. All interfaces in each EtherChannel must be the same speed and must all be configured as either L2 or L3 interfaces. The Catalyst 4000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP address, or L4 port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. The Catalyst 4000 that runs Cisco IOS Software supports ISL and 802.1Q trunking encapsulations and DTP. ISL is not available on certain modules. For a complete list of such modules, refer to the *Understanding VLAN Trunks* section of Configuring Layer 2 Ethernet Interfaces. In a future software release, detailed information on port capabilities will be available by issuing the **show interface capabilities** command. Currently this command is not available. Configuring EtherChannel at 1 -2 **Introduction** This sample configuration demonstrates how to set up a Layer 3 (L3) EtherChannel, without VLAN trunking, between a Cisco router and a Cisco Catalyst 6500 switch running Cisco IOS® System Software. EtherChannel can be called Fast EtherChannel (FEC) or Gigabit EtherChannel (GEC); the term depends on the speed of the interfaces or ports you use to form the EtherChannel. In this example, two Fast Ethernet ports from a Cisco router and a Catalyst 6500 switch have been bundled into a FEC. Throughout this document, the terms FEC, GEC, port channel, channel, and port group all refer to EtherChannel.

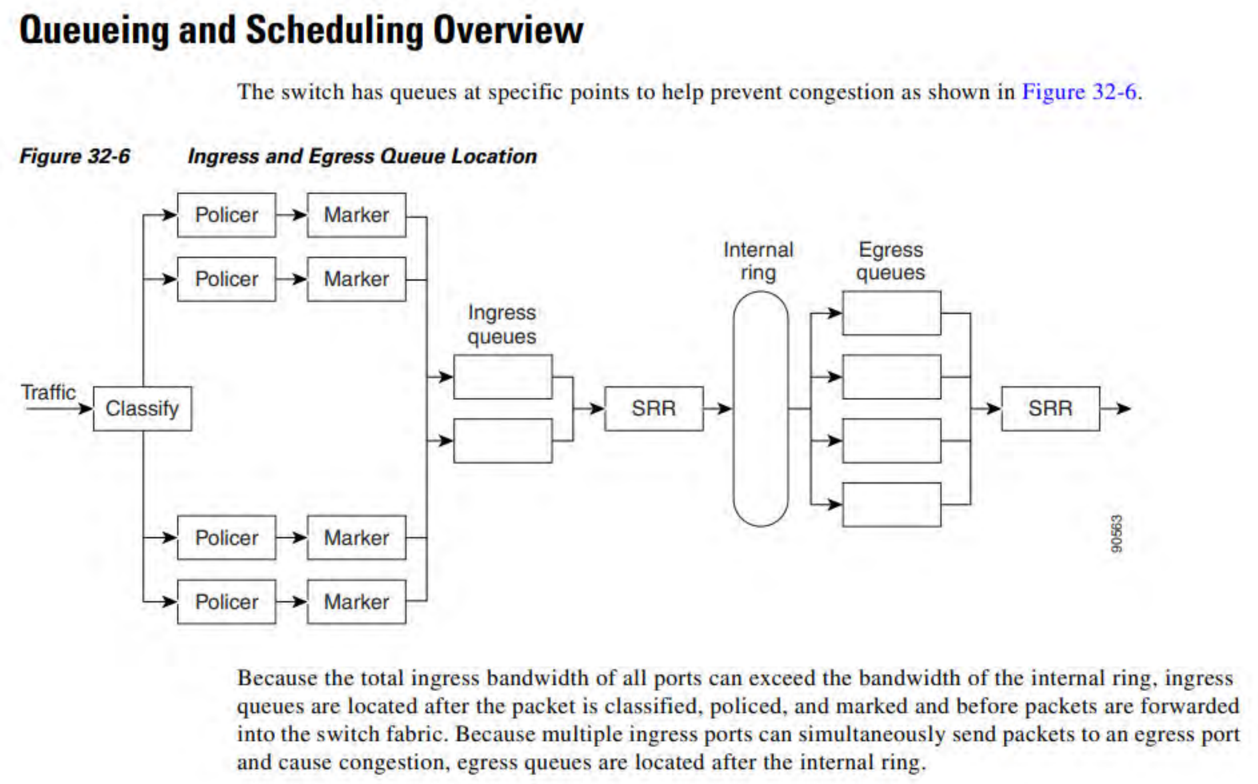
No.	’740 Patent Claim 1	Cisco EtherChannel
		 Before you attempt this configuration, ensure that you meet these requirements: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software support both Layer 2 (L2) and L3 EtherChannel, with up to eight compatibly configured Ethernet interfaces on any module. All interfaces in each EtherChannel must be the same speed. All must be configured as either L2 or L3 interfaces. • EtherChannel load balancing can use either MAC addresses, IP addresses, or the TCP port numbers. **Note:** The selected mode applies to all EtherChannels configured on the switch. • Catalyst 6500/6000 Cisco IOS Software Release 12.1E or later and Catalyst 4500/4000 Cisco IOS Software Release 12.1(8a)EW or later. • Cisco routers: • IP traffic distributes over the port channel interface while traffic from other routing protocols sends over a single link. Bridged traffic distributes on the basis of the L3 information in the packet. If the L3 information does not exist in the packet, the traffic sends over the first link. • A wide variety of Cisco routers support EtherChannel. To find a platform or version of code that supports EtherChannel on a Cisco router, use the Cisco Feature Navigator II (registered customers only) . A list of routers and Cisco IOS Software releases that support EtherChannel is found under the FEC feature.
1[f]:	selecting, in a single computation based on at least one of the frame attributes, a first physical	Cisco EtherChannel System discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. For example, Cisco EtherChannel System discloses forwarding data packets over selected ports and switch fabric interconnections based on the packet’s specific information. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Cisco EtherChannel System is found to not meet this limitation, selecting, in a single computation based on at least one of the frame

No.	’740 Patent Claim 1	Cisco EtherChannel
	link out of the first group and a second physical link out of the second group; and	attributes, a first physical link out of the first group and a second physical link out of the second group would have been obvious to a person having ordinary skill in the art, as explained below. Cisco Catalyst 6500 Data Sheet at 13 (“**Switch Fabric Modules** Designed to support distributed forwarding for interface modules with that capability, the Cisco Catalyst 6500 Series Switch Fabric Module (SFM or SFM2), in combination with the Cisco Catalyst 6000 Multilayer Switch Feature Card (MSFC2) and Cisco distributed forwarding cards (DFCs) on interface modules, increases available system bandwidth from 32 to 256 Gbps. The SFM or SFM2 supports the Cisco Catalyst 6500 CEF256 and dCEF256 interface modules. Designed to support new interface modules with 720-Gbps forwarding capabilities, the switch fabric onboard the Cisco Catalyst 6500 Series Supervisor Engine 720 increases available bandwidth to 720 Gbps and provides packet-forwarding rates up to 400 mpps. By using automatic sensing and negotiation, the switch fabric is fully interoperable with the 8- and 16-Gbps switch-fabric interconnections used by the CEF256 and dCEF256 interface modules. When a CEF256 or dCEF256 interface module is detected, the switch fabric will automatically connect those modules by offering 8 to 16 Gbps of bandwidth to each module, as applicable.”) Cisco Catalyst 6500 Data Sheet at 14 (“Using the same ASIC engine design as the central PFCx, distributed forwarding cards (DFCs) located on the interface modules forward packets between two ports, directly or across the switch fabric, without involving the supervisor engine. With the DFC, each interface module has a dedicated forwarding engine complete with the full forwarding tables. Distributed Cisco Express Forwarding (Figure 6) works like this: • As in standard Cisco Express Forwarding, the central PFCx located on the supervisor engine and the DFC engines located on the interface modules are loaded with the same Cisco Express Forwarding information derived from the forwarding table before any user traffic arrives at the switch. • As a packet arrives at an interface module, its DFC engine inspects the packet and uses the information in the Cisco Express Forwarding table (including Layer 2, Layer 3, ACLs, and QoS) to make a completely hardware-based forwarding decision for that packet.

No.	’740 Patent Claim 1	Cisco EtherChannel
		• The Distributed Cisco Express Forwarding engine manages all hardware-based forwarding for traffic on that module, including Layer 2 and Layer 3 forwarding, ACLs, QoS policing and marking, and NetFlow. • Because the DFCs make all the switching decisions locally, the central PFCx can dedicate more hardware-forwarding resources to any modules not equipped with a DFC.”) Cisco Catalyst 6500 Data Sheet at Figure 6 **Figure 6.** Distributed Cisco Express Forwarding Packet Flow  Fabric-Enabled Line Card DFC Fabric-Enabled Line Card DFC Fabric-Enabled Line Card DFC Fabric Switch Module DBus MSFC2 PFC2 Supervisor Engines Line Card Line Card 2. Packet Enters Switch or Line Card • All local ports and DFC see frame • DFC uses lookup table for local or other line card destination 3. If Destination is on another line card, DFC tells SFM to prepend tag on packet with exit SFM port info MSFC has Cisco Express Forwarding-based control plane: 1. MSFC delivers forwarding table to all DFC-enabled modules • Eliminates Supervisor Engine from forwarding path (including card to card traffic) • Enables local intelligent switching, supporting network services (security, QoS, etc.) 5. Line card takes frame from SFM and places on its own local bus • The DFC provides destination port and exit port • Packet is queued, QoS applied, and packet exits line card 4. SFM receives packet, examines tag, makes switching decision • Determines outgoing port on line card and switches packet to specified line card Cisco Catalyst 6500 Data Sheet at 13 (“How Cisco Express Forwarding Works

No.	’740 Patent Claim 1	Cisco EtherChannel
		Cisco Express Forwarding is a Layer 3 technology that provides increased forwarding scalability and performance to manage the many short-duration traffic flows common in today’s enterprise and service provider networks. To meet the needs of environments managing large amounts of short-flow, Web-based, or highly interactive types of traffic, Cisco Express Forwarding forwards all packets in hardware, and maintains its forwarding rate independent of the number of flows going though the switch. On the Cisco Catalyst 6500 Series, the Cisco Express Forwarding Layer 3 forwarding engine is located centrally on the supervisor engine’s policy feature card (PFC2 or PFC3)—the same device that performs hardware-based Layer 2 and Layer 3 forwarding, access control list (ACL) checking, QoS policing and marking, and NetFlow statistics gathering. Using the routing table that Cisco IOS Software builds to define configured interfaces and routing protocols, the Cisco Express Forwarding architecture creates Cisco Express Forwarding tables and downloads them into the hardware forwarding engine before any user traffic is sent through the switch. The Cisco Express Forwarding architecture places only the routing prefixes in its Cisco Express Forwarding tables—the only information it requires to make the Layer 3 forwarding decisions—relying on the routing protocols to do route selection. By performing a simple Cisco Express Forwarding table lookup, the switch forwards packets at wire rate, independent of the number of flows transiting the switch.”) Catalyst 3560 Configuration Guide at 1-3

No.	’740 Patent Claim 1	Cisco EtherChannel
		 **Performance Features** • Autosensing of port speed and autonegotiation of duplex mode on all switch ports for optimizing bandwidth • Automatic-medium-dependent interface crossover (Auto-MDIX) capability on 10/100 and 10/100/1000 Mbps interfaces and on 10/100/1000 BASE-T/TX SFP module interfaces that enables the interface to automatically detect the required cable connection type (straight-through or crossover) and to configure the connection appropriately • Support for routed frames up to 1546 bytes, for frames up to 9000 bytes that are bridged in hardware, and for frames up to 2000 bytes that are bridged by software • IEEE 802.3x flow control on all ports (the switch does not send pause frames) • EtherChannel for enhanced fault tolerance and for providing up to 8 Gbps (Gigabit EtherChannel) or 800 Mbps (Fast EtherChannel) full-duplex bandwidth between switches, routers, and servers • Port Aggregation Protocol (PAgP) and Link Aggregation Control Protocol (LACP) for automatic creation of EtherChannel links • Forwarding of Layer 2 and Layer 3 packets at Gigabit line rate • Per-port storm control for preventing broadcast, multicast, and unicast storms • Port blocking on forwarding unknown Layer 2 unknown unicast, multicast, and bridged broadcast traffic • Cisco Group Management Protocol (CGMP) server support and Internet Group Management Protocol (IGMP) snooping for IGMP Versions 1, 2, and 3: – (For CGMP devices) CGMP for limiting multicast traffic to specified end stations and reducing overall network traffic – (For IGMP devices) IGMP snooping for efficiently forwarding multimedia and multicast traffic • IGMP report suppression for sending only one IGMP report per multicast router query to the multicast devices (supported only for IGMPv1 or IGMPv2 queries) • IGMP snooping querier support to configure switch to generate periodic IGMP General Query messages Catalyst 3560 Configuration Guide at Figure 32-6

<table>
<tr><th>No.</th><th>’740 Patent Claim 1</th><th>Cisco EtherChannel</th></tr>
<tr><td></td><td></td><td>


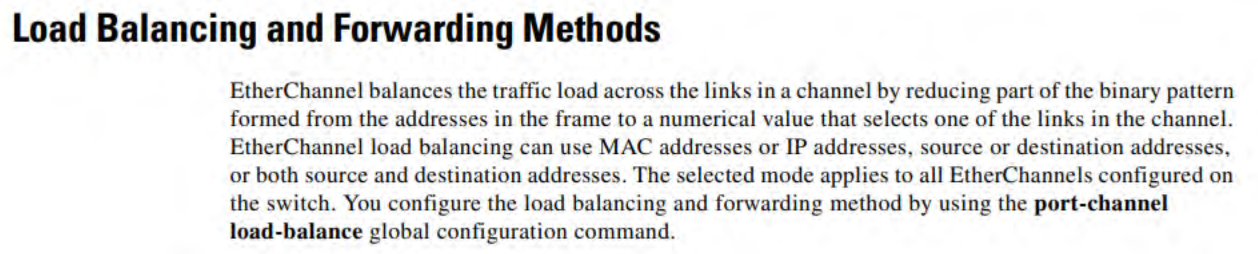
Queueing and Scheduling Overview

The switch has queues at specific points to help prevent congestion as shown in Figure 32-6.

Figure 32-6 Ingress and Egress Queue Location

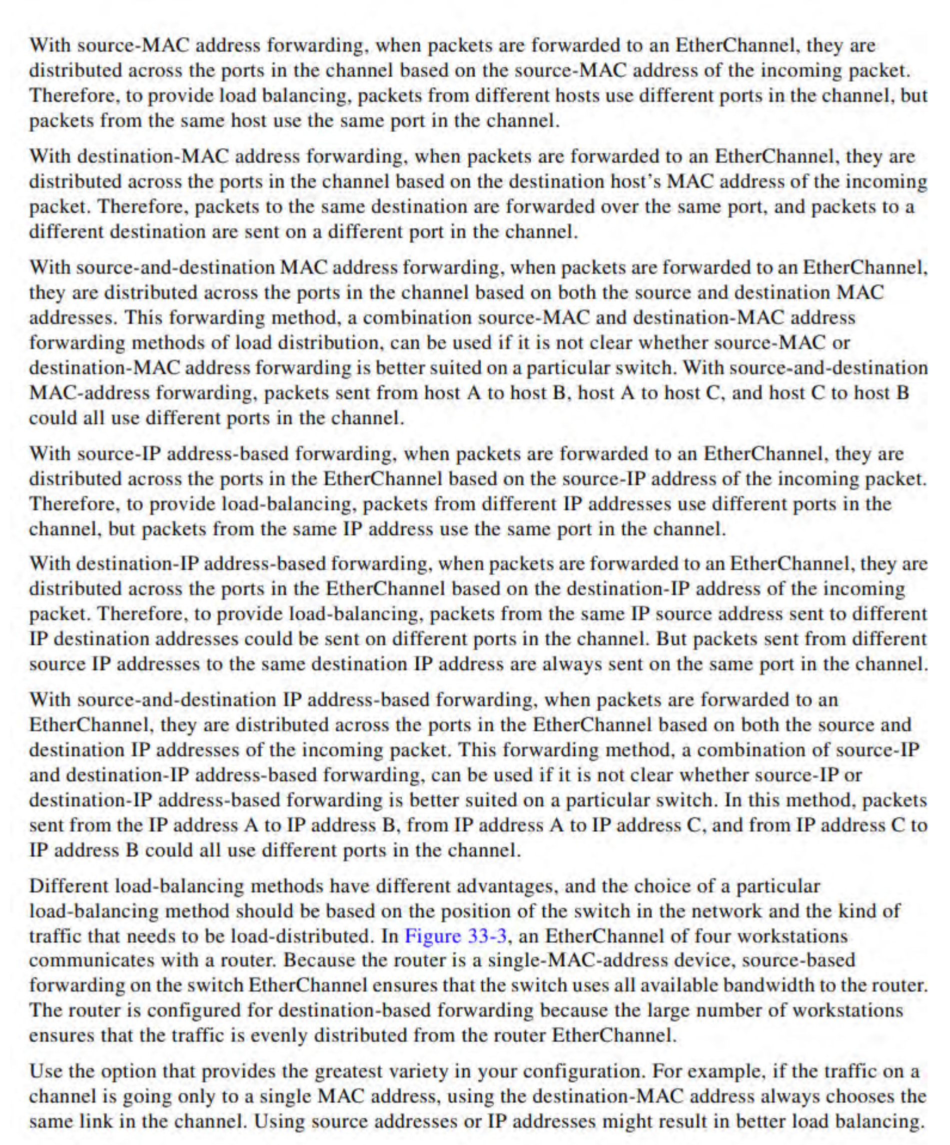
Because the total ingress bandwidth of all ports can exceed the bandwidth of the internal ring, ingress queues are located after the packet is classified, policed, and marked and before packets are forwarded into the switch fabric. Because multiple ingress ports can simultaneously send packets to an egress port and cause congestion, egress queues are located after the internal ring.

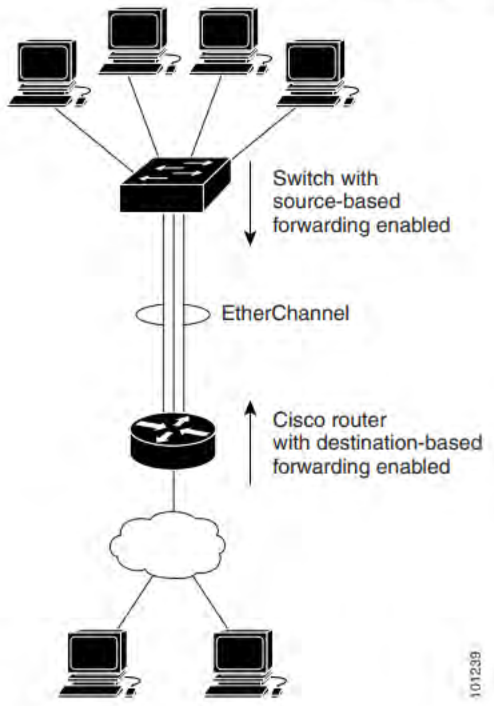
Catalyst 3560 Configuration Guide at 33-6 – 33-8

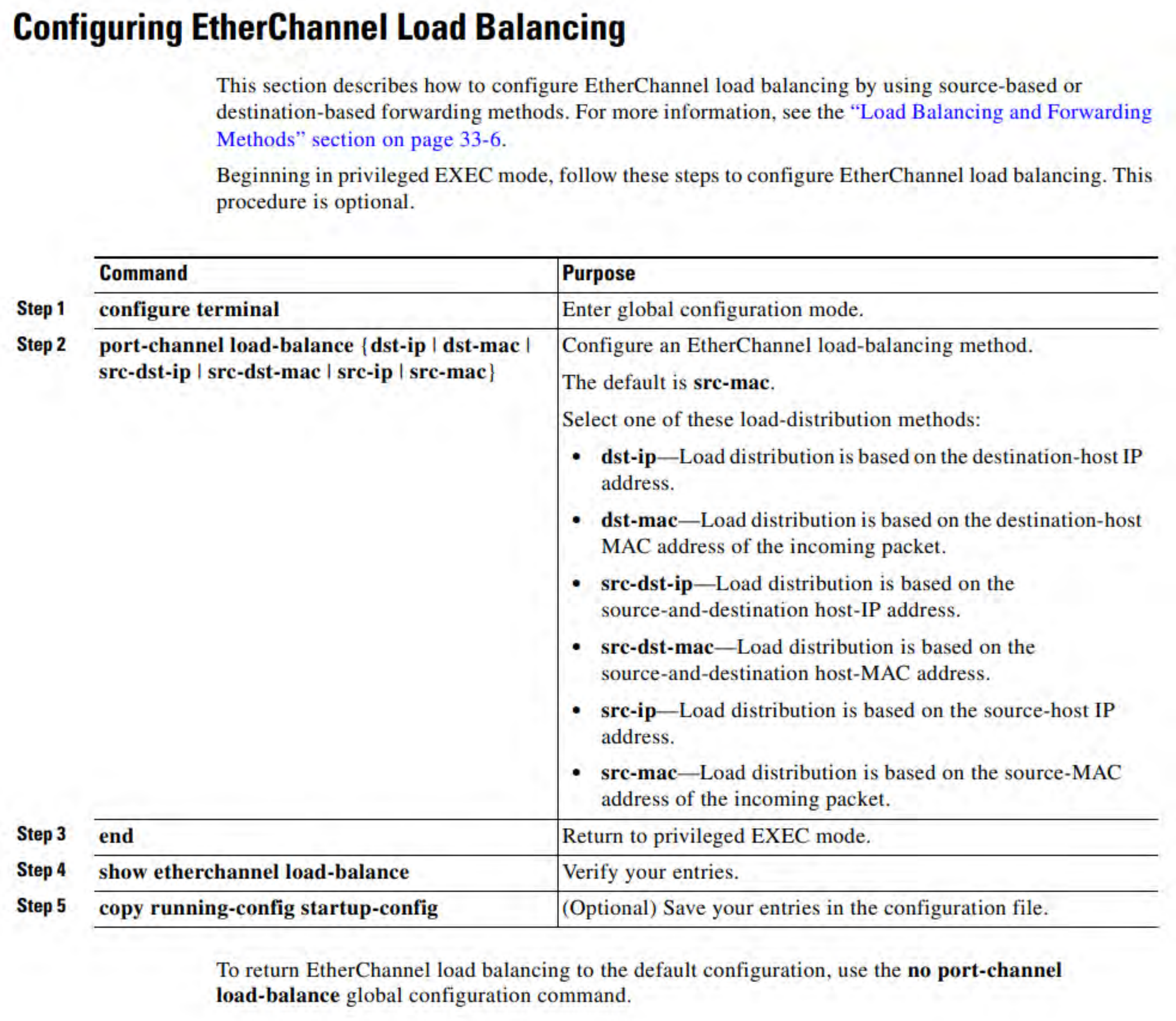


Load Balancing and Forwarding Methods

EtherChannel balances the traffic load across the links in a channel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel. EtherChannel load balancing can use MAC addresses or IP addresses, source or destination addresses, or both source and destination addresses. The selected mode applies to all EtherChannels configured on the switch. You configure the load balancing and forwarding method by using the **port-channel load-balance** global configuration command.
</td></tr>
</table>

No.	’740 Patent Claim 1	Cisco EtherChannel
		 With source-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the source-MAC address of the incoming packet. Therefore, to provide load balancing, packets from different hosts use different ports in the channel, but packets from the same host use the same port in the channel. With destination-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the destination host’s MAC address of the incoming packet. Therefore, packets to the same destination are forwarded over the same port, and packets to a different destination are sent on a different port in the channel. With source-and-destination MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on both the source and destination MAC addresses. This forwarding method, a combination source-MAC and destination-MAC address forwarding methods of load distribution, can be used if it is not clear whether source-MAC or destination-MAC address forwarding is better suited on a particular switch. With source-and-destination MAC-address forwarding, packets sent from host A to host B, host A to host C, and host C to host B could all use different ports in the channel. With source-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the source-IP address of the incoming packet. Therefore, to provide load-balancing, packets from different IP addresses use different ports in the channel, but packets from the same IP address use the same port in the channel. With destination-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the destination-IP address of the incoming packet. Therefore, to provide load-balancing, packets from the same IP source address sent to different IP destination addresses could be sent on different ports in the channel. But packets sent from different source IP addresses to the same destination IP address are always sent on the same port in the channel. With source-and-destination IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on both the source and destination IP addresses of the incoming packet. This forwarding method, a combination of source-IP and destination-IP address-based forwarding, can be used if it is not clear whether source-IP or destination-IP address-based forwarding is better suited on a particular switch. In this method, packets sent from the IP address A to IP address B, from IP address A to IP address C, and from IP address C to IP address B could all use different ports in the channel. Different load-balancing methods have different advantages, and the choice of a particular load-balancing method should be based on the position of the switch in the network and the kind of traffic that needs to be load-distributed. In Figure 33-3, an EtherChannel of four workstations communicates with a router. Because the router is a single-MAC-address device, source-based forwarding on the switch EtherChannel ensures that the switch uses all available bandwidth to the router. The router is configured for destination-based forwarding because the large number of workstations ensures that the traffic is evenly distributed from the router EtherChannel. Use the option that provides the greatest variety in your configuration. For example, if the traffic on a channel is going only to a single MAC address, using the destination-MAC address always chooses the same link in the channel. Using source addresses or IP addresses might result in better load balancing.

No.	’740 Patent Claim 1	Cisco EtherChannel
		Figure 33-3 Load Distribution and Forwarding Methods  Switch with source-based forwarding enabled EtherChannel Cisco router with destination-based forwarding enabled 101239 Catalyst 3560 Configuration Guide at 33-16

No.	’740 Patent Claim 1	Cisco EtherChannel
		

Configuring EtherChannel Load Balancing

This section describes how to configure EtherChannel load balancing by using source-based or destination-based forwarding methods. For more information, see the “Load Balancing and Forwarding Methods” section on page 33-6.

Beginning in privileged EXEC mode, follow these steps to configure EtherChannel load balancing. This procedure is optional.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**port-channel load-balance** {**dst-ip** | **dst-mac** | **src-dst-ip** | **src-dst-mac** | **src-ip** | **src-mac**}	Configure an EtherChannel load-balancing method. The default is **src-mac**. Select one of these load-distribution methods: • **dst-ip**—Load distribution is based on the destination-host IP address. • **dst-mac**—Load distribution is based on the destination-host MAC address of the incoming packet. • **src-dst-ip**—Load distribution is based on the source-and-destination host-IP address. • **src-dst-mac**—Load distribution is based on the source-and-destination host-MAC address. • **src-ip**—Load distribution is based on the source-host IP address. • **src-mac**—Load distribution is based on the source-MAC address of the incoming packet.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show etherchannel load-balance**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return EtherChannel load balancing to the default configuration, use the **no port-channel load-balance** global configuration command.

No.	’740 Patent Claim 1	Cisco EtherChannel
		Layer 2 EtherChannel **Catalyst 2950** Catalyst 2950 switches only support 802.1Q trunking and do not support ISL trunking. Catalyst 2950 switches support DTP and PAgP dynamic trunking and channel negotiation with Cisco IOS Software Release 12.1 releases and static modes only with Cisco IOS Software Release 12.0 releases. EtherChannel load balancing can use either source-MAC or destination-MAC address forwarding. You can configure the load balancing method by issuing the **port-channel load-balance** global configuration command. These switches support up to eight switch ports per channel. **Catalyst 6500 That Runs Cisco IOS Software** Catalyst 6500 switches that run Cisco IOS Software support L2 (switchport) and Layer 3 (L3) (routed port) EtherChannel configurations. A Catalyst 6500/6000 series switch supports a maximum of 64 EtherChannels (256 with Cisco IOS Software Release 12.1(2)E and earlier). You can form an EtherChannel with up to eight compatibly configured LAN ports on any module in a Catalyst 6000 series switch, with the exception of Digital Feature Card (DFC)-equipped modules (such as WS-X6816 and so on) which currently allow an L2 channel only using ports on the same DFC module. However, an L3 channel can be configured across different DFC-equipped modules. This limitation has been removed in Catalyst 6500/6000 Cisco IOS Software Release 12.1(11b)EX and later. This document configures an L2 EtherChannel. The Catalyst 6500/6000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP addresses, or Layer 4 (L4) port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. Catalyst 6500/6000 switches support both ISL and 802.1Q trunking encapsulations and DTP. Detailed information on port capabilities is available by issuing the **show interface** *interface_id* **capabilities** command.

No.	’740 Patent Claim 1	Cisco EtherChannel
		Catalyst 4000 That Runs Cisco IOS Software Catalyst 4000 switches that run Cisco IOS Software (with Supervisor Engine III and IV) support L2 (switchport) and L3 (routed port) EtherChannel configurations. A Catalyst 4000 series switch supports a maximum of 64 EtherChannels. You can form an EtherChannel with up to eight compatibly configured Ethernet interfaces on any module, and across modules in a Catalyst 4000 series switch. All interfaces in each EtherChannel must be the same speed and must all be configured as either L2 or L3 interfaces. The Catalyst 4000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP address, or L4 port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. The Catalyst 4000 that runs Cisco IOS Software supports ISL and 802.1Q trunking encapsulations and DTP. ISL is not available on certain modules. For a complete list of such modules, refer to the *Understanding VLAN Trunks* section of Configuring Layer 2 Ethernet Interfaces. In a future software release, detailed information on port capabilities will be available by issuing the **show interface capabilities** command. Currently this command is not available. Configuring EtherChannel at 1 -2 **Introduction** This sample configuration demonstrates how to set up a Layer 3 (L3) EtherChannel, without VLAN trunking, between a Cisco router and a Cisco Catalyst 6500 switch running Cisco IOS® System Software. EtherChannel can be called Fast EtherChannel (FEC) or Gigabit EtherChannel (GEC); the term depends on the speed of the interfaces or ports you use to form the EtherChannel. In this example, two Fast Ethernet ports from a Cisco router and a Catalyst 6500 switch have been bundled into a FEC. Throughout this document, the terms FEC, GEC, port channel, channel, and port group all refer to EtherChannel. Before you attempt this configuration, ensure that you meet these requirements:

No.	’740 Patent Claim 1	Cisco EtherChannel
		• Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software support both Layer 2 (L2) and L3 EtherChannel, with up to eight compatibly configured Ethernet interfaces on any module. All interfaces in each EtherChannel must be the same speed. All must be configured as either L2 or L3 interfaces. • EtherChannel load balancing can use either MAC addresses, IP addresses, or the TCP port numbers. **Note:** The selected mode applies to all EtherChannels configured on the switch. • Catalyst 6500/6000 Cisco IOS Software Release 12.1E or later and Catalyst 4500/4000 Cisco IOS Software Release 12.1(8a)EW or later. • Cisco routers: • IP traffic distributes over the port channel interface while traffic from other routing protocols sends over a single link. Bridged traffic distributes on the basis of the L3 information in the packet. If the L3 information does not exist in the packet, the traffic sends over the first link. • A wide variety of Cisco routers support EtherChannel. To find a platform or version of code that supports EtherChannel on a Cisco router, use the Cisco Feature Navigator II (registered customers only) . A list of routers and Cisco IOS Software releases that support EtherChannel is found under the FEC feature. Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Cisco EtherChannel System in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 1[f] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Basso discloses using a hash function and index table to select for blade/port combinations over which to send the packet over a user port and a switch fabric link. Basso further discloses that this selection is performed based upon packet information.

No.	’740 Patent Claim 1	Cisco EtherChannel
		Basso at [0010] (“Upon a network processor receiving a packet of data, the network processor may index into a table, commonly referred to as a forwarding table, to determine the table associated with a particular logical interface as well as the next destination address. The forwarding table may comprise a plurality of entries where each entry may comprise information indicating a particular table associated with a particular logical interface as well as the next destination address. Each logical interface may be associated with a table storing a plurality of entries containing blade/ port combinations as discussed further below. In one embodiment, an entry may be indexed in the forwarding table using a destination address in the received packet header.”) Basso at [0011] (“A hash function may then be performed on the received packet to generate a hash value. In one embodiment, a hash function may be performed on the source and destination address in the packet header to generate a hash value.”) Basso at [0012] (“The hash value generated may be used to index into the table associated with a particular logical interface. Upon indexing into the table associated with the logical interface, an appropriate blade/port combination may be identified to transmit the received packet of data. In one embodiment, a blade/port combination may be selected in the indexed entry of the table associated with the logical interface by using a portion of the bits of the hashed value. The received packet may then be transmitted through the identified blade/port combination to the next destination (next destination previously identified by the next destination address in the forwarding table).”) Basso at [0035] (“By logically grouping a plurality of ports 404 coupled to a particular network device into a logical interface 405, network processor 403 may be configured to transmit processed packets to that particular network device via any blade 402/port 404 combination grouped in that logical interface 405. For example, referring to FIG. 4, ports 404A-404I are physically connected to router 104B. If ports 404A-404I were logically grouped into logical interface 405, then a particular network processor 403, e.g., network processor 403A, may be configured to transmit processed packets that are determined to be transmitted to router 104B through any of ports 404A-404I in blades 402A-C, respectively. Network processor 403, e.g., network processor 403A, may be configured to transmit the processed packets to router 104B through ports 404, e.g., ports 404D-I, not in its blade 402, e.g., blade 402A, by forwarding the processed packets to switch fabric 401 which may then direct the processed packets to another appropriate physical blade 402/port 404 combination. Network processor 403, e.g., network processor 403A, may further be configured to transmit the processed packets to router 104B through any ports 404, e.g., ports 404A-C, in its blade 402, e.g., blade402A, instead of just one physical port 404 in its blade

No.	’740 Patent Claim 1	Cisco EtherChannel
		402, e.g., blade 402A. A more detailed description of routing packets implementing logical interface(s) 405 is provided below in FIG. 5.”) Basso at [0040] (“In step 502, network processor 403, e.g., network processor 403A, may receive a packet of data from switch fabric 401. Upon receiving the packet of data, network processor 403, in step 503, may index into a table, commonly referred to as a forwarding table, to determine the table associated with a particular logical interface 405 as well as the next destination address, i.e., the next hop address. The forwarding table may comprise a plurality of entries where each entry may comprise information indicating a particular table associated with a particular logical interface 405 as well as the next destination address. Each logical interface 405 may be associated with a table storing a plurality of entries containing blade 402/port 404 combinations as discussed further below. In one embodiment, an entry may be indexed in the forwarding table using a destination address in the received packet header. It is noted that an entry may be indexed in the forwarding table usingother means and that such means would be recognized by an artisan of ordinary skill in the art. It is further noted that embodiments implementing such means would fall within the scope of the present invention.”) Basso at [0041] (“In step 504, a hash function may be performed on the received packet to generate a hash value. In one embodiment, a hash function may be performed on the source and destination address in the packet header to generate a hash value. It is noted that in other embodiments a hash function may be performed on different fields, e.g., port, type of service, in the received packet to generate a hash value.”) Basso at [0042] (“In step 505, the hash value generated in step 504 may be used to index into the table associated with a particular logical interface 405 determined in step 503. Upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be identified in step 506 to transmit the received packet of data as explained below.”) Basso at [0043] (“As stated above, the table associated with a particular logical interface 405 may comprise a plurality of entries where each entry may comprise a threshold value associated with a particular blade 402/port 404 combination. The threshold value may represent a percentage of the total number of packets received by router 104A that may be transmitted through the blade 402/port 404 combination associated with that threshold value. In one embodiment, the threshold value may be updated periodically by a user, e.g., system administrator, in control of router 104, e.g., router 104A. For example, the threshold value, e.g., twenty percent of the number of packets

No.	’740 Patent Claim 1	Cisco EtherChannel
		received by router 104A, associated with a particular blade 402/port 404 combination may be updated by lowering the threshold value by one percent during each update. An example of an entry of the table associated with a particular logical interface 405 is shown in Table 1 below:

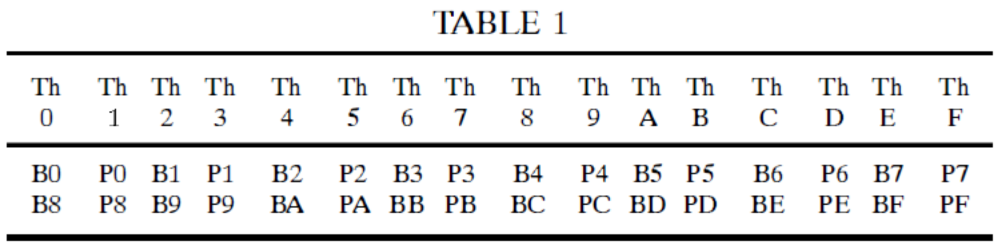


TABLE 1

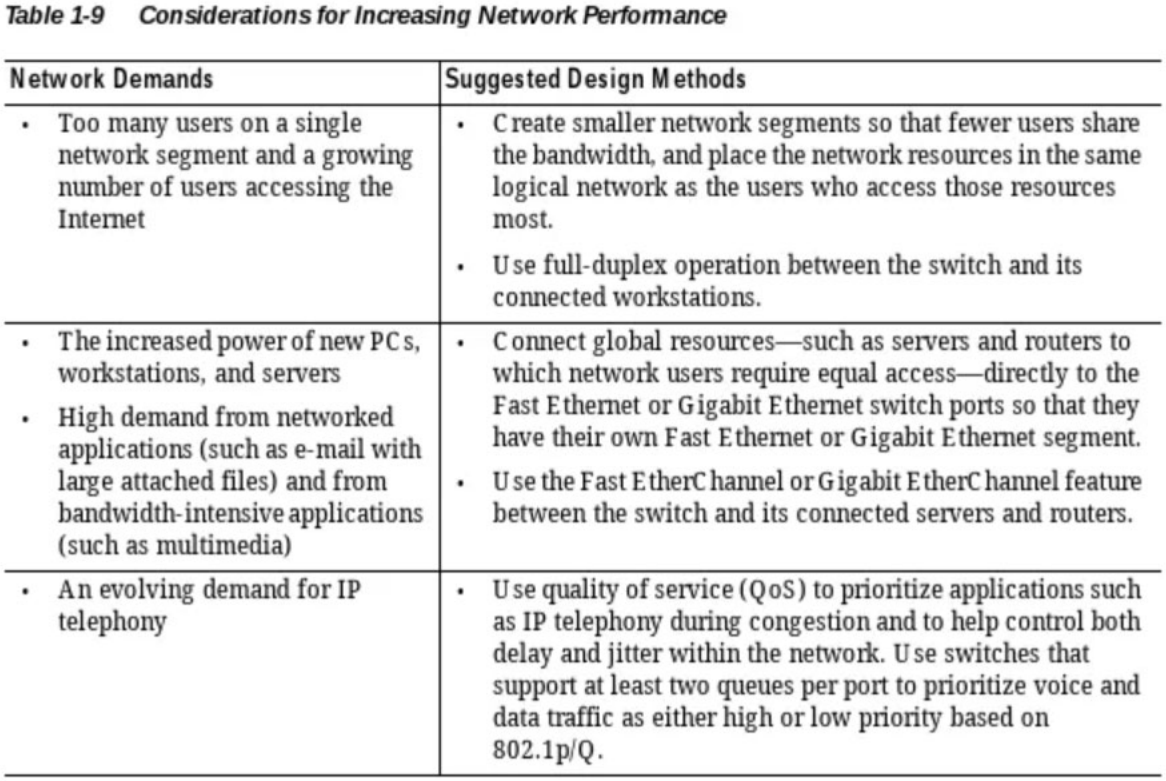
Th 0	Th 1	Th 2	Th 3	Th 4	Th 5	Th 6	Th 7	Th 8	Th 9	Th A	Th B	Th C	Th D	Th E	Th F
B0 B8	P0 P8	B1 B9	P1 P9	B2 BA	P2 PA	B3 BB	P3 PB	B4 BC	P4 PC	B5 BD	P5 PD	B6 BE	P6 PE	B7 BF	P7 PF

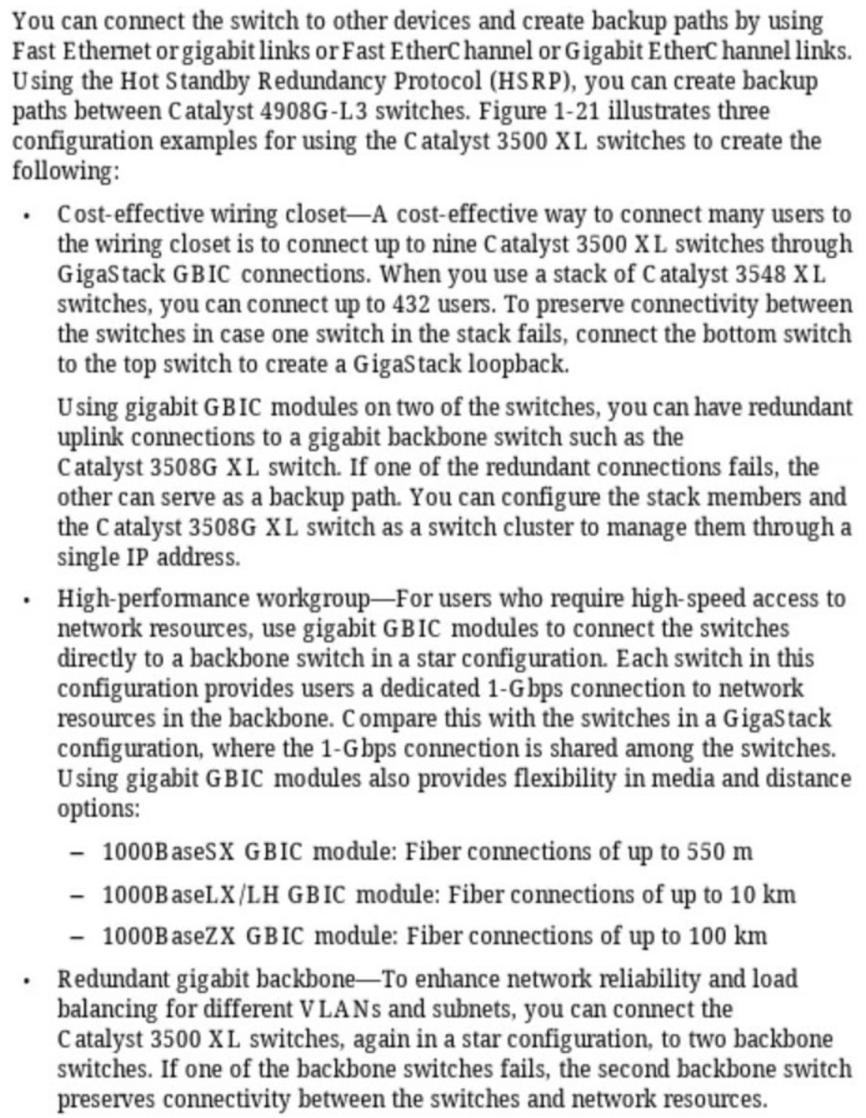
No.	’740 Patent Claim 1	Cisco EtherChannel
		Basso at [0044] (“Table 1 above illustrates an exemplary entry in the table associated with a particular logical interface 405. Each entry may comprise a plurality of threshold values (16 threshold values in exemplary Table 1) where each threshold value is associated with a particular blade 402/port 404 combination. For example, threshold value (Th0) is associated with blade B0/port P0 combination where blade BO may refer to a particular blade 402, e.g., blade 402B, and port PO may refer to a particular port 404, e.g., port 404E. Threshold value (Thl) is associated with blade Bl/port Pl combination and so forth. As stated above, each threshold value may represent a percentage of the total number of packets received by router 104A that may be transmitted through the blade 402/port 404 combination associated with that threshold value. For example, threshold value (Th0) may represent a percentage of the total number of packets received by router 104A that may be transmitted through port PO in blade BO. If port PO refers to port 404D and blade BO refers to blade 402B, then if Th0 has a value of twenty percent, a maximum of twenty percent of the total packets received by router 104A may be transmitted through port 404D in blade 402B.”)
		Basso at [0045] (“As stated above, upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be identified in step 506 to transmit the received packet of data. In one embodiment, the hash value generated in step 504 may be used to select a particular threshold value and hence a blade 402/port 404 combination associated with the selected threshold value. In one embodiment, a portion of the bits of the hash value, e.g., most significant bits, may be used to select a particular

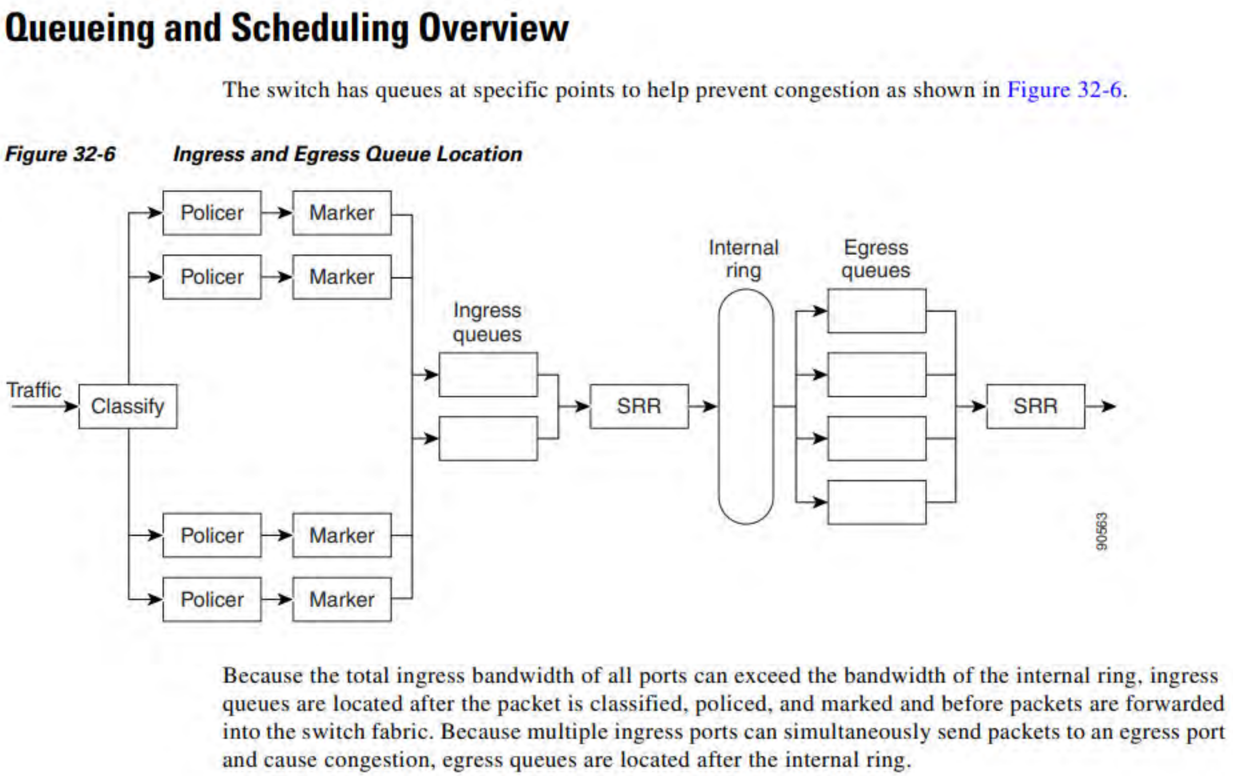
No.	’740 Patent Claim 1	Cisco EtherChannel
		threshold value in the entry indexed in step 505. For example, referring to Table 1, since there are 16 different threshold values in each entry of the table associated with logical interface 405, only four bits of the hash value generated in step 504 may be used to select a threshold value. Upon selecting a threshold value, the blade 402/port 404 combination associated with the selected threshold value may be used to transmit the received packet.”) As another example, Wiher discloses using cell header information to at each node to select and route the ATM data cell over a data link and selected backplane. Wiher at 3:43-65 (“In general, in another aspect, the invention features an apparatus for communicating data cells between a data link and a backplane. The apparatus includes transceiver circuitry to transmit and receive data cells over a data link and a plurality of backplane interfaces each including at least one cell signal terminal. Each of the backplane interface is coupled to a backplane interconnection circuit. Each backplane interconnection circuit transmits and receives cells over the cell signal terminals of its associated backplane interface. The apparatus also includes de-multiplexing circuitry coupling the transceiver circuitry to each of the backplane interconnection circuits. The de-multiplexing circuitry receives a data cell from the transceiver circuitry, select a backplane interconnection circuit associated with the data cell, and provide the data cell to the selected backplane interconnection circuit for transmission over the cell signal terminals of the associated backplane interface. The apparatus also includes multiplexing circuitry coupling the plurality of backplane interconnection circuits to the transceiver circuitry. The multiplexing circuitry receives data cells from each of the backplane interconnection circuits and provide the received data cells to the transceiver circuitry.”) Wiher at 3:66-4:22 (“Implementations of the invention may include one or more of the following features. The backplane interconnection circuits may independently receive and transmit data cells over the plurality of backplane interfaces. The de-multiplexing circuitry may select a backplane interface based on data in the header field of the data cell. The apparatus may include header translation circuitry to alter header data in cells sent between the plurality of backplane interfaces and the transceiver circuitry. Each of the plurality of backplane interfaces may include separate terminals to receive cells and separate terminals to transmit cells. The terminals to transmit cells may include a first and second control terminal and at least one outgoing cell data terminal. A backplane interface's backplane interconnection circuitry may accepts a signal on the first control terminal as indicating that a cell may be sent over the interface, asserts a 15 signal on the second control terminal to indicate

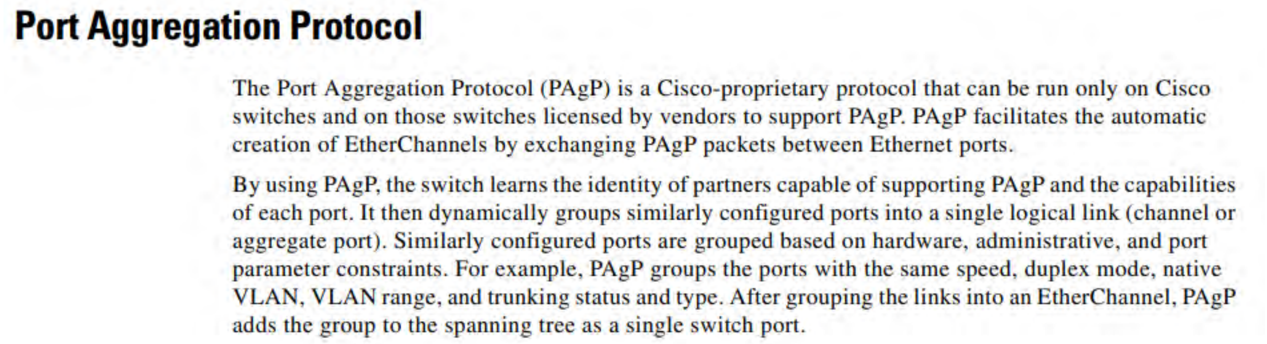
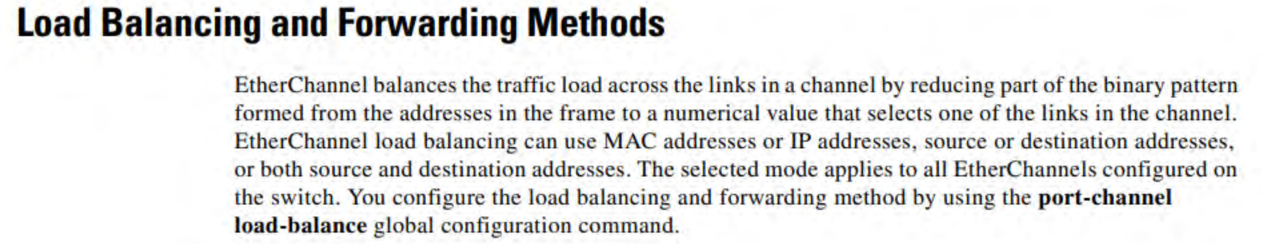
No.	’740 Patent Claim 1	Cisco EtherChannel
		that a cell is being transmitted, and transmits data bits of the cell on the outgoing cell data terminal. Each backplane interface may include a single outgoing cell data terminal and each bit of the cell may be serially transmitted over the single outgoing cell data terminal. Each backplane interface may include multiple outgoing cell data terminals and bits of the cell may be sent in parallel over the eight outgoing cell data terminals.”)
1[g]	sending the data frame over the selected first and second physical links,	Cisco EtherChannel System discloses sending the data frame over the selected first and second physical links. For example, Cisco EtherChannel System discloses forwarding the data packets over the specific port and switch fabric interconnection. Cisco Catalyst 6500 Data Sheet at 13 (“**Switch Fabric Modules** Designed to support distributed forwarding for interface modules with that capability, the Cisco Catalyst 6500 Series Switch Fabric Module (SFM or SFM2), in combination with the Cisco Catalyst 6000 Multilayer Switch Feature Card (MSFC2) and Cisco distributed forwarding cards (DFCs) on interface modules, increases available system bandwidth from 32 to 256 Gbps. The SFM or SFM2 supports the Cisco Catalyst 6500 CEF256 and dCEF256 interface modules. Designed to support new interface modules with 720-Gbps forwarding capabilities, the switch fabric onboard the Cisco Catalyst 6500 Series Supervisor Engine 720 increases available bandwidth to 720 Gbps and provides packet-forwarding rates up to 400 mpps. By using automatic sensing and negotiation, the switch fabric is fully interoperable with the 8- and 16-Gbps switch-fabric interconnections used by the CEF256 and dCEF256 interface modules. When a CEF256 or dCEF256 interface module is detected, the switch fabric will automatically connect those modules by offering 8 to 16 Gbps of bandwidth to each module, as applicable.”) Cisco Catalyst 6500 Data Sheet at 14 (“Using the same ASIC engine design as the central PFCx, distributed forwarding cards (DFCs) located on the interface modules forward packets between two ports, directly or across the switch fabric, without involving the supervisor engine. With the DFC, each interface module has a dedicated forwarding engine complete with the full forwarding tables. Distributed Cisco Express Forwarding (Figure 6) works like this:

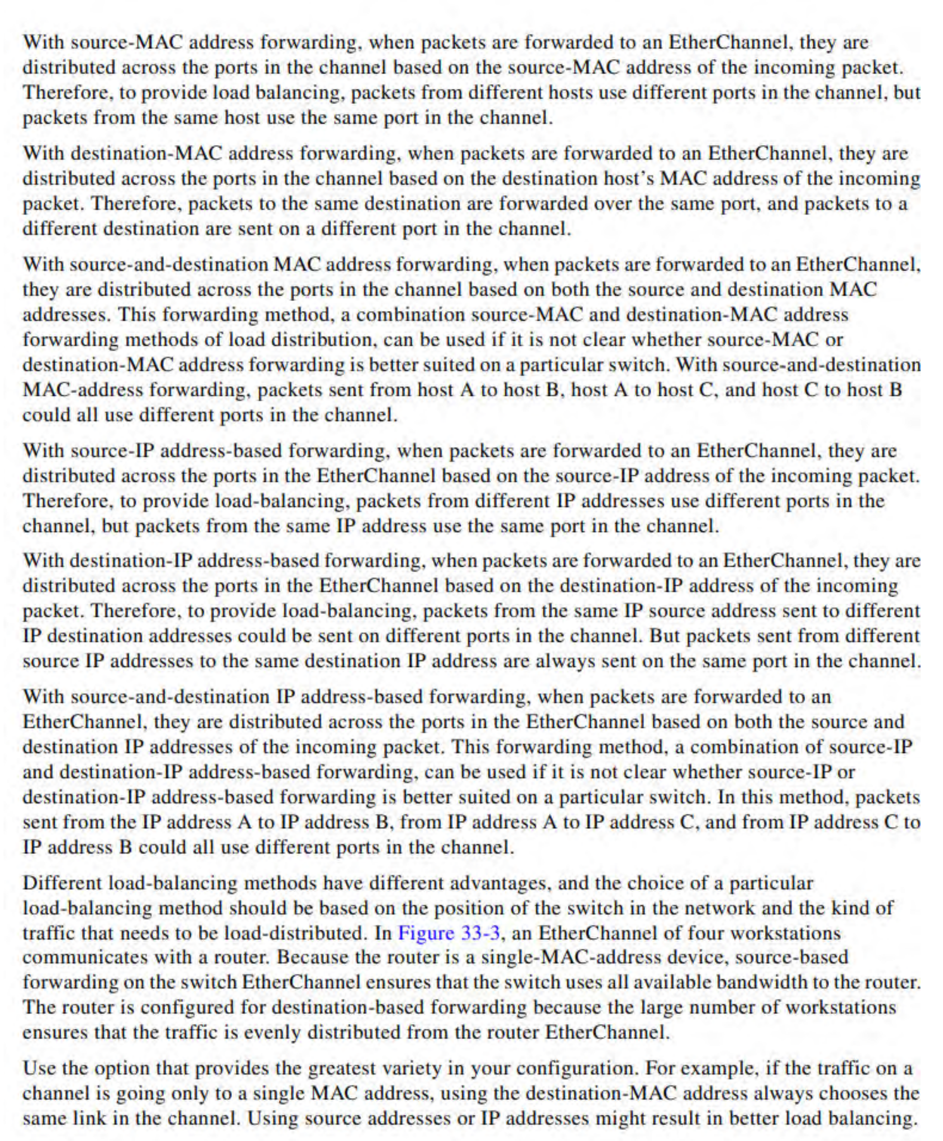
No.	’740 Patent Claim 1	Cisco EtherChannel
		• As in standard Cisco Express Forwarding, the central PFCx located on the supervisor engine and the DFC engines located on the interface modules are loaded with the same Cisco Express Forwarding information derived from the forwarding table before any user traffic arrives at the switch. • As a packet arrives at an interface module, its DFC engine inspects the packet and uses the information in the Cisco Express Forwarding table (including Layer 2, Layer 3, ACLs, and QoS) to make a completely hardware-based forwarding decision for that packet. • The Distributed Cisco Express Forwarding engine manages all hardware-based forwarding for traffic on that module, including Layer 2 and Layer 3 forwarding, ACLs, QoS policing and marking, and NetFlow. • Because the DFCs make all the switching decisions locally, the central PFCx can dedicate more hardware-forwarding resources to any modules not equipped with a DFC.”) Cisco Catalyst 6500 Data Sheet at 13 (“How Cisco Express Forwarding Works Cisco Express Forwarding is a Layer 3 technology that provides increased forwarding scalability and performance to manage the many short-duration traffic flows common in today’s enterprise and service provider networks. To meet the needs of environments managing large amounts of short-flow, Web-based, or highly interactive types of traffic, Cisco Express Forwarding forwards all packets in hardware, and maintains its forwarding rate independent of the number of flows going though the switch. On the Cisco Catalyst 6500 Series, the Cisco Express Forwarding Layer 3 forwarding engine is located centrally on the supervisor engine’s policy feature card (PFC2 or PFC3)—the same device that performs hardware-based Layer 2 and Layer 3 forwarding, access control list (ACL) checking, QoS policing and marking, and NetFlow statistics gathering. Using the routing table that Cisco IOS Software builds to define configured interfaces and routing protocols, the Cisco Express Forwarding architecture creates Cisco Express Forwarding tables and downloads them into the hardware forwarding engine before any user traffic is sent through the switch. The Cisco Express Forwarding architecture places only the routing prefixes in its Cisco Express Forwarding tables—the only information it requires to make the Layer 3 forwarding decisions—relying on the routing protocols to do route selection. By performing a simple Cisco Express Forwarding table lookup, the switch forwards packets at wire rate, independent of the number of flows transiting the switch.”)

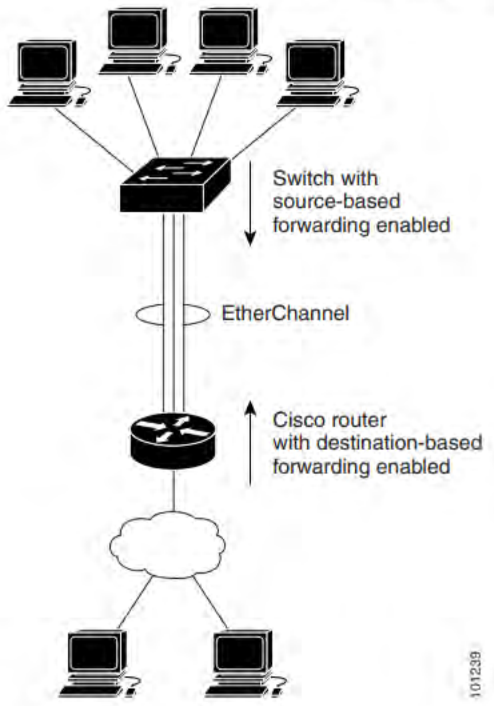
<table>
<tr><th>No.</th><th>’740 Patent Claim 1</th><th>Cisco EtherChannel</th></tr>
<tr><td></td><td></td><td>
Catalyst 3500 Installation Guide at Table 1-9

<b><i>Table 1-9 Considerations for Increasing Network Performance</i></b>
<table>
<tr><th>Network Demands</th><th>Suggested Design Methods</th></tr>
<tr><td>• Too many users on a single network segment and a growing number of users accessing the Internet</td><td>• Create smaller network segments so that fewer users share the bandwidth, and place the network resources in the same logical network as the users who access those resources most.
• Use full-duplex operation between the switch and its connected workstations.</td></tr>
<tr><td>• The increased power of new PCs, workstations, and servers
• High demand from networked applications (such as e-mail with large attached files) and from bandwidth-intensive applications (such as multimedia)</td><td>• Connect global resources—such as servers and routers to which network users require equal access—directly to the Fast Ethernet or Gigabit Ethernet switch ports so that they have their own Fast Ethernet or Gigabit Ethernet segment.
• Use the Fast EtherChannel or Gigabit EtherChannel feature between the switch and its connected servers and routers.</td></tr>
<tr><td>• An evolving demand for IP telephony</td><td>• Use quality of service (QoS) to prioritize applications such as IP telephony during congestion and to help control both delay and jitter within the network. Use switches that support at least two queues per port to prioritize voice and data traffic as either high or low priority based on 802.1p/Q.</td></tr>
</table>
Catalyst 3500 Installation Guide at 1-30
</td></tr>
</table>

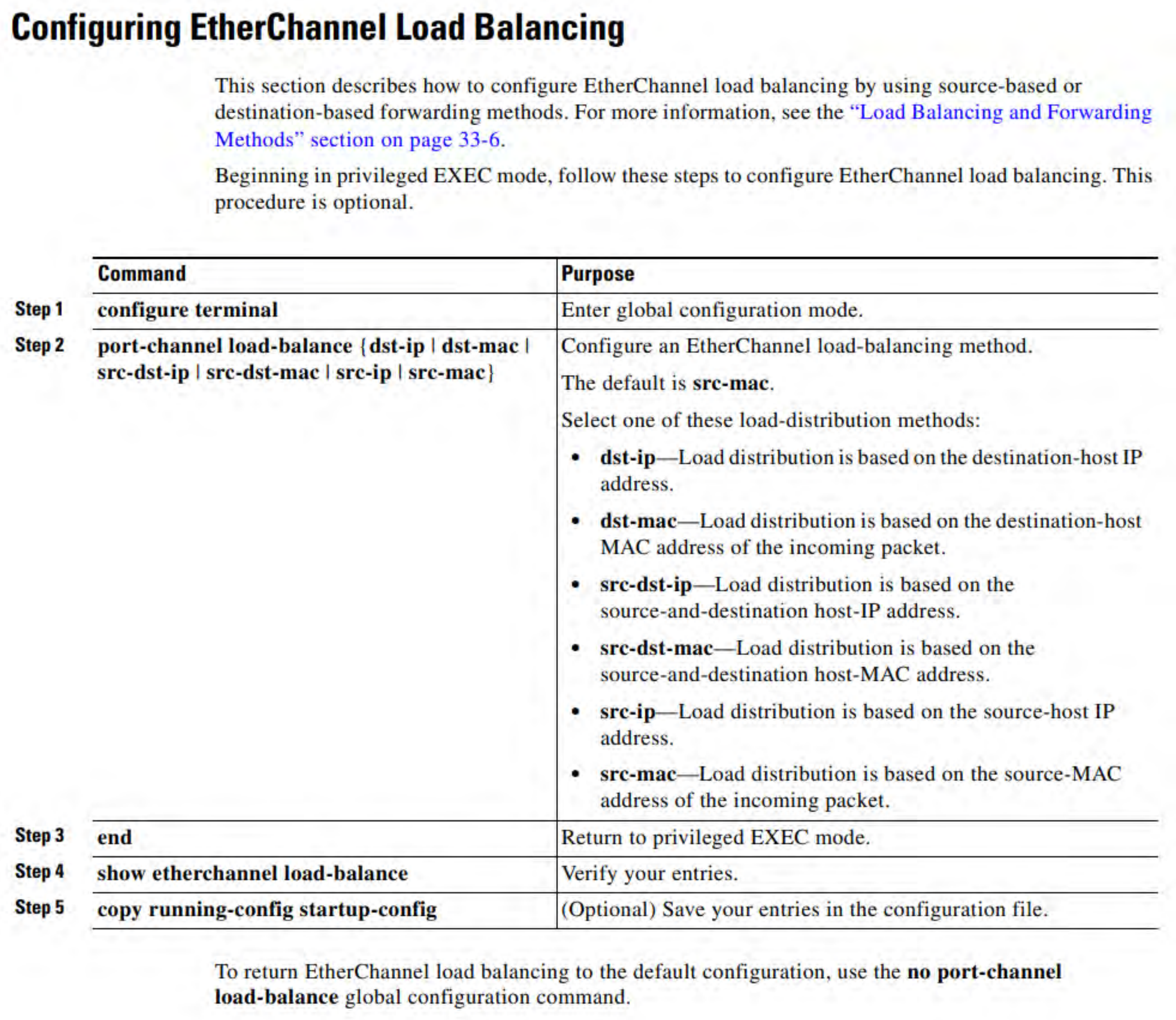
No.	’740 Patent Claim 1	Cisco EtherChannel
		 You can connect the switch to other devices and create backup paths by using Fast Ethernet or gigabit links or Fast EtherChannel or Gigabit EtherChannel links. Using the Hot Standby Redundancy Protocol (HSRP), you can create backup paths between Catalyst 4908G-L3 switches. Figure 1-21 illustrates three configuration examples for using the Catalyst 3500 XL switches to create the following: • Cost-effective wiring closet—A cost-effective way to connect many users to the wiring closet is to connect up to nine Catalyst 3500 XL switches through GigaStack GBIC connections. When you use a stack of Catalyst 3548 XL switches, you can connect up to 432 users. To preserve connectivity between the switches in case one switch in the stack fails, connect the bottom switch to the top switch to create a GigaStack loopback. Using gigabit GBIC modules on two of the switches, you can have redundant uplink connections to a gigabit backbone switch such as the Catalyst 3508G XL switch. If one of the redundant connections fails, the other can serve as a backup path. You can configure the stack members and the Catalyst 3508G XL switch as a switch cluster to manage them through a single IP address. • High-performance workgroup—For users who require high-speed access to network resources, use gigabit GBIC modules to connect the switches directly to a backbone switch in a star configuration. Each switch in this configuration provides users a dedicated 1-Gbps connection to network resources in the backbone. Compare this with the switches in a GigaStack configuration, where the 1-Gbps connection is shared among the switches. Using gigabit GBIC modules also provides flexibility in media and distance options: – 1000BaseSX GBIC module: Fiber connections of up to 550 m – 1000BaseLX/LH GBIC module: Fiber connections of up to 10 km – 1000BaseZX GBIC module: Fiber connections of up to 100 km • Redundant gigabit backbone—To enhance network reliability and load balancing for different VLANs and subnets, you can connect the Catalyst 3500 XL switches, again in a star configuration, to two backbone switches. If one of the backbone switches fails, the second backbone switch preserves connectivity between the switches and network resources.

No.	’740 Patent Claim 1	Cisco EtherChannel
		Catalyst 3560 Configuration Guide at Figure 32-6  **Queueing and Scheduling Overview** The switch has queues at specific points to help prevent congestion as shown in Figure 32-6. *Figure 32-6 Ingress and Egress Queue Location* Traffic Classify Policer Marker Policer Marker Policer Marker Policer Marker Ingress queues SRR Internal ring Egress queues SRR 90563 Because the total ingress bandwidth of all ports can exceed the bandwidth of the internal ring, ingress queues are located after the packet is classified, policed, and marked and before packets are forwarded into the switch fabric. Because multiple ingress ports can simultaneously send packets to an egress port and cause congestion, egress queues are located after the internal ring. Catalyst 3560 Configuration Guide at 33-4

No.	’740 Patent Claim 1	Cisco EtherChannel
		 **Port Aggregation Protocol** The Port Aggregation Protocol (PAgP) is a Cisco-proprietary protocol that can be run only on Cisco switches and on those switches licensed by vendors to support PAgP. PAgP facilitates the automatic creation of EtherChannels by exchanging PAgP packets between Ethernet ports. By using PAgP, the switch learns the identity of partners capable of supporting PAgP and the capabilities of each port. It then dynamically groups similarly configured ports into a single logical link (channel or aggregate port). Similarly configured ports are grouped based on hardware, administrative, and port parameter constraints. For example, PAgP groups the ports with the same speed, duplex mode, native VLAN, VLAN range, and trunking status and type. After grouping the links into an EtherChannel, PAgP adds the group to the spanning tree as a single switch port. Catalyst 3560 Configuration Guide at 33-6 – 33-8  **Load Balancing and Forwarding Methods** EtherChannel balances the traffic load across the links in a channel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel. EtherChannel load balancing can use MAC addresses or IP addresses, source or destination addresses, or both source and destination addresses. The selected mode applies to all EtherChannels configured on the switch. You configure the load balancing and forwarding method by using the **port-channel load-balance** global configuration command.

No.	’740 Patent Claim 1	Cisco EtherChannel
		 With source-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the source-MAC address of the incoming packet. Therefore, to provide load balancing, packets from different hosts use different ports in the channel, but packets from the same host use the same port in the channel. With destination-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the destination host’s MAC address of the incoming packet. Therefore, packets to the same destination are forwarded over the same port, and packets to a different destination are sent on a different port in the channel. With source-and-destination MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on both the source and destination MAC addresses. This forwarding method, a combination source-MAC and destination-MAC address forwarding methods of load distribution, can be used if it is not clear whether source-MAC or destination-MAC address forwarding is better suited on a particular switch. With source-and-destination MAC-address forwarding, packets sent from host A to host B, host A to host C, and host C to host B could all use different ports in the channel. With source-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the source-IP address of the incoming packet. Therefore, to provide load-balancing, packets from different IP addresses use different ports in the channel, but packets from the same IP address use the same port in the channel. With destination-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the destination-IP address of the incoming packet. Therefore, to provide load-balancing, packets from the same IP source address sent to different IP destination addresses could be sent on different ports in the channel. But packets sent from different source IP addresses to the same destination IP address are always sent on the same port in the channel. With source-and-destination IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on both the source and destination IP addresses of the incoming packet. This forwarding method, a combination of source-IP and destination-IP address-based forwarding, can be used if it is not clear whether source-IP or destination-IP address-based forwarding is better suited on a particular switch. In this method, packets sent from the IP address A to IP address B, from IP address A to IP address C, and from IP address C to IP address B could all use different ports in the channel. Different load-balancing methods have different advantages, and the choice of a particular load-balancing method should be based on the position of the switch in the network and the kind of traffic that needs to be load-distributed. In Figure 33-3, an EtherChannel of four workstations communicates with a router. Because the router is a single-MAC-address device, source-based forwarding on the switch EtherChannel ensures that the switch uses all available bandwidth to the router. The router is configured for destination-based forwarding because the large number of workstations ensures that the traffic is evenly distributed from the router EtherChannel. Use the option that provides the greatest variety in your configuration. For example, if the traffic on a channel is going only to a single MAC address, using the destination-MAC address always chooses the same link in the channel. Using source addresses or IP addresses might result in better load balancing.

No.	’740 Patent Claim 1	Cisco EtherChannel
		Figure 33-3 Load Distribution and Forwarding Methods  Switch with source-based forwarding enabled EtherChannel Cisco router with destination-based forwarding enabled 101239 Catalyst 3560 Configuration Guide at 33-16

No.	’740 Patent Claim 1	Cisco EtherChannel
		

Configuring EtherChannel Load Balancing

This section describes how to configure EtherChannel load balancing by using source-based or destination-based forwarding methods. For more information, see the “Load Balancing and Forwarding Methods” section on page 33-6.

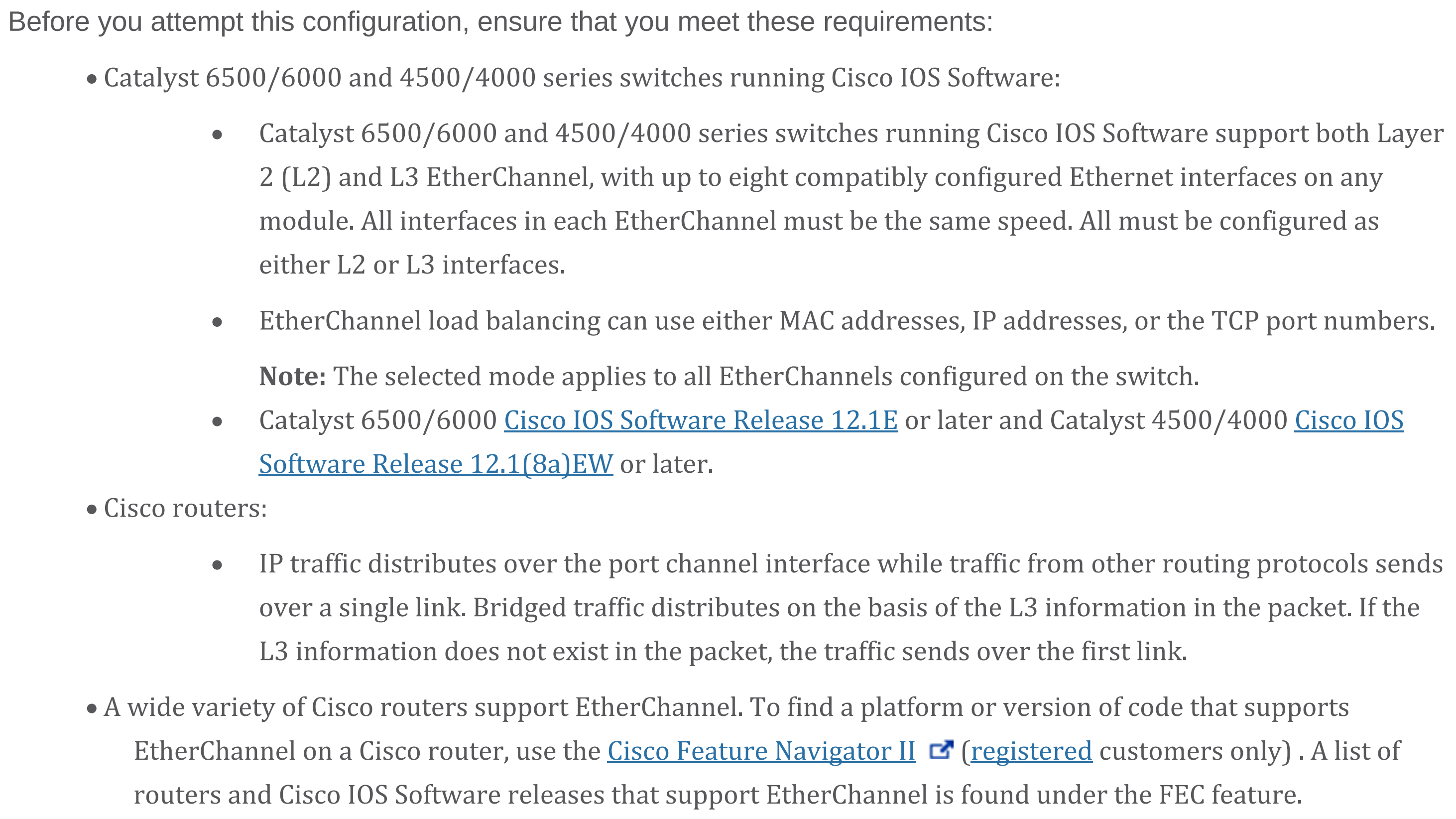
Beginning in privileged EXEC mode, follow these steps to configure EtherChannel load balancing. This procedure is optional.

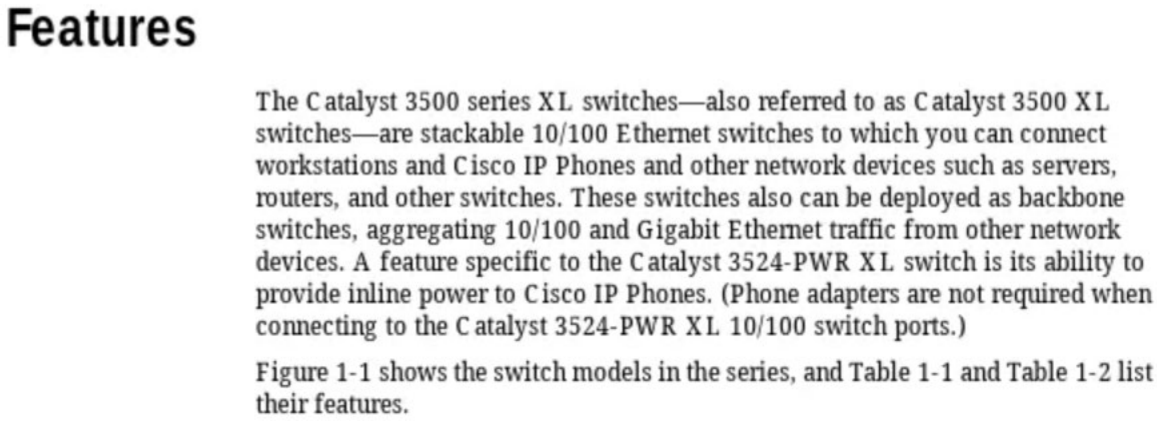
	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**port-channel load-balance** {**dst-ip** | **dst-mac** | **src-dst-ip** | **src-dst-mac** | **src-ip** | **src-mac**}	Configure an EtherChannel load-balancing method. The default is **src-mac**. Select one of these load-distribution methods: • **dst-ip**—Load distribution is based on the destination-host IP address. • **dst-mac**—Load distribution is based on the destination-host MAC address of the incoming packet. • **src-dst-ip**—Load distribution is based on the source-and-destination host-IP address. • **src-dst-mac**—Load distribution is based on the source-and-destination host-MAC address. • **src-ip**—Load distribution is based on the source-host IP address. • **src-mac**—Load distribution is based on the source-MAC address of the incoming packet.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show etherchannel load-balance**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

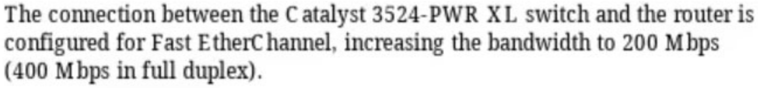
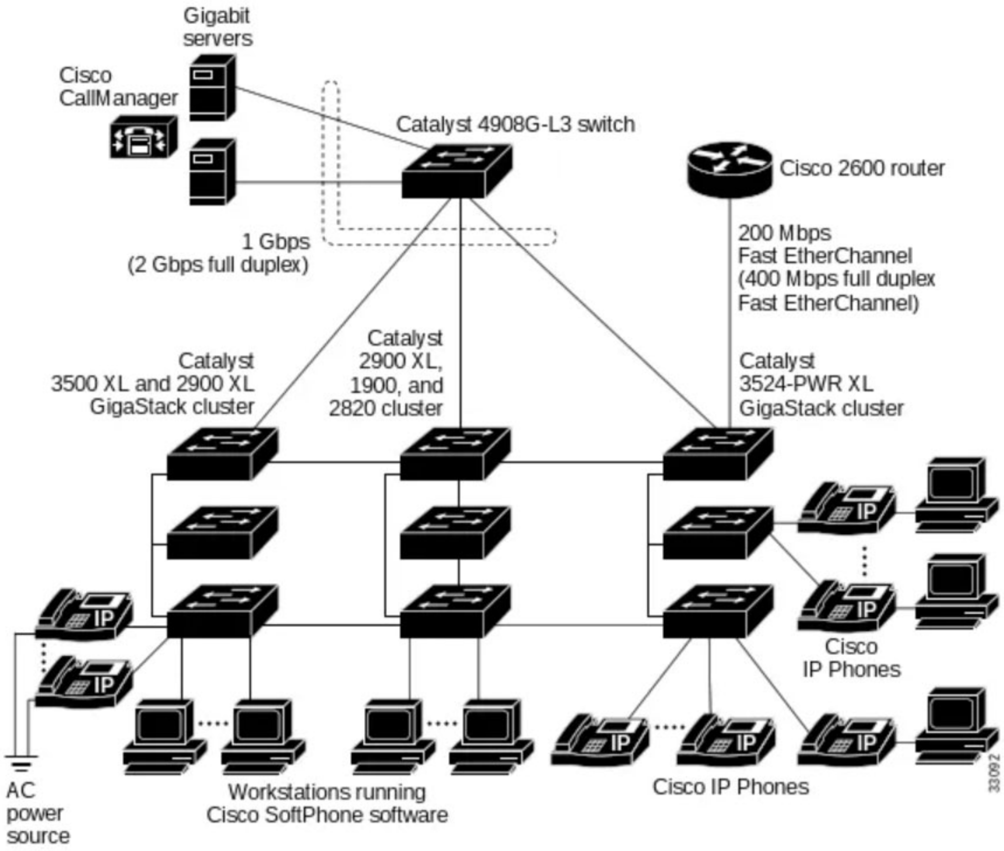
To return EtherChannel load balancing to the default configuration, use the **no port-channel load-balance** global configuration command.

No.	’740 Patent Claim 1	Cisco EtherChannel
		Layer 2 EtherChannel **Catalyst 2950** Catalyst 2950 switches only support 802.1Q trunking and do not support ISL trunking. Catalyst 2950 switches support DTP and PAgP dynamic trunking and channel negotiation with Cisco IOS Software Release 12.1 releases and static modes only with Cisco IOS Software Release 12.0 releases. EtherChannel load balancing can use either source-MAC or destination-MAC address forwarding. You can configure the load balancing method by issuing the **port-channel load-balance** global configuration command. These switches support up to eight switch ports per channel. **Catalyst 6500 That Runs Cisco IOS Software** Catalyst 6500 switches that run Cisco IOS Software support L2 (switchport) and Layer 3 (L3) (routed port) EtherChannel configurations. A Catalyst 6500/6000 series switch supports a maximum of 64 EtherChannels (256 with Cisco IOS Software Release 12.1(2)E and earlier). You can form an EtherChannel with up to eight compatibly configured LAN ports on any module in a Catalyst 6000 series switch, with the exception of Digital Feature Card (DFC)-equipped modules (such as WS-X6816 and so on) which currently allow an L2 channel only using ports on the same DFC module. However, an L3 channel can be configured across different DFC-equipped modules. This limitation has been removed in Catalyst 6500/6000 Cisco IOS Software Release 12.1(11b)EX and later. This document configures an L2 EtherChannel. The Catalyst 6500/6000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP addresses, or Layer 4 (L4) port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. Catalyst 6500/6000 switches support both ISL and 802.1Q trunking encapsulations and DTP. Detailed information on port capabilities is available by issuing the **show interface** *interface_id* **capabilities** command.

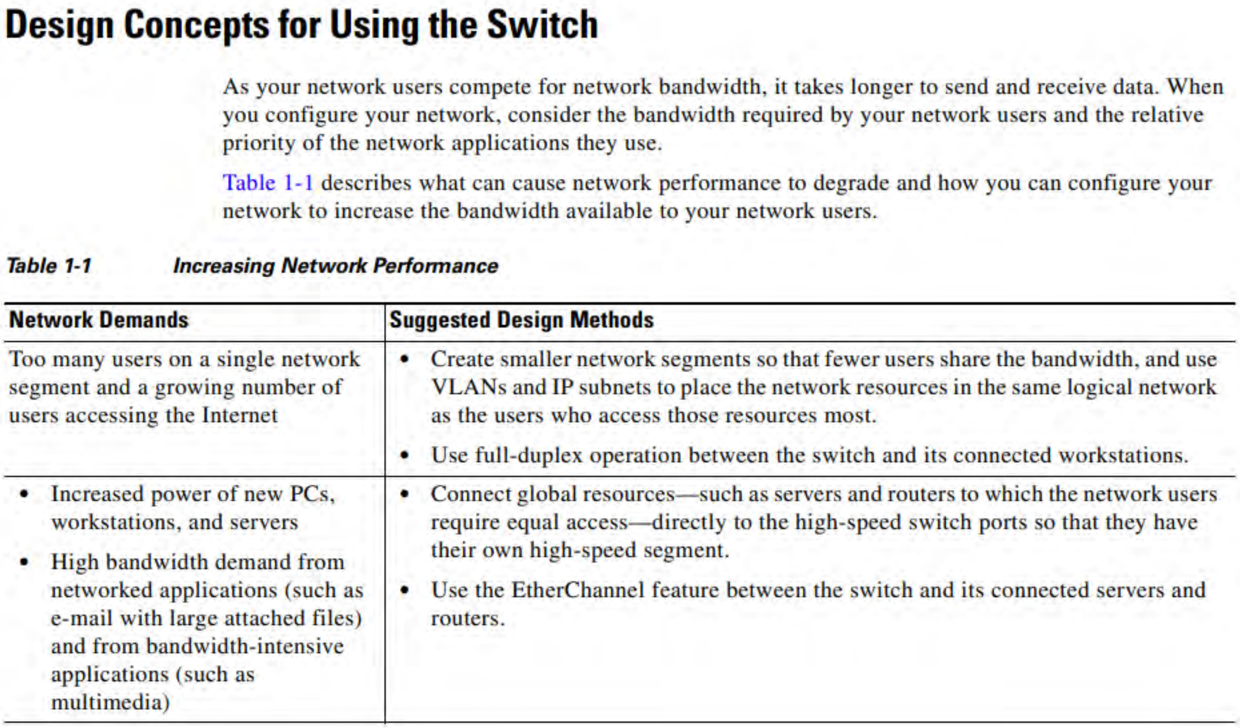
No.	’740 Patent Claim 1	Cisco EtherChannel
		Catalyst 4000 That Runs Cisco IOS Software Catalyst 4000 switches that run Cisco IOS Software (with Supervisor Engine III and IV) support L2 (switchport) and L3 (routed port) EtherChannel configurations. A Catalyst 4000 series switch supports a maximum of 64 EtherChannels. You can form an EtherChannel with up to eight compatibly configured Ethernet interfaces on any module, and across modules in a Catalyst 4000 series switch. All interfaces in each EtherChannel must be the same speed and must all be configured as either L2 or L3 interfaces. The Catalyst 4000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP address, or L4 port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. The Catalyst 4000 that runs Cisco IOS Software supports ISL and 802.1Q trunking encapsulations and DTP. ISL is not available on certain modules. For a complete list of such modules, refer to the *Understanding VLAN Trunks* section of Configuring Layer 2 Ethernet Interfaces. In a future software release, detailed information on port capabilities will be available by issuing the **show interface capabilities** command. Currently this command is not available. Configuring EtherChannel at 1 -2 **Introduction** This sample configuration demonstrates how to set up a Layer 3 (L3) EtherChannel, without VLAN trunking, between a Cisco router and a Cisco Catalyst 6500 switch running Cisco IOS® System Software. EtherChannel can be called Fast EtherChannel (FEC) or Gigabit EtherChannel (GEC); the term depends on the speed of the interfaces or ports you use to form the EtherChannel. In this example, two Fast Ethernet ports from a Cisco router and a Catalyst 6500 switch have been bundled into a FEC. Throughout this document, the terms FEC, GEC, port channel, channel, and port group all refer to EtherChannel.

No.	’740 Patent Claim 1	Cisco EtherChannel
		 Before you attempt this configuration, ensure that you meet these requirements: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software support both Layer 2 (L2) and L3 EtherChannel, with up to eight compatibly configured Ethernet interfaces on any module. All interfaces in each EtherChannel must be the same speed. All must be configured as either L2 or L3 interfaces. • EtherChannel load balancing can use either MAC addresses, IP addresses, or the TCP port numbers. **Note:** The selected mode applies to all EtherChannels configured on the switch. • Catalyst 6500/6000 Cisco IOS Software Release 12.1E or later and Catalyst 4500/4000 Cisco IOS Software Release 12.1(8a)EW or later. • Cisco routers: • IP traffic distributes over the port channel interface while traffic from other routing protocols sends over a single link. Bridged traffic distributes on the basis of the L3 information in the packet. If the L3 information does not exist in the packet, the traffic sends over the first link. • A wide variety of Cisco routers support EtherChannel. To find a platform or version of code that supports EtherChannel on a Cisco router, use the Cisco Feature Navigator II (registered customers only) . A list of routers and Cisco IOS Software releases that support EtherChannel is found under the FEC feature.
1[h]	said sending comprising communicating along at least one of said bi-directional links.	Cisco EtherChannel System discloses said sending comprising communicating along at least one of said bi-directional links. *See supra at* 1[b], 1[d], 1[g].

No.	’740 Patent Claim 2	Cisco EtherChannel
2[a]	The method according to claim 1, wherein the network node comprises a user node, and	Cisco EtherChannel System discloses the method according to claim 1, wherein the network node comprises a user node. For example, Cisco EtherChannel System discloses a phone, router, switch, or server, that can be used by a user. Cisco Catalyst 6500 Data Sheet at 1 (“The Cisco® Catalyst® 6500 and 6500-E Series sets the new standard for IP Communications and application delivery in enterprise campus and service provider networks by maximizing user productivity and enhancing operational control. As the premier intelligent, multilayer modular Cisco switch, the Catalyst 6500 Series delivers secure, converged, end-to-end services, from the wiring closet to the core network, the data center, and the WAN edge. The 6500-E series switches offer enhancements to scale beyond the 4000W power supply. Today’s 6500 series switch cannot scale beyond 4000W of total system capacity except for 6513. The 6513 can support up to 6000W of power.”) Catalyst 3500 Installation Guide at 1-1  **Features** The Catalyst 3500 series XL switches—also referred to as Catalyst 3500 XL switches—are stackable 10/100 Ethernet switches to which you can connect workstations and Cisco IP Phones and other network devices such as servers, routers, and other switches. These switches also can be deployed as backbone switches, aggregating 10/100 and Gigabit Ethernet traffic from other network devices. A feature specific to the Catalyst 3524-PWR XL switch is its ability to provide inline power to Cisco IP Phones. (Phone adapters are not required when connecting to the Catalyst 3524-PWR XL 10/100 switch ports.) Figure 1-1 shows the switch models in the series, and Table 1-1 and Table 1-2 list their features. Catalyst 3500 Installation Guide at Figure 1-23

No.	’740 Patent Claim 2	Cisco EtherChannel
		 The connection between the Catalyst 3524-PWR XL switch and the router is configured for Fast EtherChannel, increasing the bandwidth to 200 Mbps (400 Mbps in full duplex). *Figure 1-23 Collapsed Backbone and Switch Cluster Configuration*  Gigabit servers Cisco CallManager Catalyst 4908G-L3 switch Cisco 2600 router 1 Gbps (2 Gbps full duplex) 200 Mbps Fast EtherChannel (400 Mbps full duplex Fast EtherChannel) Catalyst 3500 XL and 2900 XL GigaStack cluster Catalyst 2900 XL, 1900, and 2820 cluster Catalyst 3524-PWR XL GigaStack cluster IP Cisco IP Phones AC power source Workstations running Cisco SoftPhone software Cisco IP Phones 33092 Catalyst 3560 Configuration Guide at 1-14

No.	’740 Patent Claim 2	Cisco EtherChannel
		Excerpts below



Design Concepts for Using the Switch

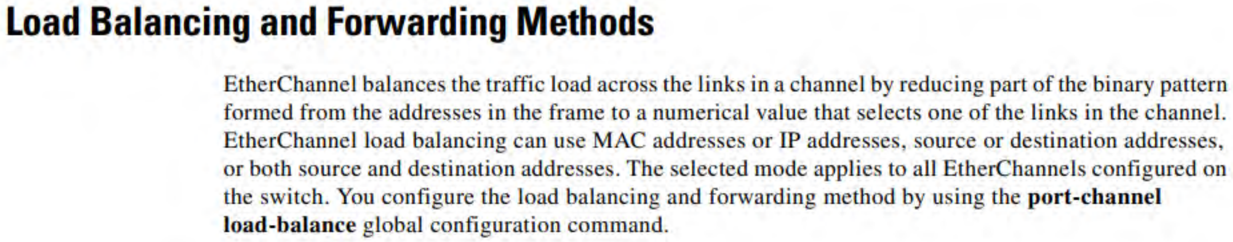
As your network users compete for network bandwidth, it takes longer to send and receive data. When you configure your network, consider the bandwidth required by your network users and the relative priority of the network applications they use.

Table 1-1 describes what can cause network performance to degrade and how you can configure your network to increase the bandwidth available to your network users.

Table 1-1 Increasing Network Performance

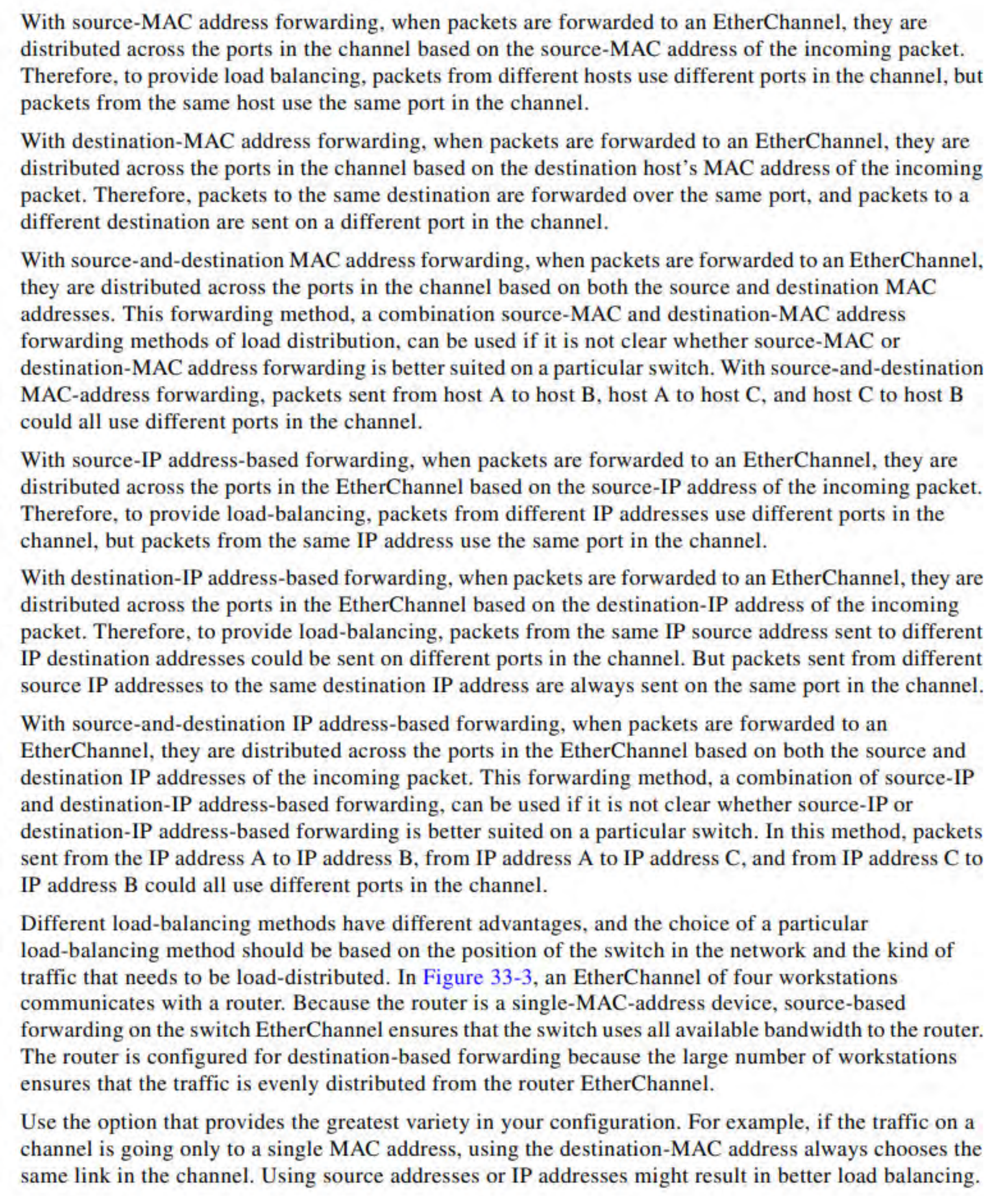
Network Demands	Suggested Design Methods
Too many users on a single network segment and a growing number of users accessing the Internet	• Create smaller network segments so that fewer users share the bandwidth, and use VLANs and IP subnets to place the network resources in the same logical network as the users who access those resources most. • Use full-duplex operation between the switch and its connected workstations.
• Increased power of new PCs, workstations, and servers • High bandwidth demand from networked applications (such as e-mail with large attached files) and from bandwidth-intensive applications (such as multimedia)	• Connect global resources—such as servers and routers to which the network users require equal access—directly to the high-speed switch ports so that they have their own high-speed segment. • Use the EtherChannel feature between the switch and its connected servers and routers.

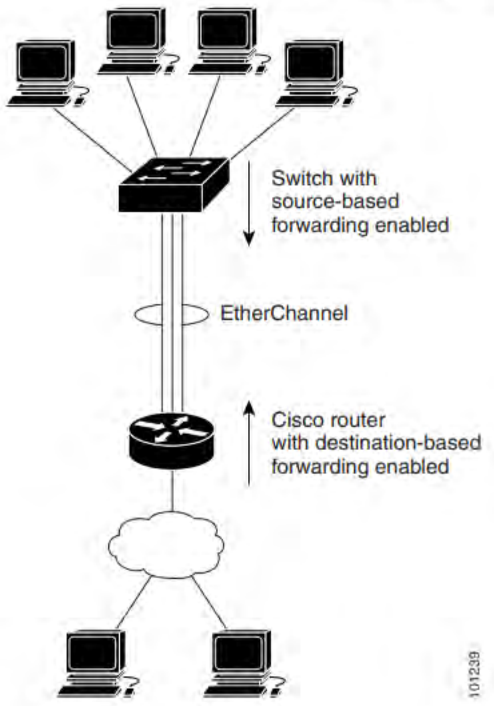
Catalyst 3560 Configuration Guide at 33-6 – 33-8



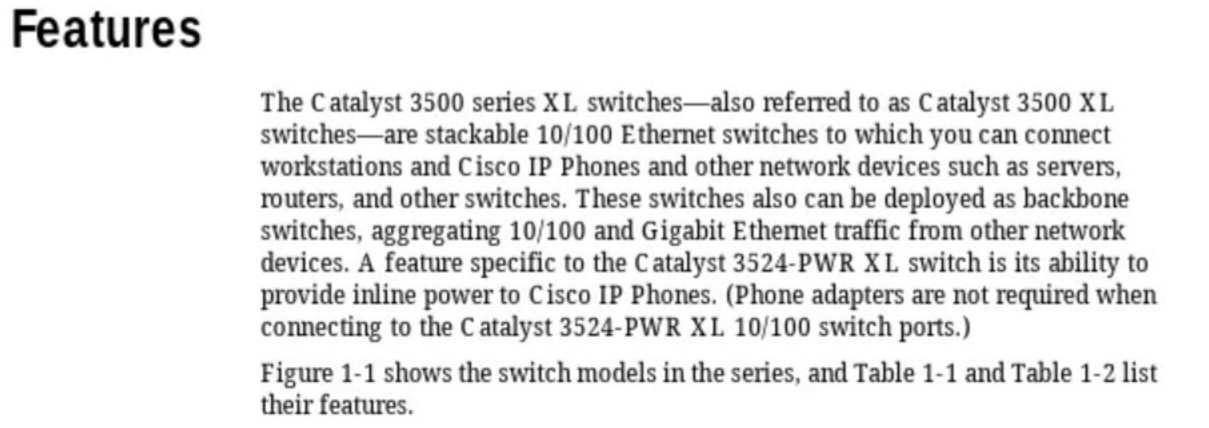
Load Balancing and Forwarding Methods

EtherChannel balances the traffic load across the links in a channel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel. EtherChannel load balancing can use MAC addresses or IP addresses, source or destination addresses, or both source and destination addresses. The selected mode applies to all EtherChannels configured on the switch. You configure the load balancing and forwarding method by using the **port-channel load-balance** global configuration command.

		 With source-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the source-MAC address of the incoming packet. Therefore, to provide load balancing, packets from different hosts use different ports in the channel, but packets from the same host use the same port in the channel. With destination-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the destination host's MAC address of the incoming packet. Therefore, packets to the same destination are forwarded over the same port, and packets to a different destination are sent on a different port in the channel. With source-and-destination MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on both the source and destination MAC addresses. This forwarding method, a combination source-MAC and destination-MAC address forwarding methods of load distribution, can be used if it is not clear whether source-MAC or destination-MAC address forwarding is better suited on a particular switch. With source-and-destination MAC-address forwarding, packets sent from host A to host B, host A to host C, and host C to host B could all use different ports in the channel. With source-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the source-IP address of the incoming packet. Therefore, to provide load-balancing, packets from different IP addresses use different ports in the channel, but packets from the same IP address use the same port in the channel. With destination-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the destination-IP address of the incoming packet. Therefore, to provide load-balancing, packets from the same IP source address sent to different IP destination addresses could be sent on different ports in the channel. But packets sent from different source IP addresses to the same destination IP address are always sent on the same port in the channel. With source-and-destination IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on both the source and destination IP addresses of the incoming packet. This forwarding method, a combination of source-IP and destination-IP address-based forwarding, can be used if it is not clear whether source-IP or destination-IP address-based forwarding is better suited on a particular switch. In this method, packets sent from the IP address A to IP address B, from IP address A to IP address C, and from IP address C to IP address B could all use different ports in the channel. Different load-balancing methods have different advantages, and the choice of a particular load-balancing method should be based on the position of the switch in the network and the kind of traffic that needs to be load-distributed. In Figure 33-3, an EtherChannel of four workstations communicates with a router. Because the router is a single-MAC-address device, source-based forwarding on the switch EtherChannel ensures that the switch uses all available bandwidth to the router. The router is configured for destination-based forwarding because the large number of workstations ensures that the traffic is evenly distributed from the router EtherChannel. Use the option that provides the greatest variety in your configuration. For example, if the traffic on a channel is going only to a single MAC address, using the destination-MAC address always chooses the same link in the channel. Using source addresses or IP addresses might result in better load balancing.

No.	’740 Patent Claim 2	Cisco EtherChannel
		Figure 33-3 Load Distribution and Forwarding Methods  Switch with source-based forwarding enabled EtherChannel Cisco router with destination-based forwarding enabled 101239
2[b]	wherein sending the data frame comprises establishing a communication service between the user node and the	Cisco EtherChannel System discloses wherein sending the data frame comprises establishing a communication service between the user node and the communication network. For example, Cisco EtherChannel System discloses forwarding data packets between a phone, router, switch, or server and the larger communication network constitutes establishing a communication service connection.

No.	’740 Patent Claim 2	Cisco EtherChannel
	communication network.	Cisco Catalyst 6500 Data Sheet at 13 (“How Cisco Express Forwarding Works Cisco Express Forwarding is a Layer 3 technology that provides increased forwarding scalability and performance to manage the many short-duration traffic flows common in today’s enterprise and service provider networks. To meet the needs of environments managing large amounts of short-flow, Web-based, or highly interactive types of traffic, Cisco Express Forwarding forwards all packets in hardware, and maintains its forwarding rate independent of the number of flows going though the switch. On the Cisco Catalyst 6500 Series, the Cisco Express Forwarding Layer 3 forwarding engine is located centrally on the supervisor engine’s policy feature card (PFC2 or PFC3)—the same device that performs hardware-based Layer 2 and Layer 3 forwarding, access control list (ACL) checking, QoS policing and marking, and NetFlow statistics gathering. Using the routing table that Cisco IOS Software builds to define configured interfaces and routing protocols, the Cisco Express Forwarding architecture creates Cisco Express Forwarding tables and downloads them into the hardware forwarding engine before any user traffic is sent through the switch. The Cisco Express Forwarding architecture places only the routing prefixes in its Cisco Express Forwarding tables—the only information it requires to make the Layer 3 forwarding decisions—relying on the routing protocols to do route selection. By performing a simple Cisco Express Forwarding table lookup, the switch forwards packets at wire rate, independent of the number of flows transiting the switch.”) Catalyst 3500 Installation Guide at 1-1

No.	’740 Patent Claim 2	Cisco EtherChannel
		 **Features** The Catalyst 3500 series XL switches—also referred to as Catalyst 3500 XL switches—are stackable 10/100 Ethernet switches to which you can connect workstations and Cisco IP Phones and other network devices such as servers, routers, and other switches. These switches also can be deployed as backbone switches, aggregating 10/100 and Gigabit Ethernet traffic from other network devices. A feature specific to the Catalyst 3524-PWR XL switch is its ability to provide inline power to Cisco IP Phones. (Phone adapters are not required when connecting to the Catalyst 3524-PWR XL 10/100 switch ports.) Figure 1-1 shows the switch models in the series, and Table 1-1 and Table 1-2 list their features. Catalyst 3500 Installation Guide at Table 1-1

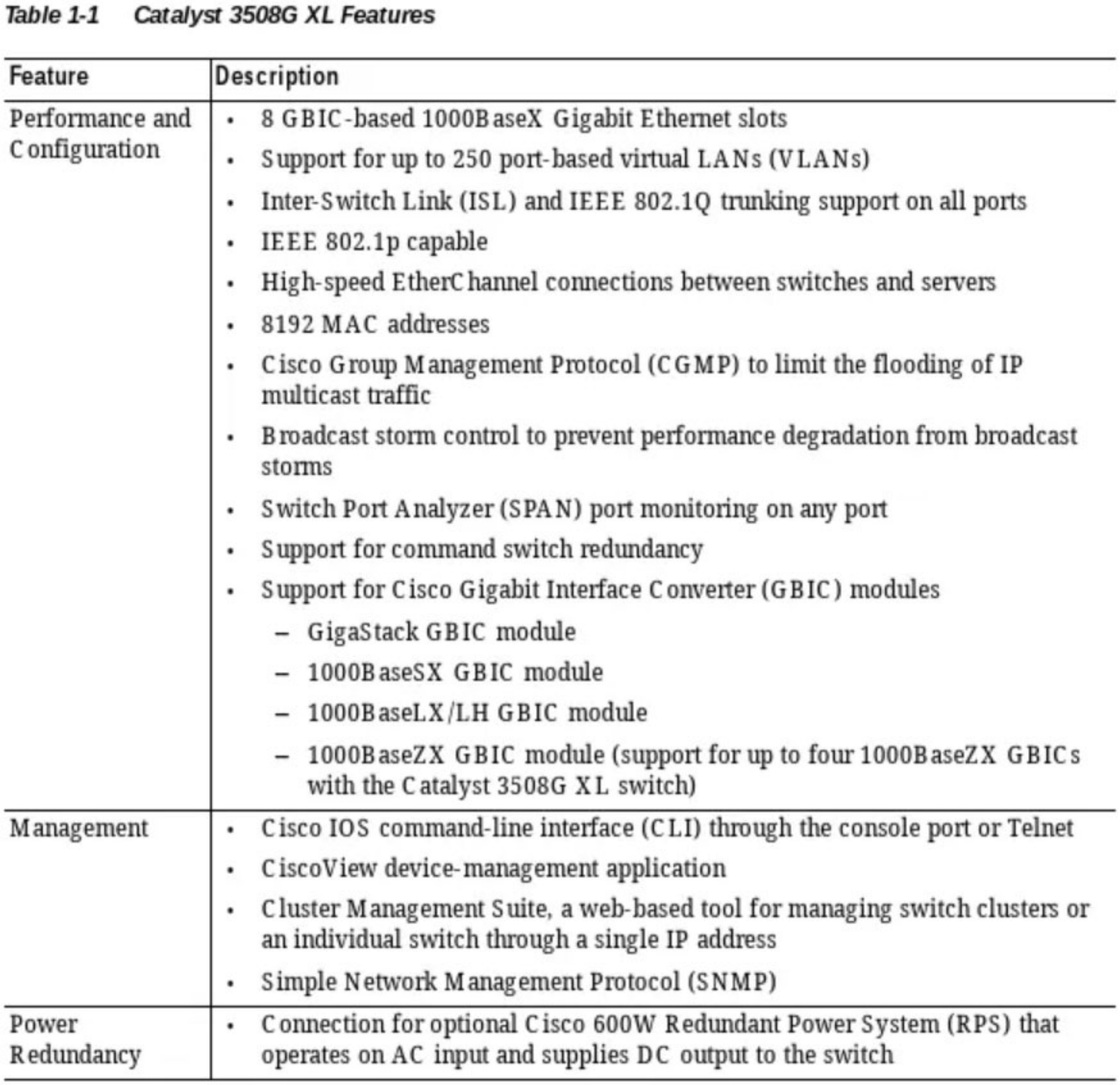
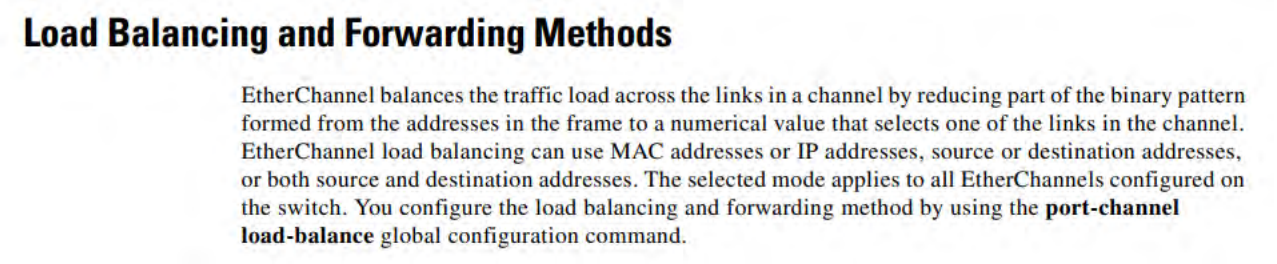
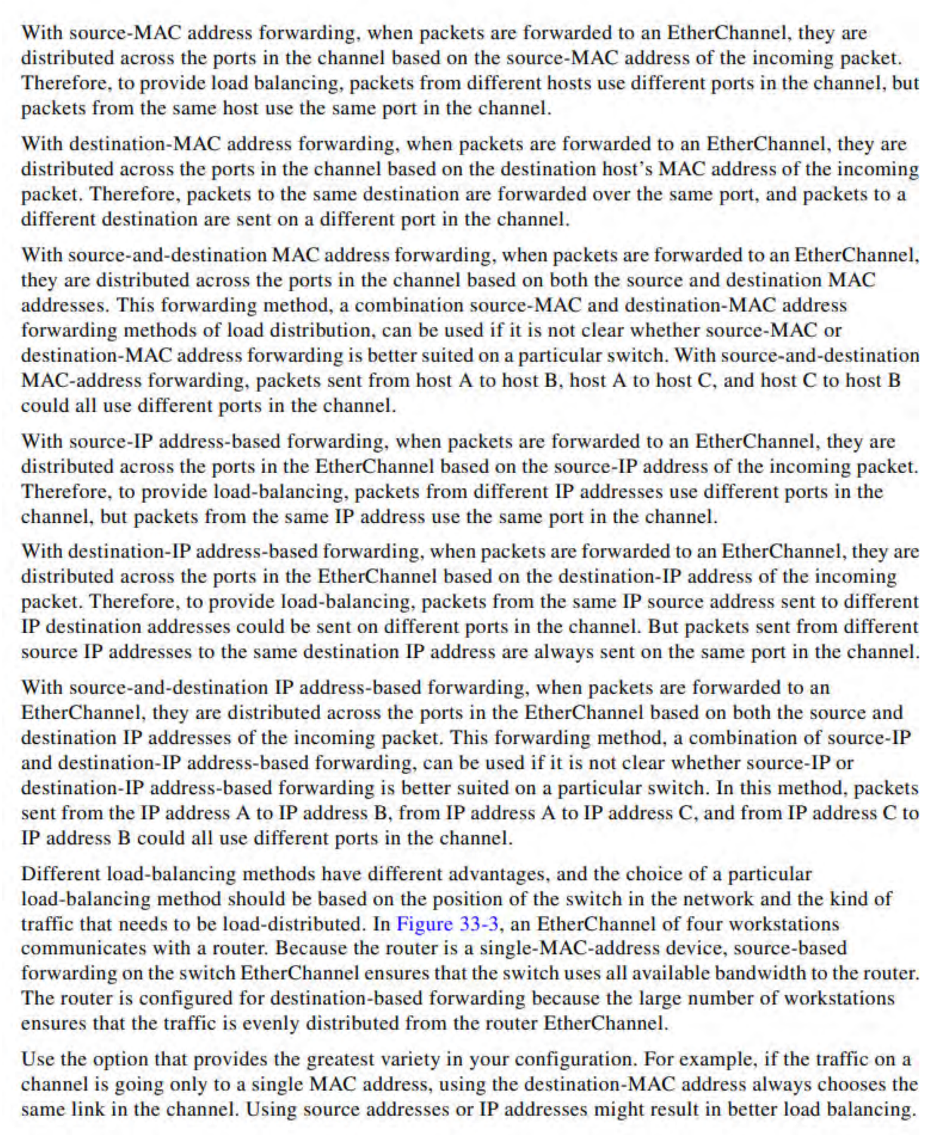
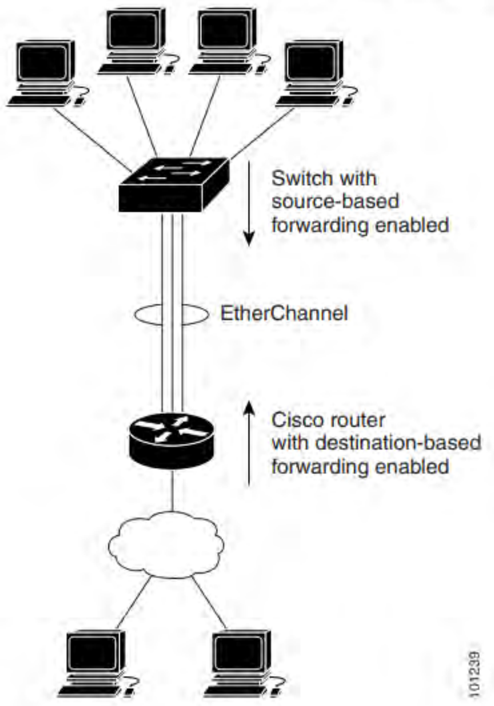
No.	’740 Patent Claim 2	Cisco EtherChannel
		

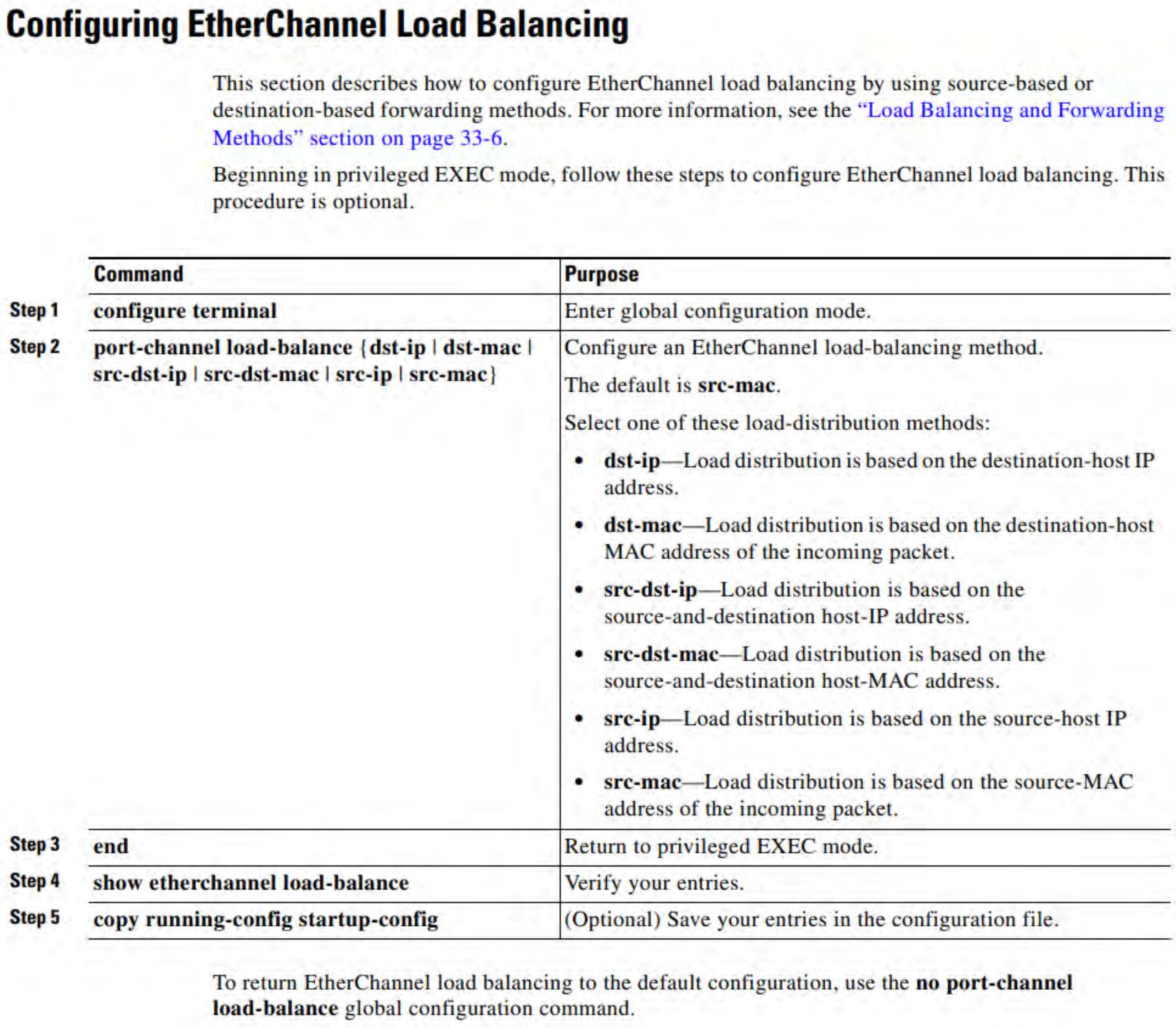
Table 1-1 Catalyst 3508G XL Features

Feature	Description
Performance and Configuration	• 8 GBIC-based 1000BaseX Gigabit Ethernet slots • Support for up to 250 port-based virtual LANs (VLANs) • Inter-Switch Link (ISL) and IEEE 802.1Q trunking support on all ports • IEEE 802.1p capable • High-speed EtherChannel connections between switches and servers • 8192 MAC addresses • Cisco Group Management Protocol (CGMP) to limit the flooding of IP multicast traffic • Broadcast storm control to prevent performance degradation from broadcast storms • Switch Port Analyzer (SPAN) port monitoring on any port • Support for command switch redundancy • Support for Cisco Gigabit Interface Converter (GBIC) modules – GigaStack GBIC module – 1000BaseSX GBIC module – 1000BaseLX/LH GBIC module – 1000BaseZX GBIC module (support for up to four 1000BaseZX GBICs with the Catalyst 3508G XL switch)
Management	• Cisco IOS command-line interface (CLI) through the console port or Telnet • CiscoView device-management application • Cluster Management Suite, a web-based tool for managing switch clusters or an individual switch through a single IP address • Simple Network Management Protocol (SNMP)
Power Redundancy	• Connection for optional Cisco 600W Redundant Power System (RPS) that operates on AC input and supplies DC output to the switch

No.	’740 Patent Claim 2	Cisco EtherChannel
		Catalyst 3560 Configuration Guide at 33-6 – 33-8  **Load Balancing and Forwarding Methods** EtherChannel balances the traffic load across the links in a channel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel. EtherChannel load balancing can use MAC addresses or IP addresses, source or destination addresses, or both source and destination addresses. The selected mode applies to all EtherChannels configured on the switch. You configure the load balancing and forwarding method by using the **port-channel load-balance** global configuration command.

No.	’740 Patent Claim 2	Cisco EtherChannel
		 With source-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the source-MAC address of the incoming packet. Therefore, to provide load balancing, packets from different hosts use different ports in the channel, but packets from the same host use the same port in the channel. With destination-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the destination host’s MAC address of the incoming packet. Therefore, packets to the same destination are forwarded over the same port, and packets to a different destination are sent on a different port in the channel. With source-and-destination MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on both the source and destination MAC addresses. This forwarding method, a combination source-MAC and destination-MAC address forwarding methods of load distribution, can be used if it is not clear whether source-MAC or destination-MAC address forwarding is better suited on a particular switch. With source-and-destination MAC-address forwarding, packets sent from host A to host B, host A to host C, and host C to host B could all use different ports in the channel. With source-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the source-IP address of the incoming packet. Therefore, to provide load-balancing, packets from different IP addresses use different ports in the channel, but packets from the same IP address use the same port in the channel. With destination-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the destination-IP address of the incoming packet. Therefore, to provide load-balancing, packets from the same IP source address sent to different IP destination addresses could be sent on different ports in the channel. But packets sent from different source IP addresses to the same destination IP address are always sent on the same port in the channel. With source-and-destination IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on both the source and destination IP addresses of the incoming packet. This forwarding method, a combination of source-IP and destination-IP address-based forwarding, can be used if it is not clear whether source-IP or destination-IP address-based forwarding is better suited on a particular switch. In this method, packets sent from the IP address A to IP address B, from IP address A to IP address C, and from IP address C to IP address B could all use different ports in the channel. Different load-balancing methods have different advantages, and the choice of a particular load-balancing method should be based on the position of the switch in the network and the kind of traffic that needs to be load-distributed. In Figure 33-3, an EtherChannel of four workstations communicates with a router. Because the router is a single-MAC-address device, source-based forwarding on the switch EtherChannel ensures that the switch uses all available bandwidth to the router. The router is configured for destination-based forwarding because the large number of workstations ensures that the traffic is evenly distributed from the router EtherChannel. Use the option that provides the greatest variety in your configuration. For example, if the traffic on a channel is going only to a single MAC address, using the destination-MAC address always chooses the same link in the channel. Using source addresses or IP addresses might result in better load balancing.

No.	’740 Patent Claim 2	Cisco EtherChannel
		Figure 33-3 Load Distribution and Forwarding Methods  Switch with source-based forwarding enabled EtherChannel Cisco router with destination-based forwarding enabled 101239 Catalyst 3560 Configuration Guide at 33-16

No.	’740 Patent Claim 2	Cisco EtherChannel
		

Configuring EtherChannel Load Balancing

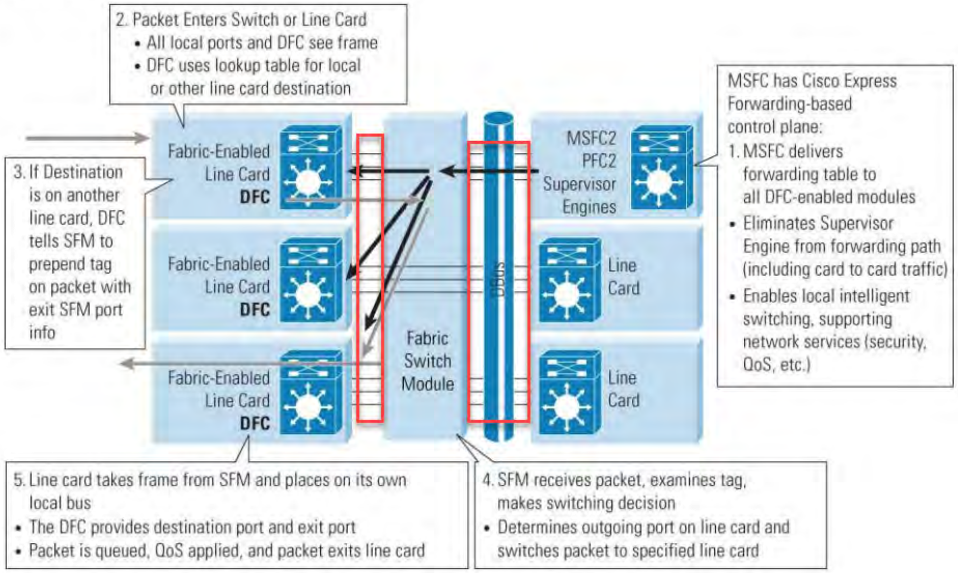
This section describes how to configure EtherChannel load balancing by using source-based or destination-based forwarding methods. For more information, see the “Load Balancing and Forwarding Methods” section on page 33-6.

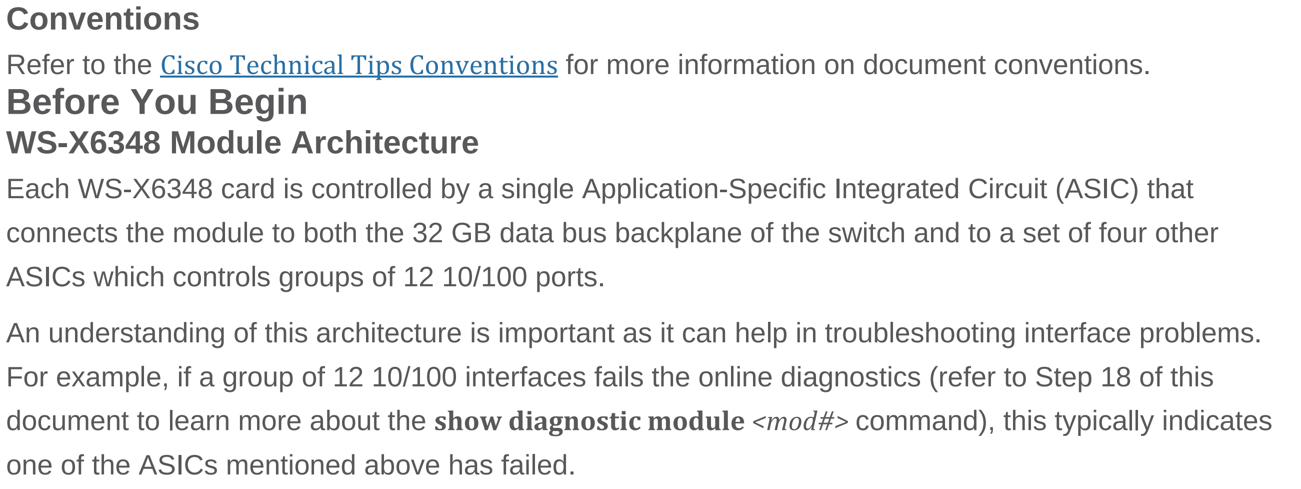
Beginning in privileged EXEC mode, follow these steps to configure EtherChannel load balancing. This procedure is optional.

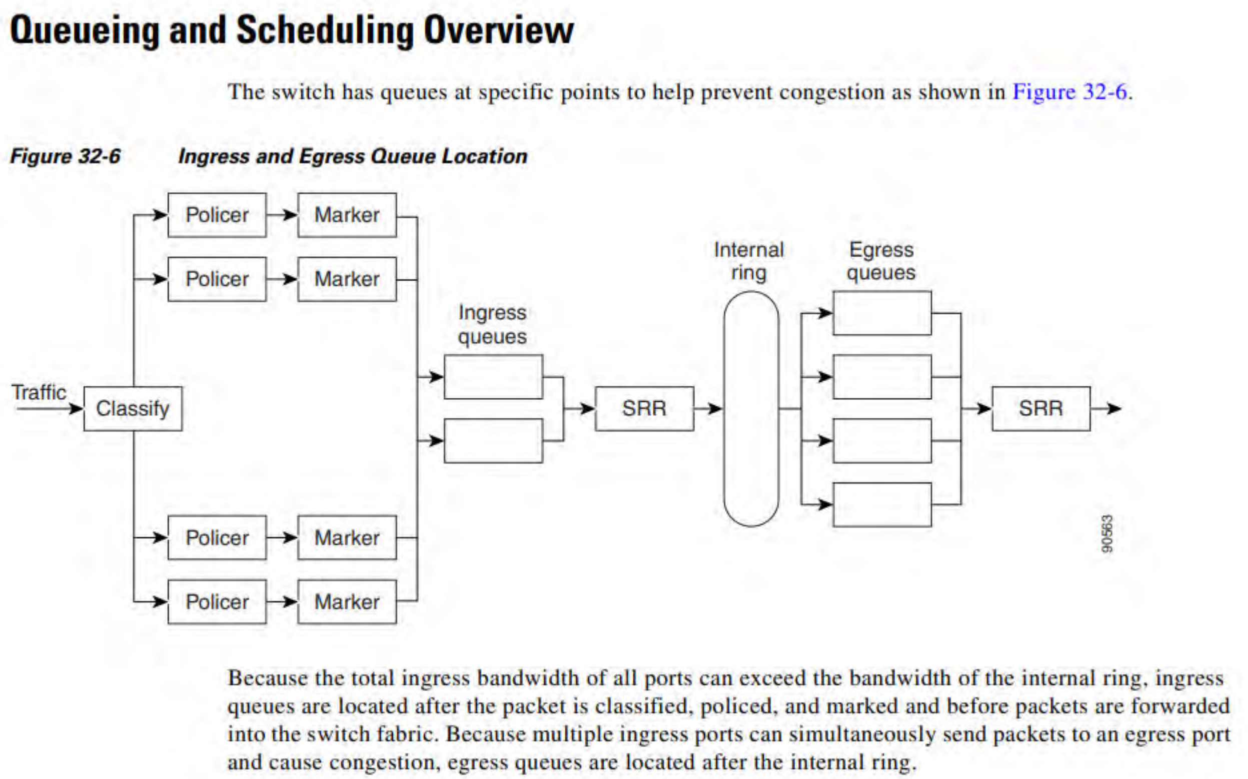
	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**port-channel load-balance** {**dst-ip** | **dst-mac** | **src-dst-ip** | **src-dst-mac** | **src-ip** | **src-mac**}	Configure an EtherChannel load-balancing method. The default is **src-mac**. Select one of these load-distribution methods: • **dst-ip**—Load distribution is based on the destination-host IP address. • **dst-mac**—Load distribution is based on the destination-host MAC address of the incoming packet. • **src-dst-ip**—Load distribution is based on the source-and-destination host-IP address. • **src-dst-mac**—Load distribution is based on the source-and-destination host-MAC address. • **src-ip**—Load distribution is based on the source-host IP address. • **src-mac**—Load distribution is based on the source-MAC address of the incoming packet.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show etherchannel load-balance**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return EtherChannel load balancing to the default configuration, use the **no port-channel load-balance** global configuration command.

No.	’740 Patent Claim 3	Cisco EtherChannel
3	The method according to claim 1, wherein the second physical links comprise backplane traces formed on a backplane to which the one or more interface modules are coupled.	Cisco EtherChannel System discloses the method according to claim 1, wherein the second physical links comprise backplane traces formed on a backplane to which the one or more interface modules are coupled. For example, Cisco EtherChannel System discloses switch fabric interconnections which form on the switch fabric and connect it to the interface modules or line cards. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Cisco EtherChannel System is found to not meet this limitation, wherein the second physical links comprise backplane traces formed on a back plane to which the one or more interface modules are coupled would have been obvious to a person having ordinary skill in the art, as explained below. Cisco Catalyst 6500 Data Sheet at 13 (“**Switch Fabric Modules** Designed to support distributed forwarding for interface modules with that capability, the Cisco Catalyst 6500 Series Switch Fabric Module (SFM or SFM2), in combination with the Cisco Catalyst 6000 Multilayer Switch Feature Card (MSFC2) and Cisco distributed forwarding cards (DFCs) on interface modules, increases available system bandwidth from 32 to 256 Gbps. The SFM or SFM2 supports the Cisco Catalyst 6500 CEF256 and dCEF256 interface modules. Designed to support new interface modules with 720-Gbps forwarding capabilities, the switch fabric onboard the Cisco Catalyst 6500 Series Supervisor Engine 720 increases available bandwidth to 720 Gbps and provides packet-forwarding rates up to 400 mpps. By using automatic sensing and negotiation, the switch fabric is fully interoperable with the 8- and 16-Gbps switch-fabric interconnections used by the CEF256 and dCEF256 interface modules. When a CEF256 or dCEF256 interface module is detected, the switch fabric will automatically connect those modules by offering 8 to 16 Gbps of bandwidth to each module, as applicable.”) Cisco Catalyst 6500 Data Sheet at Figure 6

No.	’740 Patent Claim 3	Cisco EtherChannel
		Figure 6. Distributed Cisco Express Forwarding Packet Flow  2. Packet Enters Switch or Line Card • All local ports and DFC see frame • DFC uses lookup table for local or other line card destination 3. If Destination is on another line card, DFC tells SFM to prepend tag on packet with exit SFM port info Fabric-Enabled Line Card DFC Fabric-Enabled Line Card DFC Fabric-Enabled Line Card DFC Fabric Switch Module DBus MSFC2 PFC2 Supervisor Engines Line Card Line Card MSFC has Cisco Express Forwarding-based control plane: 1. MSFC delivers forwarding table to all DFC-enabled modules • Eliminates Supervisor Engine from forwarding path (including card to card traffic) • Enables local intelligent switching, supporting network services (security, QoS, etc.) 5. Line card takes frame from SFM and places on its own local bus • The DFC provides destination port and exit port • Packet is queued, QoS applied, and packet exits line card 4. SFM receives packet, examines tag, makes switching decision • Determines outgoing port on line card and switches packet to specified line card Catalyst 6500/6000 Module **Components Used** The information in this document is based on these software and hardware versions: • Catalyst 6500 with Supervisor II with Multilayer Switch Feature Card 2 (MSFC2) • WS-X6348 module • Cisco IOS version 12.1(11b)E4 The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

No.	’740 Patent Claim 3	Cisco EtherChannel
		 **Conventions** Refer to the Cisco Technical Tips Conventions for more information on document conventions. **Before You Begin** **WS-X6348 Module Architecture** Each WS-X6348 card is controlled by a single Application-Specific Integrated Circuit (ASIC) that connects the module to both the 32 GB data bus backplane of the switch and to a set of four other ASICs which controls groups of 12 10/100 ports. An understanding of this architecture is important as it can help in troubleshooting interface problems. For example, if a group of 12 10/100 interfaces fails the online diagnostics (refer to Step 18 of this document to learn more about the **show diagnostic module** *<mod#>* command), this typically indicates one of the ASICs mentioned above has failed. Catalyst 3560 Configuration Guide at Figure 32-6

No.	’740 Patent Claim 3	Cisco EtherChannel
		 **Queueing and Scheduling Overview** The switch has queues at specific points to help prevent congestion as shown in Figure 32-6. *Figure 32-6 Ingress and Egress Queue Location* Policer Marker Policer Marker Internal ring Egress queues Ingress queues Traffic Classify SRR SRR Policer Marker Policer Marker 90563 Because the total ingress bandwidth of all ports can exceed the bandwidth of the internal ring, ingress queues are located after the packet is classified, policed, and marked and before packets are forwarded into the switch fabric. Because multiple ingress ports can simultaneously send packets to an egress port and cause congestion, egress queues are located after the internal ring. Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Cisco EtherChannel System in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 3 of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of three such references. For example, Ghosh discloses connecting the line cards to the active supervisor via the backplane using parallel interface circuitry.

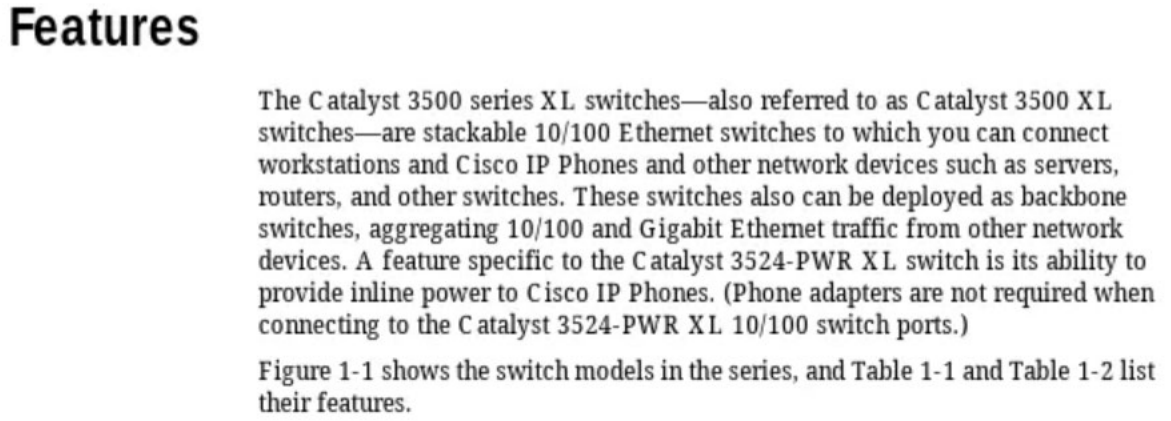
No.	’740 Patent Claim 3	Cisco EtherChannel
		Ghosh at [0059] (“Line cards 803, 805, and 807 can communicate with an active supervisor 811 through interface circuitry 883, 885, and 887 and the backplane 815. According to various embodiments, each line card includes a plurality of ports that can act as either input ports or output ports for communication with external fibre channel network entities 851 and 853. The backplane 815 can provide a communi-cations channel for all traffic between line cards and super- visors. Individual line cards 803 and 807 can also be coupled to external fibre channel network entities 851 and 853 through fibre channel ports 843 and 847.”) For example, Bruckman discloses connecting the line cards to the network using traces comprising a backplane. Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”) For example, Basso discloses coupling the blades to the communication network via link connections to the switch fabric. Basso at [0009] (“A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more ports. These ports may be connected to another one or more network devices. The switch fabric may be configured to direct incoming packets of data to particular blades where one or more

No.	’740 Patent Claim 3	Cisco EtherChannel
		of the network proces-sors in the recipient blade may be configured to process the received packets.”)

No.	’740 Patent Claim 4	Cisco EtherChannel
4[preamble]	A method for communication, comprising:	Cisco EtherChannel System discloses a method for communication. *See supra* at 1[preamble].
4[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	Cisco EtherChannel System discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].
4[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Cisco EtherChannel System discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra* at 1[c].
4[c]	receiving a data frame having	Cisco EtherChannel System discloses receiving a data frame having frame attributes sent between the communication network and the network node.

No.	’740 Patent Claim 4	Cisco EtherChannel
	frame attributes sent between the communication network and the network node:	*See supra* at 1[e].
4[d]	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	Cisco EtherChannel System discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. *See supra* at 1[f].
4[e]	sending the data frame over the selected first and second physical links,	Cisco EtherChannel System discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].
4[f]	at least one of the first and second groups of physical links	Cisco EtherChannel System discloses at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group. For example, Cisco EtherChannel System discloses link aggregation technology in which at least Ethernet ports are aggregated to form a single logical Ethernet channel. Thus, at least under the

No.	’740 Patent Claim 4	Cisco EtherChannel
	comprising an Ethernet link aggregation (LAG) group.	apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Cisco EtherChannel System is found to not meet this limitation, at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group would have been obvious to a person having ordinary skill in the art, as explained below. Cisco Catalyst 6500 Data Sheet at 2 (“Offers optional, redundant, high-performance Cisco Catalyst 6500 Series Supervisor Engine 720, passive backplane, multimodule Cisco EtherChannel® technology, IEEE 802.3ad link aggregation, IEEE 802.1s only, and Hot Standby Router Protocol/Virtual Router Redundancy Protocol (HSRP/VRRP) high-availability features”) Cisco Catalyst 6500 Data Sheet at 4 (“Ideal for Metro Ethernet WAN Services • 802.1Q and 802.1Q tunneling (QinQ), providing point-to-point and multipoint Ethernet services • Ethernet over MPLS in MPLS backbone networks for superior network scaling, providing VLAN translation capability • Layer 2 and Layer 3 quality of service (QoS), facilitating tiered Ethernet service offerings through rate limiting and traffic shaping • Superior high-availability features, including enhanced Spanning Tree Protocol, IEEE 802.1s, IEEE 802.1w, and Cisco EtherChannel IEEE 802.3ad link aggregation”) Catalyst 3500 Installation Guide at 1-1

No.	’740 Patent Claim 4	Cisco EtherChannel
		 Features The Catalyst 3500 series XL switches—also referred to as Catalyst 3500 XL switches—are stackable 10/100 Ethernet switches to which you can connect workstations and Cisco IP Phones and other network devices such as servers, routers, and other switches. These switches also can be deployed as backbone switches, aggregating 10/100 and Gigabit Ethernet traffic from other network devices. A feature specific to the Catalyst 3524-PWR XL switch is its ability to provide inline power to Cisco IP Phones. (Phone adapters are not required when connecting to the Catalyst 3524-PWR XL 10/100 switch ports.) Figure 1-1 shows the switch models in the series, and Table 1-1 and Table 1-2 list their features. Catalyst 3500 Installation Guide at Table 1-1

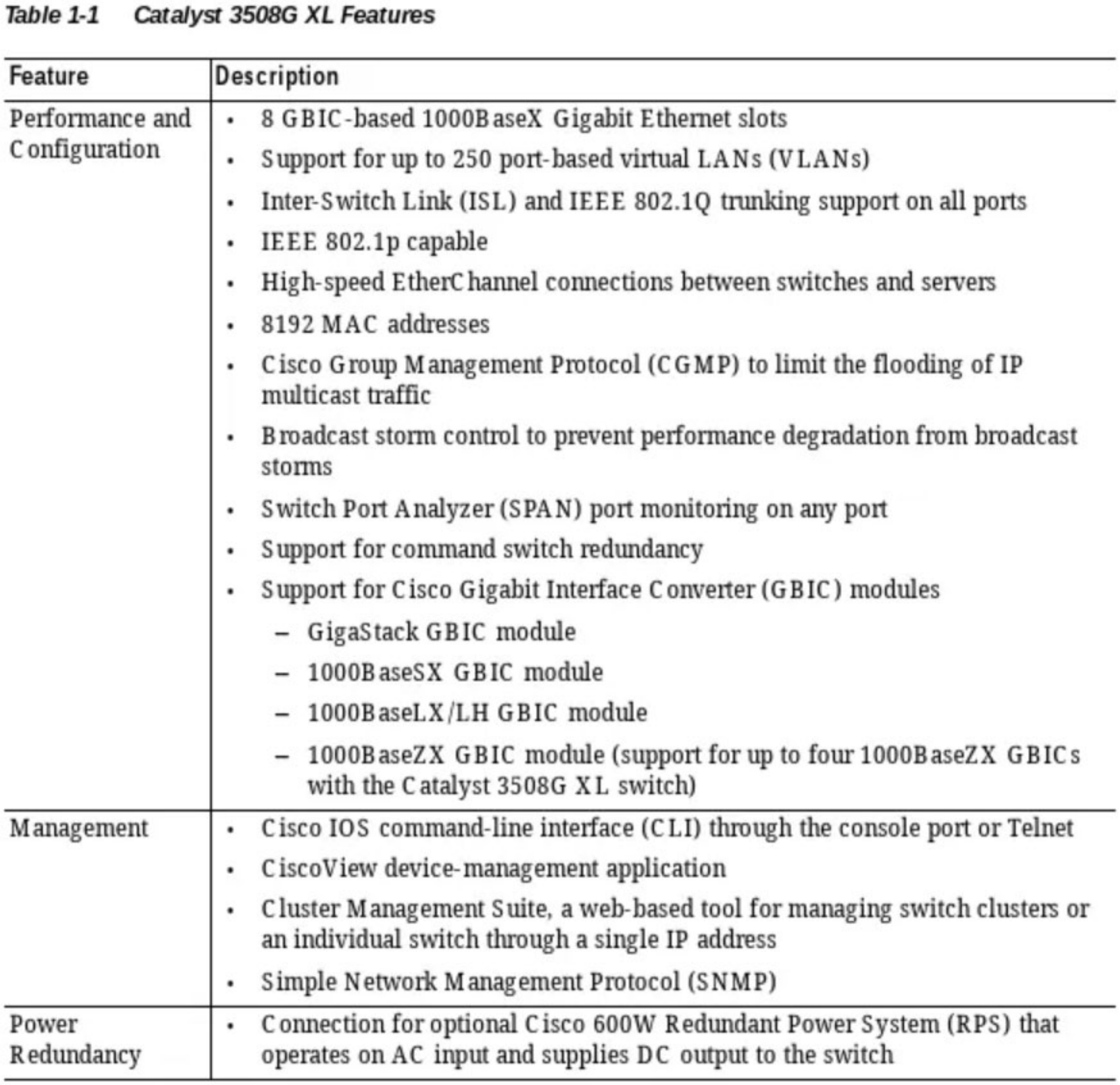
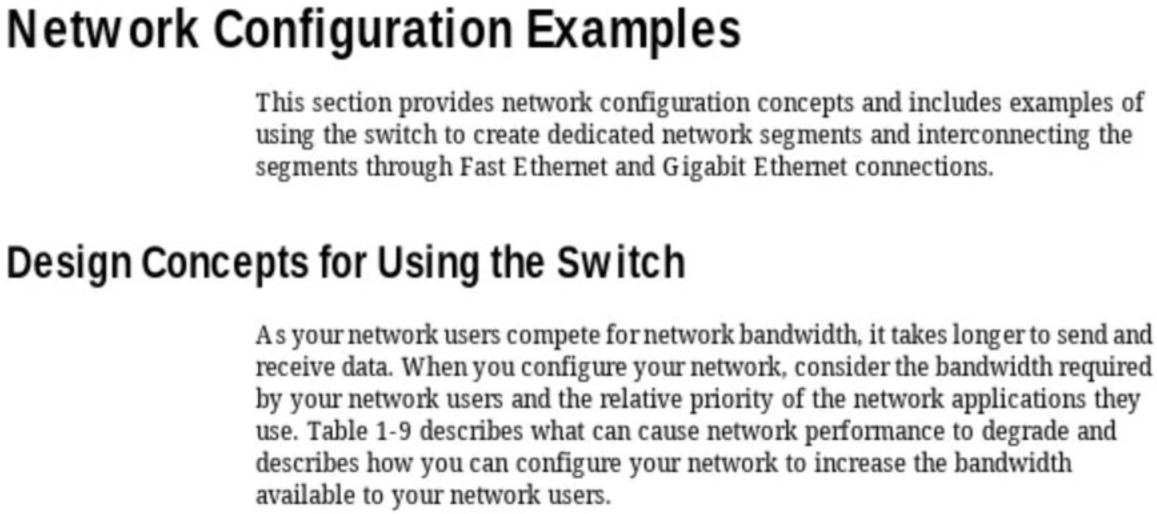
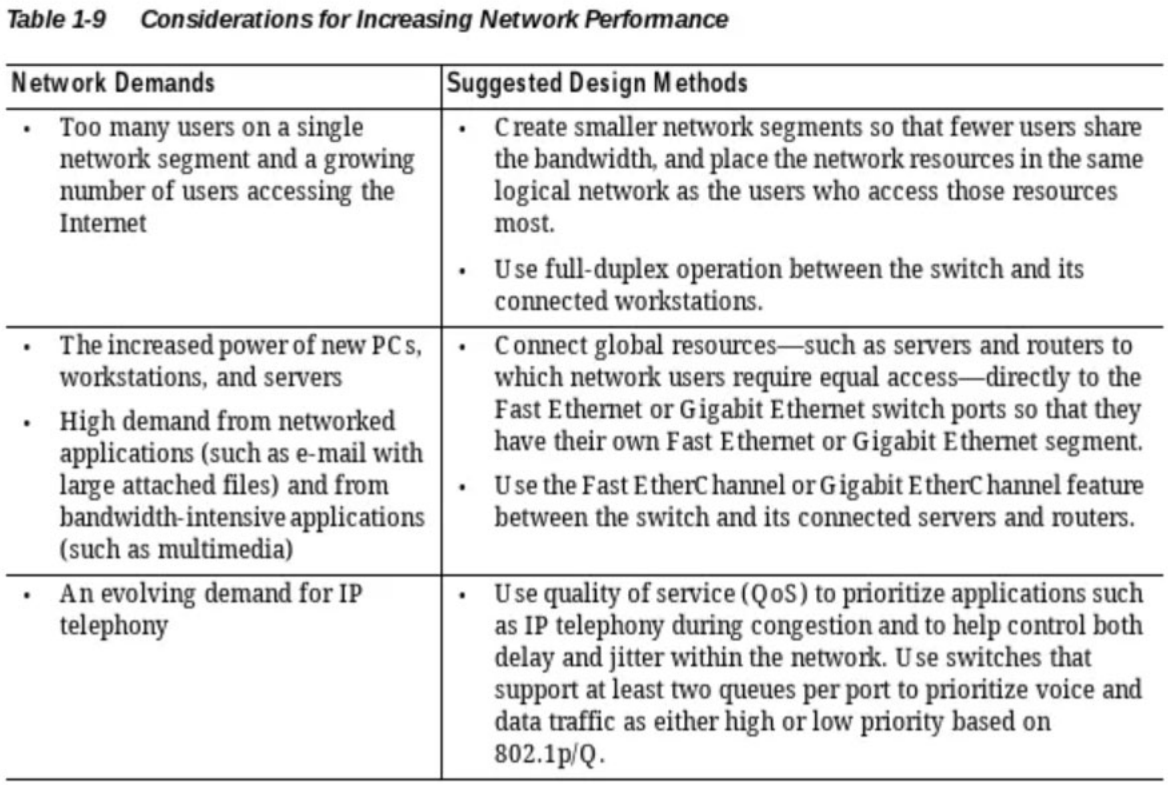
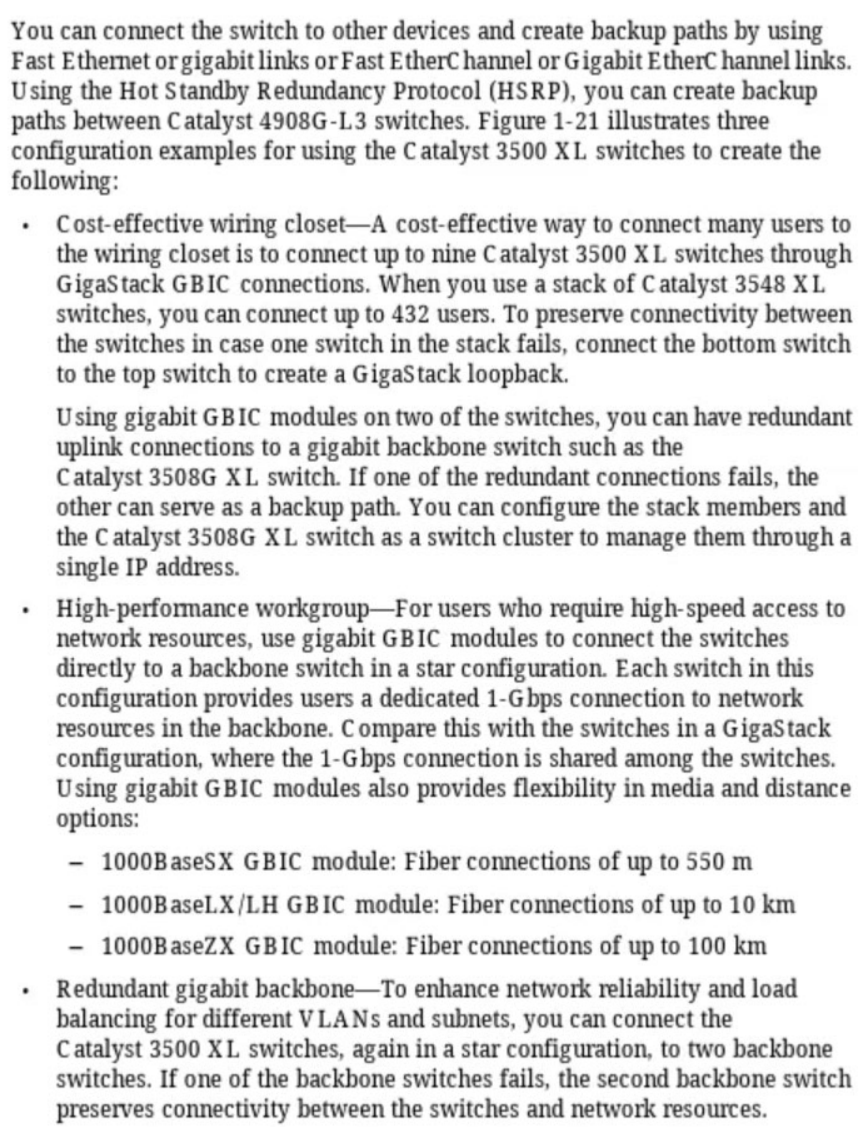
No.	’740 Patent Claim 4	Cisco EtherChannel
		

Table 1-1 Catalyst 3508G XL Features

Feature	Description
Performance and Configuration	• 8 GBIC-based 1000BaseX Gigabit Ethernet slots • Support for up to 250 port-based virtual LANs (VLANs) • Inter-Switch Link (ISL) and IEEE 802.1Q trunking support on all ports • IEEE 802.1p capable • High-speed EtherChannel connections between switches and servers • 8192 MAC addresses • Cisco Group Management Protocol (CGMP) to limit the flooding of IP multicast traffic • Broadcast storm control to prevent performance degradation from broadcast storms • Switch Port Analyzer (SPAN) port monitoring on any port • Support for command switch redundancy • Support for Cisco Gigabit Interface Converter (GBIC) modules – GigaStack GBIC module – 1000BaseSX GBIC module – 1000BaseLX/LH GBIC module – 1000BaseZX GBIC module (support for up to four 1000BaseZX GBICs with the Catalyst 3508G XL switch)
Management	• Cisco IOS command-line interface (CLI) through the console port or Telnet • CiscoView device-management application • Cluster Management Suite, a web-based tool for managing switch clusters or an individual switch through a single IP address • Simple Network Management Protocol (SNMP)
Power Redundancy	• Connection for optional Cisco 600W Redundant Power System (RPS) that operates on AC input and supplies DC output to the switch

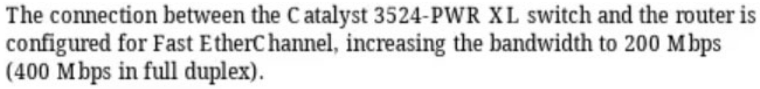
No.	’740 Patent Claim 4	Cisco EtherChannel
		Catalyst 3500 Installation Guide at 1-28  **Network Configuration Examples** This section provides network configuration concepts and includes examples of using the switch to create dedicated network segments and interconnecting the segments through Fast Ethernet and Gigabit Ethernet connections. **Design Concepts for Using the Switch** As your network users compete for network bandwidth, it takes longer to send and receive data. When you configure your network, consider the bandwidth required by your network users and the relative priority of the network applications they use. Table 1-9 describes what can cause network performance to degrade and describes how you can configure your network to increase the bandwidth available to your network users. Catalyst 3500 Installation Guide at Table 1-9

<table>
<tr><th>No.</th><th>’740 Patent Claim 4</th><th>Cisco EtherChannel</th></tr>
<tr><td></td><td></td><td>

<i>Table 1-9 Considerations for Increasing Network Performance</i>
<table>
<tr><th>Network Demands</th><th>Suggested Design Methods</th></tr>
<tr><td>• Too many users on a single network segment and a growing number of users accessing the Internet</td><td>• Create smaller network segments so that fewer users share the bandwidth, and place the network resources in the same logical network as the users who access those resources most.
• Use full-duplex operation between the switch and its connected workstations.</td></tr>
<tr><td>• The increased power of new PCs, workstations, and servers
• High demand from networked applications (such as e-mail with large attached files) and from bandwidth-intensive applications (such as multimedia)</td><td>• Connect global resources—such as servers and routers to which network users require equal access—directly to the Fast Ethernet or Gigabit Ethernet switch ports so that they have their own Fast Ethernet or Gigabit Ethernet segment.
• Use the Fast EtherChannel or Gigabit EtherChannel feature between the switch and its connected servers and routers.</td></tr>
<tr><td>• An evolving demand for IP telephony</td><td>• Use quality of service (QoS) to prioritize applications such as IP telephony during congestion and to help control both delay and jitter within the network. Use switches that support at least two queues per port to prioritize voice and data traffic as either high or low priority based on 802.1p/Q.</td></tr>
</table>
Catalyst 3500 Installation Guide at 1-30
</td></tr>
</table>

No.	’740 Patent Claim 4	Cisco EtherChannel
		 You can connect the switch to other devices and create backup paths by using Fast Ethernet or gigabit links or Fast EtherChannel or Gigabit EtherChannel links. Using the Hot Standby Redundancy Protocol (HSRP), you can create backup paths between Catalyst 4908G-L3 switches. Figure 1-21 illustrates three configuration examples for using the Catalyst 3500 XL switches to create the following: • Cost-effective wiring closet—A cost-effective way to connect many users to the wiring closet is to connect up to nine Catalyst 3500 XL switches through GigaStack GBIC connections. When you use a stack of Catalyst 3548 XL switches, you can connect up to 432 users. To preserve connectivity between the switches in case one switch in the stack fails, connect the bottom switch to the top switch to create a GigaStack loopback. Using gigabit GBIC modules on two of the switches, you can have redundant uplink connections to a gigabit backbone switch such as the Catalyst 3508G XL switch. If one of the redundant connections fails, the other can serve as a backup path. You can configure the stack members and the Catalyst 3508G XL switch as a switch cluster to manage them through a single IP address. • High-performance workgroup—For users who require high-speed access to network resources, use gigabit GBIC modules to connect the switches directly to a backbone switch in a star configuration. Each switch in this configuration provides users a dedicated 1-Gbps connection to network resources in the backbone. Compare this with the switches in a GigaStack configuration, where the 1-Gbps connection is shared among the switches. Using gigabit GBIC modules also provides flexibility in media and distance options: – 1000BaseSX GBIC module: Fiber connections of up to 550 m – 1000BaseLX/LH GBIC module: Fiber connections of up to 10 km – 1000BaseZX GBIC module: Fiber connections of up to 100 km • Redundant gigabit backbone—To enhance network reliability and load balancing for different VLANs and subnets, you can connect the Catalyst 3500 XL switches, again in a star configuration, to two backbone switches. If one of the backbone switches fails, the second backbone switch preserves connectivity between the switches and network resources.

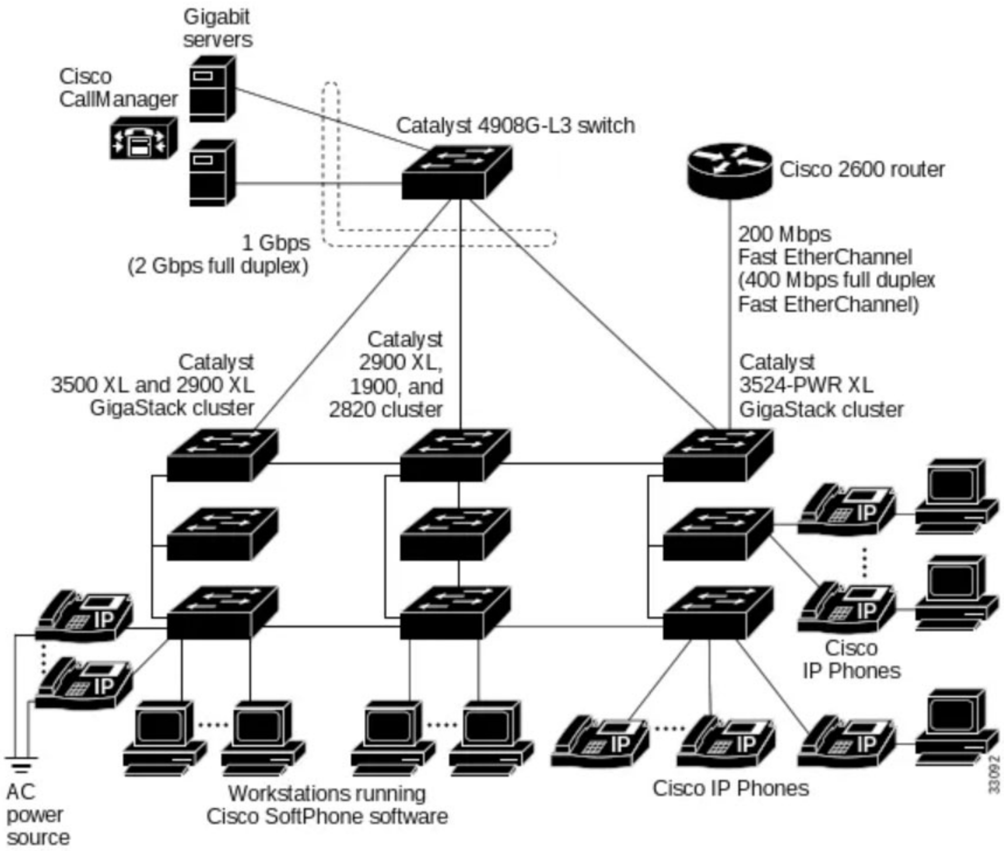
<table>
<tr><th>No.</th><th>’740 Patent Claim 4</th><th>Cisco EtherChannel</th></tr>
<tr><td></td><td></td><td>

Catalyst 3500 Installation Guide at Figure 1-23

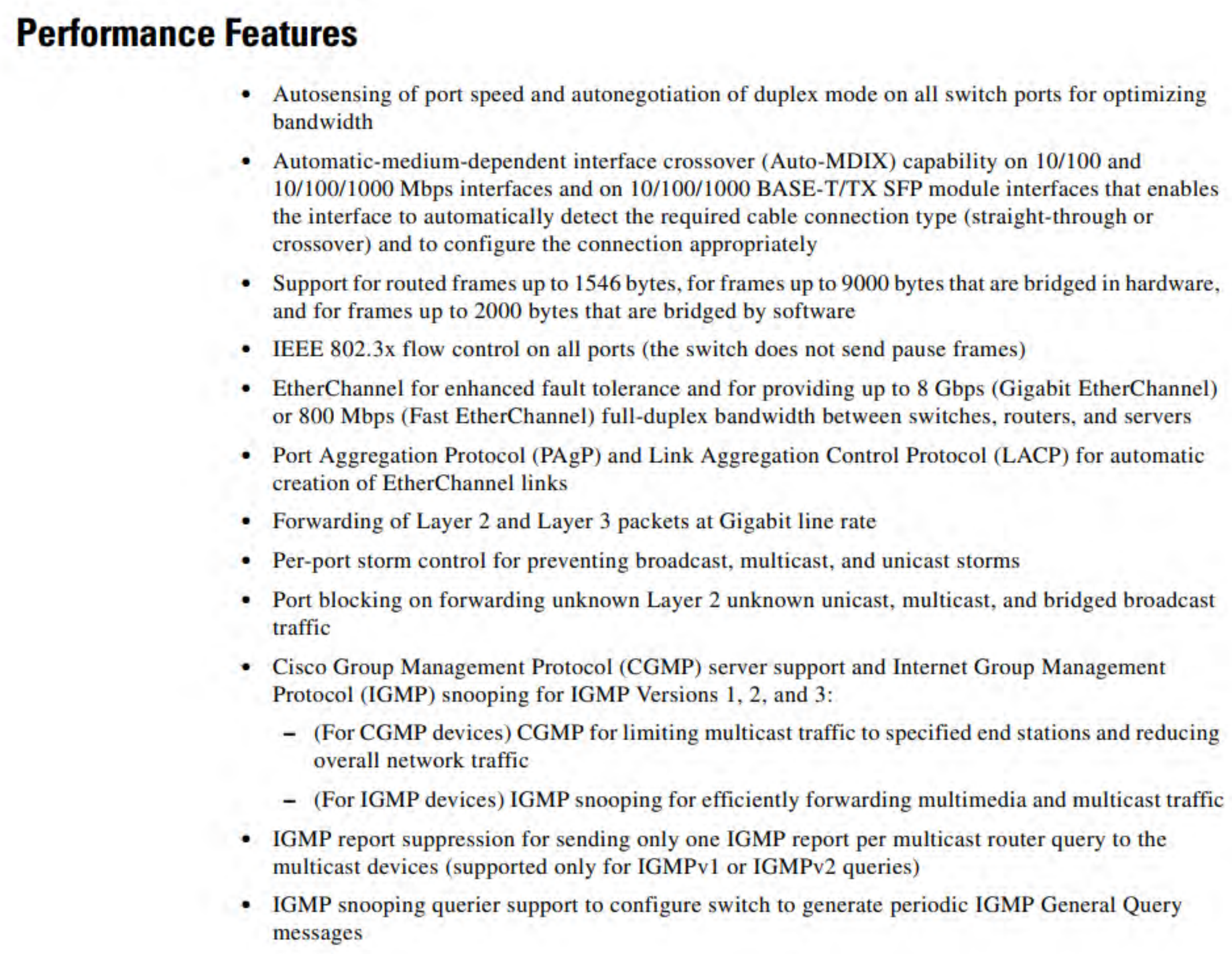


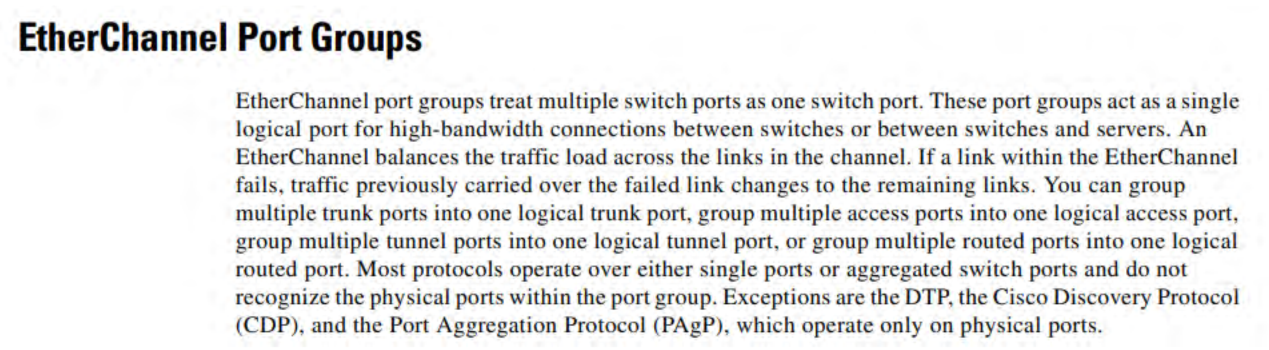
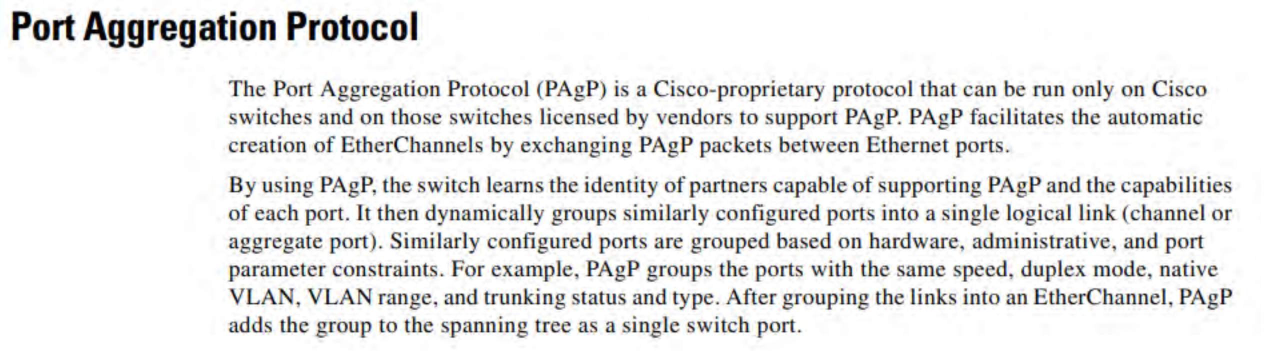
The connection between the Catalyst 3524-PWR XL switch and the router is configured for Fast EtherChannel, increasing the bandwidth to 200 Mbps (400 Mbps in full duplex).

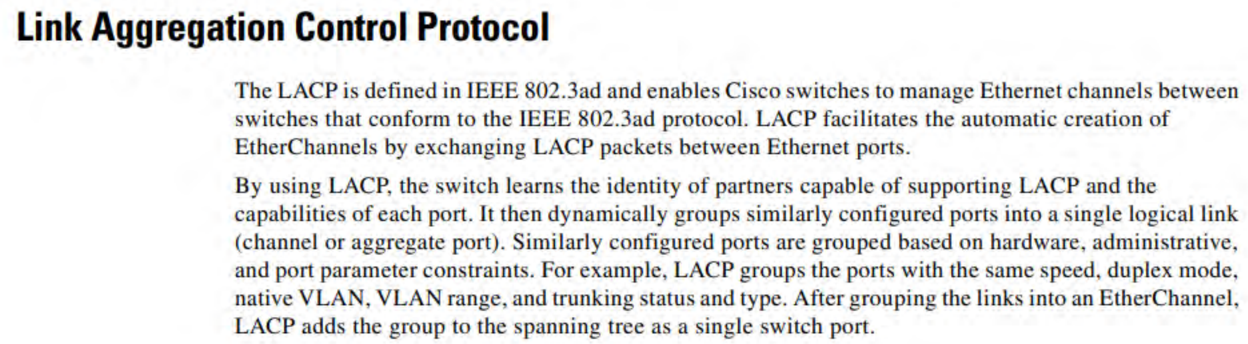
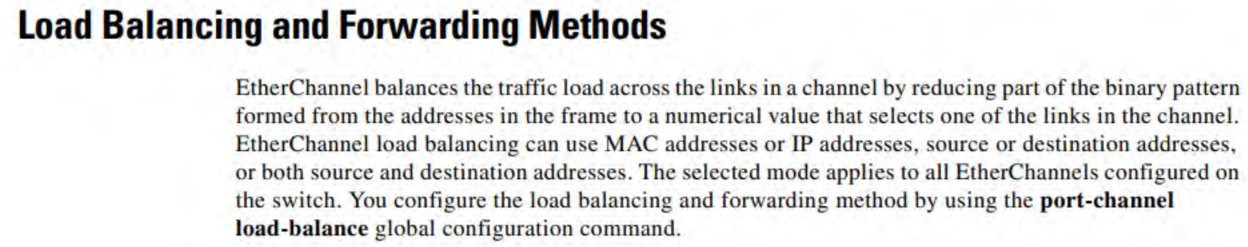
Figure 1-23 Collapsed Backbone and Switch Cluster Configuration

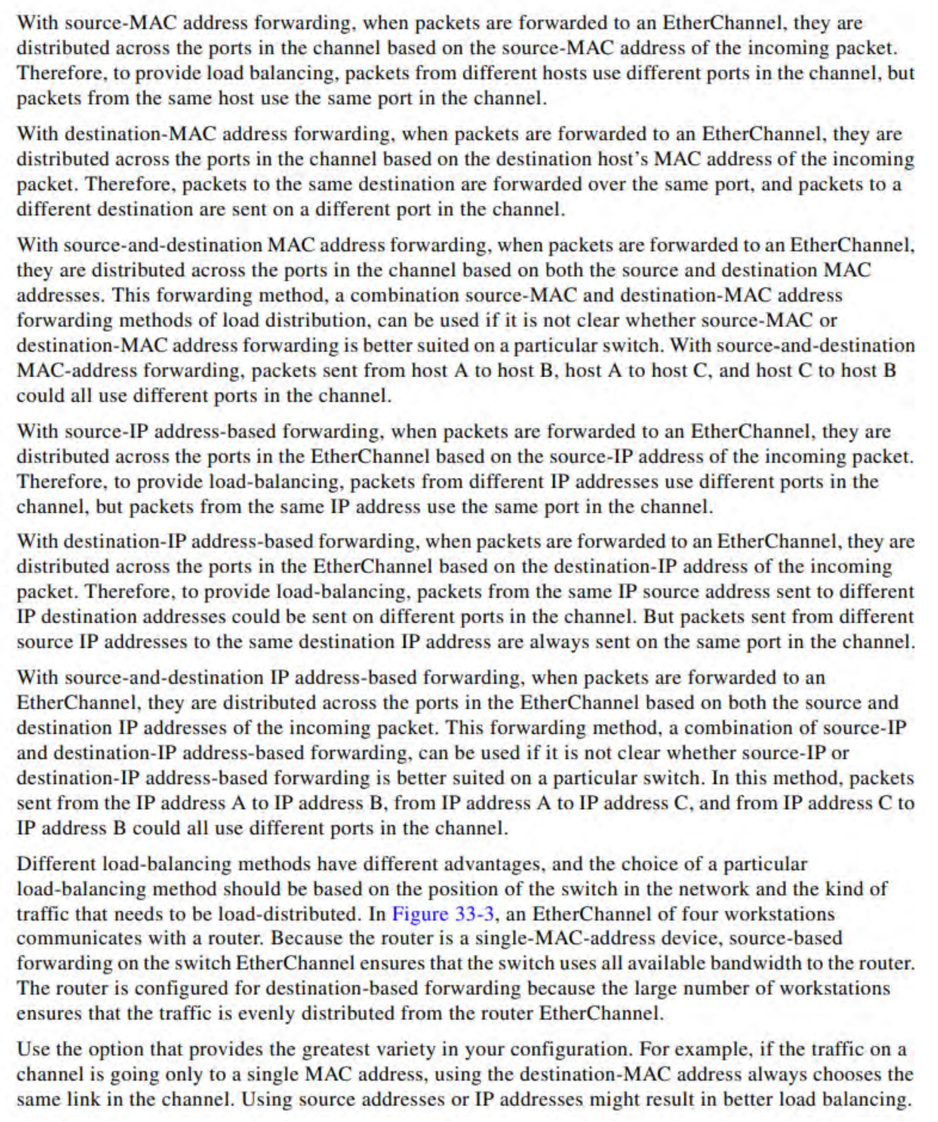

Gigabit servers
Cisco CallManager
Catalyst 4908G-L3 switch
Cisco 2600 router
1 Gbps (2 Gbps full duplex)
200 Mbps Fast EtherChannel (400 Mbps full duplex Fast EtherChannel)
Catalyst 3500 XL and 2900 XL GigaStack cluster
Catalyst 2900 XL, 1900, and 2820 cluster
Catalyst 3524-PWR XL GigaStack cluster
IP
Cisco IP Phones
AC power source
Workstations running Cisco SoftPhone software
Cisco IP Phones
33092

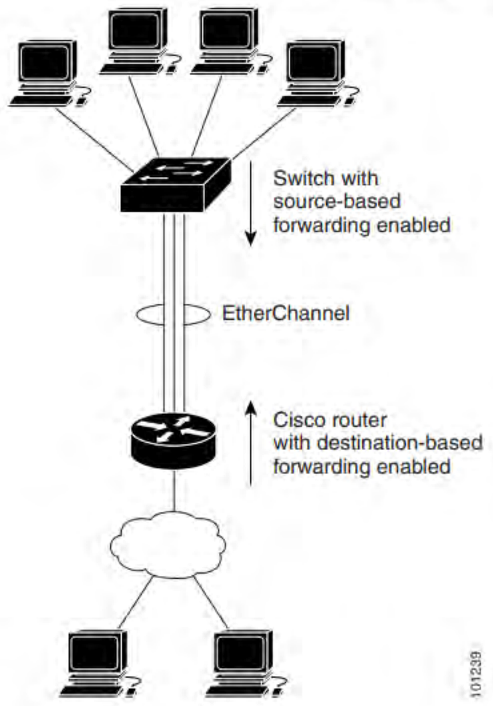
</td></tr>
</table>

No.	’740 Patent Claim 4	Cisco EtherChannel
		Catalyst 3560 Configuration Guide at 1-3  **Performance Features** • Autosensing of port speed and autonegotiation of duplex mode on all switch ports for optimizing bandwidth • Automatic-medium-dependent interface crossover (Auto-MDIX) capability on 10/100 and 10/100/1000 Mbps interfaces and on 10/100/1000 BASE-T/TX SFP module interfaces that enables the interface to automatically detect the required cable connection type (straight-through or crossover) and to configure the connection appropriately • Support for routed frames up to 1546 bytes, for frames up to 9000 bytes that are bridged in hardware, and for frames up to 2000 bytes that are bridged by software • IEEE 802.3x flow control on all ports (the switch does not send pause frames) • EtherChannel for enhanced fault tolerance and for providing up to 8 Gbps (Gigabit EtherChannel) or 800 Mbps (Fast EtherChannel) full-duplex bandwidth between switches, routers, and servers • Port Aggregation Protocol (PAgP) and Link Aggregation Control Protocol (LACP) for automatic creation of EtherChannel links • Forwarding of Layer 2 and Layer 3 packets at Gigabit line rate • Per-port storm control for preventing broadcast, multicast, and unicast storms • Port blocking on forwarding unknown Layer 2 unknown unicast, multicast, and bridged broadcast traffic • Cisco Group Management Protocol (CGMP) server support and Internet Group Management Protocol (IGMP) snooping for IGMP Versions 1, 2, and 3: – (For CGMP devices) CGMP for limiting multicast traffic to specified end stations and reducing overall network traffic – (For IGMP devices) IGMP snooping for efficiently forwarding multimedia and multicast traffic • IGMP report suppression for sending only one IGMP report per multicast router query to the multicast devices (supported only for IGMPv1 or IGMPv2 queries) • IGMP snooping querier support to configure switch to generate periodic IGMP General Query messages Catalyst 3560 Configuration Guide at 10-5

No.	’740 Patent Claim 4	Cisco EtherChannel
		 **EtherChannel Port Groups** EtherChannel port groups treat multiple switch ports as one switch port. These port groups act as a single logical port for high-bandwidth connections between switches or between switches and servers. An EtherChannel balances the traffic load across the links in the channel. If a link within the EtherChannel fails, traffic previously carried over the failed link changes to the remaining links. You can group multiple trunk ports into one logical trunk port, group multiple access ports into one logical access port, group multiple tunnel ports into one logical tunnel port, or group multiple routed ports into one logical routed port. Most protocols operate over either single ports or aggregated switch ports and do not recognize the physical ports within the port group. Exceptions are the DTP, the Cisco Discovery Protocol (CDP), and the Port Aggregation Protocol (PAgP), which operate only on physical ports. Catalyst 3560 Configuration Guide at 33-4  **Port Aggregation Protocol** The Port Aggregation Protocol (PAgP) is a Cisco-proprietary protocol that can be run only on Cisco switches and on those switches licensed by vendors to support PAgP. PAgP facilitates the automatic creation of EtherChannels by exchanging PAgP packets between Ethernet ports. By using PAgP, the switch learns the identity of partners capable of supporting PAgP and the capabilities of each port. It then dynamically groups similarly configured ports into a single logical link (channel or aggregate port). Similarly configured ports are grouped based on hardware, administrative, and port parameter constraints. For example, PAgP groups the ports with the same speed, duplex mode, native VLAN, VLAN range, and trunking status and type. After grouping the links into an EtherChannel, PAgP adds the group to the spanning tree as a single switch port. Catalyst 3560 Configuration Guide at 33-5

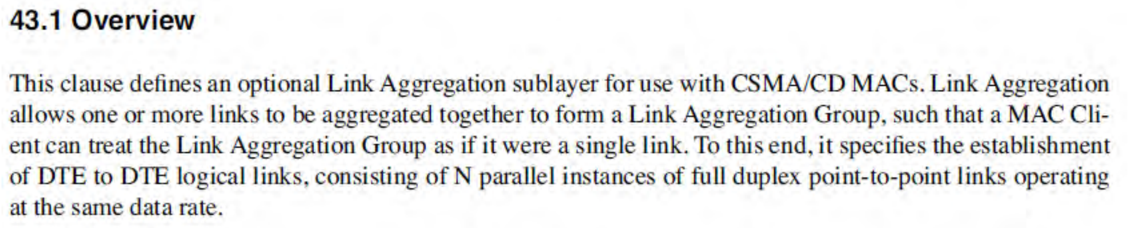
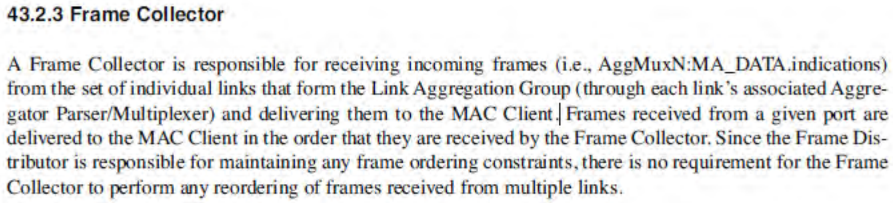
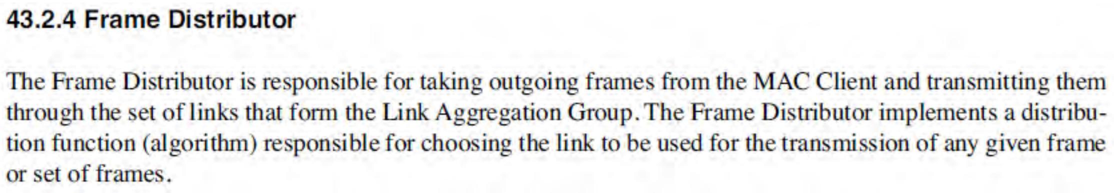
No.	’740 Patent Claim 4	Cisco EtherChannel
		 **Link Aggregation Control Protocol** The LACP is defined in IEEE 802.3ad and enables Cisco switches to manage Ethernet channels between switches that conform to the IEEE 802.3ad protocol. LACP facilitates the automatic creation of EtherChannels by exchanging LACP packets between Ethernet ports. By using LACP, the switch learns the identity of partners capable of supporting LACP and the capabilities of each port. It then dynamically groups similarly configured ports into a single logical link (channel or aggregate port). Similarly configured ports are grouped based on hardware, administrative, and port parameter constraints. For example, LACP groups the ports with the same speed, duplex mode, native VLAN, VLAN range, and trunking status and type. After grouping the links into an EtherChannel, LACP adds the group to the spanning tree as a single switch port. Catalyst 3560 Configuration Guide at 33-6 – 33-8  **Load Balancing and Forwarding Methods** EtherChannel balances the traffic load across the links in a channel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel. EtherChannel load balancing can use MAC addresses or IP addresses, source or destination addresses, or both source and destination addresses. The selected mode applies to all EtherChannels configured on the switch. You configure the load balancing and forwarding method by using the **port-channel load-balance** global configuration command.

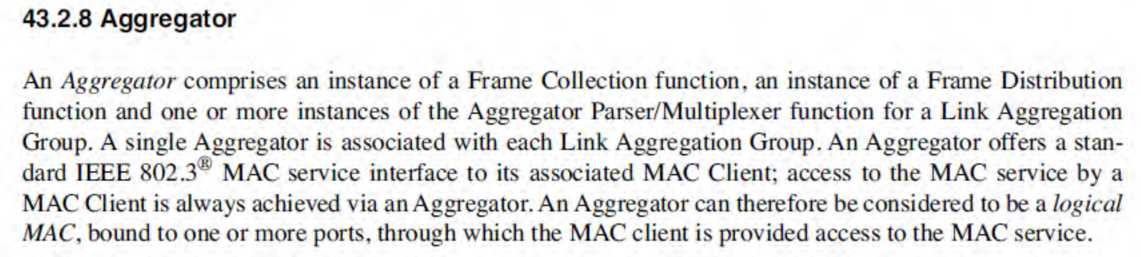
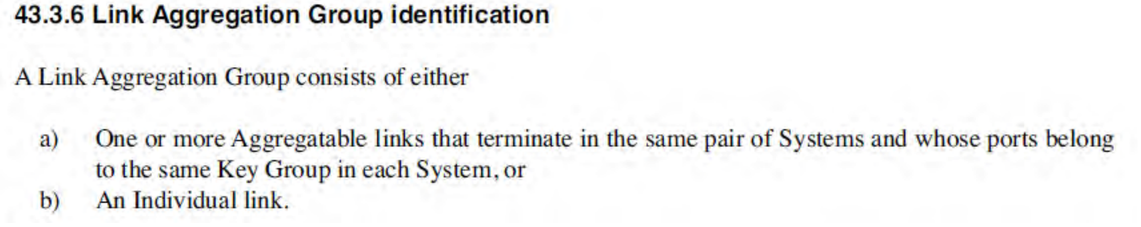
No.	’740 Patent Claim 4	Cisco EtherChannel
		 With source-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the source-MAC address of the incoming packet. Therefore, to provide load balancing, packets from different hosts use different ports in the channel, but packets from the same host use the same port in the channel. With destination-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the destination host’s MAC address of the incoming packet. Therefore, packets to the same destination are forwarded over the same port, and packets to a different destination are sent on a different port in the channel. With source-and-destination MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on both the source and destination MAC addresses. This forwarding method, a combination source-MAC and destination-MAC address forwarding methods of load distribution, can be used if it is not clear whether source-MAC or destination-MAC address forwarding is better suited on a particular switch. With source-and-destination MAC-address forwarding, packets sent from host A to host B, host A to host C, and host C to host B could all use different ports in the channel. With source-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the source-IP address of the incoming packet. Therefore, to provide load-balancing, packets from different IP addresses use different ports in the channel, but packets from the same IP address use the same port in the channel. With destination-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the destination-IP address of the incoming packet. Therefore, to provide load-balancing, packets from the same IP source address sent to different IP destination addresses could be sent on different ports in the channel. But packets sent from different source IP addresses to the same destination IP address are always sent on the same port in the channel. With source-and-destination IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on both the source and destination IP addresses of the incoming packet. This forwarding method, a combination of source-IP and destination-IP address-based forwarding, can be used if it is not clear whether source-IP or destination-IP address-based forwarding is better suited on a particular switch. In this method, packets sent from the IP address A to IP address B, from IP address A to IP address C, and from IP address C to IP address B could all use different ports in the channel. Different load-balancing methods have different advantages, and the choice of a particular load-balancing method should be based on the position of the switch in the network and the kind of traffic that needs to be load-distributed. In Figure 33-3, an EtherChannel of four workstations communicates with a router. Because the router is a single-MAC-address device, source-based forwarding on the switch EtherChannel ensures that the switch uses all available bandwidth to the router. The router is configured for destination-based forwarding because the large number of workstations ensures that the traffic is evenly distributed from the router EtherChannel. Use the option that provides the greatest variety in your configuration. For example, if the traffic on a channel is going only to a single MAC address, using the destination-MAC address always chooses the same link in the channel. Using source addresses or IP addresses might result in better load balancing.

No.	’740 Patent Claim 4	Cisco EtherChannel
		Figure 33-3 Load Distribution and Forwarding Methods  Switch with source-based forwarding enabled EtherChannel Cisco router with destination-based forwarding enabled 101239 Layer 2 EtherChannel **Catalyst 2950** Catalyst 2950 switches only support 802.1Q trunking and do not support ISL trunking. Catalyst 2950 switches support DTP and PAgP dynamic trunking and channel negotiation with Cisco IOS Software Release 12.1 releases and static modes only with Cisco IOS Software Release 12.0 releases. EtherChannel load balancing can use either source-MAC or destination-MAC address forwarding. You can configure the load balancing method by issuing the **port-channel load-**

No.	’740 Patent Claim 4	Cisco EtherChannel
		balance global configuration command. These switches support up to eight switch ports per channel. **Catalyst 6500 That Runs Cisco IOS Software** Catalyst 6500 switches that run Cisco IOS Software support L2 (switchport) and Layer 3 (L3) (routed port) EtherChannel configurations. A Catalyst 6500/6000 series switch supports a maximum of 64 EtherChannels (256 with Cisco IOS Software Release 12.1(2)E and earlier). You can form an EtherChannel with up to eight compatibly configured LAN ports on any module in a Catalyst 6000 series switch, with the exception of Digital Feature Card (DFC)-equipped modules (such as WS-X6816 and so on) which currently allow an L2 channel only using ports on the same DFC module. However, an L3 channel can be configured across different DFC-equipped modules. This limitation has been removed in Catalyst 6500/6000 Cisco IOS Software Release 12.1(11b)EX and later. This document configures an L2 EtherChannel. The Catalyst 6500/6000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP addresses, or Layer 4 (L4) port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. Catalyst 6500/6000 switches support both ISL and 802.1Q trunking encapsulations and DTP. Detailed information on port capabilities is available by issuing the **show interface** *interface_id* **capabilities** command. **Catalyst 4000 That Runs Cisco IOS Software** Catalyst 4000 switches that run Cisco IOS Software (with Supervisor Engine III and IV) support L2 (switchport) and L3 (routed port) EtherChannel configurations. A Catalyst 4000 series switch

No.	’740 Patent Claim 4	Cisco EtherChannel
		supports a maximum of 64 EtherChannels. You can form an EtherChannel with up to eight compatibly configured Ethernet interfaces on any module, and across modules in a Catalyst 4000 series switch. All interfaces in each EtherChannel must be the same speed and must all be configured as either L2 or L3 interfaces. The Catalyst 4000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP address, or L4 port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. The Catalyst 4000 that runs Cisco IOS Software supports ISL and 802.1Q trunking encapsulations and DTP. ISL is not available on certain modules. For a complete list of such modules, refer to the *Understanding VLAN Trunks* section of Configuring Layer 2 Ethernet Interfaces. In a future software release, detailed information on port capabilities will be available by issuing the **show interface capabilities** command. Currently this command is not available. Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Cisco EtherChannel System in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 4[f] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, IEEE 802.3 discloses the aggregation of one or more links together to form a Link Aggregation Group. IEEE 802.3 at 1465

No.	’740 Patent Claim 4	Cisco EtherChannel
		 **43.1 Overview** This clause defines an optional Link Aggregation sublayer for use with CSMA/CD MACs. Link Aggregation allows one or more links to be aggregated together to form a Link Aggregation Group, such that a MAC Client can treat the Link Aggregation Group as if it were a single link. To this end, it specifies the establishment of DTE to DTE logical links, consisting of N parallel instances of full duplex point-to-point links operating at the same data rate. IEEE 802.3 at 1470  **43.2.3 Frame Collector** A Frame Collector is responsible for receiving incoming frames (i.e., AggMuxN:MA_DATA.indications) from the set of individual links that form the Link Aggregation Group (through each link’s associated Aggregator Parser/Multiplexer) and delivering them to the MAC Client. Frames received from a given port are delivered to the MAC Client in the order that they are received by the Frame Collector. Since the Frame Distributor is responsible for maintaining any frame ordering constraints, there is no requirement for the Frame Collector to perform any reordering of frames received from multiple links. IEEE 802.3 at 1471  **43.2.4 Frame Distributor** The Frame Distributor is responsible for taking outgoing frames from the MAC Client and transmitting them through the set of links that form the Link Aggregation Group. The Frame Distributor implements a distribution function (algorithm) responsible for choosing the link to be used for the transmission of any given frame or set of frames. IEEE 802.3 at 1474

No.	’740 Patent Claim 4	Cisco EtherChannel
		 **43.2.8 Aggregator** An *Aggregator* comprises an instance of a Frame Collection function, an instance of a Frame Distribution function and one or more instances of the Aggregator Parser/Multiplexer function for a Link Aggregation Group. A single Aggregator is associated with each Link Aggregation Group. An Aggregator offers a standard IEEE 802.3® MAC service interface to its associated MAC Client; access to the MAC service by a MAC Client is always achieved via an Aggregator. An Aggregator can therefore be considered to be a *logical MAC*, bound to one or more ports, through which the MAC client is provided access to the MAC service. IEEE 802.3 at 1481  **43.3.6 Link Aggregation Group identification** A Link Aggregation Group consists of either a) One or more Aggregatable links that terminate in the same pair of Systems and whose ports belong to the same Key Group in each System, or b) An Individual link. For example, Ghosh discloses aggregating physical links, including ports, into aggregate port channels that form a single logical link to increase bandwidth. Ghosh at Abstract (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggregation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent gen-eration of port channel map tables.”) Ghosh at [0004] (“Neighboring nodes in a fibre channel network are typically interconnected through multiple physical links. For example, a local fibre channel switch may be connected to a remote fibre channel switch through four physical links. In many instances, it may be beneficial to aggregate some of the physical links into logical links. That is, multiple physi-cal links can be combined to form a logical interface to provide higher aggregate bandwidth, load balancing, and

No.	’740 Patent Claim 4	Cisco EtherChannel
		link redundancy. When a frame is being transmitted over a logical link, it does not matter what particular physical link is being used as long as all the frames of a given flow are transmitted through the same link. If a constituent physical link goes down, the logical link can still remain operational.”) Ghosh at [0007] (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggre-gation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent generation of port channel map tables.”) Ghosh at [0008] (“In one embodiment, a method for aggregating ports in a fibre channel fabric is provided. It is determined that a plurality of local ports at a local fibre channel switch are compatible. Identifiers for the plurality of local ports are sent to a remote fibre channel switch. The remote fibre channel switch determines if a plurality of remote ports are compat-ible, the plurality of remote ports corresponding to the plurality of local ports. An indication that one or more of the remote physical ports are compatible is received. A port channel including one or more of the local ports correspond-ing to the compatible remote ports is created.”) Ghosh at [0010] (“In another embodiment, a fibre channel network is described. The fibre channel network includes a local fibre channel switch and a remote fibre channel switch. The local fibre channel switch aggregates a compatible subset of the plurality of local ports and sends identifiers for the compat-ible subset of the plurality of local ports to the remote fibre channel switch. The remote fibre channel switch determines if a subset of the plurality of remote ports are compatible. The subset of the plurality of remote ports corresponds to the compatible subset of the plurality of local ports.”) Ghosh at [0022] (“Switches in a fibre channel network are typically interconnected using multiple physical links. The physical links connecting a pair of switches allows transmission of data and control signals. In some instances, it is useful to aggregate multiple physical links into a logical link. Physi-cal links are also referred to herein as physical interfaces and channels while logical links are

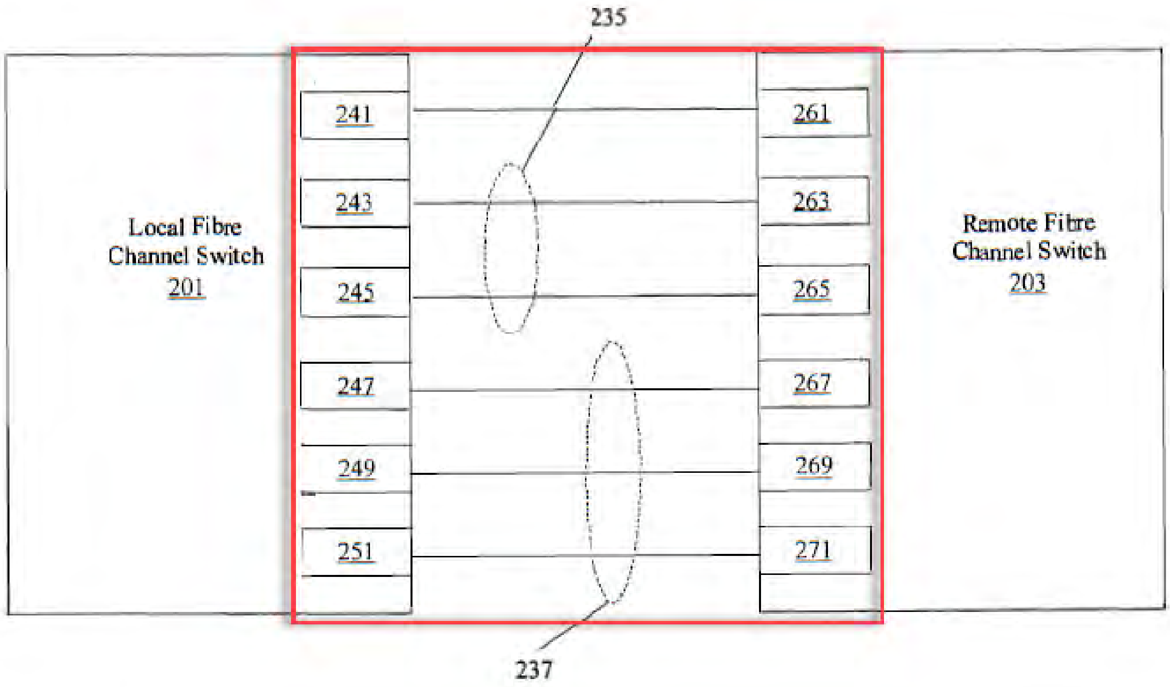
No.	’740 Patent Claim 4	Cisco EtherChannel
		also referred to herein as logical interfaces and port channels. For example, a local switch may be connected to a remote switch through four physical links. Instead of having to transmit data through a particular physical link, the physical links can be aggregated to form one or more logical links. In one example, all four physical links are aggregated into a single logical link. Instead of having data transmitted through a particular physical link, the data can merely be transmitted over a particular logical link without regard to the particular physi-cal interface used. Aggregating physical links into a logical link allows for higher aggregated bandwidth, load balancing, and link redundancy. For example, if a particular physical link fails or is overloaded, data can still be transmitted over the logical link.”) Ghosh at [0029] (“FIG. 2 is a diagrammatic representation showing links between two switches, such as two fibre channel switches shown in FIG. 1. A local fibre channel switch 201 includes local ports 241, 243, 245, 247, 249, and 251. A remote fibre channel switch 203 includes remote ports 261, 263, 265, 267, 269, and 271. Local port 241 is coupled to remote port 261 through an individual physical link or channel. Connected ports are also referred to herein as peer ports. Local port 243 is coupled to remote port 263 and local port 245 is coupled to remote port 265. The two resulting physical links are aggregated to form port channel 235. Local ports 247, 249, and 251 are coupled to remote ports 267, 269, and 271 respectively. The three resulting physical links are aggregated to form port channel 237.”) Ghosh at [0030] (“According to various embodiments, local fibre channel switch 201 and remote fibre channel switch both have associated identifiers. In some examples, the identifiers are globally unique identifiers such as a global switch world wide names (WWNs). Each local port 241, 243, 245, 247, 249, and 251 and each remote port 261,263,265,267, 269, and 271 can also be associated with identifiers. In some examples, the identifiers are port WWNs. The port WWNs are typically used for debugging or identifying the peer port in alert or warning messages. However, according to various embodiments, the techniques of the present invention use WWNs as globally unique identifiers to aggregate ports instead of using compatibility keys which are only locally unique. Compatibility keys are mechanisms typically used by other protocols such as Ethernet for aggregation.”)

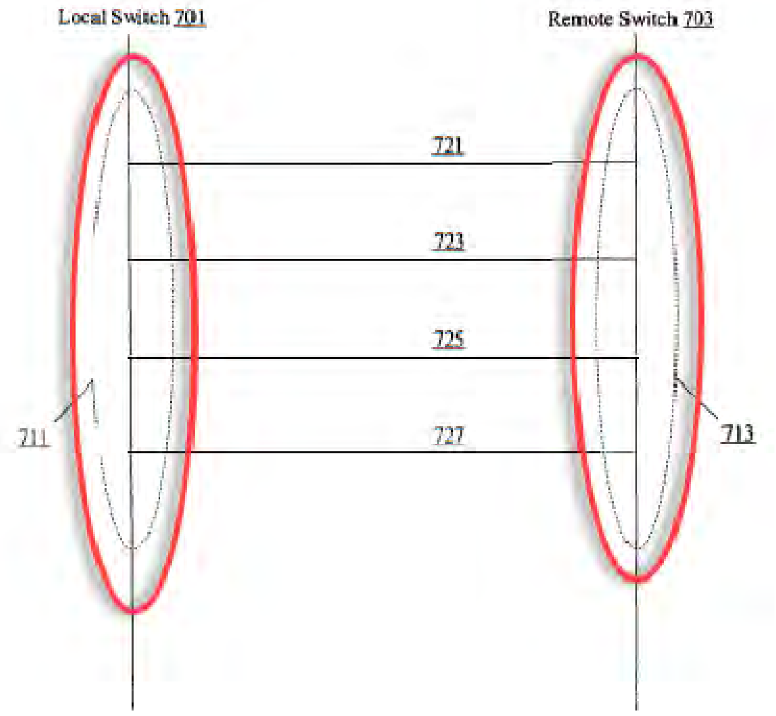
No.	’740 Patent Claim 4	Cisco EtherChannel
		Ghosh at [0033] (“A variety of parameters can be used to aggregate physical ports. FIG. 3 is a flow process diagram showing one technique for aggregating physical ports into a logical port. And 301, it is determined if auto create functionality is enabled. According to various embodiments, auto create functionality allows automatic configuration and detection of compatible physical ports as well as aggregation into one or more logical ports. Auto creation does not require user intervention. In other examples, administrators can manu-ally arrange ports for aggregation.”) Ghosh at [0037] (“FIG. 4 is an exchange diagram showing one example of a bring up procedure used for a port creating a new port channel. A local switch 401 is coupled to a remote switch 403. The local switch 401 includes a physical port Al coupled to physical port Bl included in remote switch 403. When two peer ports Al and Bl are being aggregated into a port channel, the peer switches 401 and 403 typically already know the world wide names of the individual physical peer ports. However, the peer switches only know the world wide name of their own logical port or port channel. That is, both switches have the individual physical link configured, but the link is not yet part of a port channel. At 421, a local switch 401 sends a synchronize (sync) message 411 to the remote switch 403 to begin the process of creating a port channel including ports Al and Bl.”) Ghosh at [0038] (“In some examples, the sync message 411 includes a local port channel identifier and a remote port channel identifier. In one particular example, the local port channel identifier is set to the world wide name of the local port channel assigned by the local switch 401. The remote port channel identifier is left blank to indicate that the port Al is being aggregated as part of a new port channel. The sync message 411 can also include other parameters such as channel status, channel model, or channel intent.”) Ghosh at [0042] (“When two peer ports A2 and B2 are aggregated into a port channel Cl, the peer switches 501 and 503 typically already know the world wide names of the indi-vidual physical peer ports A2 and B2 as well as the world wide name information of the port channel Cl. Conse-quently, the port channel is already successfully established. According to various embodiments, local switch 501 and remote switch 503 perform parameter checking to ensure that the new physical port A2 and B2 can be safely added to the existing port channel Cl. At 521, a local switch can check configuration parameters to ensure that physical ports Al and A2 at the local switch 501 are

No.	’740 Patent Claim 4	Cisco EtherChannel
		compatible. The compatibility checking can be performed anytime. In some examples, compatibility checking is checked before a local switch 501 sends a synchronize (sync) message 511 to the remote switch 503 to begin the process of aggregating ports A2 and B2 into the port channel.) Ghosh at [0043] (“In some examples, the sync message 511 includes local port channel identifier and a remote port channel identifier. In one particular example, the local port channel identifier is set to the world wide name of the local port channel assigned by the local switch 501. The remote port channel identifier is filled with the existing port channel identifier to indicate that the port A2 is being aggregated into existing port channel C2. The sync message 511 can also include other parameters such as channel status, channel model, or channel intent.”) Ghosh at [0044] (“At 531, remote switch 503 uses the information received from the local switch 501 to verify port B2 is compatible with other port in port channel C2. In one example, configuration parameters associated with B2 are checked against configuration parameters associated with Bl. The remote switch 503 can also check if the port B2 is already assigned to a different port channel. If the port B2 is compatible with port Bl, the remote switch 503 can proceed and send a sync accept message 513 in response to the sync message 511 to indicate that the port B2 can be aggregated into the port channel. The sync accept message indicates that a port channel can now be modified. At 523, local switch 501 uses the information to update its own port channel database. However, the port channel may not yet be fully operational until the hardware configuration is completed. The local switch 501 continues hardware configuration such as line card configuration to make the port A2 part of the port channel Cl. An acknowledgment 527 is sent and received by remote switch 503 at 529. In some examples, the local switch 501 sends a commit signal 515 when hardware configuration is complete.”) Ghosh at [0045] (“The remote switch 503 receives the commit signal at 533 and begins its own hardware configuration. On completion of its hardware configuration, remote switch 503 sends out a commit accept signal 517 to indicate to local switch 501 that hardware configuration is completed. According to various embodiments, local switch 501 receives the commit accept signal 517 and notifies relevant applications that the port channel is now fully operational at 525 and that port A2 has been aggregated into port channel Cl. The local switch 501 can also send out an acknowledge

No.	’740 Patent Claim 4	Cisco EtherChannel
		message 519. When the remote switch 503 receives the acknowledge, it notifies relevant applications that the port channel is operational at 535 and that port B2 has been aggregated into port channel Cl. In one embodiments, the techniques of the present invention contemplate using a two phase SYNC and COMMIT mechanism similar to the mechanism used in EPP.”) Ghosh at [0046] (“FIGS. 4 and 5 show examples of ports being aggregated into a port channel. At a particular switch, ports can be selected for aggregation into a port channel in a variety of manners. FIG. 6 is an exchange diagram showing automatic selection of ports at a switch for aggregation into a port channel. A local switch 601 is coupled to a remote switch 603. In one example, the local switch 601 includes physical ports Al, A2, A3, and A4 while remote switch 603 includes physical ports Bl, B2, B3, and B4. No port chan-nels have been formed.”) Ghosh at [0049] (“At 631, remote switch 603 uses the information received from the local switch 601 to verify that the peer ports of Al, A2, and A4 are compatible. That is, ports Bl, B2, and B4 are checked for compatibility. In one example, only ports Bl and B2 may be compatible, and consequently only ports Al, A2, Bl, and B2 can be included in the port channel. In another example, ports Bl, B2, and B4 are compatible, so ports Al, A2, A4, Bl, B2, and B4 can be aggregated into port channel Cl. According to various embodiments, if the port B2 is compatible with port Bl, the remote switch 603 can proceed and send a sync accept message 613 in response to the sync message 611 to indicate that the port B2 can be aggregated into the port channel. It should be noted that remote switch 603 can send a list indicating that ports B2 and B4 are compatible with Bl. However, the remote switch 603 sends only one compatible port B2 back for several reasons, and in the process of selection compatible port channels get priority over com-patible individual ports.”) Ghosh at [0050] (“One reason is that aggregation mechanisms and techniques can be implemented more elegantly by handling ports on an individual basis. Any individual port will either start a new port channel, be added to an existing port channel, or operate stand alone. There is no need to keep track of groups of ports to be aggregated. Another reason is that fewer ports need to be locked if only a single port is being aggregated at any one time. The sync accept message indicates that a port channel can now be modified. At 623, local switch 601 receives the information and recognizes that Al and A2 can now be aggregated into port channel Cl. However, the port channel may not yet be

No.	’740 Patent Claim 4	Cisco EtherChannel
		fully operational until the hardware configuration is completed. An acknowl-edgment 627 is sent and received by remote switch 603 at 629. In some examples, the local switch 601 sends a commit signal 615 when hardware configuration is complete.”) Ghosh at [0051] (“The remote switch 603 receives the commit signal at 633 to create port channel Cl including ports Bl and B2. Hardware configuration can now be performed. On comple-tion of its hardware configuration, remote switch 603 sends out a commit accept signal 617 to indicate to local switch 601 that hardware configuration is completed. According to various embodiments, local switch 601 receives the commit accept signal 617 and notifies relevant applications that the port channel is now fully operational at 625 and that ports Al and A2 have been aggregated into port channel Cl. The local switch 601 can also send out an acknowledge message 619. When the remote switch 603 receives the acknowledge, it notifies relevant applications that the port channel is fully operational at 635 and that ports Bl and B2 have been aggregated into port channel Cl.”) Ghosh at [0053] (“FIG. 7 is a diagrammatic representation showing synchronous aggregation of ports into a port channel. A local switch 701 is coupled to a remote switch 703 through links 721, 723, 725, and 727. According to various embodiments, the links are being aggregated into port channel 711 at the local switch 701 and port channel 713 at the remote switch 703 in a synchronous manner. That is the peer ports corre-sponding to each link are brought up in the same order at both the local switch 701 and the remote switch 703.”) Ghosh at Figure 2 (annotation added)

No.	’740 Patent Claim 4	Cisco EtherChannel
		 235 Local Fibre Channel Switch 201 241 243 245 247 249 251 261 263 265 267 269 271 Remote Fibre Channel Switch 203 237 Figure 2 Ghosh at Figure 7 (annotation added)

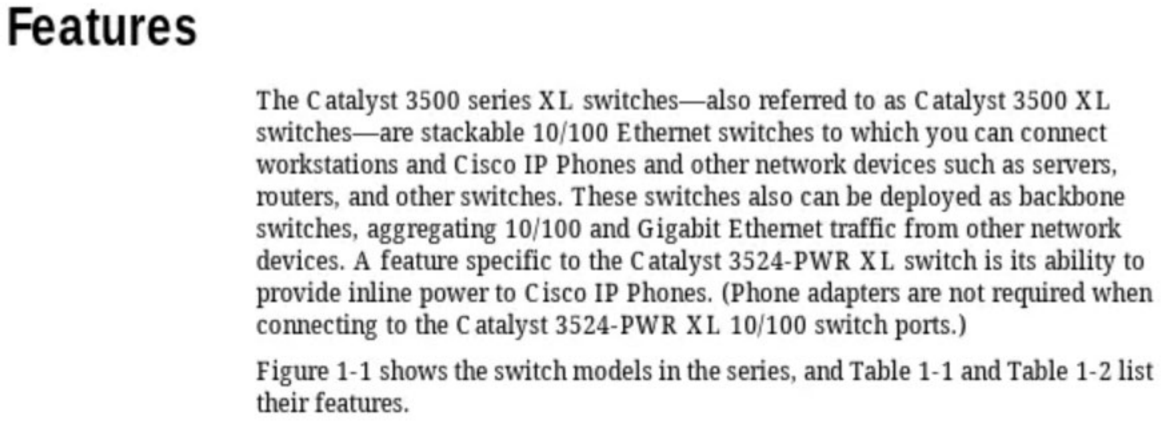
No.	’740 Patent Claim 4	Cisco EtherChannel
		 Local Switch 701 Remote Switch 703 721 723 725 727 711 713 Figure 7

No.	’740 Patent Claim 5	Cisco EtherChannel
5[preamble]	A method for communication, comprising:	Cisco EtherChannel System discloses a method for communication. *See supra at* 1[preamble].

No.	’740 Patent Claim 5	Cisco EtherChannel
5[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	Cisco EtherChannel System discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].
5[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Cisco EtherChannel System discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra at* 1[c].
5[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	Cisco EtherChannel System discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra at* 1[e].
5[d]	selecting, in a single computation based on at	Cisco EtherChannel System discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group.

No.	’740 Patent Claim 5	Cisco EtherChannel
	least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	*See supra at* 1[f].
5[e]	sending the data frame over the selected first and second physical links,	Cisco EtherChannel System discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].
5[f]	coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group so as to increase a	Cisco EtherChannel System discloses coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node. For example, Cisco EtherChannel System discloses link aggregation technology in which at least Ethernet ports are aggregated to form a single logical Ethernet channel. Cisco EtherChannel System further discloses aggregating the ports in order to aggregate and increase the total bandwidth available over the single logical Ethernet channel. Cisco Catalyst 6500 Data Sheet at 2 (“Offers optional, redundant, high-performance Cisco Catalyst 6500 Series Supervisor Engine 720, passive backplane, multimodule Cisco EtherChannel® technology, IEEE 802.3ad link aggregation, IEEE 802.1s only, and Hot Standby Router Protocol/Virtual Router Redundancy Protocol (HSRP/VRRP) high-availability features”) Cisco Catalyst 6500 Data Sheet at 4 (“Ideal for Metro Ethernet WAN Services

No.	’740 Patent Claim 5	Cisco EtherChannel
	data bandwidth provided to the network node.	• 802.1Q and 802.1Q tunneling (QinQ), providing point-to-point and multipoint Ethernet services • Ethernet over MPLS in MPLS backbone networks for superior network scaling, providing VLAN translation capability • Layer 2 and Layer 3 quality of service (QoS), facilitating tiered Ethernet service offerings through rate limiting and traffic shaping • Superior high-availability features, including enhanced Spanning Tree Protocol, IEEE 802.1s, IEEE 802.1w, and Cisco EtherChannel IEEE 802.3ad link aggregation”) Cisco Catalyst 6500 Data Sheet at 2 (“Software features such as Network-Based Application Recognition (NBAR) enhance network management and control of bandwidth utilization.”) Cisco Catalyst 6500 Data Sheet at 13 (“**Switch Fabric Modules** Designed to support distributed forwarding for interface modules with that capability, the Cisco Catalyst 6500 Series Switch Fabric Module (SFM or SFM2), in combination with the Cisco Catalyst 6000 Multilayer Switch Feature Card (MSFC2) and Cisco distributed forwarding cards (DFCs) on interface modules, increases available system bandwidth from 32 to 256 Gbps. The SFM or SFM2 supports the Cisco Catalyst 6500 CEF256 and dCEF256 interface modules. Designed to support new interface modules with 720-Gbps forwarding capabilities, the switch fabric onboard the Cisco Catalyst 6500 Series Supervisor Engine 720 increases available bandwidth to 720 Gbps and provides packet-forwarding rates up to 400 mpps. By using automatic sensing and negotiation, the switch fabric is fully interoperable with the 8- and 16-Gbps switch-fabric interconnections used by the CEF256 and dCEF256 interface modules. When a CEF256 or dCEF256 interface module is detected, the switch fabric will automatically connect those modules by offering 8 to 16 Gbps of bandwidth to each module, as applicable.”) Catalyst 3500 Installation Guide at 1-1

No.	’740 Patent Claim 5	Cisco EtherChannel
		 **Features** The Catalyst 3500 series XL switches—also referred to as Catalyst 3500 XL switches—are stackable 10/100 Ethernet switches to which you can connect workstations and Cisco IP Phones and other network devices such as servers, routers, and other switches. These switches also can be deployed as backbone switches, aggregating 10/100 and Gigabit Ethernet traffic from other network devices. A feature specific to the Catalyst 3524-PWR XL switch is its ability to provide inline power to Cisco IP Phones. (Phone adapters are not required when connecting to the Catalyst 3524-PWR XL 10/100 switch ports.) Figure 1-1 shows the switch models in the series, and Table 1-1 and Table 1-2 list their features. Catalyst 3500 Installation Guide at Table 1-1

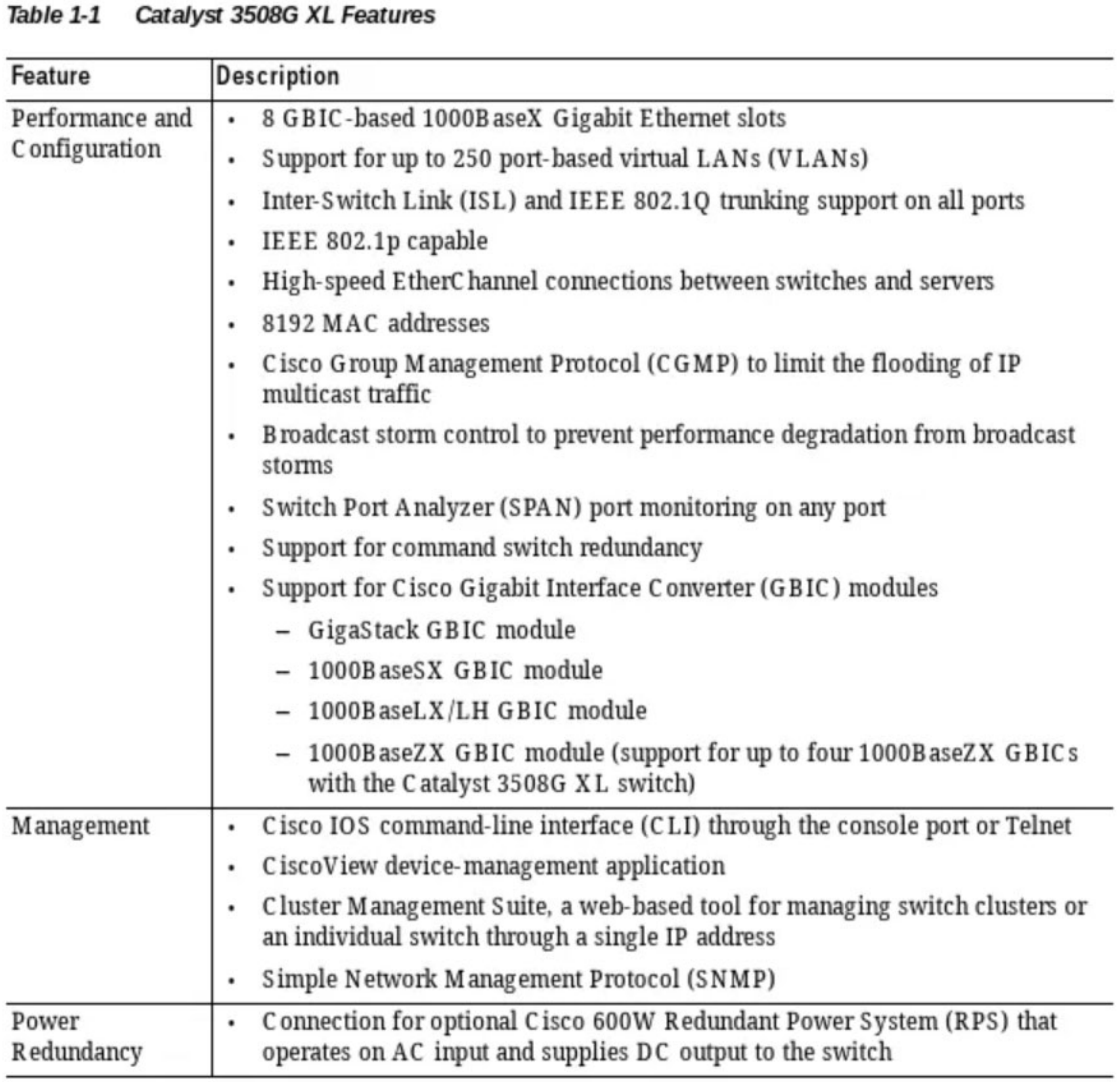
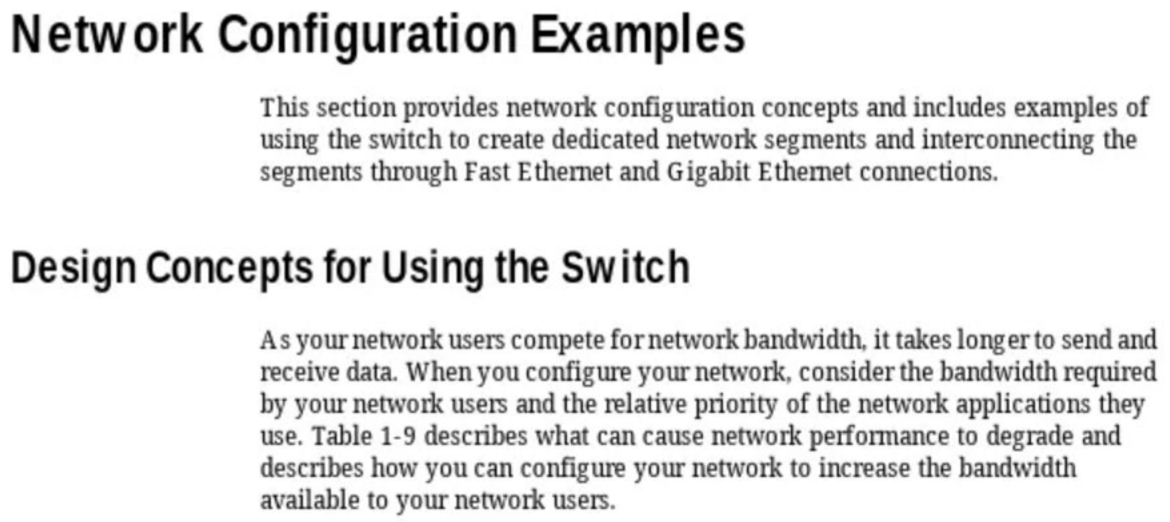
No.	’740 Patent Claim 5	Cisco EtherChannel
		

Table 1-1 Catalyst 3508G XL Features

Feature	Description
Performance and Configuration	• 8 GBIC-based 1000BaseX Gigabit Ethernet slots • Support for up to 250 port-based virtual LANs (VLANs) • Inter-Switch Link (ISL) and IEEE 802.1Q trunking support on all ports • IEEE 802.1p capable • High-speed EtherChannel connections between switches and servers • 8192 MAC addresses • Cisco Group Management Protocol (CGMP) to limit the flooding of IP multicast traffic • Broadcast storm control to prevent performance degradation from broadcast storms • Switch Port Analyzer (SPAN) port monitoring on any port • Support for command switch redundancy • Support for Cisco Gigabit Interface Converter (GBIC) modules – GigaStack GBIC module – 1000BaseSX GBIC module – 1000BaseLX/LH GBIC module – 1000BaseZX GBIC module (support for up to four 1000BaseZX GBICs with the Catalyst 3508G XL switch)
Management	• Cisco IOS command-line interface (CLI) through the console port or Telnet • CiscoView device-management application • Cluster Management Suite, a web-based tool for managing switch clusters or an individual switch through a single IP address • Simple Network Management Protocol (SNMP)
Power Redundancy	• Connection for optional Cisco 600W Redundant Power System (RPS) that operates on AC input and supplies DC output to the switch

No.	’740 Patent Claim 5	Cisco EtherChannel
		Catalyst 3500 Installation Guide at 1-28  **Network Configuration Examples** This section provides network configuration concepts and includes examples of using the switch to create dedicated network segments and interconnecting the segments through Fast Ethernet and Gigabit Ethernet connections. **Design Concepts for Using the Switch** As your network users compete for network bandwidth, it takes longer to send and receive data. When you configure your network, consider the bandwidth required by your network users and the relative priority of the network applications they use. Table 1-9 describes what can cause network performance to degrade and describes how you can configure your network to increase the bandwidth available to your network users. Catalyst 3500 Installation Guide at Table 1-9

<table>
<tr><th>No.</th><th>’740 Patent Claim 5</th><th>Cisco EtherChannel</th></tr>
<tr><td></td><td></td><td>

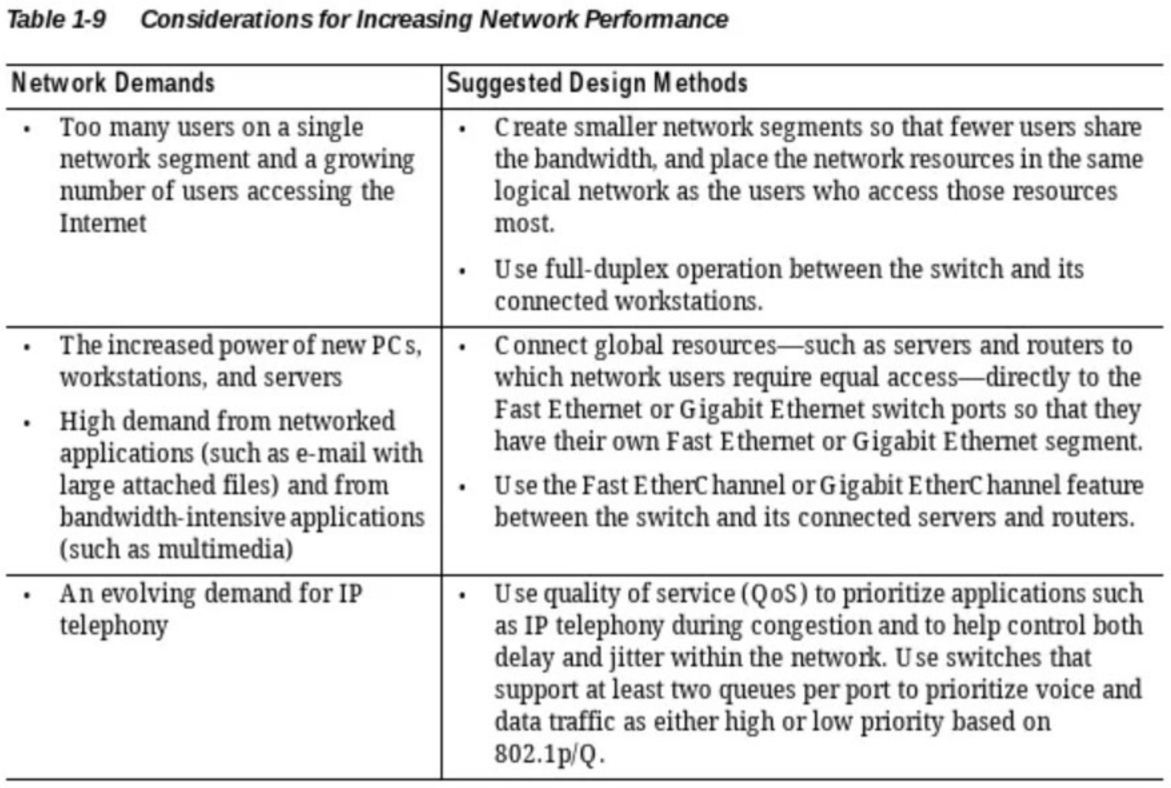
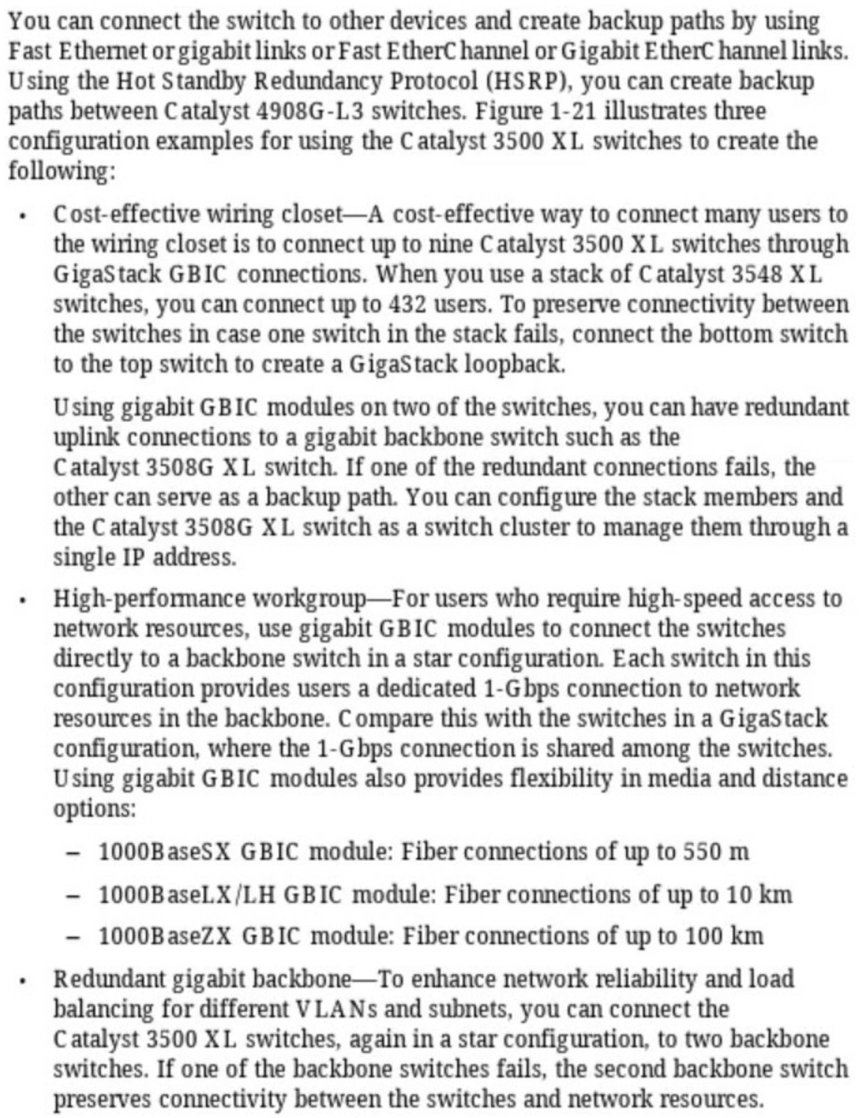
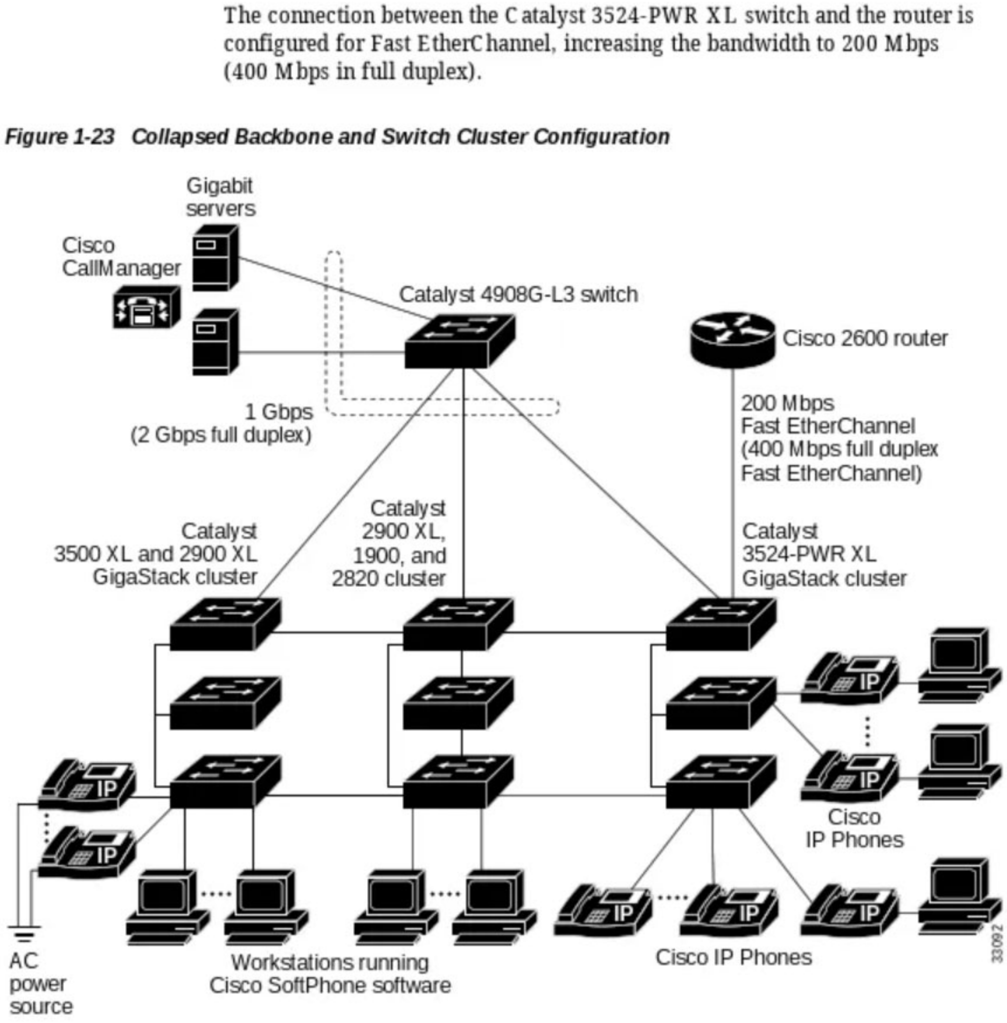


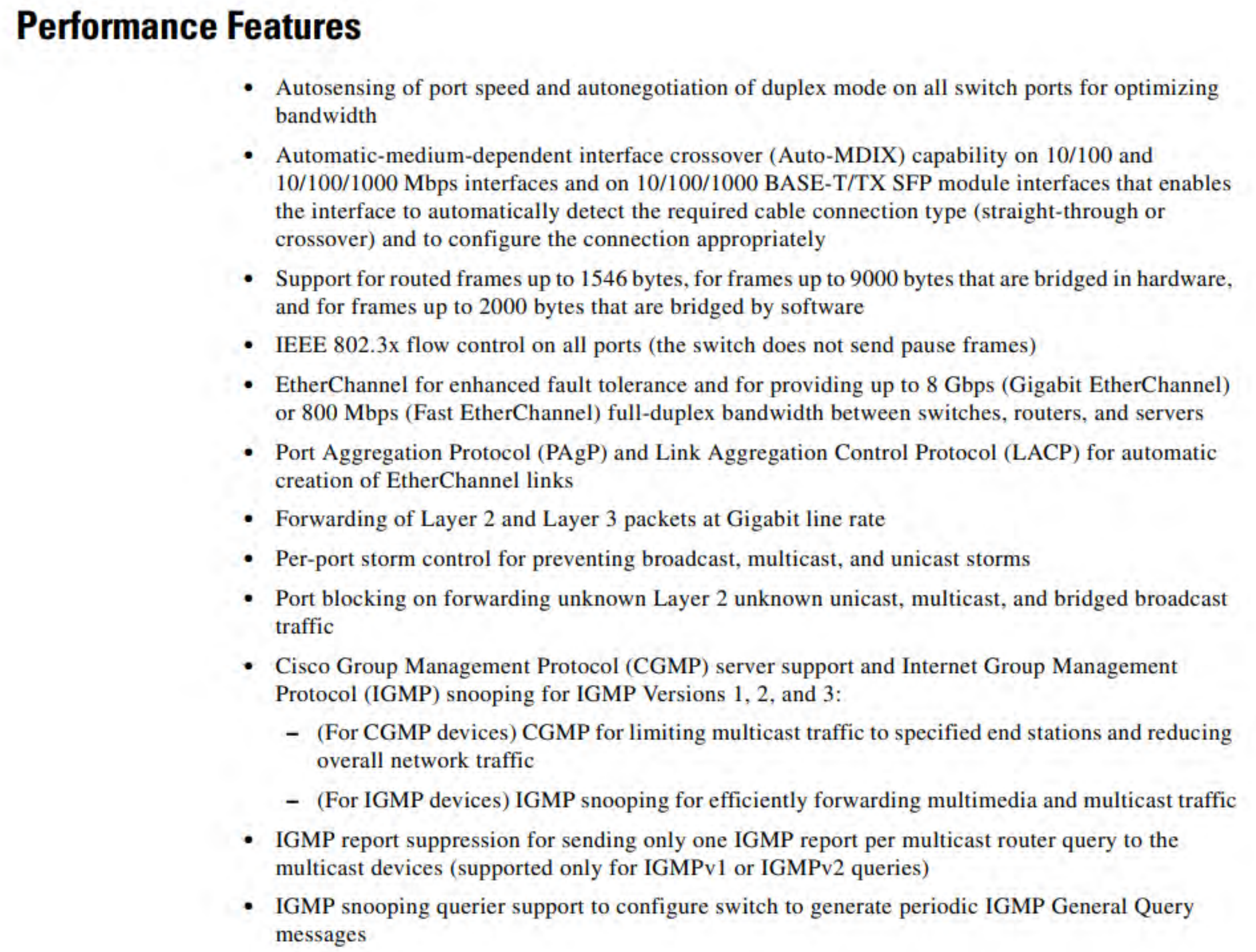
Table 1-9 Considerations for Increasing Network Performance

<table>
<tr><th>Network Demands</th><th>Suggested Design Methods</th></tr>
<tr><td>• Too many users on a single network segment and a growing number of users accessing the Internet</td><td>• Create smaller network segments so that fewer users share the bandwidth, and place the network resources in the same logical network as the users who access those resources most.
• Use full-duplex operation between the switch and its connected workstations.</td></tr>
<tr><td>• The increased power of new PCs, workstations, and servers
• High demand from networked applications (such as e-mail with large attached files) and from bandwidth-intensive applications (such as multimedia)</td><td>• Connect global resources—such as servers and routers to which network users require equal access—directly to the Fast Ethernet or Gigabit Ethernet switch ports so that they have their own Fast Ethernet or Gigabit Ethernet segment.
• Use the Fast EtherChannel or Gigabit EtherChannel feature between the switch and its connected servers and routers.</td></tr>
<tr><td>• An evolving demand for IP telephony</td><td>• Use quality of service (QoS) to prioritize applications such as IP telephony during congestion and to help control both delay and jitter within the network. Use switches that support at least two queues per port to prioritize voice and data traffic as either high or low priority based on 802.1p/Q.</td></tr>
</table>

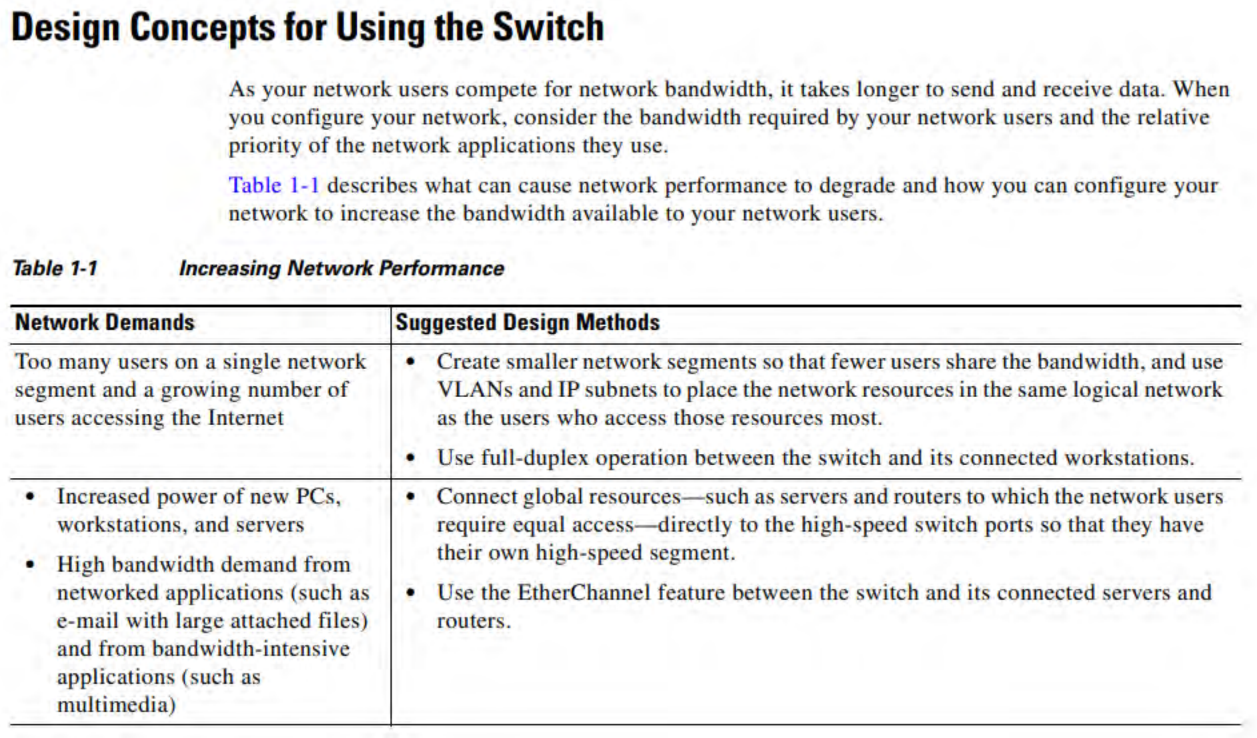
Catalyst 3500 Installation Guide at 1-30</td></tr>
</table>

No.	’740 Patent Claim 5	Cisco EtherChannel
		 You can connect the switch to other devices and create backup paths by using Fast Ethernet or gigabit links or Fast EtherChannel or Gigabit EtherChannel links. Using the Hot Standby Redundancy Protocol (HSRP), you can create backup paths between Catalyst 4908G-L3 switches. Figure 1-21 illustrates three configuration examples for using the Catalyst 3500 XL switches to create the following: • Cost-effective wiring closet—A cost-effective way to connect many users to the wiring closet is to connect up to nine Catalyst 3500 XL switches through GigaStack GBIC connections. When you use a stack of Catalyst 3548 XL switches, you can connect up to 432 users. To preserve connectivity between the switches in case one switch in the stack fails, connect the bottom switch to the top switch to create a GigaStack loopback. Using gigabit GBIC modules on two of the switches, you can have redundant uplink connections to a gigabit backbone switch such as the Catalyst 3508G XL switch. If one of the redundant connections fails, the other can serve as a backup path. You can configure the stack members and the Catalyst 3508G XL switch as a switch cluster to manage them through a single IP address. • High-performance workgroup—For users who require high-speed access to network resources, use gigabit GBIC modules to connect the switches directly to a backbone switch in a star configuration. Each switch in this configuration provides users a dedicated 1-Gbps connection to network resources in the backbone. Compare this with the switches in a GigaStack configuration, where the 1-Gbps connection is shared among the switches. Using gigabit GBIC modules also provides flexibility in media and distance options: – 1000BaseSX GBIC module: Fiber connections of up to 550 m – 1000BaseLX/LH GBIC module: Fiber connections of up to 10 km – 1000BaseZX GBIC module: Fiber connections of up to 100 km • Redundant gigabit backbone—To enhance network reliability and load balancing for different VLANs and subnets, you can connect the Catalyst 3500 XL switches, again in a star configuration, to two backbone switches. If one of the backbone switches fails, the second backbone switch preserves connectivity between the switches and network resources.

No.	’740 Patent Claim 5	Cisco EtherChannel
		Catalyst 3500 Installation Guide at Figure 1-23  The connection between the Catalyst 3524-PWR XL switch and the router is configured for Fast EtherChannel, increasing the bandwidth to 200 Mbps (400 Mbps in full duplex). *Figure 1-23 Collapsed Backbone and Switch Cluster Configuration* Gigabit servers Cisco CallManager Catalyst 4908G-L3 switch Cisco 2600 router 1 Gbps (2 Gbps full duplex) 200 Mbps Fast EtherChannel (400 Mbps full duplex Fast EtherChannel) Catalyst 3500 XL and 2900 XL GigaStack cluster Catalyst 2900 XL, 1900, and 2820 cluster Catalyst 3524-PWR XL GigaStack cluster IP Cisco IP Phones AC power source Workstations running Cisco SoftPhone software Cisco IP Phones 33092 Cisco EtherChannel at

No.	’740 Patent Claim 5	Cisco EtherChannel
		Catalyst 3560 Configuration Guide at 1-3  **Performance Features** • Autosensing of port speed and autonegotiation of duplex mode on all switch ports for optimizing bandwidth • Automatic-medium-dependent interface crossover (Auto-MDIX) capability on 10/100 and 10/100/1000 Mbps interfaces and on 10/100/1000 BASE-T/TX SFP module interfaces that enables the interface to automatically detect the required cable connection type (straight-through or crossover) and to configure the connection appropriately • Support for routed frames up to 1546 bytes, for frames up to 9000 bytes that are bridged in hardware, and for frames up to 2000 bytes that are bridged by software • IEEE 802.3x flow control on all ports (the switch does not send pause frames) • EtherChannel for enhanced fault tolerance and for providing up to 8 Gbps (Gigabit EtherChannel) or 800 Mbps (Fast EtherChannel) full-duplex bandwidth between switches, routers, and servers • Port Aggregation Protocol (PAgP) and Link Aggregation Control Protocol (LACP) for automatic creation of EtherChannel links • Forwarding of Layer 2 and Layer 3 packets at Gigabit line rate • Per-port storm control for preventing broadcast, multicast, and unicast storms • Port blocking on forwarding unknown Layer 2 unknown unicast, multicast, and bridged broadcast traffic • Cisco Group Management Protocol (CGMP) server support and Internet Group Management Protocol (IGMP) snooping for IGMP Versions 1, 2, and 3: – (For CGMP devices) CGMP for limiting multicast traffic to specified end stations and reducing overall network traffic – (For IGMP devices) IGMP snooping for efficiently forwarding multimedia and multicast traffic • IGMP report suppression for sending only one IGMP report per multicast router query to the multicast devices (supported only for IGMPv1 or IGMPv2 queries) • IGMP snooping querier support to configure switch to generate periodic IGMP General Query messages Catalyst 3560 Configuration Guide at 1-14

No.	’740 Patent Claim 5	Cisco EtherChannel



Design Concepts for Using the Switch

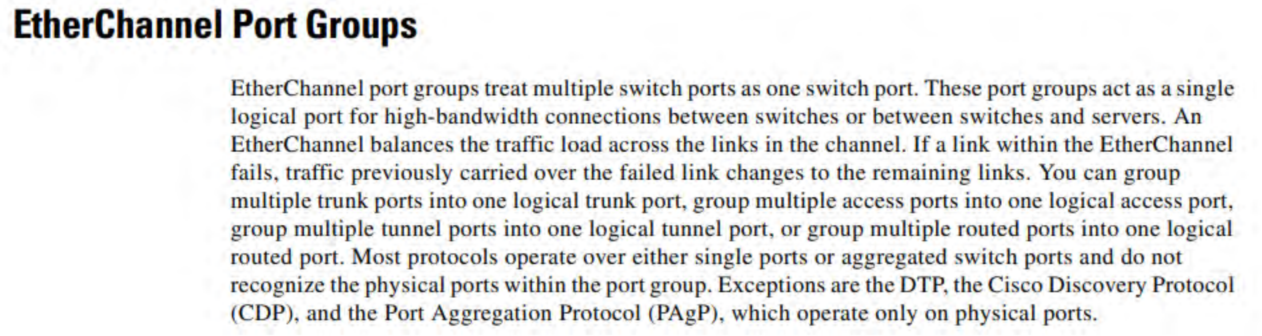
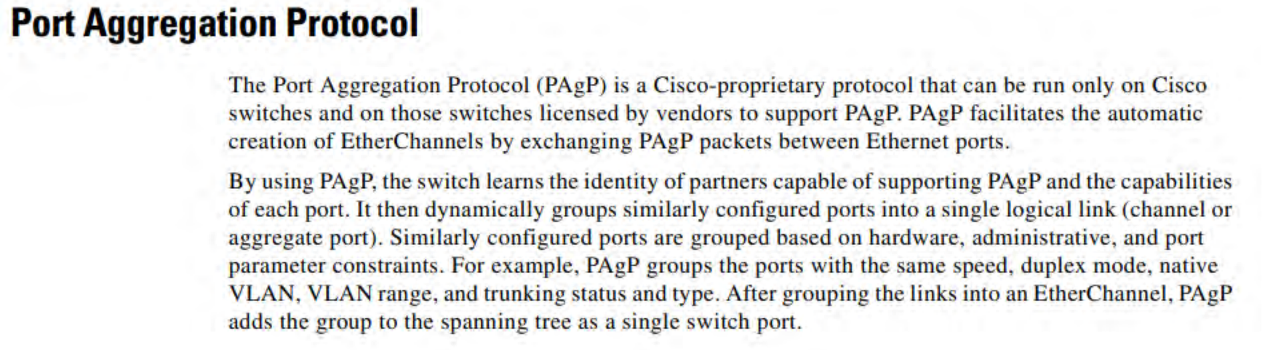
As your network users compete for network bandwidth, it takes longer to send and receive data. When you configure your network, consider the bandwidth required by your network users and the relative priority of the network applications they use.

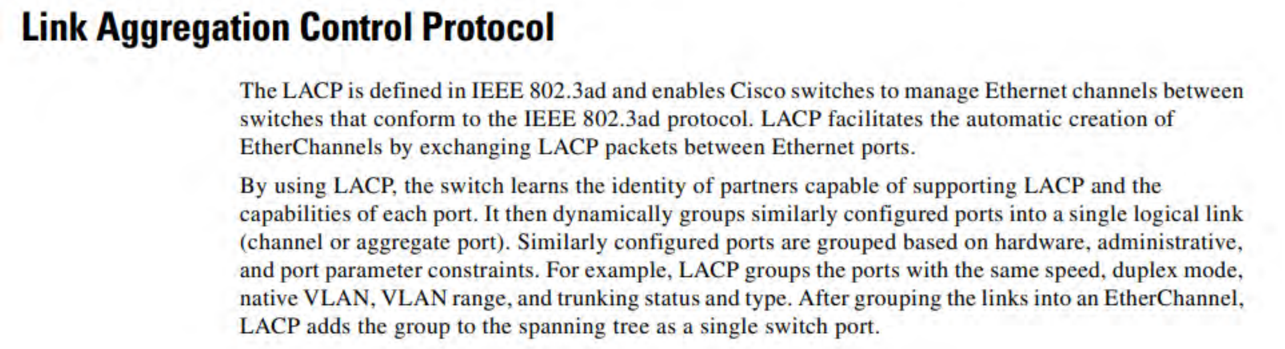
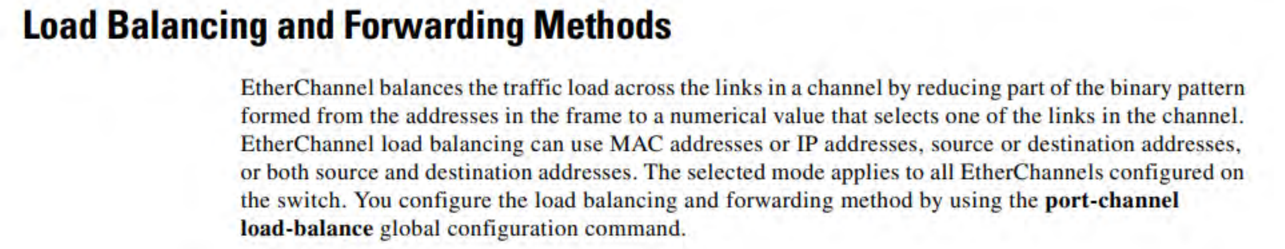
Table 1-1 describes what can cause network performance to degrade and how you can configure your network to increase the bandwidth available to your network users.

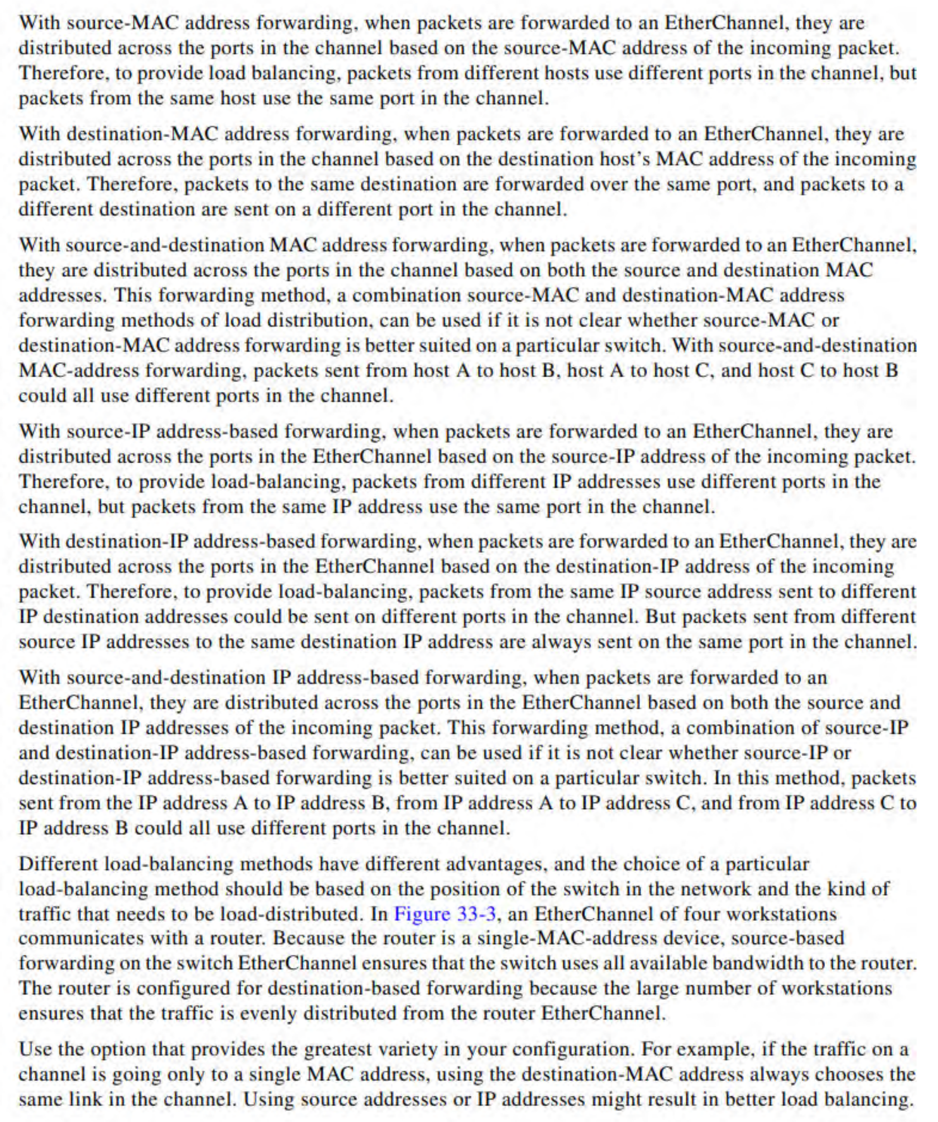
Table 1-1 Increasing Network Performance

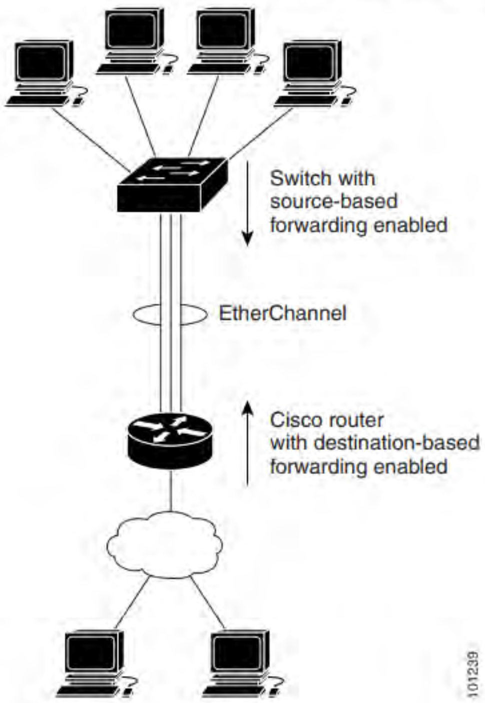
Network Demands	Suggested Design Methods
Too many users on a single network segment and a growing number of users accessing the Internet	• Create smaller network segments so that fewer users share the bandwidth, and use VLANs and IP subnets to place the network resources in the same logical network as the users who access those resources most. • Use full-duplex operation between the switch and its connected workstations.
• Increased power of new PCs, workstations, and servers • High bandwidth demand from networked applications (such as e-mail with large attached files) and from bandwidth-intensive applications (such as multimedia)	• Connect global resources—such as servers and routers to which the network users require equal access—directly to the high-speed switch ports so that they have their own high-speed segment. • Use the EtherChannel feature between the switch and its connected servers and routers.

Catalyst 3560 Configuration Guide at 10-5

No.	’740 Patent Claim 5	Cisco EtherChannel
		 EtherChannel Port Groups EtherChannel port groups treat multiple switch ports as one switch port. These port groups act as a single logical port for high-bandwidth connections between switches or between switches and servers. An EtherChannel balances the traffic load across the links in the channel. If a link within the EtherChannel fails, traffic previously carried over the failed link changes to the remaining links. You can group multiple trunk ports into one logical trunk port, group multiple access ports into one logical access port, group multiple tunnel ports into one logical tunnel port, or group multiple routed ports into one logical routed port. Most protocols operate over either single ports or aggregated switch ports and do not recognize the physical ports within the port group. Exceptions are the DTP, the Cisco Discovery Protocol (CDP), and the Port Aggregation Protocol (PAgP), which operate only on physical ports. Catalyst 3560 Configuration Guide at 33-4  Port Aggregation Protocol The Port Aggregation Protocol (PAgP) is a Cisco-proprietary protocol that can be run only on Cisco switches and on those switches licensed by vendors to support PAgP. PAgP facilitates the automatic creation of EtherChannels by exchanging PAgP packets between Ethernet ports. By using PAgP, the switch learns the identity of partners capable of supporting PAgP and the capabilities of each port. It then dynamically groups similarly configured ports into a single logical link (channel or aggregate port). Similarly configured ports are grouped based on hardware, administrative, and port parameter constraints. For example, PAgP groups the ports with the same speed, duplex mode, native VLAN, VLAN range, and trunking status and type. After grouping the links into an EtherChannel, PAgP adds the group to the spanning tree as a single switch port. Catalyst 3560 Configuration Guide at 33-5

No.	’740 Patent Claim 5	Cisco EtherChannel
		 **Link Aggregation Control Protocol** The LACP is defined in IEEE 802.3ad and enables Cisco switches to manage Ethernet channels between switches that conform to the IEEE 802.3ad protocol. LACP facilitates the automatic creation of EtherChannels by exchanging LACP packets between Ethernet ports. By using LACP, the switch learns the identity of partners capable of supporting LACP and the capabilities of each port. It then dynamically groups similarly configured ports into a single logical link (channel or aggregate port). Similarly configured ports are grouped based on hardware, administrative, and port parameter constraints. For example, LACP groups the ports with the same speed, duplex mode, native VLAN, VLAN range, and trunking status and type. After grouping the links into an EtherChannel, LACP adds the group to the spanning tree as a single switch port. Catalyst 3560 Configuration Guide at 33-6 – 33-8  **Load Balancing and Forwarding Methods** EtherChannel balances the traffic load across the links in a channel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel. EtherChannel load balancing can use MAC addresses or IP addresses, source or destination addresses, or both source and destination addresses. The selected mode applies to all EtherChannels configured on the switch. You configure the load balancing and forwarding method by using the **port-channel load-balance** global configuration command.

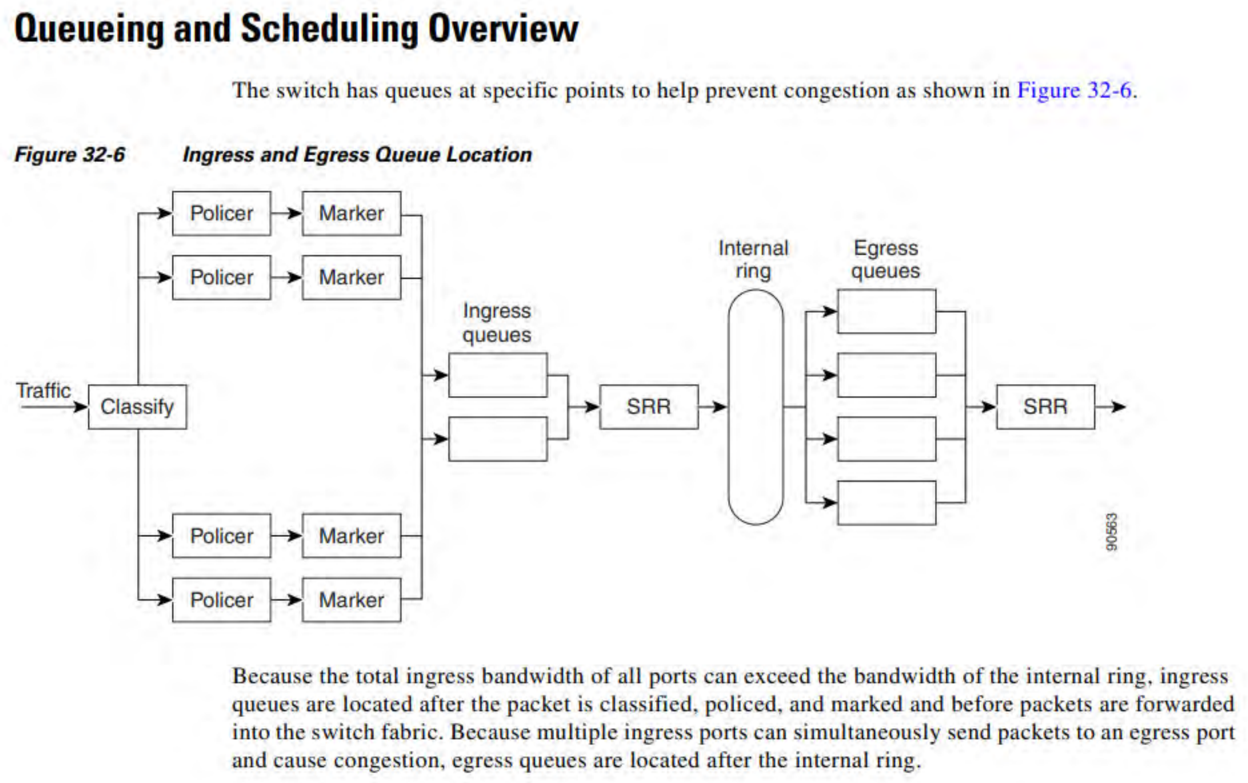
No.	’740 Patent Claim 5	Cisco EtherChannel
		 With source-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the source-MAC address of the incoming packet. Therefore, to provide load balancing, packets from different hosts use different ports in the channel, but packets from the same host use the same port in the channel. With destination-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the destination host’s MAC address of the incoming packet. Therefore, packets to the same destination are forwarded over the same port, and packets to a different destination are sent on a different port in the channel. With source-and-destination MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on both the source and destination MAC addresses. This forwarding method, a combination source-MAC and destination-MAC address forwarding methods of load distribution, can be used if it is not clear whether source-MAC or destination-MAC address forwarding is better suited on a particular switch. With source-and-destination MAC-address forwarding, packets sent from host A to host B, host A to host C, and host C to host B could all use different ports in the channel. With source-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the source-IP address of the incoming packet. Therefore, to provide load-balancing, packets from different IP addresses use different ports in the channel, but packets from the same IP address use the same port in the channel. With destination-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the destination-IP address of the incoming packet. Therefore, to provide load-balancing, packets from the same IP source address sent to different IP destination addresses could be sent on different ports in the channel. But packets sent from different source IP addresses to the same destination IP address are always sent on the same port in the channel. With source-and-destination IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on both the source and destination IP addresses of the incoming packet. This forwarding method, a combination of source-IP and destination-IP address-based forwarding, can be used if it is not clear whether source-IP or destination-IP address-based forwarding is better suited on a particular switch. In this method, packets sent from the IP address A to IP address B, from IP address A to IP address C, and from IP address C to IP address B could all use different ports in the channel. Different load-balancing methods have different advantages, and the choice of a particular load-balancing method should be based on the position of the switch in the network and the kind of traffic that needs to be load-distributed. In Figure 33-3, an EtherChannel of four workstations communicates with a router. Because the router is a single-MAC-address device, source-based forwarding on the switch EtherChannel ensures that the switch uses all available bandwidth to the router. The router is configured for destination-based forwarding because the large number of workstations ensures that the traffic is evenly distributed from the router EtherChannel. Use the option that provides the greatest variety in your configuration. For example, if the traffic on a channel is going only to a single MAC address, using the destination-MAC address always chooses the same link in the channel. Using source addresses or IP addresses might result in better load balancing.

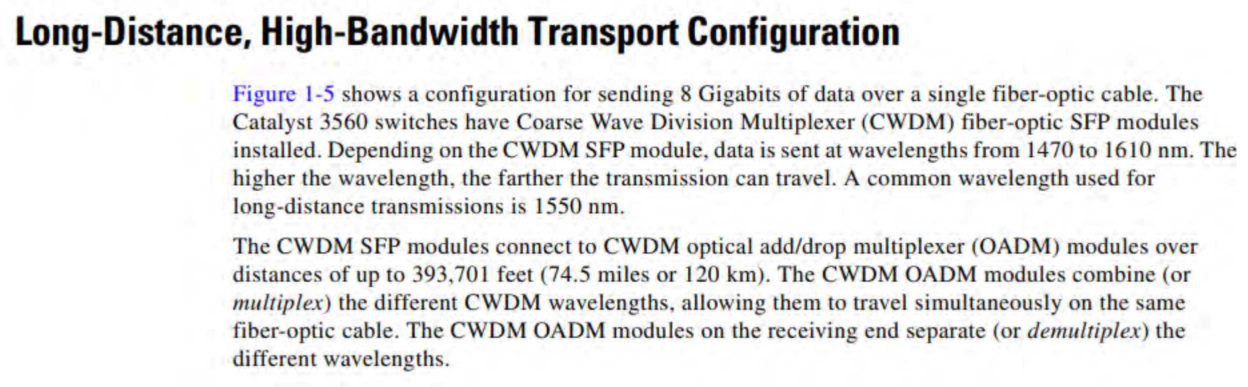
No.	’740 Patent Claim 5	Cisco EtherChannel
		Figure 33-3 Load Distribution and Forwarding Methods  Switch with source-based forwarding enabled EtherChannel Cisco router with destination-based forwarding enabled 101239 Layer 2 EtherChannel **Catalyst 2950** Catalyst 2950 switches only support 802.1Q trunking and do not support ISL trunking. Catalyst 2950 switches support DTP and PAgP dynamic trunking and channel negotiation with Cisco IOS Software Release 12.1 releases and static modes only with Cisco IOS Software Release 12.0 releases. EtherChannel load balancing can use either source-MAC or destination-MAC address forwarding. You

No.	’740 Patent Claim 5	Cisco EtherChannel
		can configure the load balancing method by issuing the **port-channel load-balance** global configuration command. These switches support up to eight switch ports per channel. **Catalyst 6500 That Runs Cisco IOS Software** Catalyst 6500 switches that run Cisco IOS Software support L2 (switchport) and Layer 3 (L3) (routed port) EtherChannel configurations. A Catalyst 6500/6000 series switch supports a maximum of 64 EtherChannels (256 with Cisco IOS Software Release 12.1(2)E and earlier). You can form an EtherChannel with up to eight compatibly configured LAN ports on any module in a Catalyst 6000 series switch, with the exception of Digital Feature Card (DFC)-equipped modules (such as WS-X6816 and so on) which currently allow an L2 channel only using ports on the same DFC module. However, an L3 channel can be configured across different DFC-equipped modules. This limitation has been removed in Catalyst 6500/6000 Cisco IOS Software Release 12.1(11b)EX and later. This document configures an L2 EtherChannel. The Catalyst 6500/6000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP addresses, or Layer 4 (L4) port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. Catalyst 6500/6000 switches support both ISL and 802.1Q trunking encapsulations and DTP. Detailed information on port capabilities is available by issuing the **show interface** *interface_id* **capabilities** command. **Catalyst 4000 That Runs Cisco IOS Software** Catalyst 4000 switches that run Cisco IOS Software (with Supervisor Engine III and IV) support L2 (switchport) and L3 (routed port) EtherChannel configurations. A Catalyst 4000 series switch supports a

No.	’740 Patent Claim 5	Cisco EtherChannel
		maximum of 64 EtherChannels. You can form an EtherChannel with up to eight compatibly configured Ethernet interfaces on any module, and across modules in a Catalyst 4000 series switch. All interfaces in each EtherChannel must be the same speed and must all be configured as either L2 or L3 interfaces. The Catalyst 4000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP address, or L4 port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. The Catalyst 4000 that runs Cisco IOS Software supports ISL and 802.1Q trunking encapsulations and DTP. ISL is not available on certain modules. For a complete list of such modules, refer to the *Understanding VLAN Trunks* section of Configuring Layer 2 Ethernet Interfaces. In a future software release, detailed information on port capabilities will be available by issuing the **show interface capabilities** command. Currently this command is not available.

No.	’740 Patent Claim 6	Cisco EtherChannel
6	The method according to claim 1, wherein coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from	Cisco EtherChannel System discloses the method according to claim 1, wherein coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node. For example, Cisco EtherChannel System discloses multiplexing and demultiplexing data packets sent between a phone, router, switch, or server and the larger network into ingress and egress queues. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Cisco EtherChannel System is found to not meet this limitation, wherein coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the

No.	’740 Patent Claim 6	Cisco EtherChannel
	the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node.	communication network to the network node would have been obvious to a person having ordinary skill in the art, as explained below. *See supra* Claim 1. Catalyst 3560 Configuration Guide at Figure 32-6  **Queueing and Scheduling Overview** The switch has queues at specific points to help prevent congestion as shown in Figure 32-6. ***Figure 32-6 Ingress and Egress Queue Location*** Traffic Classify Policer Marker Policer Marker Policer Marker Policer Marker Ingress queues SRR Internal ring Egress queues SRR 90563 Because the total ingress bandwidth of all ports can exceed the bandwidth of the internal ring, ingress queues are located after the packet is classified, policed, and marked and before packets are forwarded into the switch fabric. Because multiple ingress ports can simultaneously send packets to an egress port and cause congestion, egress queues are located after the internal ring. Catalyst 3560 Configuration Guide at 1-20

No.	’740 Patent Claim 6	Cisco EtherChannel
		 **Long-Distance, High-Bandwidth Transport Configuration** Figure 1-5 shows a configuration for sending 8 Gigabits of data over a single fiber-optic cable. The Catalyst 3560 switches have Coarse Wave Division Multiplexer (CWDM) fiber-optic SFP modules installed. Depending on the CWDM SFP module, data is sent at wavelengths from 1470 to 1610 nm. The higher the wavelength, the farther the transmission can travel. A common wavelength used for long-distance transmissions is 1550 nm. The CWDM SFP modules connect to CWDM optical add/drop multiplexer (OADM) modules over distances of up to 393,701 feet (74.5 miles or 120 km). The CWDM OADM modules combine (or *multiplex*) the different CWDM wavelengths, allowing them to travel simultaneously on the same fiber-optic cable. The CWDM OADM modules on the receiving end separate (or *demultiplex*) the different wavelengths. Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Cisco EtherChannel System in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 6 of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Wiher discloses multiplexing and demultiplexing circuitry to transmit and receive ATM data cells over a data link between the ATM network and network access equipment. Wiher at 3:43-65 (“In general, in another aspect, the invention features an apparatus for communicating data cells between a data link and a backplane. The apparatus includes transceiver circuitry to transmit and receive data cells over a data link and a plurality of backplane interfaces each including at least one cell signal terminal. Each of the backplane interface is coupled to a backplane interconnection circuit. Each backplane interconnection circuit transmits and receives cells over the cell signal terminals of its associated backplane interface. The apparatus also includes de-multiplexing circuitry coupling the transceiver circuitry to each of the backplane interconnection circuits. The de-multiplexing circuitry receives a data cell from the transceiver circuitry, select a backplane interconnection circuit associated with the data cell, and provide the data cell to the selected backplane interconnection circuit for transmission over the cell signal terminals of the

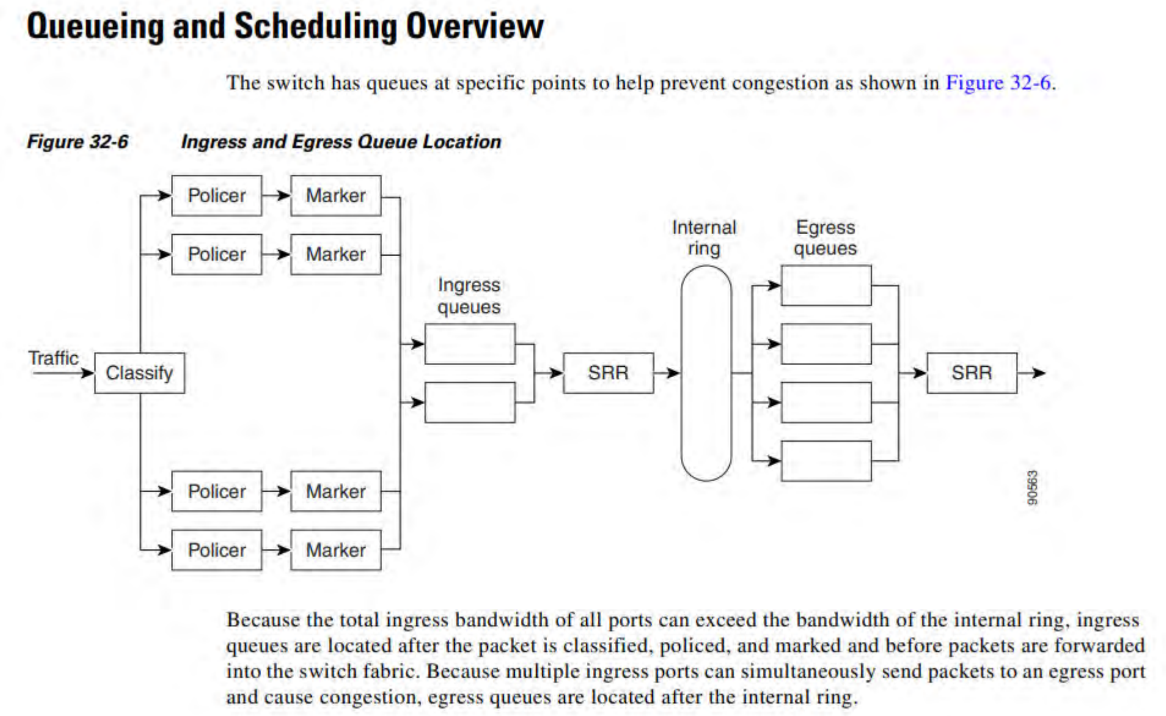
No.	’740 Patent Claim 6	Cisco EtherChannel
		associated backplane interface. The apparatus also includes multiplexing circuitry coupling the plurality of backplane interconnection circuits to the transceiver circuitry. The multiplexing circuitry receives data cells from each of the backplane interconnection circuits and provide the received data cells to the transceiver circuitry.”) Wiher at 3:66-4:22 (“Implementations of the invention may include one or more of the following features. The backplane interconnection circuits may independently receive and transmit data cells over the plurality of backplane interfaces. The de-multiplexing circuitry may select a backplane interface based on data in the header field of the data cell. The apparatus may include header translation circuitry to alter header data in cells sent between the plurality of backplane interfaces and the transceiver circuitry. Each of the plurality of backplane interfaces may include separate terminals to receive cells and separate terminals to transmit cells. The terminals to transmit cells may include a first and second control terminal and at least one outgoing cell data terminal. A backplane interface's backplane interconnection circuitry may accepts a signal on the first control terminal as indicating that a cell may be sent over the interface, asserts a 15 signal on the second control terminal to indicate that a cell is being transmitted, and transmits data bits of the cell on the outgoing cell data terminal. Each backplane interface may include a single outgoing cell data terminal and each bit of the cell may be serially transmitted over the single outgoing cell data terminal. Each backplane interface may include multiple outgoing cell data terminals and bits of the cell may be sent in parallel over the eight outgoing cell data terminals.”) For example, Lebizay discloses an optical add/drop multiplexer that multiplexers and demultiplexes data packets sent between the network boards and network. Lebizay at [0043] (“InfiniBand offers link layer Virtual Lanes (VLs) to support multiple logical channels (i.e. multiplexing) on the same physical link. Infiniband offers up to 16 virtual lanes per link. VLs provide a mechanism to avoid head-of-line blocking and the ability to support Quality of Service (QoS). The difference between a Virtual Lane and a Service Level (SL) is that a Virtual Lane is the actual logical lane (mul-tiplexed) used on a given point-to-point link. The Service Level stays constant as a packet traverses the fabric, and specifies the desired service level within a subnet. The SL (AF, EF or BE) is included in the link header, and each switch maps the SL to a VL

No.	’740 Patent Claim 6	Cisco EtherChannel
		supported by the destination link. A switch supporting a limited number of virtual lanes will map the SL field to a VL it supports. Without preserving the SL, the desired SL (AF, EF or BE) would be lost in this mapping, and later in the path, a switch supporting more VLs would be unable to recover finer granularity of SLs between two packets mapped to the same VL.”) Lebizay at [0050] (“The issue with using a ring, however, is how to map the addressing of multiple boards across these fibers. One solution is to employ Wavelength Division Multiplex-ing (WDM). A WDM optical mesh defines a meshed-topology in the wavelength space as opposed to the physical fiber space. By utilizing multiple discrete lambda-waves as optical carriers such that by meshing dedicated optical wavelengths between every two boards, layer 2 protocols are eliminated, thereby creating a dramatic improvement in the efficiency of the transport. Today, every packet transport requires a protocol that allows the end point (and interme-diate points) to decipher the intended path (or consumer) of the packet. This protocol increases the amount of overhead required in the packet bus, allowing less room for actual data to be sent. By moving the protocol into the wavelength of the actual optical signal, the destination is implied by the wavelength and no additional bandwidth needs to be sur-rendered on the signal to provide this information. This makes the efficiency of the transport better and also speeds the routing of the packet through the network. In addition, the use of optical interconnects in a backplane environment greatly increases chassis bandwidth as well as reducing electrical radiation that often accompanies copper intercon-nects. The components involved include an optical back-plane in a physical ring topology, and the necessary trans-mitters and receivers for the size of the installation (i.e., number of slots in the chassis). In addition, optical add/drop multiplexer devices are required.”)

No.	’740 Patent Claim 7	Cisco EtherChannel
7	The method according to claim 1,	Cisco EtherChannel System discloses the method according to claim 1, wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the first and second physical links.

No.	’740 Patent Claim 7	Cisco EtherChannel
	wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the first and second physical links.	For example, Cisco EtherChannel System discloses forwarding data packets by selecting a specific port and switch fabric interconnection using load balancing considerations to distribute the packets at rates over the ports and switch fabric interconnections. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met *See supra* Claim 1. Cisco Catalyst 6500 Data Sheet at 13 (“How Cisco Express Forwarding Works Cisco Express Forwarding is a Layer 3 technology that provides increased forwarding scalability and performance to manage the many short-duration traffic flows common in today’s enterprise and service provider networks. To meet the needs of environments managing large amounts of short-flow, Web-based, or highly interactive types of traffic, Cisco Express Forwarding forwards all packets in hardware, and maintains its forwarding rate independent of the number of flows going though the switch. On the Cisco Catalyst 6500 Series, the Cisco Express Forwarding Layer 3 forwarding engine is located centrally on the supervisor engine’s policy feature card (PFC2 or PFC3)—the same device that performs hardware-based Layer 2 and Layer 3 forwarding, access control list (ACL) checking, QoS policing and marking, and NetFlow statistics gathering. Using the routing table that Cisco IOS Software builds to define configured interfaces and routing protocols, the Cisco Express Forwarding architecture creates Cisco Express Forwarding tables and downloads them into the hardware forwarding engine before any user traffic is sent through the switch. The Cisco Express Forwarding architecture places only the routing prefixes in its Cisco Express Forwarding tables—the only information it requires to make the Layer 3 forwarding decisions—relying on the routing protocols to do route selection. By performing a simple Cisco Express Forwarding table lookup, the switch forwards packets at wire rate, independent of the number of flows transiting the switch.”) Catalyst 3560 Configuration Guide at Figure 32-6

<table>
<tr><th>No.</th><th>’740 Patent Claim 7</th><th>Cisco EtherChannel</th></tr>
<tr><td></td><td></td><td>



Queueing and Scheduling Overview

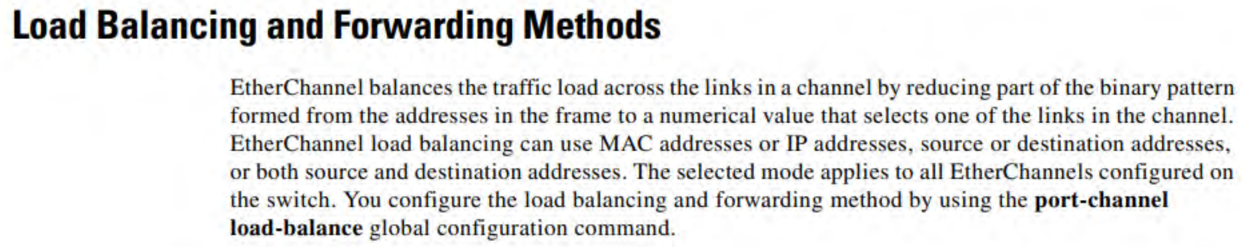
The switch has queues at specific points to help prevent congestion as shown in Figure 32-6.

Figure 32-6 Ingress and Egress Queue Location

Traffic
Classify
Policer
Marker
Policer
Marker
Policer
Marker
Policer
Marker
Ingress queues
SRR
Internal ring
Egress queues
SRR
90563

Because the total ingress bandwidth of all ports can exceed the bandwidth of the internal ring, ingress queues are located after the packet is classified, policed, and marked and before packets are forwarded into the switch fabric. Because multiple ingress ports can simultaneously send packets to an egress port and cause congestion, egress queues are located after the internal ring.

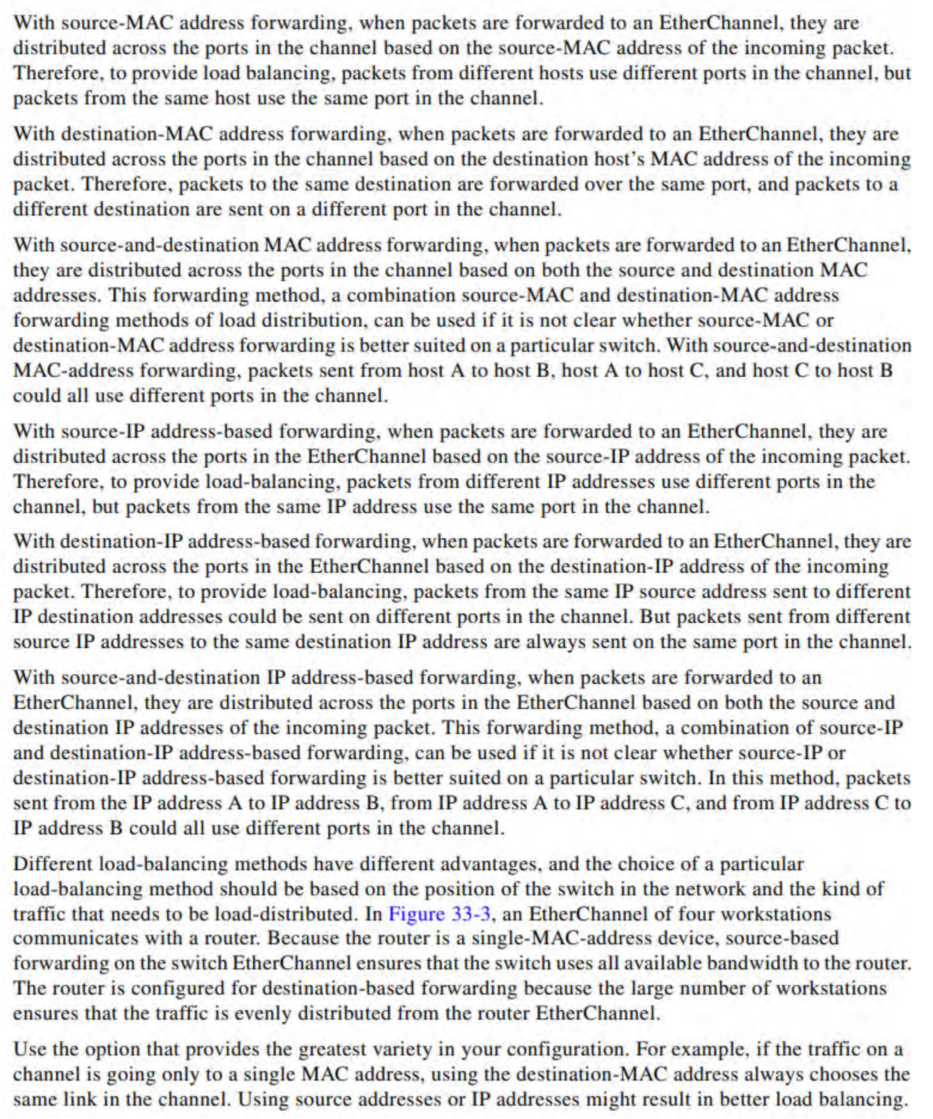
Catalyst 3560 Configuration Guide at 33-6 – 33-8

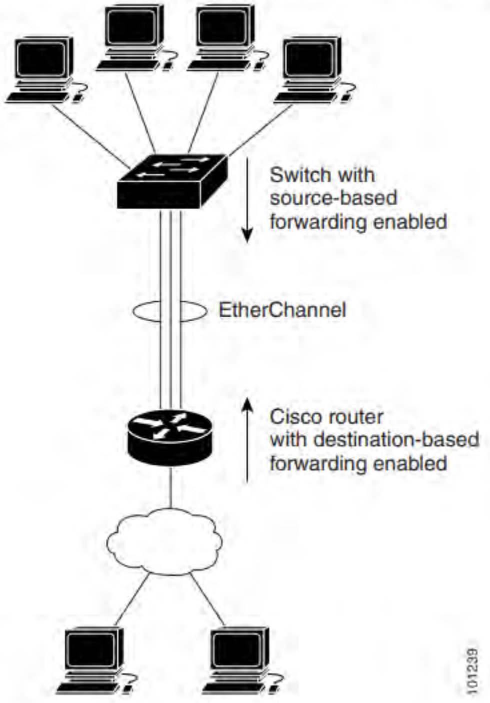


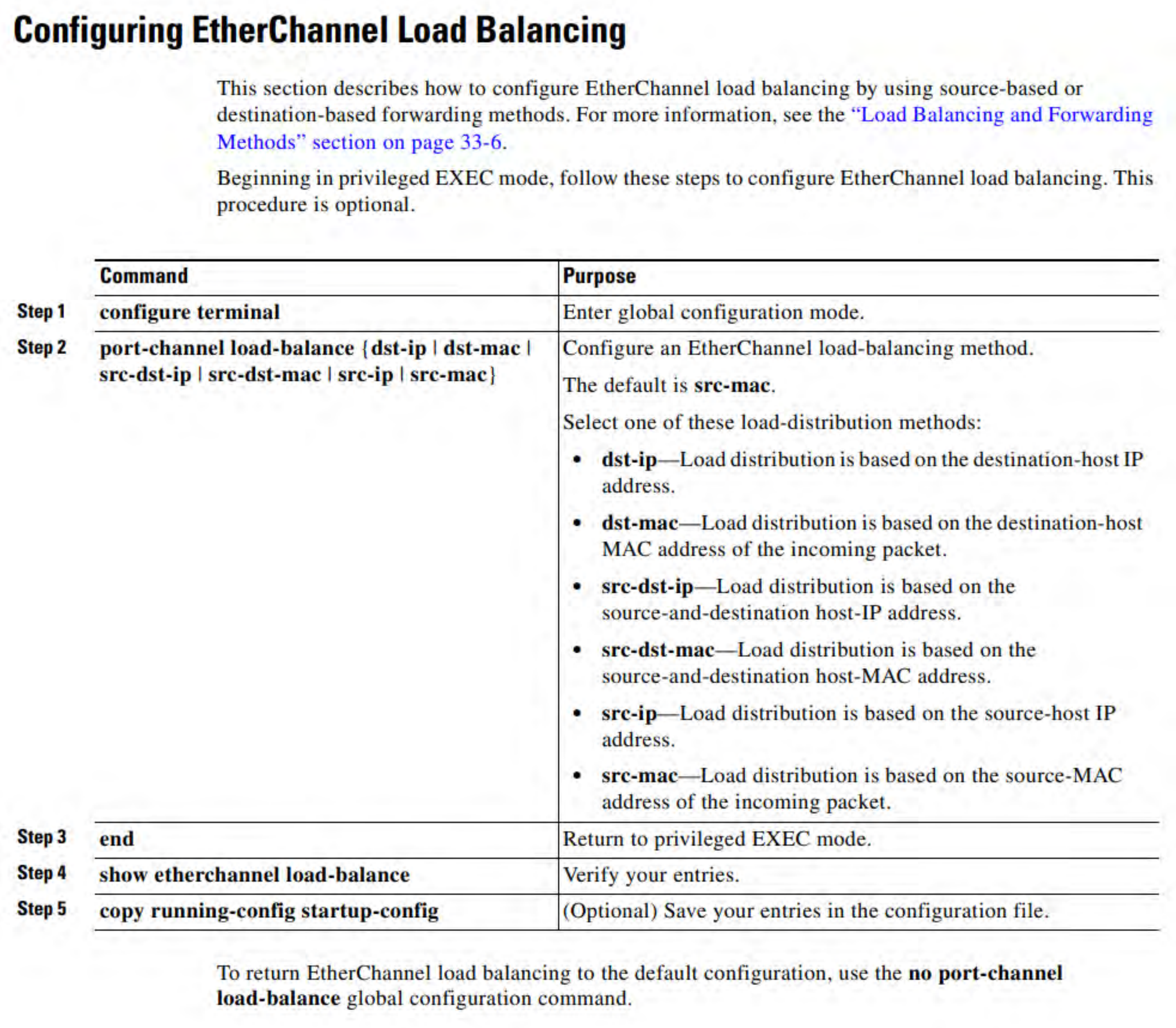
Load Balancing and Forwarding Methods

EtherChannel balances the traffic load across the links in a channel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel. EtherChannel load balancing can use MAC addresses or IP addresses, source or destination addresses, or both source and destination addresses. The selected mode applies to all EtherChannels configured on the switch. You configure the load balancing and forwarding method by using the **port-channel load-balance** global configuration command.

</td></tr>
</table>

No.	’740 Patent Claim 7	Cisco EtherChannel
		 With source-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the source-MAC address of the incoming packet. Therefore, to provide load balancing, packets from different hosts use different ports in the channel, but packets from the same host use the same port in the channel. With destination-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the destination host’s MAC address of the incoming packet. Therefore, packets to the same destination are forwarded over the same port, and packets to a different destination are sent on a different port in the channel. With source-and-destination MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on both the source and destination MAC addresses. This forwarding method, a combination source-MAC and destination-MAC address forwarding methods of load distribution, can be used if it is not clear whether source-MAC or destination-MAC address forwarding is better suited on a particular switch. With source-and-destination MAC-address forwarding, packets sent from host A to host B, host A to host C, and host C to host B could all use different ports in the channel. With source-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the source-IP address of the incoming packet. Therefore, to provide load-balancing, packets from different IP addresses use different ports in the channel, but packets from the same IP address use the same port in the channel. With destination-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the destination-IP address of the incoming packet. Therefore, to provide load-balancing, packets from the same IP source address sent to different IP destination addresses could be sent on different ports in the channel. But packets sent from different source IP addresses to the same destination IP address are always sent on the same port in the channel. With source-and-destination IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on both the source and destination IP addresses of the incoming packet. This forwarding method, a combination of source-IP and destination-IP address-based forwarding, can be used if it is not clear whether source-IP or destination-IP address-based forwarding is better suited on a particular switch. In this method, packets sent from the IP address A to IP address B, from IP address A to IP address C, and from IP address C to IP address B could all use different ports in the channel. Different load-balancing methods have different advantages, and the choice of a particular load-balancing method should be based on the position of the switch in the network and the kind of traffic that needs to be load-distributed. In Figure 33-3, an EtherChannel of four workstations communicates with a router. Because the router is a single-MAC-address device, source-based forwarding on the switch EtherChannel ensures that the switch uses all available bandwidth to the router. The router is configured for destination-based forwarding because the large number of workstations ensures that the traffic is evenly distributed from the router EtherChannel. Use the option that provides the greatest variety in your configuration. For example, if the traffic on a channel is going only to a single MAC address, using the destination-MAC address always chooses the same link in the channel. Using source addresses or IP addresses might result in better load balancing.

No.	’740 Patent Claim 7	Cisco EtherChannel
		Figure 33-3 Load Distribution and Forwarding Methods  Switch with source-based forwarding enabled EtherChannel Cisco router with destination-based forwarding enabled 101239 Catalyst 3560 Configuration Guide at 33-16

No.	’740 Patent Claim 7	Cisco EtherChannel
		

Configuring EtherChannel Load Balancing

This section describes how to configure EtherChannel load balancing by using source-based or destination-based forwarding methods. For more information, see the “Load Balancing and Forwarding Methods” section on page 33-6.

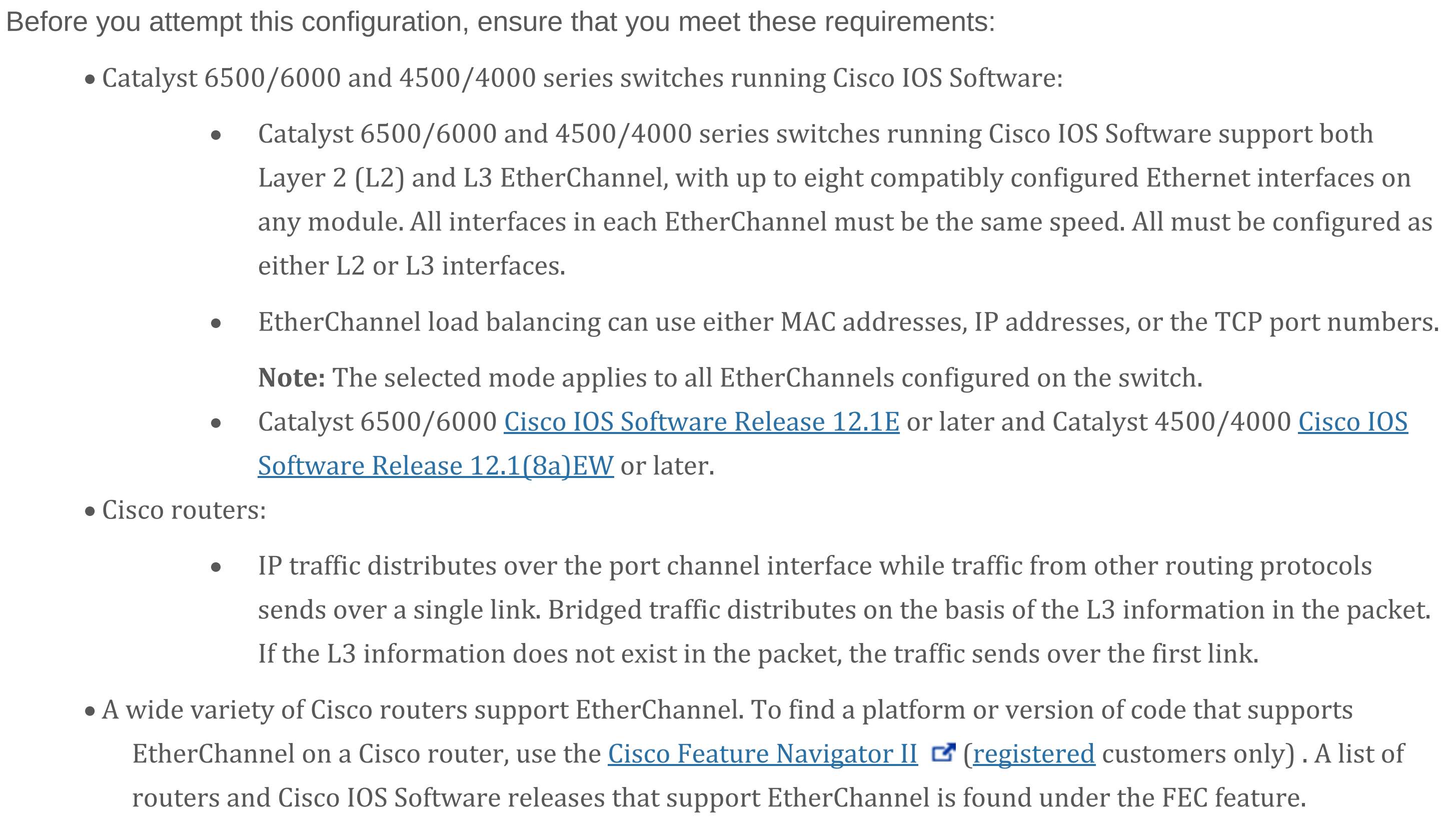
Beginning in privileged EXEC mode, follow these steps to configure EtherChannel load balancing. This procedure is optional.

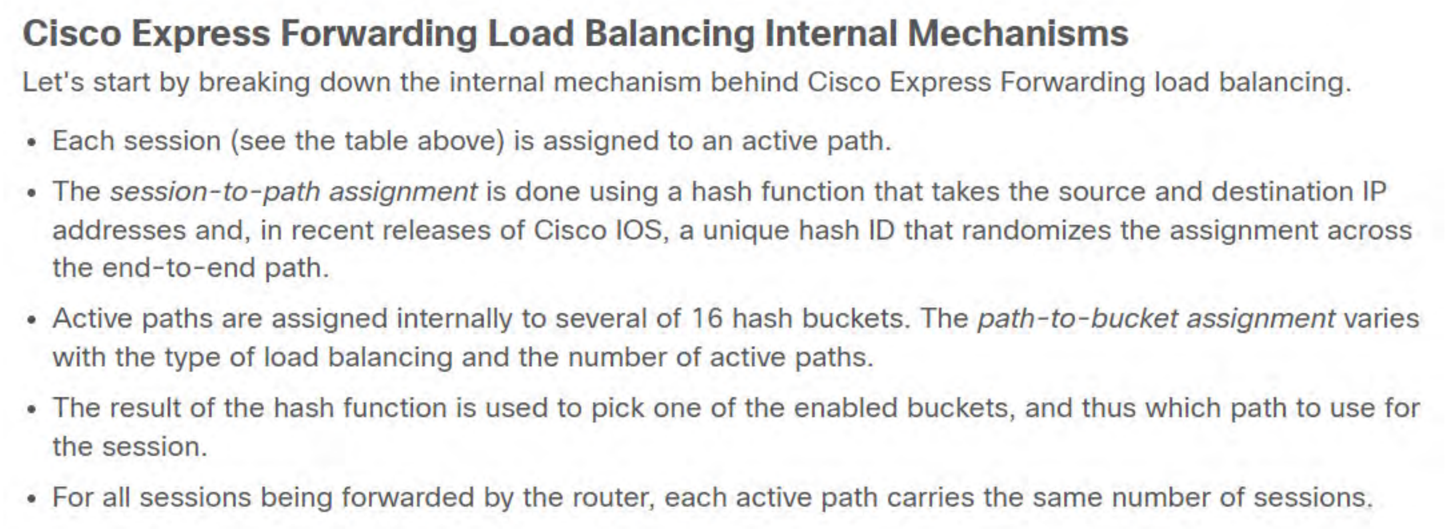
	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**port-channel load-balance** {**dst-ip** | **dst-mac** | **src-dst-ip** | **src-dst-mac** | **src-ip** | **src-mac**}	Configure an EtherChannel load-balancing method. The default is **src-mac**. Select one of these load-distribution methods: • **dst-ip**—Load distribution is based on the destination-host IP address. • **dst-mac**—Load distribution is based on the destination-host MAC address of the incoming packet. • **src-dst-ip**—Load distribution is based on the source-and-destination host-IP address. • **src-dst-mac**—Load distribution is based on the source-and-destination host-MAC address. • **src-ip**—Load distribution is based on the source-host IP address. • **src-mac**—Load distribution is based on the source-MAC address of the incoming packet.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show etherchannel load-balance**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

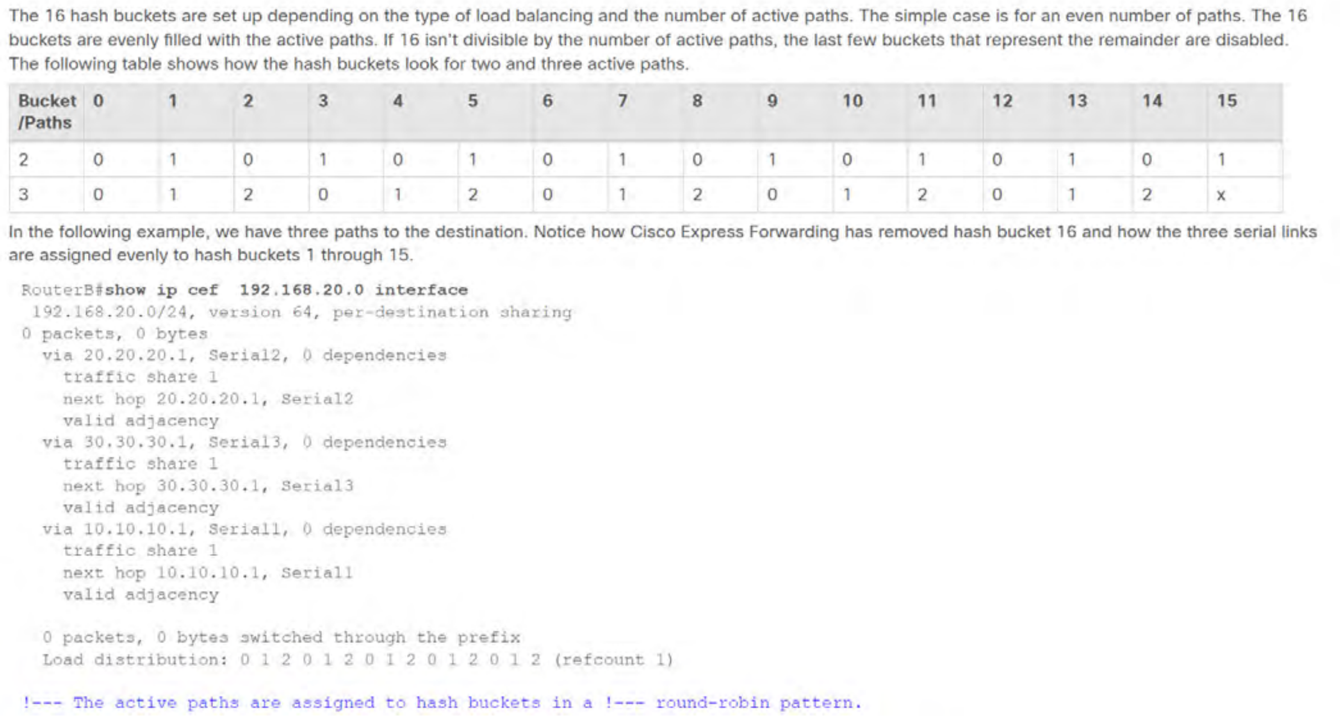
To return EtherChannel load balancing to the default configuration, use the **no port-channel load-balance** global configuration command.

No.	’740 Patent Claim 7	Cisco EtherChannel
		Layer 2 EtherChannel **Catalyst 2950** Catalyst 2950 switches only support 802.1Q trunking and do not support ISL trunking. Catalyst 2950 switches support DTP and PAgP dynamic trunking and channel negotiation with Cisco IOS Software Release 12.1 releases and static modes only with Cisco IOS Software Release 12.0 releases. EtherChannel load balancing can use either source-MAC or destination-MAC address forwarding. You can configure the load balancing method by issuing the **port-channel load-balance** global configuration command. These switches support up to eight switch ports per channel. **Catalyst 6500 That Runs Cisco IOS Software** Catalyst 6500 switches that run Cisco IOS Software support L2 (switchport) and Layer 3 (L3) (routed port) EtherChannel configurations. A Catalyst 6500/6000 series switch supports a maximum of 64 EtherChannels (256 with Cisco IOS Software Release 12.1(2)E and earlier). You can form an EtherChannel with up to eight compatibly configured LAN ports on any module in a Catalyst 6000 series switch, with the exception of Digital Feature Card (DFC)-equipped modules (such as WS-X6816 and so on) which currently allow an L2 channel only using ports on the same DFC module. However, an L3 channel can be configured across different DFC-equipped modules. This limitation has been removed in Catalyst 6500/6000 Cisco IOS Software Release 12.1(11b)EX and later. This document configures an L2 EtherChannel. The Catalyst 6500/6000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP addresses, or Layer 4 (L4) port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. Catalyst 6500/6000 switches support both ISL and 802.1Q trunking encapsulations and DTP. Detailed information on port capabilities is available by issuing the **show interface** *interface_id* **capabilities** command.

No.	’740 Patent Claim 7	Cisco EtherChannel
		Catalyst 4000 That Runs Cisco IOS Software Catalyst 4000 switches that run Cisco IOS Software (with Supervisor Engine III and IV) support L2 (switchport) and L3 (routed port) EtherChannel configurations. A Catalyst 4000 series switch supports a maximum of 64 EtherChannels. You can form an EtherChannel with up to eight compatibly configured Ethernet interfaces on any module, and across modules in a Catalyst 4000 series switch. All interfaces in each EtherChannel must be the same speed and must all be configured as either L2 or L3 interfaces. The Catalyst 4000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP address, or L4 port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. The Catalyst 4000 that runs Cisco IOS Software supports ISL and 802.1Q trunking encapsulations and DTP. ISL is not available on certain modules. For a complete list of such modules, refer to the *Understanding VLAN Trunks* section of Configuring Layer 2 Ethernet Interfaces. In a future software release, detailed information on port capabilities will be available by issuing the **show interface capabilities** command. Currently this command is not available. Configuring EtherChannel at 1 -2 **Introduction** This sample configuration demonstrates how to set up a Layer 3 (L3) EtherChannel, without VLAN trunking, between a Cisco router and a Cisco Catalyst 6500 switch running Cisco IOS® System Software. EtherChannel can be called Fast EtherChannel (FEC) or Gigabit EtherChannel (GEC); the term depends on the speed of the interfaces or ports you use to form the EtherChannel. In this example, two Fast Ethernet ports from a Cisco router and a Catalyst 6500 switch have been bundled into a FEC. Throughout this document, the terms FEC, GEC, port channel, channel, and port group all refer to EtherChannel.

No.	’740 Patent Claim 7	Cisco EtherChannel
		 Before you attempt this configuration, ensure that you meet these requirements: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software support both Layer 2 (L2) and L3 EtherChannel, with up to eight compatibly configured Ethernet interfaces on any module. All interfaces in each EtherChannel must be the same speed. All must be configured as either L2 or L3 interfaces. • EtherChannel load balancing can use either MAC addresses, IP addresses, or the TCP port numbers. **Note:** The selected mode applies to all EtherChannels configured on the switch. • Catalyst 6500/6000 Cisco IOS Software Release 12.1E or later and Catalyst 4500/4000 Cisco IOS Software Release 12.1(8a)EW or later. • Cisco routers: • IP traffic distributes over the port channel interface while traffic from other routing protocols sends over a single link. Bridged traffic distributes on the basis of the L3 information in the packet. If the L3 information does not exist in the packet, the traffic sends over the first link. • A wide variety of Cisco routers support EtherChannel. To find a platform or version of code that supports EtherChannel on a Cisco router, use the Cisco Feature Navigator II (registered customers only) . A list of routers and Cisco IOS Software releases that support EtherChannel is found under the FEC feature. Load Balancing at 7-9

No.	’740 Patent Claim 7	Cisco EtherChannel
		 **Cisco Express Forwarding Load Balancing Internal Mechanisms** Let's start by breaking down the internal mechanism behind Cisco Express Forwarding load balancing. • Each session (see the table above) is assigned to an active path. • The *session-to-path assignment* is done using a hash function that takes the source and destination IP addresses and, in recent releases of Cisco IOS, a unique hash ID that randomizes the assignment across the end-to-end path. • Active paths are assigned internally to several of 16 hash buckets. The *path-to-bucket assignment* varies with the type of load balancing and the number of active paths. • The result of the hash function is used to pick one of the enabled buckets, and thus which path to use for the session. • For all sessions being forwarded by the router, each active path carries the same number of sessions.

<table>
<tr><th>No.</th><th>’740 Patent Claim 7</th><th>Cisco EtherChannel</th></tr>
<tr><td></td><td></td><td>


The 16 hash buckets are set up depending on the type of load balancing and the number of active paths. The simple case is for an even number of paths. The 16 buckets are evenly filled with the active paths. If 16 isn't divisible by the number of active paths, the last few buckets that represent the remainder are disabled. The following table shows how the hash buckets look for two and three active paths.

<table>
<tr><th>Bucket /Paths</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th></tr>
<tr><td>2</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>3</td><td>0</td><td>1</td><td>2</td><td>0</td><td>1</td><td>2</td><td>0</td><td>1</td><td>2</td><td>0</td><td>1</td><td>2</td><td>0</td><td>1</td><td>2</td><td>x</td></tr>
</table>

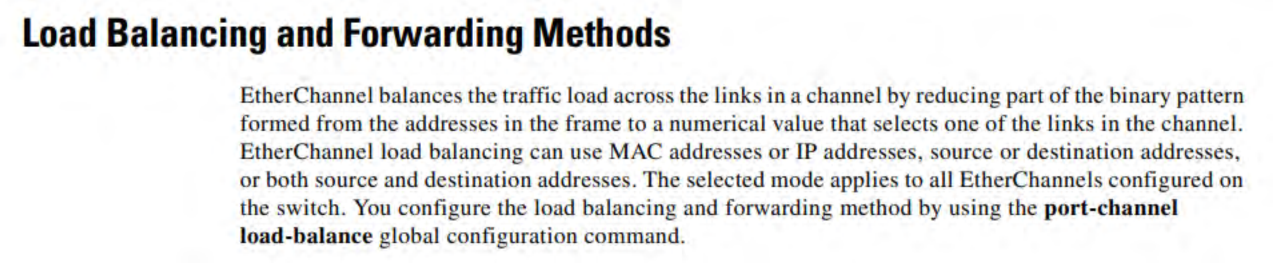
In the following example, we have three paths to the destination. Notice how Cisco Express Forwarding has removed hash bucket 16 and how the three serial links are assigned evenly to hash buckets 1 through 15.

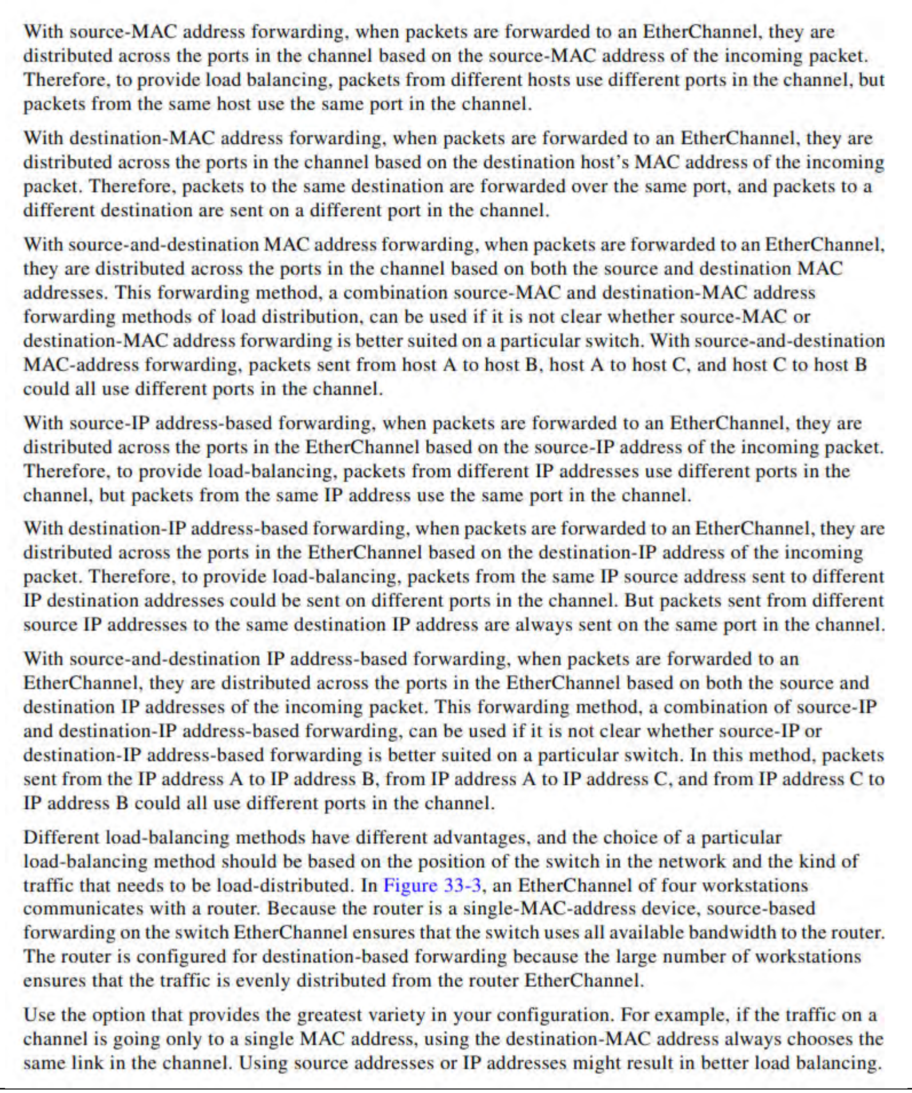
<pre>
RouterB#show ip cef 192.168.20.0 interface
 192.168.20.0/24, version 64, per-destination sharing
0 packets, 0 bytes
  via 20.20.20.1, Serial2, 0 dependencies
    traffic share 1
    next hop 20.20.20.1, Serial2
    valid adjacency
  via 30.30.30.1, Serial3, 0 dependencies
    traffic share 1
    next hop 30.30.30.1, Serial3
    valid adjacency
  via 10.10.10.1, Serial1, 0 dependencies
    traffic share 1
    next hop 10.10.10.1, Serial1
    valid adjacency

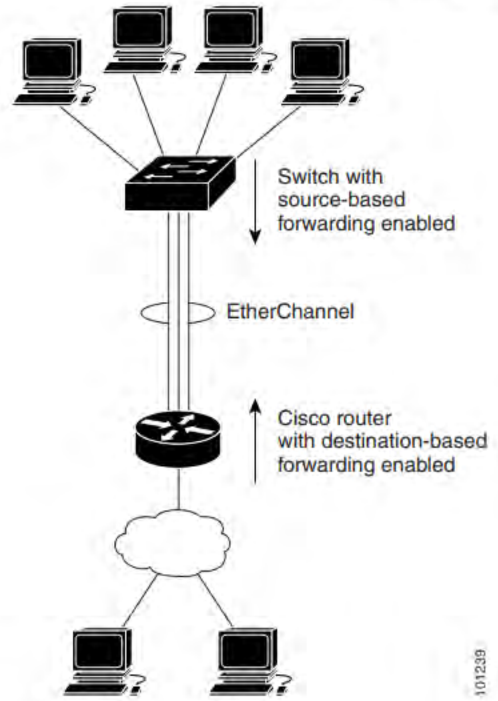
  0 packets, 0 bytes switched through the prefix
  Load distribution: 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 (refcount 1)

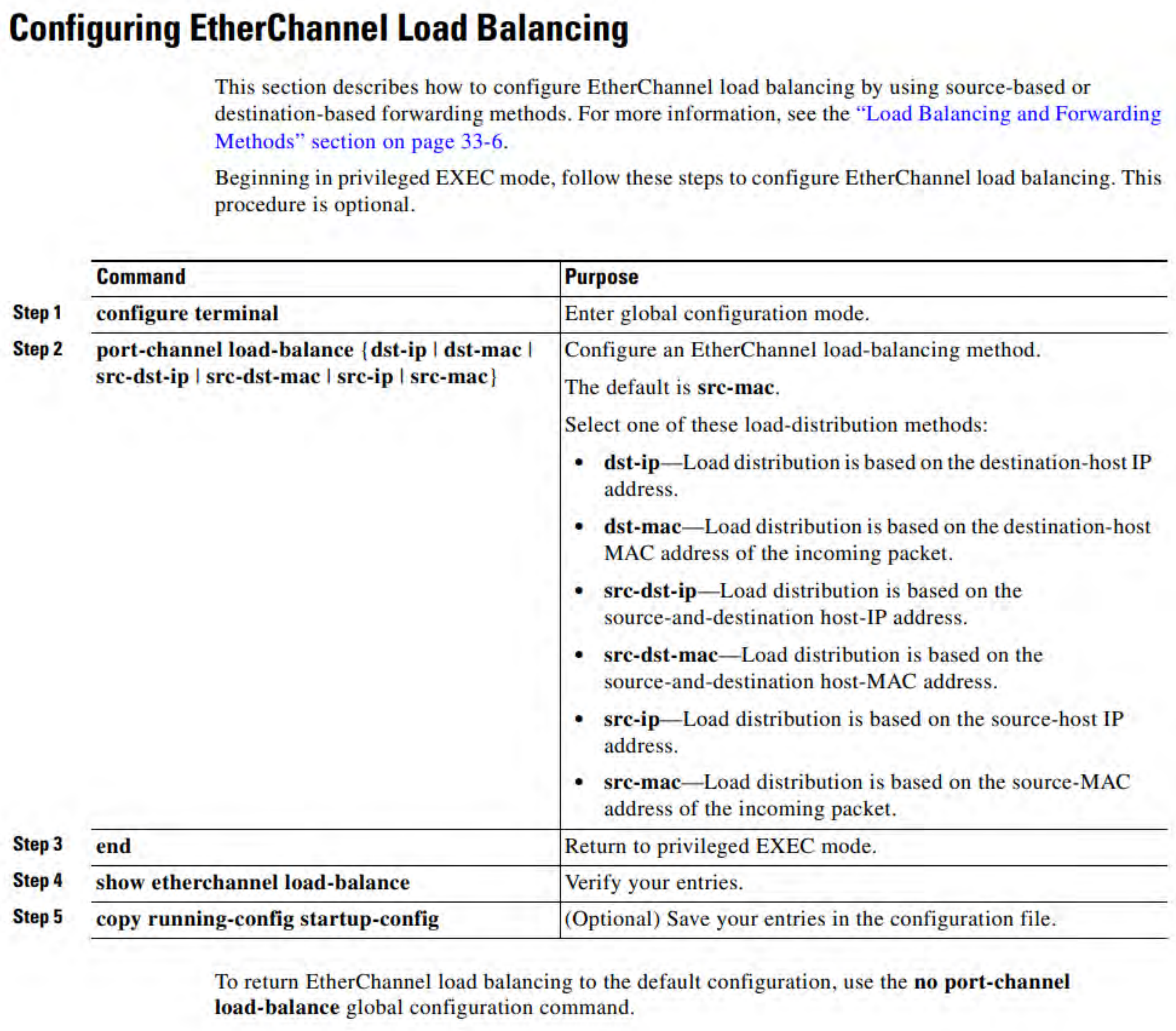
!--- The active paths are assigned to hash buckets in a !--- round-robin pattern.
</pre>
</td></tr>
</table>

No.	’740 Patent Claim 8	Cisco EtherChannel
8	The method according to claim 1, wherein selecting the first	Cisco EtherChannel System discloses the method according to claim 1, wherein selecting the first and second physical links comprises applying a mapping function to the at least one of the frame attributes. For example, Cisco EtherChannel System discloses selecting ports and switch fabric interconnections over which to send a data packet using the packet header information to forward the packet. *See supra* Claim 1.

No.	’740 Patent Claim 8	Cisco EtherChannel
	and second physical links comprises applying a mapping function to the at least one of the frame attributes.	Cisco Catalyst 6500 Data Sheet at 14 (“Using the same ASIC engine design as the central PFCx, distributed forwarding cards (DFCs) located on the interface modules forward packets between two ports, directly or across the switch fabric, without involving the supervisor engine. With the DFC, each interface module has a dedicated forwarding engine complete with the full forwarding tables. Distributed Cisco Express Forwarding (Figure 6) works like this: • As in standard Cisco Express Forwarding, the central PFCx located on the supervisor engine and the DFC engines located on the interface modules are loaded with the same Cisco Express Forwarding information derived from the forwarding table before any user traffic arrives at the switch. • As a packet arrives at an interface module, its DFC engine inspects the packet and uses the information in the Cisco Express Forwarding table (including Layer 2, Layer 3, ACLs, and QoS) to make a completely hardware-based forwarding decision for that packet. • The Distributed Cisco Express Forwarding engine manages all hardware-based forwarding for traffic on that module, including Layer 2 and Layer 3 forwarding, ACLs, QoS policing and marking, and NetFlow. • Because the DFCs make all the switching decisions locally, the central PFCx can dedicate more hardware-forwarding resources to any modules not equipped with a DFC.”) Catalyst 3560 Configuration Guide at 33-6 – 33-8  **Load Balancing and Forwarding Methods** EtherChannel balances the traffic load across the links in a channel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel. EtherChannel load balancing can use MAC addresses or IP addresses, source or destination addresses, or both source and destination addresses. The selected mode applies to all EtherChannels configured on the switch. You configure the load balancing and forwarding method by using the **port-channel load-balance** global configuration command.

No.	’740 Patent Claim 8	Cisco EtherChannel
		 With source-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the source-MAC address of the incoming packet. Therefore, to provide load balancing, packets from different hosts use different ports in the channel, but packets from the same host use the same port in the channel. With destination-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the destination host’s MAC address of the incoming packet. Therefore, packets to the same destination are forwarded over the same port, and packets to a different destination are sent on a different port in the channel. With source-and-destination MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on both the source and destination MAC addresses. This forwarding method, a combination source-MAC and destination-MAC address forwarding methods of load distribution, can be used if it is not clear whether source-MAC or destination-MAC address forwarding is better suited on a particular switch. With source-and-destination MAC-address forwarding, packets sent from host A to host B, host A to host C, and host C to host B could all use different ports in the channel. With source-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the source-IP address of the incoming packet. Therefore, to provide load-balancing, packets from different IP addresses use different ports in the channel, but packets from the same IP address use the same port in the channel. With destination-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the destination-IP address of the incoming packet. Therefore, to provide load-balancing, packets from the same IP source address sent to different IP destination addresses could be sent on different ports in the channel. But packets sent from different source IP addresses to the same destination IP address are always sent on the same port in the channel. With source-and-destination IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on both the source and destination IP addresses of the incoming packet. This forwarding method, a combination of source-IP and destination-IP address-based forwarding, can be used if it is not clear whether source-IP or destination-IP address-based forwarding is better suited on a particular switch. In this method, packets sent from the IP address A to IP address B, from IP address A to IP address C, and from IP address C to IP address B could all use different ports in the channel. Different load-balancing methods have different advantages, and the choice of a particular load-balancing method should be based on the position of the switch in the network and the kind of traffic that needs to be load-distributed. In Figure 33-3, an EtherChannel of four workstations communicates with a router. Because the router is a single-MAC-address device, source-based forwarding on the switch EtherChannel ensures that the switch uses all available bandwidth to the router. The router is configured for destination-based forwarding because the large number of workstations ensures that the traffic is evenly distributed from the router EtherChannel. Use the option that provides the greatest variety in your configuration. For example, if the traffic on a channel is going only to a single MAC address, using the destination-MAC address always chooses the same link in the channel. Using source addresses or IP addresses might result in better load balancing.

No.	’740 Patent Claim 8	Cisco EtherChannel
		Figure 33-3 Load Distribution and Forwarding Methods  Switch with source-based forwarding enabled EtherChannel Cisco router with destination-based forwarding enabled 101239 Catalyst 3560 Configuration Guide at 33-16



Configuring EtherChannel Load Balancing

This section describes how to configure EtherChannel load balancing by using source-based or destination-based forwarding methods. For more information, see the "Load Balancing and Forwarding Methods" section on page 33-6.

Beginning in privileged EXEC mode, follow these steps to configure EtherChannel load balancing. This procedure is optional.

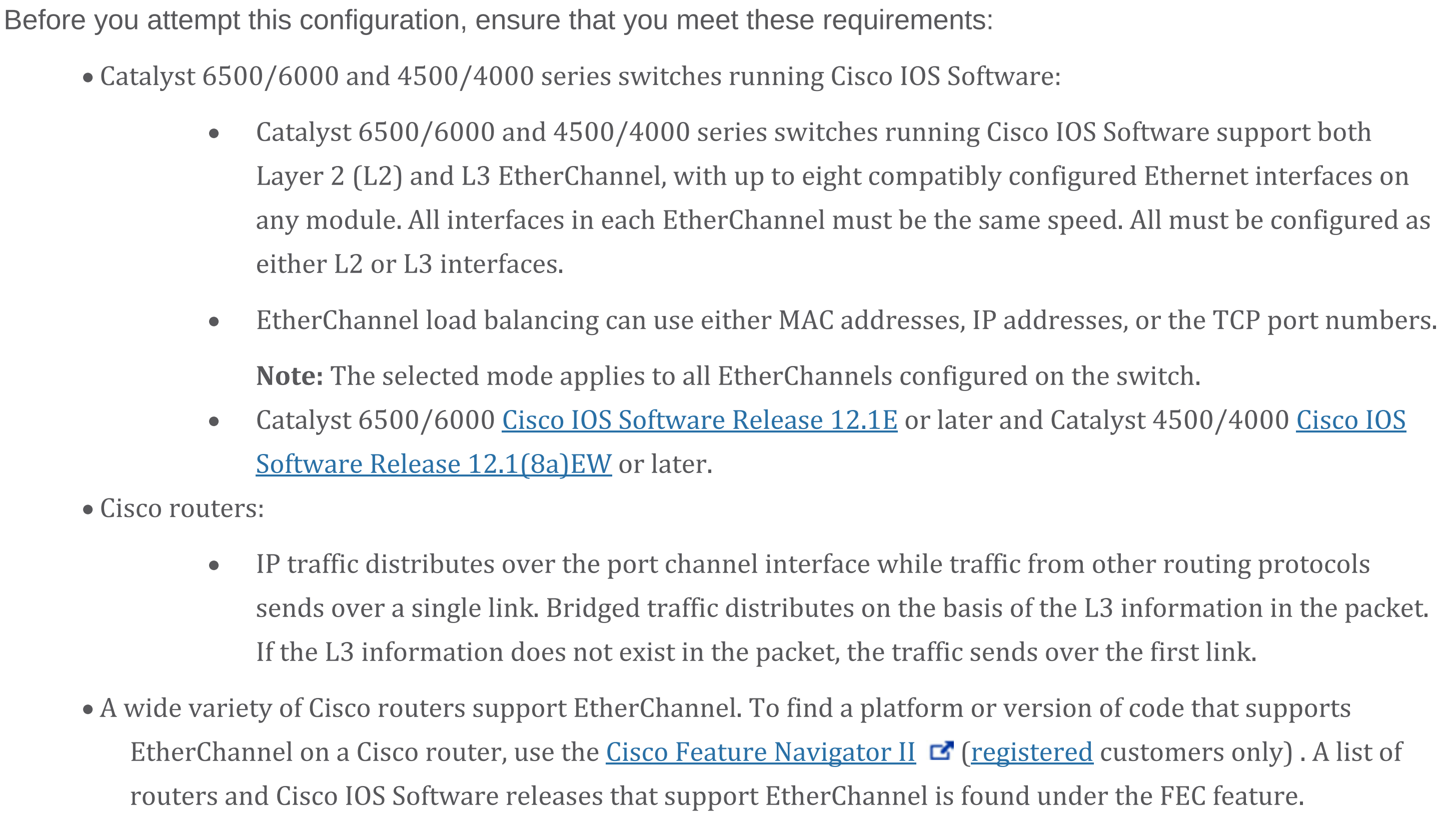
	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**port-channel load-balance** {**dst-ip** | **dst-mac** | **src-dst-ip** | **src-dst-mac** | **src-ip** | **src-mac**}	Configure an EtherChannel load-balancing method. The default is **src-mac**. Select one of these load-distribution methods: • **dst-ip**—Load distribution is based on the destination-host IP address. • **dst-mac**—Load distribution is based on the destination-host MAC address of the incoming packet. • **src-dst-ip**—Load distribution is based on the source-and-destination host-IP address. • **src-dst-mac**—Load distribution is based on the source-and-destination host-MAC address. • **src-ip**—Load distribution is based on the source-host IP address. • **src-mac**—Load distribution is based on the source-MAC address of the incoming packet.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show etherchannel load-balance**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

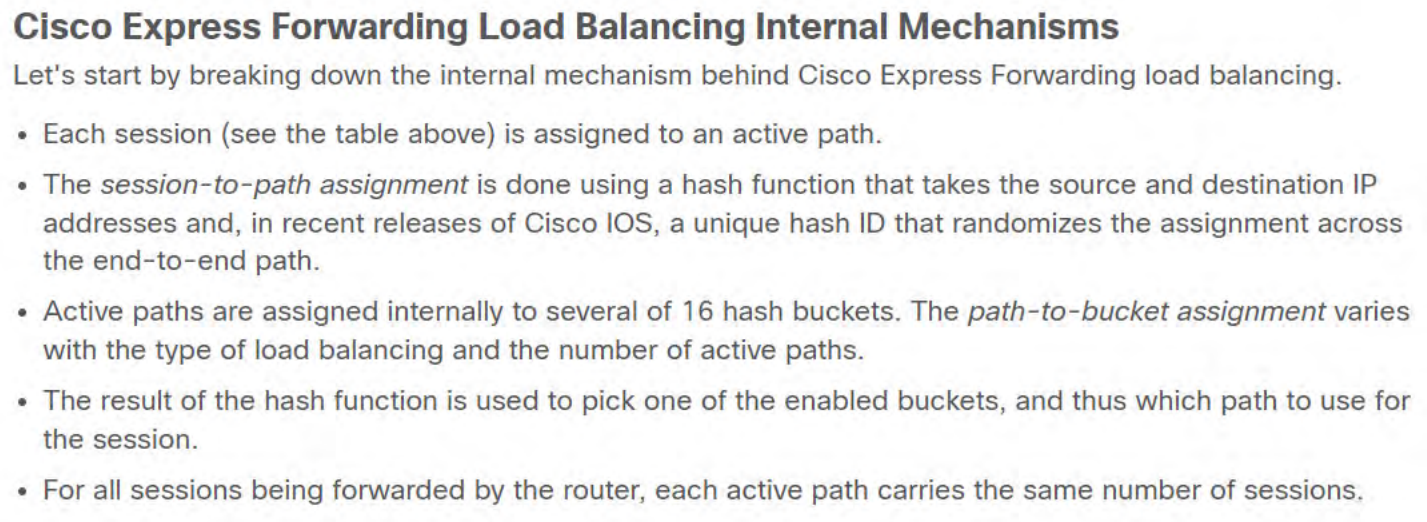
To return EtherChannel load balancing to the default configuration, use the **no port-channel load-balance** global configuration command.

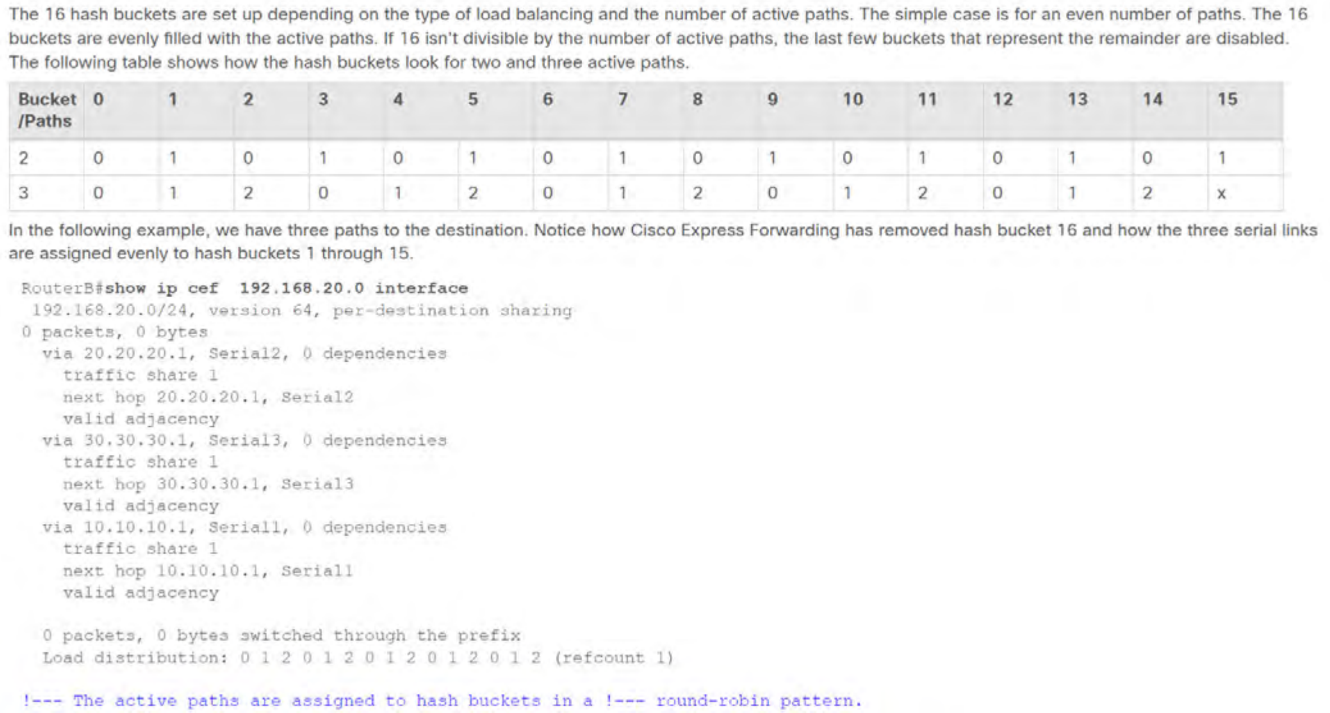
Layer 2 EtherChannel

No.	’740 Patent Claim 8	Cisco EtherChannel
		Catalyst 2950 Catalyst 2950 switches only support 802.1Q trunking and do not support ISL trunking. Catalyst 2950 switches support DTP and PAgP dynamic trunking and channel negotiation with Cisco IOS Software Release 12.1 releases and static modes only with Cisco IOS Software Release 12.0 releases. EtherChannel load balancing can use either source-MAC or destination-MAC address forwarding. You can configure the load balancing method by issuing the **port-channel load-balance** global configuration command. These switches support up to eight switch ports per channel. **Catalyst 6500 That Runs Cisco IOS Software** Catalyst 6500 switches that run Cisco IOS Software support L2 (switchport) and Layer 3 (L3) (routed port) EtherChannel configurations. A Catalyst 6500/6000 series switch supports a maximum of 64 EtherChannels (256 with Cisco IOS Software Release 12.1(2)E and earlier). You can form an EtherChannel with up to eight compatibly configured LAN ports on any module in a Catalyst 6000 series switch, with the exception of Digital Feature Card (DFC)-equipped modules (such as WS-X6816 and so on) which currently allow an L2 channel only using ports on the same DFC module. However, an L3 channel can be configured across different DFC-equipped modules. This limitation has been removed in Catalyst 6500/6000 Cisco IOS Software Release 12.1(11b)EX and later. This document configures an L2 EtherChannel. The Catalyst 6500/6000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP addresses, or Layer 4 (L4) port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. Catalyst 6500/6000 switches support both ISL and 802.1Q trunking encapsulations and DTP. Detailed information on port capabilities is available by issuing the **show interface** *interface_id* **capabilities** command.

No.	’740 Patent Claim 8	Cisco EtherChannel
		Catalyst 4000 That Runs Cisco IOS Software Catalyst 4000 switches that run Cisco IOS Software (with Supervisor Engine III and IV) support L2 (switchport) and L3 (routed port) EtherChannel configurations. A Catalyst 4000 series switch supports a maximum of 64 EtherChannels. You can form an EtherChannel with up to eight compatibly configured Ethernet interfaces on any module, and across modules in a Catalyst 4000 series switch. All interfaces in each EtherChannel must be the same speed and must all be configured as either L2 or L3 interfaces. The Catalyst 4000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP address, or L4 port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. The Catalyst 4000 that runs Cisco IOS Software supports ISL and 802.1Q trunking encapsulations and DTP. ISL is not available on certain modules. For a complete list of such modules, refer to the *Understanding VLAN Trunks* section of Configuring Layer 2 Ethernet Interfaces. In a future software release, detailed information on port capabilities will be available by issuing the **show interface capabilities** command. Currently this command is not available. Configuring EtherChannel at 1 -2 **Introduction** This sample configuration demonstrates how to set up a Layer 3 (L3) EtherChannel, without VLAN trunking, between a Cisco router and a Cisco Catalyst 6500 switch running Cisco IOS® System Software. EtherChannel can be called Fast EtherChannel (FEC) or Gigabit EtherChannel (GEC); the term depends on the speed of the interfaces or ports you use to form the EtherChannel. In this example, two Fast Ethernet ports from a Cisco router and a Catalyst 6500 switch have been bundled into a FEC. Throughout this document, the terms FEC, GEC, port channel, channel, and port group all refer to EtherChannel.

No.	’740 Patent Claim 8	Cisco EtherChannel
		 Before you attempt this configuration, ensure that you meet these requirements: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software support both Layer 2 (L2) and L3 EtherChannel, with up to eight compatibly configured Ethernet interfaces on any module. All interfaces in each EtherChannel must be the same speed. All must be configured as either L2 or L3 interfaces. • EtherChannel load balancing can use either MAC addresses, IP addresses, or the TCP port numbers. **Note:** The selected mode applies to all EtherChannels configured on the switch. • Catalyst 6500/6000 Cisco IOS Software Release 12.1E or later and Catalyst 4500/4000 Cisco IOS Software Release 12.1(8a)EW or later. • Cisco routers: • IP traffic distributes over the port channel interface while traffic from other routing protocols sends over a single link. Bridged traffic distributes on the basis of the L3 information in the packet. If the L3 information does not exist in the packet, the traffic sends over the first link. • A wide variety of Cisco routers support EtherChannel. To find a platform or version of code that supports EtherChannel on a Cisco router, use the Cisco Feature Navigator II (registered customers only) . A list of routers and Cisco IOS Software releases that support EtherChannel is found under the FEC feature. Load Balancing at 7-9

No.	’740 Patent Claim 8	Cisco EtherChannel
		 **Cisco Express Forwarding Load Balancing Internal Mechanisms** Let's start by breaking down the internal mechanism behind Cisco Express Forwarding load balancing. • Each session (see the table above) is assigned to an active path. • The *session-to-path assignment* is done using a hash function that takes the source and destination IP addresses and, in recent releases of Cisco IOS, a unique hash ID that randomizes the assignment across the end-to-end path. • Active paths are assigned internally to several of 16 hash buckets. The *path-to-bucket assignment* varies with the type of load balancing and the number of active paths. • The result of the hash function is used to pick one of the enabled buckets, and thus which path to use for the session. • For all sessions being forwarded by the router, each active path carries the same number of sessions.

No.	’740 Patent Claim 8	Cisco EtherChannel
		

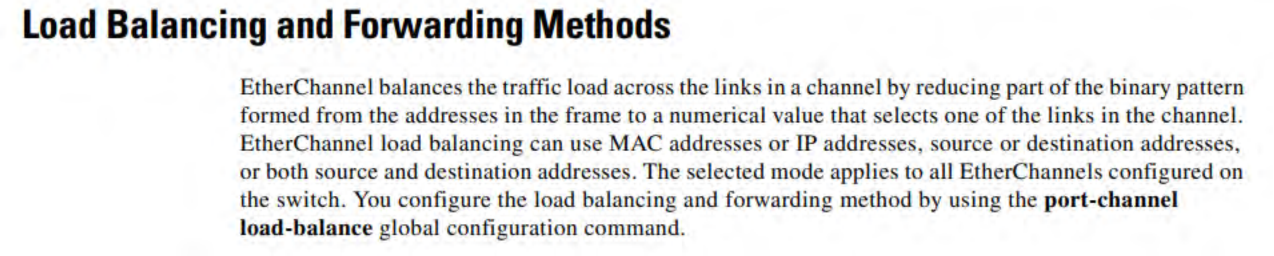
The 16 hash buckets are set up depending on the type of load balancing and the number of active paths. The simple case is for an even number of paths. The 16 buckets are evenly filled with the active paths. If 16 isn't divisible by the number of active paths, the last few buckets that represent the remainder are disabled. The following table shows how the hash buckets look for two and three active paths.

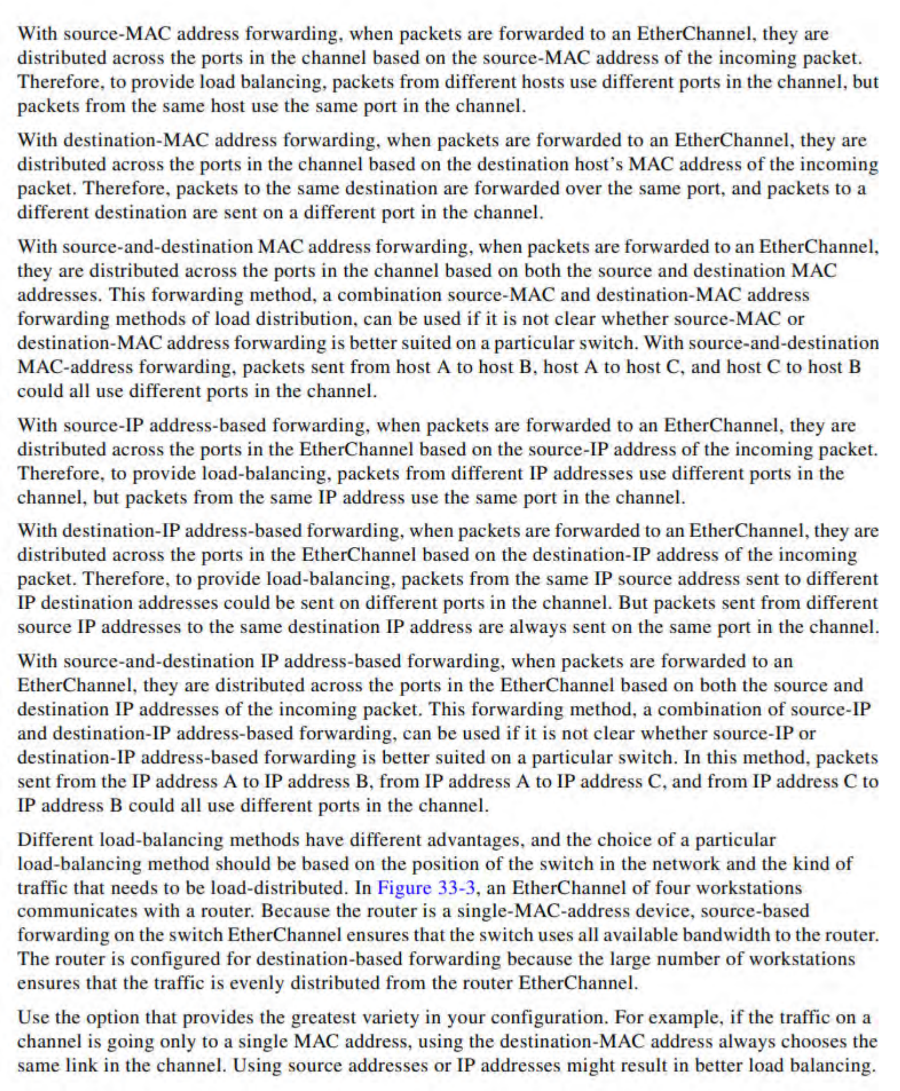
Bucket /Paths	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
3	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	x

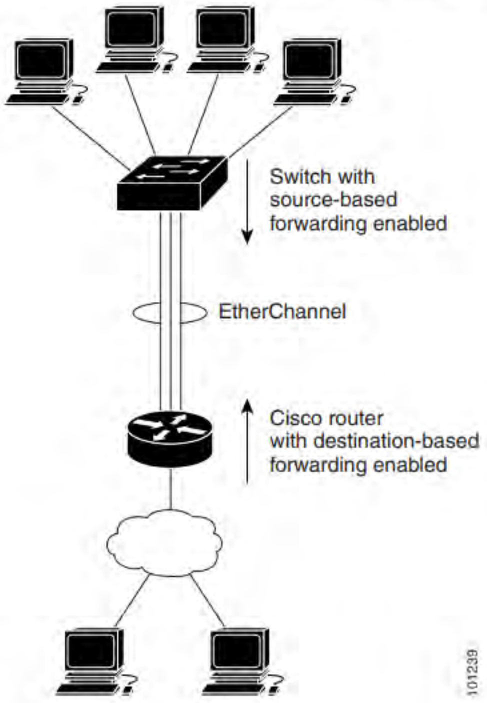
In the following example, we have three paths to the destination. Notice how Cisco Express Forwarding has removed hash bucket 16 and how the three serial links are assigned evenly to hash buckets 1 through 15.

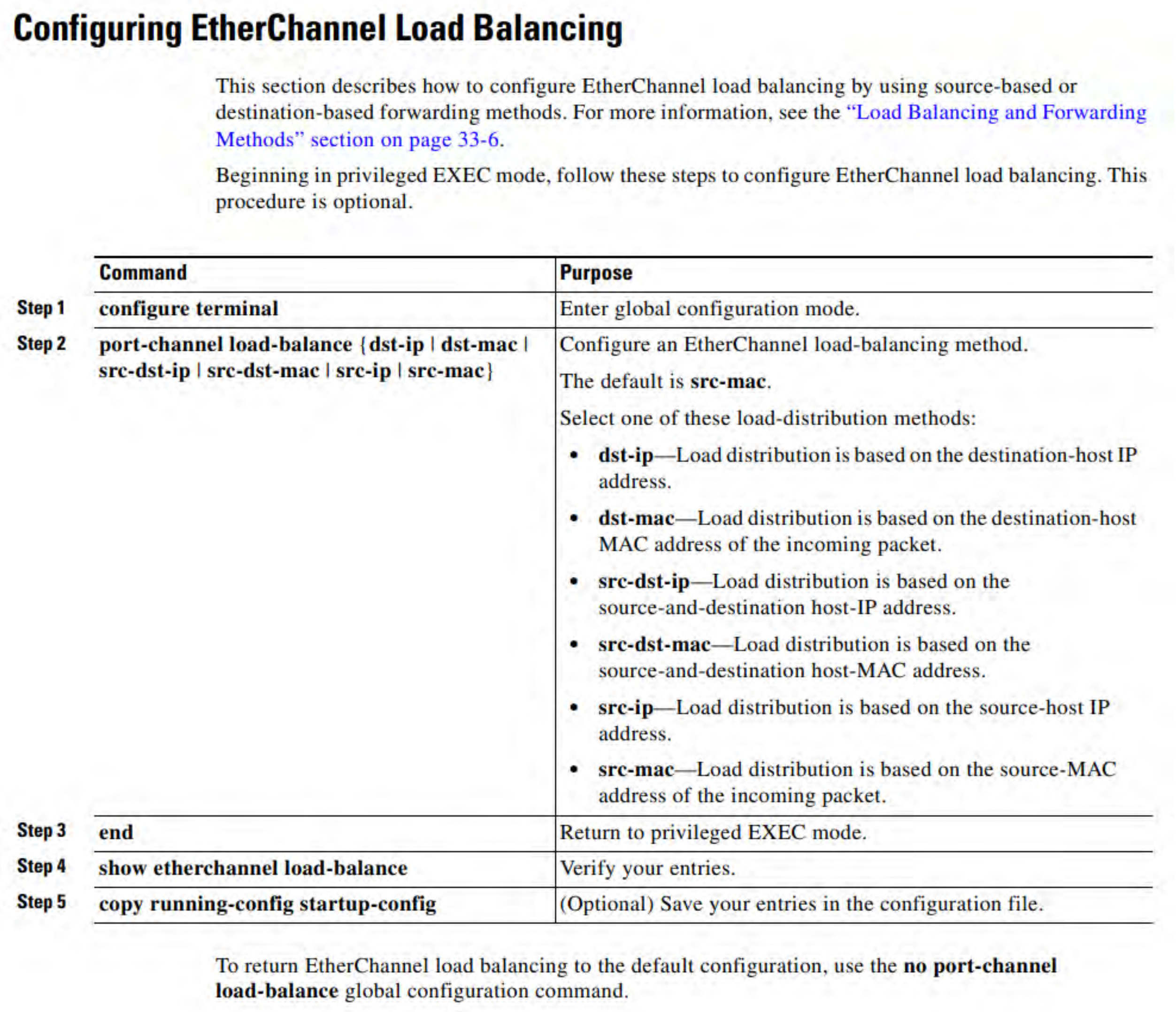
```
RouterB#show ip cef  192.168.20.0 interface
 192.168.20.0/24, version 64, per-destination sharing
0 packets, 0 bytes
  via 20.20.20.1, Serial2, 0 dependencies
    traffic share 1
    next hop 20.20.20.1, Serial2
    valid adjacency
  via 30.30.30.1, Serial3, 0 dependencies
    traffic share 1
    next hop 30.30.30.1, Serial3
    valid adjacency
  via 10.10.10.1, Serial1, 0 dependencies
    traffic share 1
    next hop 10.10.10.1, Serial1
    valid adjacency

  0 packets, 0 bytes switched through the prefix
  Load distribution: 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 (refcount 1)

 !--- The active paths are assigned to hash buckets in a !--- round-robin pattern.
```

No.	’740 Patent Claim 9	Cisco EtherChannel
9	The method according to claim 8, wherein applying	Cisco EtherChannel System discloses the method according to claim 8, wherein applying the mapping function comprises applying a hashing function. For example, Cisco EtherChannel System discloses forwarding data packets by using a port-channel load-balance global configuration command as a hash function on the packet information to select the channel over which to send the packet.

No.	’740 Patent Claim 9	Cisco EtherChannel
	the mapping function comprises applying a hashing function.	*See supra* Claim 8 Catalyst 3560 Configuration Guide at 33-6 – 33-8  **Load Balancing and Forwarding Methods** EtherChannel balances the traffic load across the links in a channel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel. EtherChannel load balancing can use MAC addresses or IP addresses, source or destination addresses, or both source and destination addresses. The selected mode applies to all EtherChannels configured on the switch. You configure the load balancing and forwarding method by using the **port-channel load-balance** global configuration command.

No.	’740 Patent Claim 9	Cisco EtherChannel
		 With source-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the source-MAC address of the incoming packet. Therefore, to provide load balancing, packets from different hosts use different ports in the channel, but packets from the same host use the same port in the channel. With destination-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the destination host’s MAC address of the incoming packet. Therefore, packets to the same destination are forwarded over the same port, and packets to a different destination are sent on a different port in the channel. With source-and-destination MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on both the source and destination MAC addresses. This forwarding method, a combination source-MAC and destination-MAC address forwarding methods of load distribution, can be used if it is not clear whether source-MAC or destination-MAC address forwarding is better suited on a particular switch. With source-and-destination MAC-address forwarding, packets sent from host A to host B, host A to host C, and host C to host B could all use different ports in the channel. With source-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the source-IP address of the incoming packet. Therefore, to provide load-balancing, packets from different IP addresses use different ports in the channel, but packets from the same IP address use the same port in the channel. With destination-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the destination-IP address of the incoming packet. Therefore, to provide load-balancing, packets from the same IP source address sent to different IP destination addresses could be sent on different ports in the channel. But packets sent from different source IP addresses to the same destination IP address are always sent on the same port in the channel. With source-and-destination IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on both the source and destination IP addresses of the incoming packet. This forwarding method, a combination of source-IP and destination-IP address-based forwarding, can be used if it is not clear whether source-IP or destination-IP address-based forwarding is better suited on a particular switch. In this method, packets sent from the IP address A to IP address B, from IP address A to IP address C, and from IP address C to IP address B could all use different ports in the channel. Different load-balancing methods have different advantages, and the choice of a particular load-balancing method should be based on the position of the switch in the network and the kind of traffic that needs to be load-distributed. In Figure 33-3, an EtherChannel of four workstations communicates with a router. Because the router is a single-MAC-address device, source-based forwarding on the switch EtherChannel ensures that the switch uses all available bandwidth to the router. The router is configured for destination-based forwarding because the large number of workstations ensures that the traffic is evenly distributed from the router EtherChannel. Use the option that provides the greatest variety in your configuration. For example, if the traffic on a channel is going only to a single MAC address, using the destination-MAC address always chooses the same link in the channel. Using source addresses or IP addresses might result in better load balancing.

No.	’740 Patent Claim 9	Cisco EtherChannel
		Figure 33-3 Load Distribution and Forwarding Methods  Switch with source-based forwarding enabled EtherChannel Cisco router with destination-based forwarding enabled 101239 Catalyst 3560 Configuration Guide at 33-16

No.	’740 Patent Claim 9	Cisco EtherChannel
		

Configuring EtherChannel Load Balancing

This section describes how to configure EtherChannel load balancing by using source-based or destination-based forwarding methods. For more information, see the “Load Balancing and Forwarding Methods” section on page 33-6.

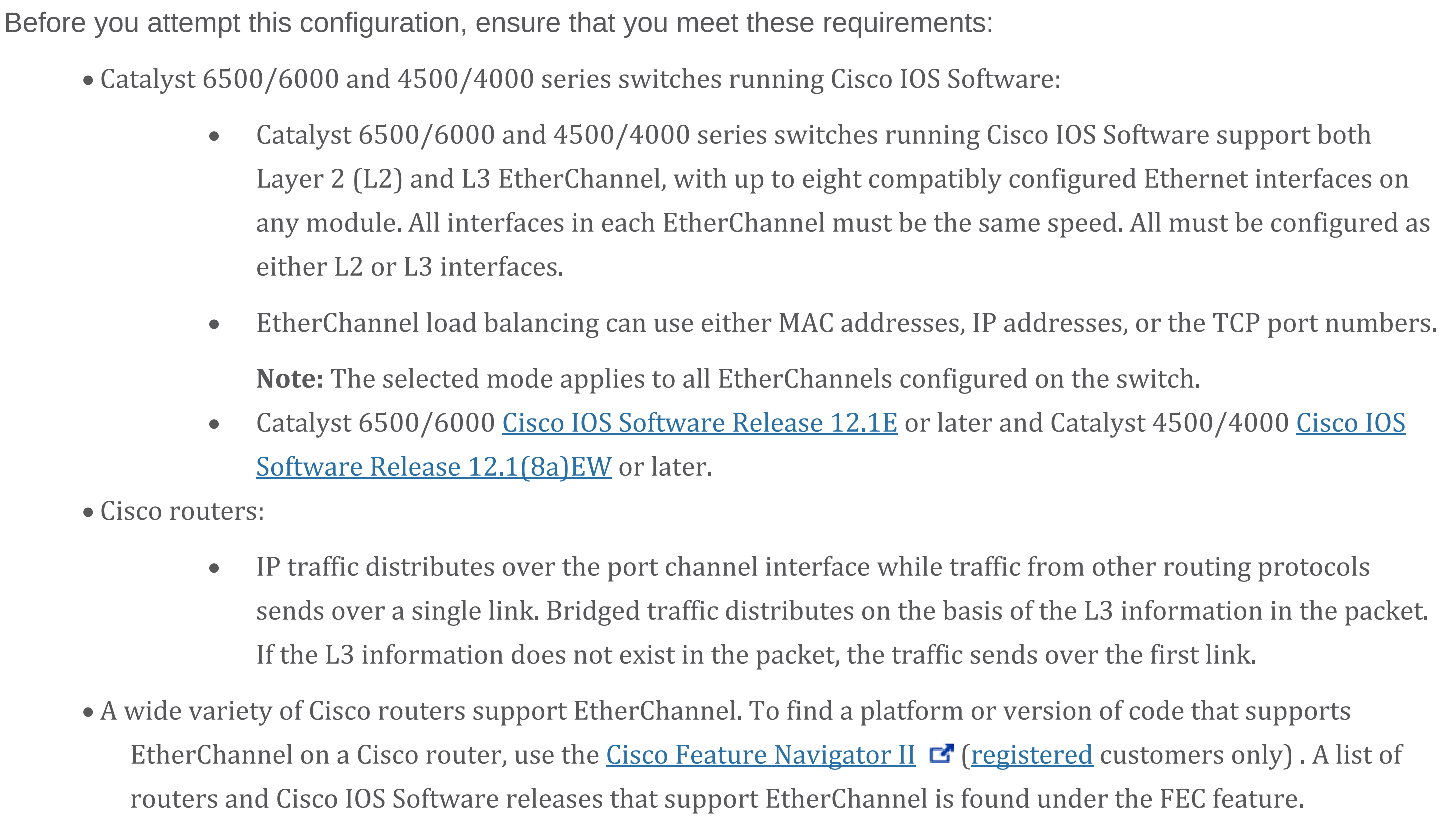
Beginning in privileged EXEC mode, follow these steps to configure EtherChannel load balancing. This procedure is optional.

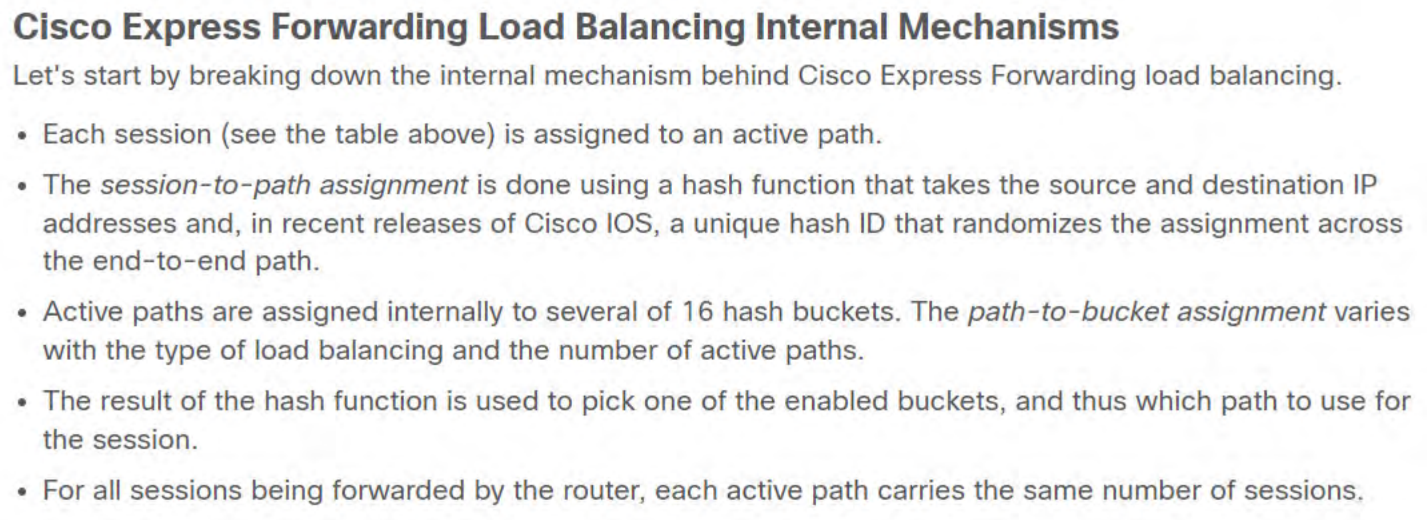
	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**port-channel load-balance** {**dst-ip** | **dst-mac** | **src-dst-ip** | **src-dst-mac** | **src-ip** | **src-mac**}	Configure an EtherChannel load-balancing method. The default is **src-mac**. Select one of these load-distribution methods: • **dst-ip**—Load distribution is based on the destination-host IP address. • **dst-mac**—Load distribution is based on the destination-host MAC address of the incoming packet. • **src-dst-ip**—Load distribution is based on the source-and-destination host-IP address. • **src-dst-mac**—Load distribution is based on the source-and-destination host-MAC address. • **src-ip**—Load distribution is based on the source-host IP address. • **src-mac**—Load distribution is based on the source-MAC address of the incoming packet.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show etherchannel load-balance**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

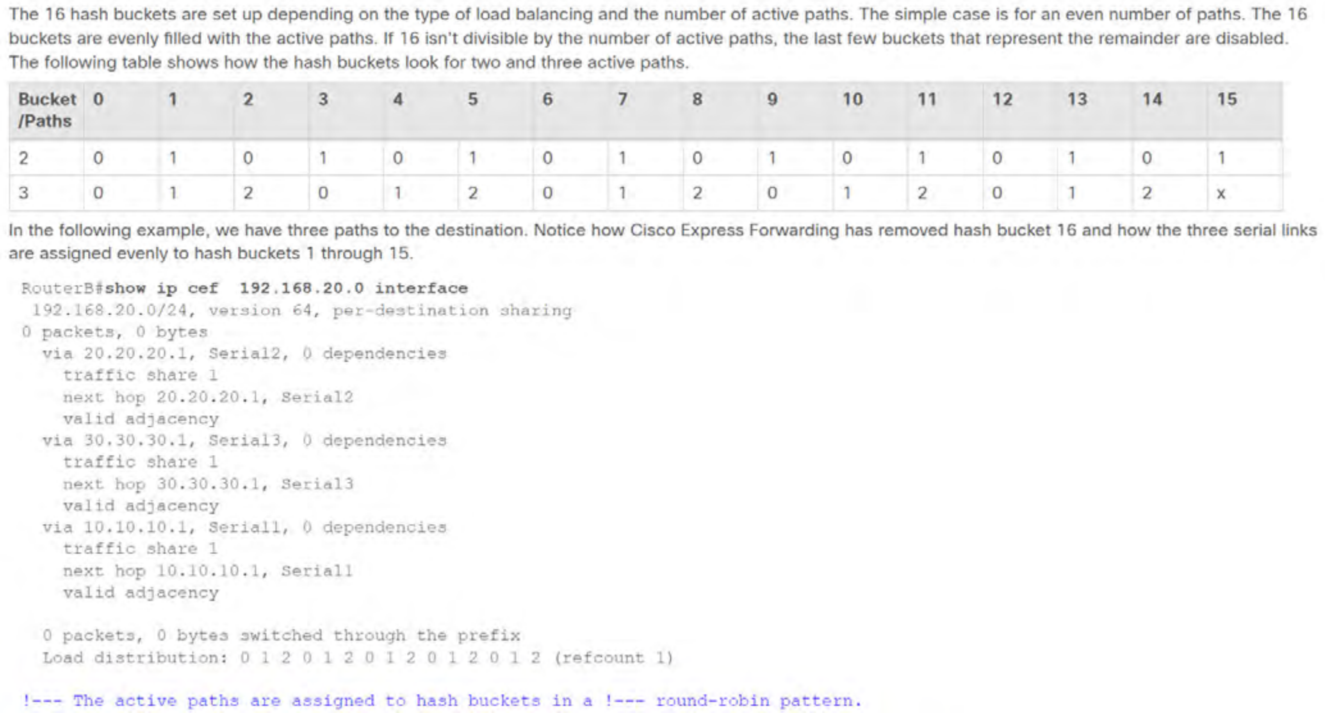
To return EtherChannel load balancing to the default configuration, use the **no port-channel load-balance** global configuration command.

No.	’740 Patent Claim 9	Cisco EtherChannel
		Layer 2 EtherChannel **Catalyst 2950** Catalyst 2950 switches only support 802.1Q trunking and do not support ISL trunking. Catalyst 2950 switches support DTP and PAgP dynamic trunking and channel negotiation with Cisco IOS Software Release 12.1 releases and static modes only with Cisco IOS Software Release 12.0 releases. EtherChannel load balancing can use either source-MAC or destination-MAC address forwarding. You can configure the load balancing method by issuing the **port-channel load-balance** global configuration command. These switches support up to eight switch ports per channel. **Catalyst 6500 That Runs Cisco IOS Software** Catalyst 6500 switches that run Cisco IOS Software support L2 (switchport) and Layer 3 (L3) (routed port) EtherChannel configurations. A Catalyst 6500/6000 series switch supports a maximum of 64 EtherChannels (256 with Cisco IOS Software Release 12.1(2)E and earlier). You can form an EtherChannel with up to eight compatibly configured LAN ports on any module in a Catalyst 6000 series switch, with the exception of Digital Feature Card (DFC)-equipped modules (such as WS-X6816 and so on) which currently allow an L2 channel only using ports on the same DFC module. However, an L3 channel can be configured across different DFC-equipped modules. This limitation has been removed in Catalyst 6500/6000 Cisco IOS Software Release 12.1(11b)EX and later. This document configures an L2 EtherChannel. The Catalyst 6500/6000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP addresses, or Layer 4 (L4) port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. Catalyst 6500/6000 switches support both ISL and 802.1Q trunking encapsulations and DTP. Detailed information on port capabilities is available by issuing the **show interface** *interface_id* **capabilities** command.

No.	’740 Patent Claim 9	Cisco EtherChannel
		Catalyst 4000 That Runs Cisco IOS Software Catalyst 4000 switches that run Cisco IOS Software (with Supervisor Engine III and IV) support L2 (switchport) and L3 (routed port) EtherChannel configurations. A Catalyst 4000 series switch supports a maximum of 64 EtherChannels. You can form an EtherChannel with up to eight compatibly configured Ethernet interfaces on any module, and across modules in a Catalyst 4000 series switch. All interfaces in each EtherChannel must be the same speed and must all be configured as either L2 or L3 interfaces. The Catalyst 4000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP address, or L4 port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. The Catalyst 4000 that runs Cisco IOS Software supports ISL and 802.1Q trunking encapsulations and DTP. ISL is not available on certain modules. For a complete list of such modules, refer to the *Understanding VLAN Trunks* section of Configuring Layer 2 Ethernet Interfaces. In a future software release, detailed information on port capabilities will be available by issuing the **show interface capabilities** command. Currently this command is not available. Configuring EtherChannel at 1 -2 **Introduction** This sample configuration demonstrates how to set up a Layer 3 (L3) EtherChannel, without VLAN trunking, between a Cisco router and a Cisco Catalyst 6500 switch running Cisco IOS® System Software. EtherChannel can be called Fast EtherChannel (FEC) or Gigabit EtherChannel (GEC); the term depends on the speed of the interfaces or ports you use to form the EtherChannel. In this example, two Fast Ethernet ports from a Cisco router and a Catalyst 6500 switch have been bundled into a FEC. Throughout this document, the terms FEC, GEC, port channel, channel, and port group all refer to EtherChannel.

No.	’740 Patent Claim 9	Cisco EtherChannel
		 Before you attempt this configuration, ensure that you meet these requirements: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software support both Layer 2 (L2) and L3 EtherChannel, with up to eight compatibly configured Ethernet interfaces on any module. All interfaces in each EtherChannel must be the same speed. All must be configured as either L2 or L3 interfaces. • EtherChannel load balancing can use either MAC addresses, IP addresses, or the TCP port numbers. **Note:** The selected mode applies to all EtherChannels configured on the switch. • Catalyst 6500/6000 Cisco IOS Software Release 12.1E or later and Catalyst 4500/4000 Cisco IOS Software Release 12.1(8a)EW or later. • Cisco routers: • IP traffic distributes over the port channel interface while traffic from other routing protocols sends over a single link. Bridged traffic distributes on the basis of the L3 information in the packet. If the L3 information does not exist in the packet, the traffic sends over the first link. • A wide variety of Cisco routers support EtherChannel. To find a platform or version of code that supports EtherChannel on a Cisco router, use the Cisco Feature Navigator II (registered customers only) . A list of routers and Cisco IOS Software releases that support EtherChannel is found under the FEC feature. Load Balancing at 7-9

No.	’740 Patent Claim 9	Cisco EtherChannel
		 **Cisco Express Forwarding Load Balancing Internal Mechanisms** Let's start by breaking down the internal mechanism behind Cisco Express Forwarding load balancing. • Each session (see the table above) is assigned to an active path. • The *session-to-path assignment* is done using a hash function that takes the source and destination IP addresses and, in recent releases of Cisco IOS, a unique hash ID that randomizes the assignment across the end-to-end path. • Active paths are assigned internally to several of 16 hash buckets. The *path-to-bucket assignment* varies with the type of load balancing and the number of active paths. • The result of the hash function is used to pick one of the enabled buckets, and thus which path to use for the session. • For all sessions being forwarded by the router, each active path carries the same number of sessions.

No.	’740 Patent Claim 9	Cisco EtherChannel
		

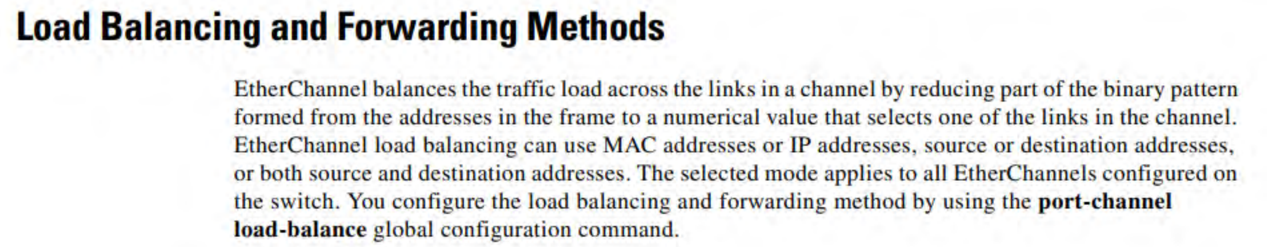
The 16 hash buckets are set up depending on the type of load balancing and the number of active paths. The simple case is for an even number of paths. The 16 buckets are evenly filled with the active paths. If 16 isn't divisible by the number of active paths, the last few buckets that represent the remainder are disabled. The following table shows how the hash buckets look for two and three active paths.

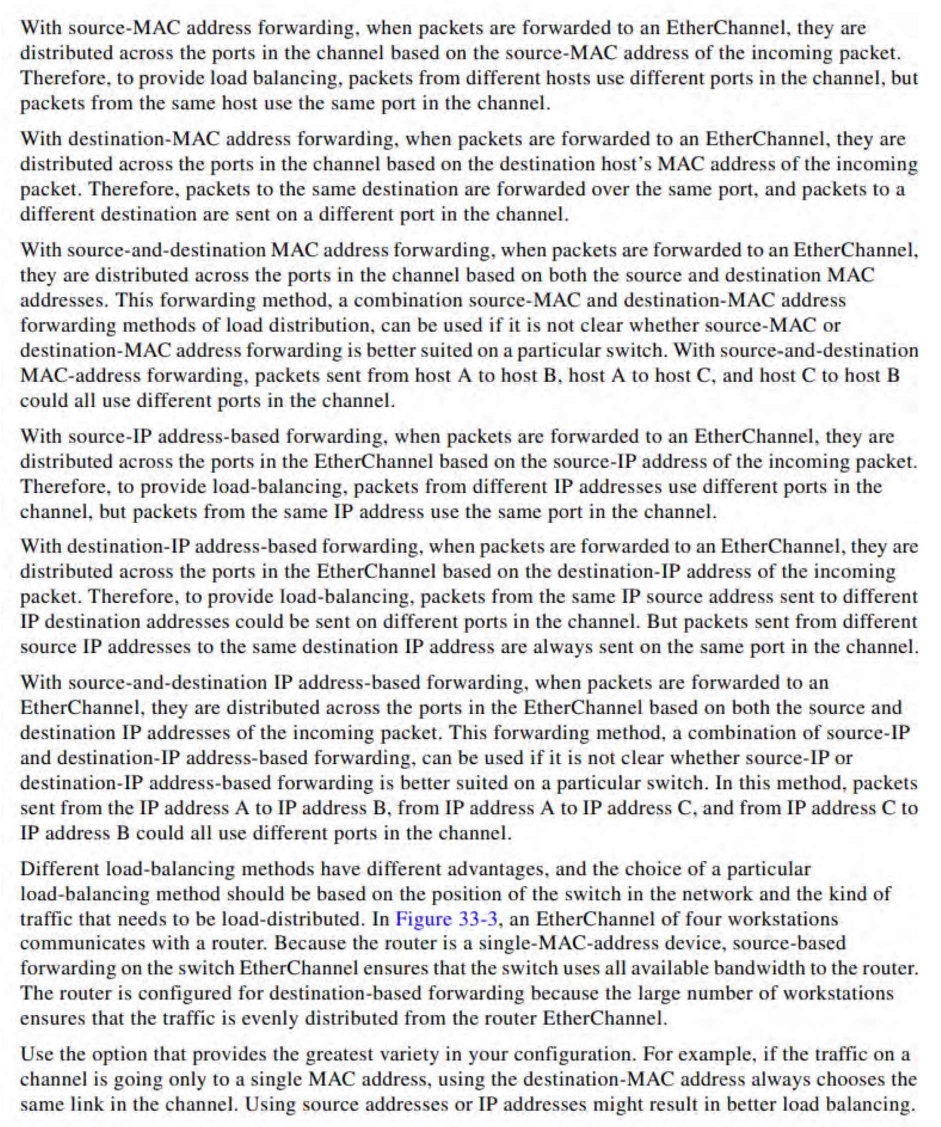
Bucket /Paths	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
3	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	x

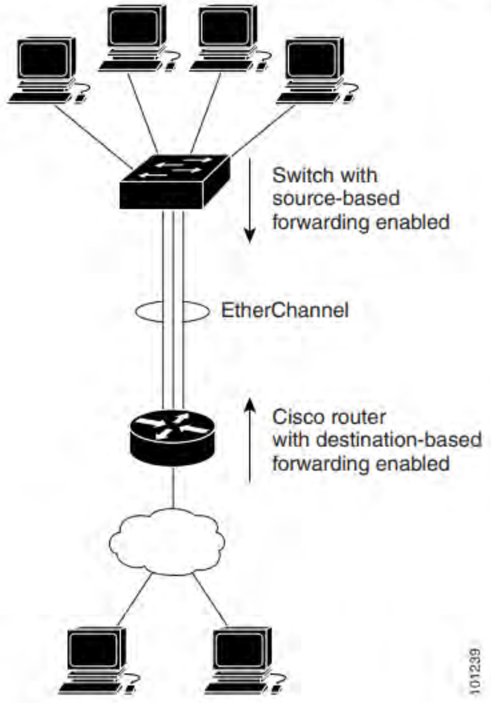
In the following example, we have three paths to the destination. Notice how Cisco Express Forwarding has removed hash bucket 16 and how the three serial links are assigned evenly to hash buckets 1 through 15.

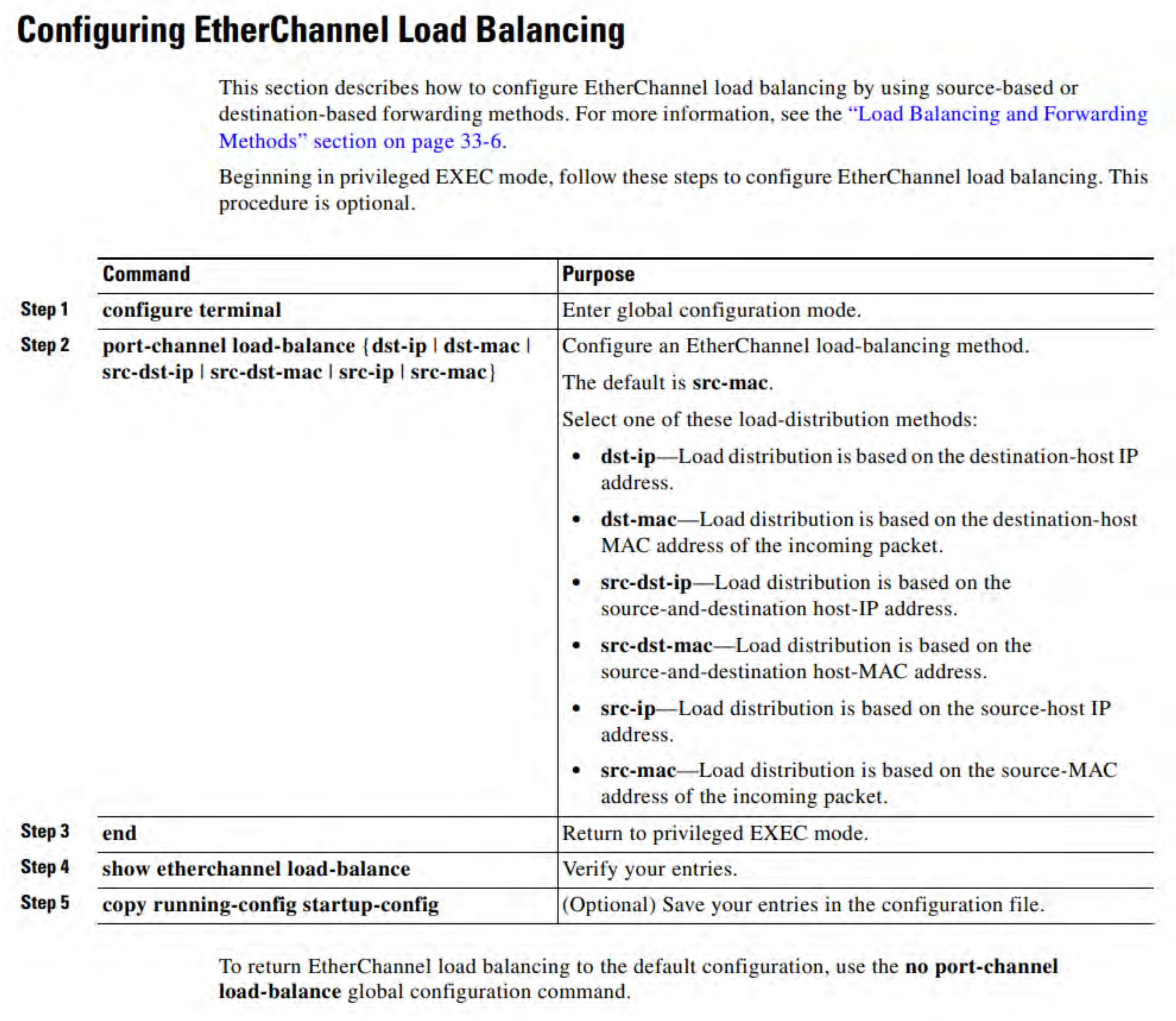
```
RouterB#show ip cef  192.168.20.0 interface
 192.168.20.0/24, version 64, per-destination sharing
0 packets, 0 bytes
  via 20.20.20.1, Serial2, 0 dependencies
    traffic share 1
    next hop 20.20.20.1, Serial2
    valid adjacency
  via 30.30.30.1, Serial3, 0 dependencies
    traffic share 1
    next hop 30.30.30.1, Serial3
    valid adjacency
  via 10.10.10.1, Serial1, 0 dependencies
    traffic share 1
    next hop 10.10.10.1, Serial1
    valid adjacency

  0 packets, 0 bytes switched through the prefix
  Load distribution: 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 (refcount 1)

 !--- The active paths are assigned to hash buckets in a !--- round-robin pattern.
```

No.	’740 Patent Claim 10	Cisco EtherChannel
10[a]	The method according to claim 9, wherein applying the hashing	Cisco EtherChannel System discloses the method according to claim 9, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second physical links. For example, Cisco EtherChannel System discloses a hash function in which the number of active paths is used to determined. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this

No.	’740 Patent Claim 10	Cisco EtherChannel
	function comprises determining a hashing size responsively to a number of at least some of the first and second physical links,	limitation is met. To the extent that the Cisco EtherChannel System is found to not meet this limitation, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second physical links would have been obvious to a person having ordinary skill in the art, as explained below. *See supra at* Claim 9. Catalyst 3560 Configuration Guide at 33-6 – 33-8  **Load Balancing and Forwarding Methods** EtherChannel balances the traffic load across the links in a channel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel. EtherChannel load balancing can use MAC addresses or IP addresses, source or destination addresses, or both source and destination addresses. The selected mode applies to all EtherChannels configured on the switch. You configure the load balancing and forwarding method by using the **port-channel load-balance** global configuration command.

No.	’740 Patent Claim 10	Cisco EtherChannel
		 With source-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the source-MAC address of the incoming packet. Therefore, to provide load balancing, packets from different hosts use different ports in the channel, but packets from the same host use the same port in the channel. With destination-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the destination host’s MAC address of the incoming packet. Therefore, packets to the same destination are forwarded over the same port, and packets to a different destination are sent on a different port in the channel. With source-and-destination MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on both the source and destination MAC addresses. This forwarding method, a combination source-MAC and destination-MAC address forwarding methods of load distribution, can be used if it is not clear whether source-MAC or destination-MAC address forwarding is better suited on a particular switch. With source-and-destination MAC-address forwarding, packets sent from host A to host B, host A to host C, and host C to host B could all use different ports in the channel. With source-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the source-IP address of the incoming packet. Therefore, to provide load-balancing, packets from different IP addresses use different ports in the channel, but packets from the same IP address use the same port in the channel. With destination-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the destination-IP address of the incoming packet. Therefore, to provide load-balancing, packets from the same IP source address sent to different IP destination addresses could be sent on different ports in the channel. But packets sent from different source IP addresses to the same destination IP address are always sent on the same port in the channel. With source-and-destination IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on both the source and destination IP addresses of the incoming packet. This forwarding method, a combination of source-IP and destination-IP address-based forwarding, can be used if it is not clear whether source-IP or destination-IP address-based forwarding is better suited on a particular switch. In this method, packets sent from the IP address A to IP address B, from IP address A to IP address C, and from IP address C to IP address B could all use different ports in the channel. Different load-balancing methods have different advantages, and the choice of a particular load-balancing method should be based on the position of the switch in the network and the kind of traffic that needs to be load-distributed. In Figure 33-3, an EtherChannel of four workstations communicates with a router. Because the router is a single-MAC-address device, source-based forwarding on the switch EtherChannel ensures that the switch uses all available bandwidth to the router. The router is configured for destination-based forwarding because the large number of workstations ensures that the traffic is evenly distributed from the router EtherChannel. Use the option that provides the greatest variety in your configuration. For example, if the traffic on a channel is going only to a single MAC address, using the destination-MAC address always chooses the same link in the channel. Using source addresses or IP addresses might result in better load balancing.

No.	’740 Patent Claim 10	Cisco EtherChannel
		Figure 33-3 Load Distribution and Forwarding Methods  Switch with source-based forwarding enabled EtherChannel Cisco router with destination-based forwarding enabled 101239 Catalyst 3560 Configuration Guide at 33-16

No.	’740 Patent Claim 10	Cisco EtherChannel
		

Configuring EtherChannel Load Balancing

This section describes how to configure EtherChannel load balancing by using source-based or destination-based forwarding methods. For more information, see the “Load Balancing and Forwarding Methods” section on page 33-6.

Beginning in privileged EXEC mode, follow these steps to configure EtherChannel load balancing. This procedure is optional.

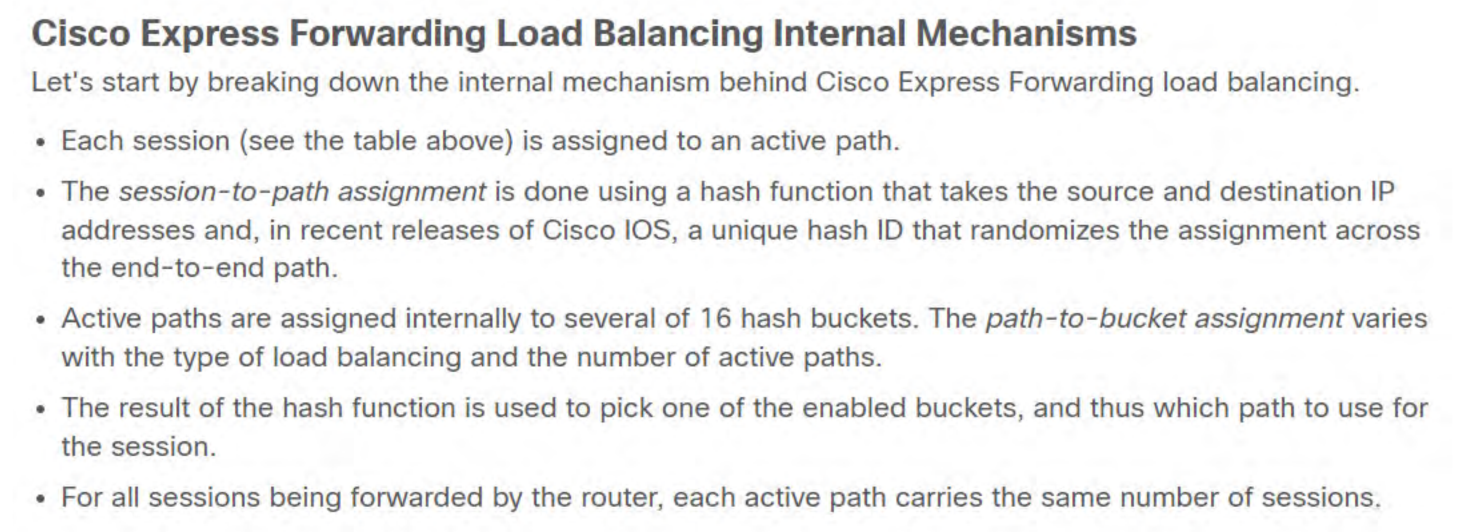
	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**port-channel load-balance** {**dst-ip** | **dst-mac** | **src-dst-ip** | **src-dst-mac** | **src-ip** | **src-mac**}	Configure an EtherChannel load-balancing method. The default is **src-mac**. Select one of these load-distribution methods: • **dst-ip**—Load distribution is based on the destination-host IP address. • **dst-mac**—Load distribution is based on the destination-host MAC address of the incoming packet. • **src-dst-ip**—Load distribution is based on the source-and-destination host-IP address. • **src-dst-mac**—Load distribution is based on the source-and-destination host-MAC address. • **src-ip**—Load distribution is based on the source-host IP address. • **src-mac**—Load distribution is based on the source-MAC address of the incoming packet.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show etherchannel load-balance**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return EtherChannel load balancing to the default configuration, use the **no port-channel load-balance** global configuration command.

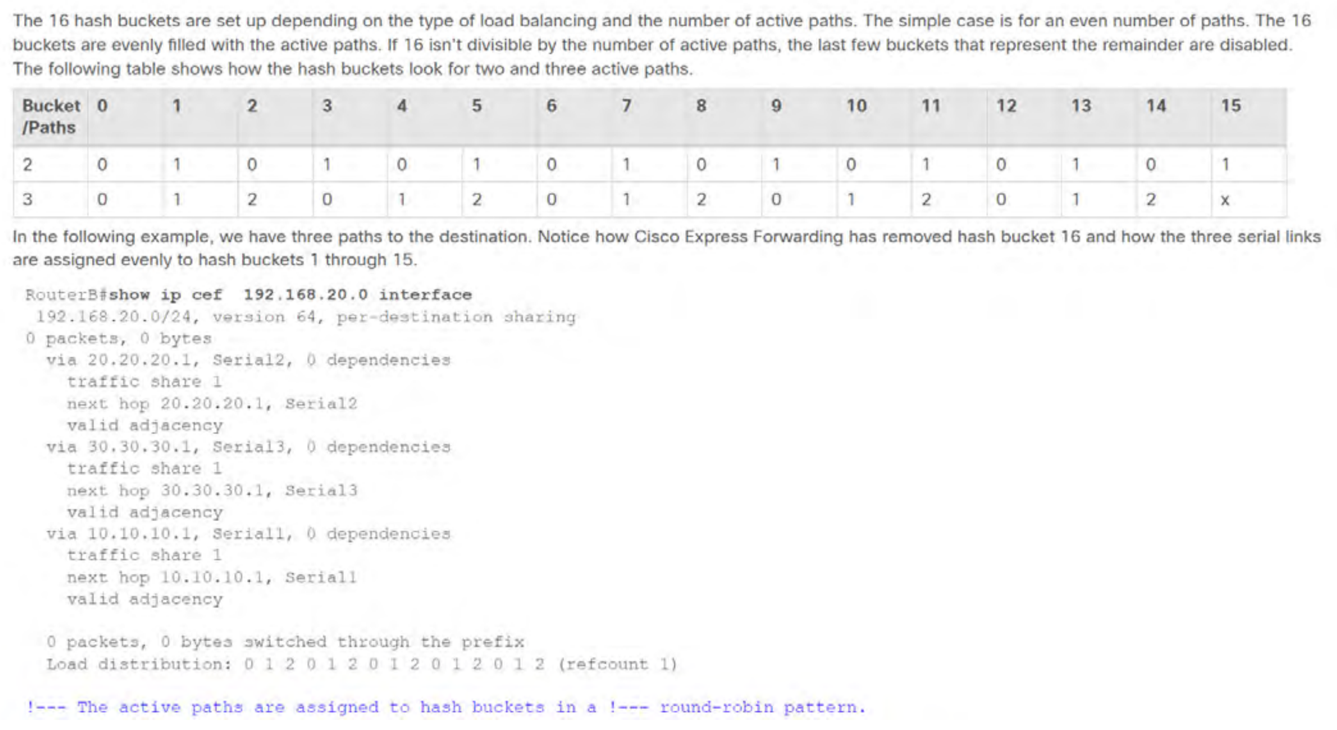
No.	’740 Patent Claim 10	Cisco EtherChannel
		Layer 2 EtherChannel **Catalyst 2950** Catalyst 2950 switches only support 802.1Q trunking and do not support ISL trunking. Catalyst 2950 switches support DTP and PAgP dynamic trunking and channel negotiation with Cisco IOS Software Release 12.1 releases and static modes only with Cisco IOS Software Release 12.0 releases. EtherChannel load balancing can use either source-MAC or destination-MAC address forwarding. You can configure the load balancing method by issuing the **port-channel load-balance** global configuration command. These switches support up to eight switch ports per channel. **Catalyst 6500 That Runs Cisco IOS Software** Catalyst 6500 switches that run Cisco IOS Software support L2 (switchport) and Layer 3 (L3) (routed port) EtherChannel configurations. A Catalyst 6500/6000 series switch supports a maximum of 64 EtherChannels (256 with Cisco IOS Software Release 12.1(2)E and earlier). You can form an EtherChannel with up to eight compatibly configured LAN ports on any module in a Catalyst 6000 series switch, with the exception of Digital Feature Card (DFC)-equipped modules (such as WS-X6816 and so on) which currently allow an L2 channel only using ports on the same DFC module. However, an L3 channel can be configured across different DFC-equipped modules. This limitation has been removed in Catalyst 6500/6000 Cisco IOS Software Release 12.1(11b)EX and later. This document configures an L2 EtherChannel. The Catalyst 6500/6000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP addresses, or Layer 4 (L4) port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. Catalyst 6500/6000 switches support both ISL and 802.1Q trunking encapsulations and DTP. Detailed information on port capabilities is available by issuing the **show interface** *interface_id* **capabilities** command.

No.	’740 Patent Claim 10	Cisco EtherChannel
		Catalyst 4000 That Runs Cisco IOS Software Catalyst 4000 switches that run Cisco IOS Software (with Supervisor Engine III and IV) support L2 (switchport) and L3 (routed port) EtherChannel configurations. A Catalyst 4000 series switch supports a maximum of 64 EtherChannels. You can form an EtherChannel with up to eight compatibly configured Ethernet interfaces on any module, and across modules in a Catalyst 4000 series switch. All interfaces in each EtherChannel must be the same speed and must all be configured as either L2 or L3 interfaces. The Catalyst 4000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP address, or L4 port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. The Catalyst 4000 that runs Cisco IOS Software supports ISL and 802.1Q trunking encapsulations and DTP. ISL is not available on certain modules. For a complete list of such modules, refer to the *Understanding VLAN Trunks* section of Configuring Layer 2 Ethernet Interfaces. In a future software release, detailed information on port capabilities will be available by issuing the **show interface capabilities** command. Currently this command is not available. Configuring EtherChannel at 1 -2 **Introduction** This sample configuration demonstrates how to set up a Layer 3 (L3) EtherChannel, without VLAN trunking, between a Cisco router and a Cisco Catalyst 6500 switch running Cisco IOS® System Software. EtherChannel can be called Fast EtherChannel (FEC) or Gigabit EtherChannel (GEC); the term depends on the speed of the interfaces or ports you use to form the EtherChannel. In this example, two Fast Ethernet ports from a Cisco router and a Catalyst 6500 switch have been bundled into a FEC. Throughout this document, the terms FEC, GEC, port channel, channel, and port group all refer to EtherChannel.

No.	’740 Patent Claim 10	Cisco EtherChannel
		Before you attempt this configuration, ensure that you meet these requirements: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software support both Layer 2 (L2) and L3 EtherChannel, with up to eight compatibly configured Ethernet interfaces on any module. All interfaces in each EtherChannel must be the same speed. All must be configured as either L2 or L3 interfaces. • EtherChannel load balancing can use either MAC addresses, IP addresses, or the TCP port numbers. **Note:** The selected mode applies to all EtherChannels configured on the switch. • Catalyst 6500/6000 Cisco IOS Software Release 12.1E or later and Catalyst 4500/4000 Cisco IOS Software Release 12.1(8a)EW or later. • Cisco routers: • IP traffic distributes over the port channel interface while traffic from other routing protocols sends over a single link. Bridged traffic distributes on the basis of the L3 information in the packet. If the L3 information does not exist in the packet, the traffic sends over the first link. • A wide variety of Cisco routers support EtherChannel. To find a platform or version of code that supports EtherChannel on a Cisco router, use the Cisco Feature Navigator II (registered customers only) . A list of routers and Cisco IOS Software releases that support EtherChannel is found under the FEC feature. Load Balancing at 7-9

No.	’740 Patent Claim 10	Cisco EtherChannel
		 **Cisco Express Forwarding Load Balancing Internal Mechanisms** Let's start by breaking down the internal mechanism behind Cisco Express Forwarding load balancing. • Each session (see the table above) is assigned to an active path. • The *session-to-path assignment* is done using a hash function that takes the source and destination IP addresses and, in recent releases of Cisco IOS, a unique hash ID that randomizes the assignment across the end-to-end path. • Active paths are assigned internally to several of 16 hash buckets. The *path-to-bucket assignment* varies with the type of load balancing and the number of active paths. • The result of the hash function is used to pick one of the enabled buckets, and thus which path to use for the session. • For all sessions being forwarded by the router, each active path carries the same number of sessions.

No.	’740 Patent Claim 10	Cisco EtherChannel



The 16 hash buckets are set up depending on the type of load balancing and the number of active paths. The simple case is for an even number of paths. The 16 buckets are evenly filled with the active paths. If 16 isn't divisible by the number of active paths, the last few buckets that represent the remainder are disabled. The following table shows how the hash buckets look for two and three active paths.

Bucket /Paths	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
3	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	x

In the following example, we have three paths to the destination. Notice how Cisco Express Forwarding has removed hash bucket 16 and how the three serial links are assigned evenly to hash buckets 1 through 15.

```
RouterB#show ip cef  192.168.20.0 interface
 192.168.20.0/24, version 64, per-destination sharing
0 packets, 0 bytes
  via 20.20.20.1, Serial2, 0 dependencies
    traffic share 1
    next hop 20.20.20.1, Serial2
    valid adjacency
  via 30.30.30.1, Serial3, 0 dependencies
    traffic share 1
    next hop 30.30.30.1, Serial3
    valid adjacency
  via 10.10.10.1, Serial1, 0 dependencies
    traffic share 1
    next hop 10.10.10.1, Serial1
    valid adjacency

  0 packets, 0 bytes switched through the prefix
  Load distribution: 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 (refcount 1)

!--- The active paths are assigned to hash buckets in a !--- round-robin pattern.
```

Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Cisco EtherChannel System in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[a] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references.

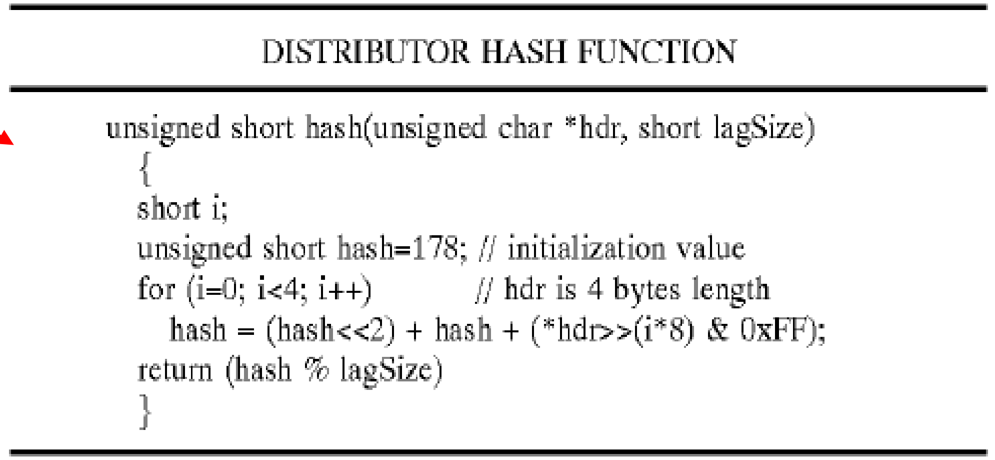
For example, Bruckman discloses applying a distributor hash function to the frame information which includes determining a number of the plurality of physical links.

Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some

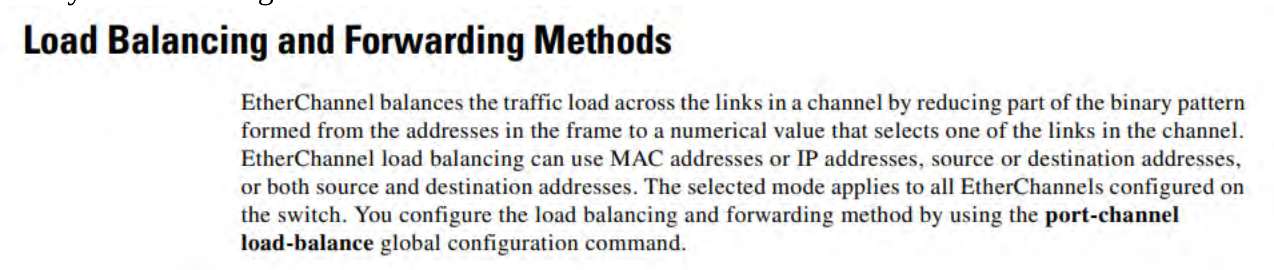
No.	’740 Patent Claim 10	Cisco EtherChannel
		measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links.

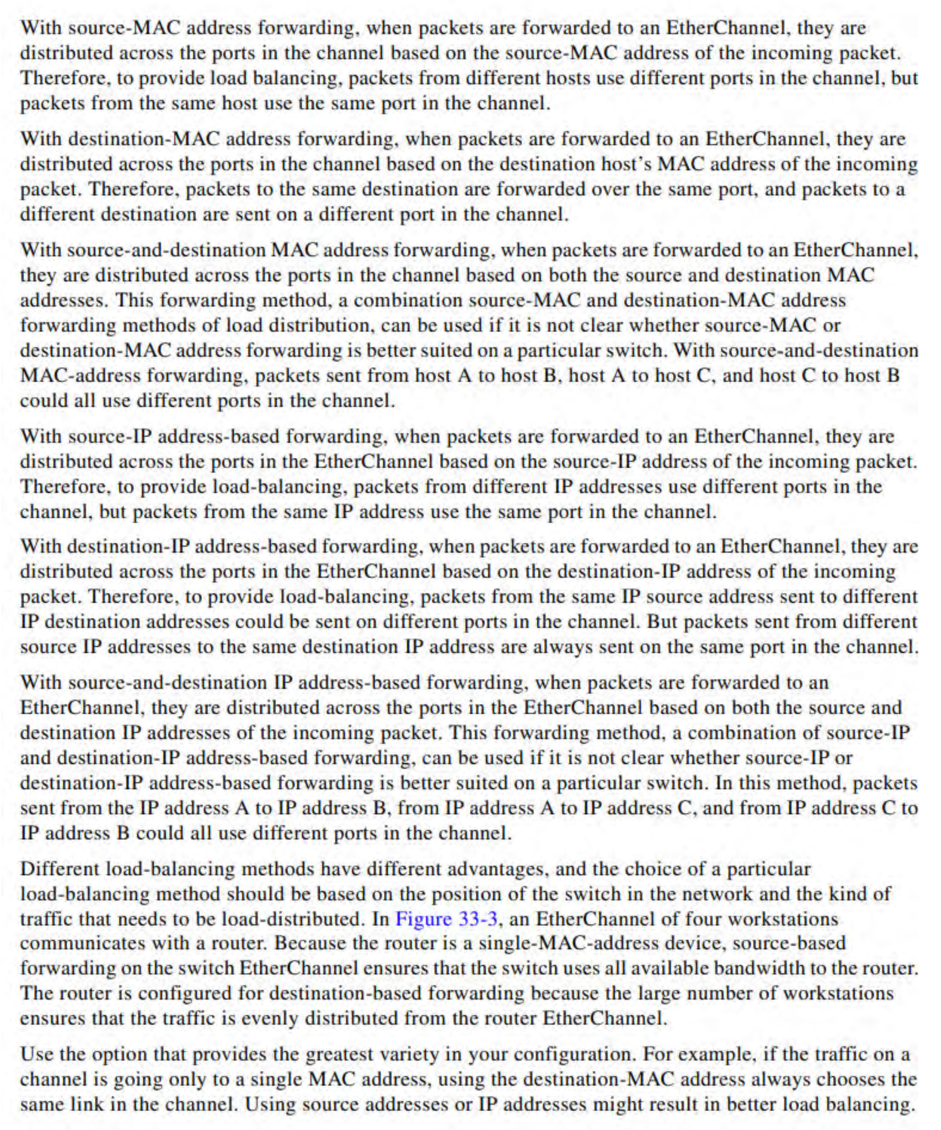
No.	’740 Patent Claim 10	Cisco EtherChannel
		Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”) Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I: Bruckman at Table 1 (annotated)

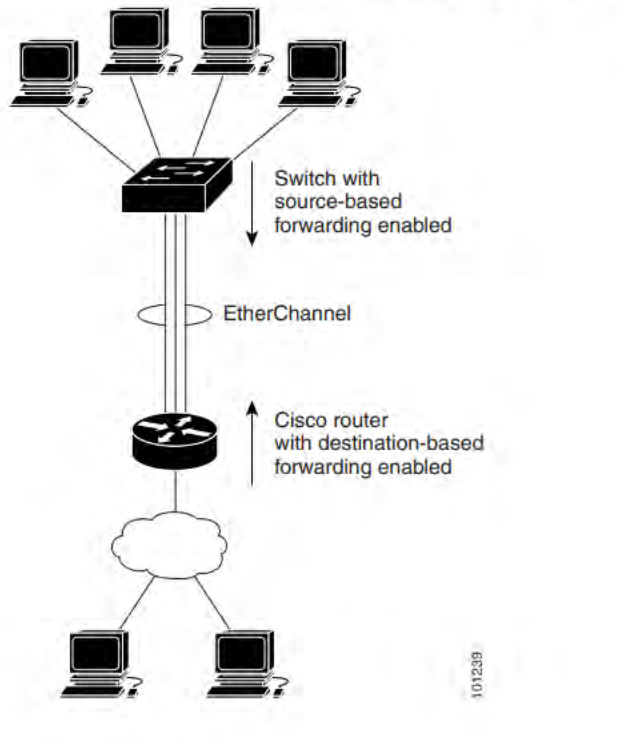
No.	’740 Patent Claim 10	Cisco EtherChannel

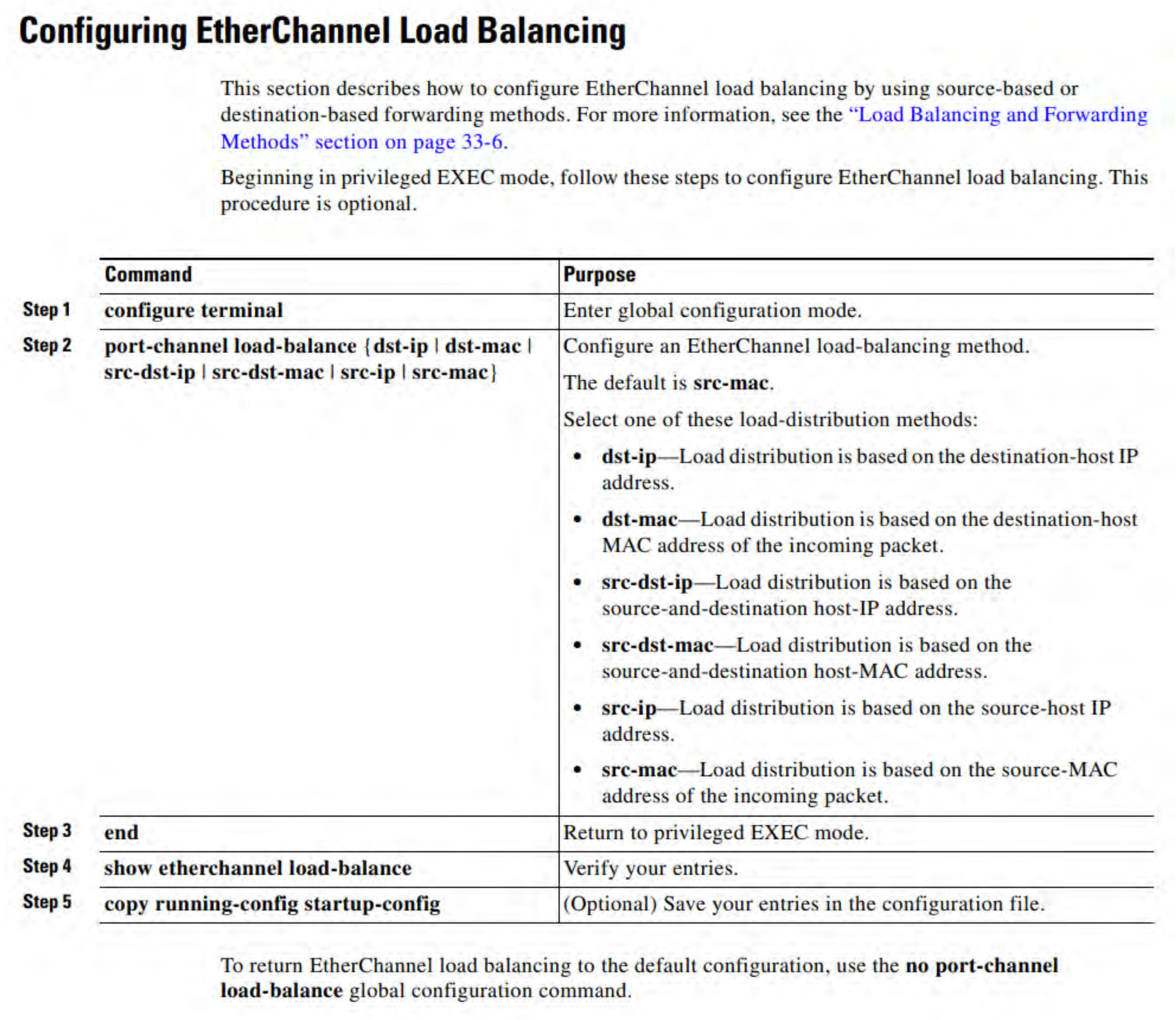
		hashing function *“mapping function”*
		
		DISTRIBUTOR HASH FUNCTION
		`unsigned short hash(unsigned char *hdr, short lagSize)`
		`{`
		`short i;`
		`unsigned short hash=178; // initialization value`
		`for (i=0; i<4; i++)          // hdr is 4 bytes length`
		`    hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);`
		`return (hash % lagSize)`
		`}`
		
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		
		For example, Solomon discloses applying a distributor hash function to the frame information which includes determining a number of the plurality of physical links.
		
		Solomon at [0024] (“In another embodiment, switching the data packets includes mapping the data packets to the selected port responsively to the label. Additionally or alternatively, map-ping the data packets includes applying a hashing function to the label so as to determine a number of the selected port, and choosing the label includes applying an inverse of the hashing function to the number of the selected port.”)
		
		Solomon at [0048] (“The mapping function typically uses MPLS label 52 for mapping, since the MPLS label uniquely identifies MPLS tunnel 28, and it is required that all MPLS packets belonging to the same tunnel be switched through the same physical port 24. Additionally or alternatively, the mapping function uses a "PW"

No.	’740 Patent Claim 10	Cisco EtherChannel
		label (pseudo wire label, formerly known as a virtual connection, or VC label), which is optionally added to MPLS header 50. The PW label com-prises information that the egress node requires for deliver-ing the packet to its destination, and is optionally added during the encapsulation of MPLS packets. Additional details regarding the VC label can be found in an IETF draft by Martini et al. entitled "Encapsulation Methods for Trans-port of Ethernet Frames Over IP/MPLS Networks" (IETF draft-ietf-pwe3-ethernet-encap-07.txt, May, 2004), which is incorporated herein by reference. In some embodiments, mapper 34 applies a hashing function to the MPLS and/or PW label, as will be described below.”) Solomon at [0059] (“In this method, the mapping function used by mapper 34 of switch A is a hashing function. Various hashing functions are known in the art, and any suitable hashing function may be used in mapper 34. Since the hashing operation is performed for each packet, it is desirable to have a hashing function that is computationally simple.”) Solomon at [0060] (“As mentioned above, the hashing function typically hashes the value of MPLS label 52 to determine the selected physical port, as the MPLS label uniquely identifies tunnel 28. For example, the following hashing function may be used by mapper 34: Selected port number=l+((MPLS label) mod N), wherein N denotes the number of physical Ethernet ports in LAG group 25, and "mod" denotes the modulus operator. Assuming the values of MPLS labels are distrib-uted uniformly over a certain range, this function achieves a uniform distribution of port allocations for the different MPLS labels. It can also be seen that all packets carrying the same MPLS label (in other words-belonging to the same MPLS tunnel) will be mapped to the same physical port.”) Solomon at [0065] (“Mapper 34 of switch A maps each received packet to the selected physical port of LAG group 25 using the hashing function, at a hashing step 90. Mapper 34 extracts the MPLS label from each received packet and uses the hashing function to calculate the serial number of the selected physical port, which was selected by the CAC processor at step 82. Following the numerical example given above, the mapper extracts MPLS label=65647 from the packet. Substituting this value and N=3 into the hashing function gives: Selected port number=1+(65647 mod 3)=2, which is indeed the port number selected in the example above.”)
10[b]	applying the hashing	Cisco EtherChannel System discloses applying the hashing function to the at least one of the frame attributes to produce a hashing key.

No.	’740 Patent Claim 10	Cisco EtherChannel
	function to the at least one of the frame attributes to produce a hashing key,	For example, Cisco EtherChannel System discloses applying a hash function to the packet information to reduce the information to a numerical value that selects the forwarding path. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Cisco EtherChannel System is found to not meet this limitation, applying the hashing function to the at least one of the frame attributes to produce a hashing key would have been obvious to a person having ordinary skill in the art, as explained below. Catalyst 3560 Configuration Guide at 33-6 – 33-8  **Load Balancing and Forwarding Methods** EtherChannel balances the traffic load across the links in a channel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel. EtherChannel load balancing can use MAC addresses or IP addresses, source or destination addresses, or both source and destination addresses. The selected mode applies to all EtherChannels configured on the switch. You configure the load balancing and forwarding method by using the **port-channel load-balance** global configuration command.

No.	’740 Patent Claim 10	Cisco EtherChannel
		 With source-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the source-MAC address of the incoming packet. Therefore, to provide load balancing, packets from different hosts use different ports in the channel, but packets from the same host use the same port in the channel. With destination-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the destination host’s MAC address of the incoming packet. Therefore, packets to the same destination are forwarded over the same port, and packets to a different destination are sent on a different port in the channel. With source-and-destination MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on both the source and destination MAC addresses. This forwarding method, a combination source-MAC and destination-MAC address forwarding methods of load distribution, can be used if it is not clear whether source-MAC or destination-MAC address forwarding is better suited on a particular switch. With source-and-destination MAC-address forwarding, packets sent from host A to host B, host A to host C, and host C to host B could all use different ports in the channel. With source-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the source-IP address of the incoming packet. Therefore, to provide load-balancing, packets from different IP addresses use different ports in the channel, but packets from the same IP address use the same port in the channel. With destination-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the destination-IP address of the incoming packet. Therefore, to provide load-balancing, packets from the same IP source address sent to different IP destination addresses could be sent on different ports in the channel. But packets sent from different source IP addresses to the same destination IP address are always sent on the same port in the channel. With source-and-destination IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on both the source and destination IP addresses of the incoming packet. This forwarding method, a combination of source-IP and destination-IP address-based forwarding, can be used if it is not clear whether source-IP or destination-IP address-based forwarding is better suited on a particular switch. In this method, packets sent from the IP address A to IP address B, from IP address A to IP address C, and from IP address C to IP address B could all use different ports in the channel. Different load-balancing methods have different advantages, and the choice of a particular load-balancing method should be based on the position of the switch in the network and the kind of traffic that needs to be load-distributed. In Figure 33-3, an EtherChannel of four workstations communicates with a router. Because the router is a single-MAC-address device, source-based forwarding on the switch EtherChannel ensures that the switch uses all available bandwidth to the router. The router is configured for destination-based forwarding because the large number of workstations ensures that the traffic is evenly distributed from the router EtherChannel. Use the option that provides the greatest variety in your configuration. For example, if the traffic on a channel is going only to a single MAC address, using the destination-MAC address always chooses the same link in the channel. Using source addresses or IP addresses might result in better load balancing.

No.	’740 Patent Claim 10	Cisco EtherChannel
		Figure 33-3 Load Distribution and Forwarding Methods  Switch with source-based forwarding enabled EtherChannel Cisco router with destination-based forwarding enabled 101239 Catalyst 3560 Configuration Guide at 33-16

No.	’740 Patent Claim 10	Cisco EtherChannel
		

Configuring EtherChannel Load Balancing

This section describes how to configure EtherChannel load balancing by using source-based or destination-based forwarding methods. For more information, see the “Load Balancing and Forwarding Methods” section on page 33-6.

Beginning in privileged EXEC mode, follow these steps to configure EtherChannel load balancing. This procedure is optional.

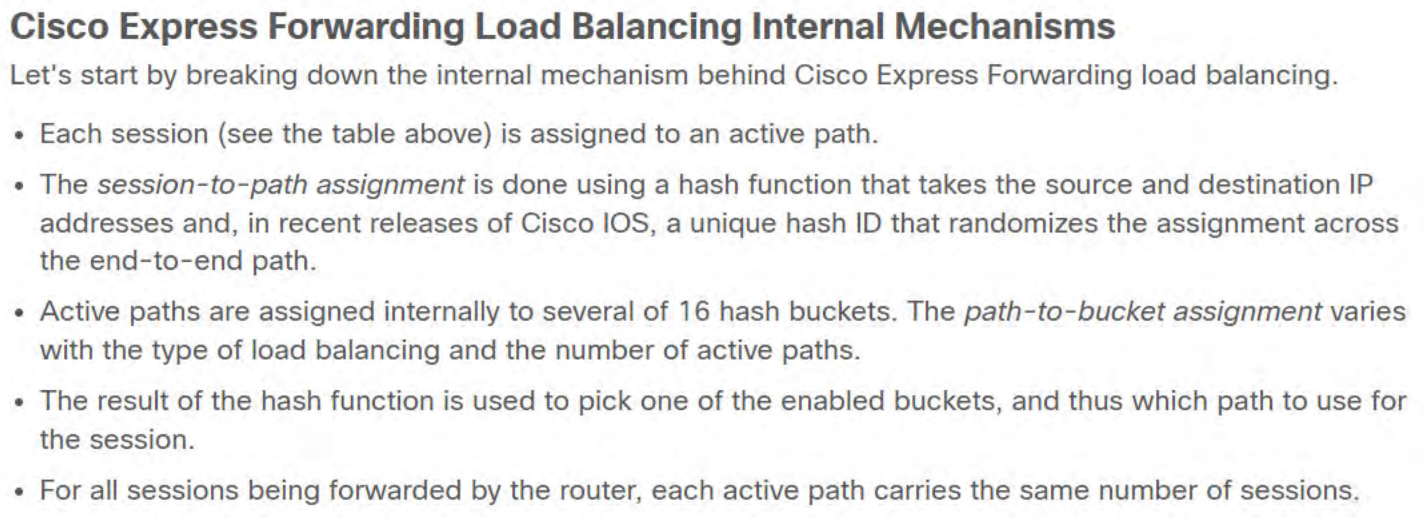
	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**port-channel load-balance** {**dst-ip** | **dst-mac** | **src-dst-ip** | **src-dst-mac** | **src-ip** | **src-mac**}	Configure an EtherChannel load-balancing method. The default is **src-mac**. Select one of these load-distribution methods: • **dst-ip**—Load distribution is based on the destination-host IP address. • **dst-mac**—Load distribution is based on the destination-host MAC address of the incoming packet. • **src-dst-ip**—Load distribution is based on the source-and-destination host-IP address. • **src-dst-mac**—Load distribution is based on the source-and-destination host-MAC address. • **src-ip**—Load distribution is based on the source-host IP address. • **src-mac**—Load distribution is based on the source-MAC address of the incoming packet.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show etherchannel load-balance**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return EtherChannel load balancing to the default configuration, use the **no port-channel load-balance** global configuration command.

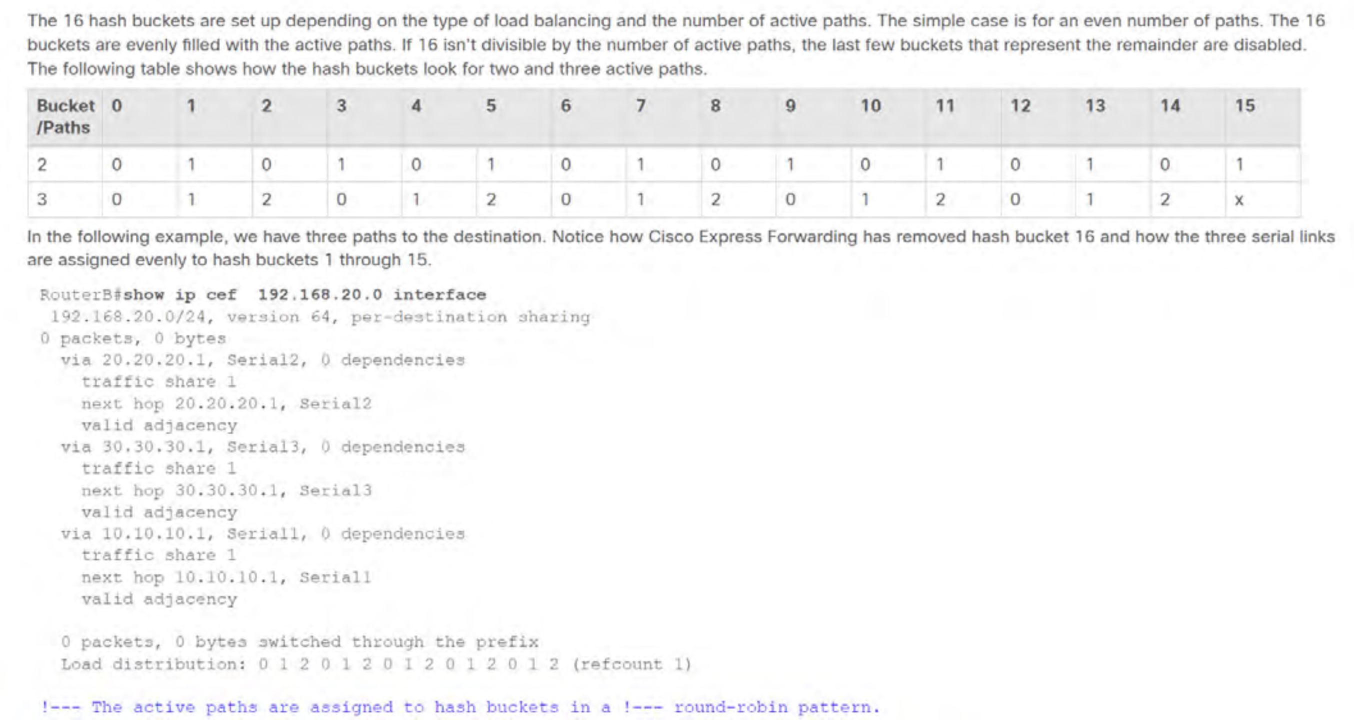
No.	’740 Patent Claim 10	Cisco EtherChannel
		Layer 2 EtherChannel **Catalyst 2950** Catalyst 2950 switches only support 802.1Q trunking and do not support ISL trunking. Catalyst 2950 switches support DTP and PAgP dynamic trunking and channel negotiation with Cisco IOS Software Release 12.1 releases and static modes only with Cisco IOS Software Release 12.0 releases. EtherChannel load balancing can use either source-MAC or destination-MAC address forwarding. You can configure the load balancing method by issuing the **port-channel load-balance** global configuration command. These switches support up to eight switch ports per channel. **Catalyst 6500 That Runs Cisco IOS Software** Catalyst 6500 switches that run Cisco IOS Software support L2 (switchport) and Layer 3 (L3) (routed port) EtherChannel configurations. A Catalyst 6500/6000 series switch supports a maximum of 64 EtherChannels (256 with Cisco IOS Software Release 12.1(2)E and earlier). You can form an EtherChannel with up to eight compatibly configured LAN ports on any module in a Catalyst 6000 series switch, with the exception of Digital Feature Card (DFC)-equipped modules (such as WS-X6816 and so on) which currently allow an L2 channel only using ports on the same DFC module. However, an L3 channel can be configured across different DFC-equipped modules. This limitation has been removed in Catalyst 6500/6000 Cisco IOS Software Release 12.1(11b)EX and later. This document configures an L2 EtherChannel. The Catalyst 6500/6000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP addresses, or Layer 4 (L4) port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. Catalyst 6500/6000 switches support both ISL and 802.1Q trunking encapsulations and DTP. Detailed information on port capabilities is available by issuing the **show interface** *interface_id* **capabilities** command.

No.	’740 Patent Claim 10	Cisco EtherChannel
		Catalyst 4000 That Runs Cisco IOS Software Catalyst 4000 switches that run Cisco IOS Software (with Supervisor Engine III and IV) support L2 (switchport) and L3 (routed port) EtherChannel configurations. A Catalyst 4000 series switch supports a maximum of 64 EtherChannels. You can form an EtherChannel with up to eight compatibly configured Ethernet interfaces on any module, and across modules in a Catalyst 4000 series switch. All interfaces in each EtherChannel must be the same speed and must all be configured as either L2 or L3 interfaces. The Catalyst 4000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP address, or L4 port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. The Catalyst 4000 that runs Cisco IOS Software supports ISL and 802.1Q trunking encapsulations and DTP. ISL is not available on certain modules. For a complete list of such modules, refer to the *Understanding VLAN Trunks* section of Configuring Layer 2 Ethernet Interfaces. In a future software release, detailed information on port capabilities will be available by issuing the **show interface capabilities** command. Currently this command is not available. Configuring EtherChannel at 1 -2 **Introduction** This sample configuration demonstrates how to set up a Layer 3 (L3) EtherChannel, without VLAN trunking, between a Cisco router and a Cisco Catalyst 6500 switch running Cisco IOS® System Software. EtherChannel can be called Fast EtherChannel (FEC) or Gigabit EtherChannel (GEC); the term depends on the speed of the interfaces or ports you use to form the EtherChannel. In this example, two Fast Ethernet ports from a Cisco router and a Catalyst 6500 switch have been bundled into a FEC. Throughout this document, the terms FEC, GEC, port channel, channel, and port group all refer to EtherChannel.

No.	’740 Patent Claim 10	Cisco EtherChannel
		Before you attempt this configuration, ensure that you meet these requirements: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software support both Layer 2 (L2) and L3 EtherChannel, with up to eight compatibly configured Ethernet interfaces on any module. All interfaces in each EtherChannel must be the same speed. All must be configured as either L2 or L3 interfaces. • EtherChannel load balancing can use either MAC addresses, IP addresses, or the TCP port numbers. **Note:** The selected mode applies to all EtherChannels configured on the switch. • Catalyst 6500/6000 Cisco IOS Software Release 12.1E or later and Catalyst 4500/4000 Cisco IOS Software Release 12.1(8a)EW or later. • Cisco routers: • IP traffic distributes over the port channel interface while traffic from other routing protocols sends over a single link. Bridged traffic distributes on the basis of the L3 information in the packet. If the L3 information does not exist in the packet, the traffic sends over the first link. • A wide variety of Cisco routers support EtherChannel. To find a platform or version of code that supports EtherChannel on a Cisco router, use the Cisco Feature Navigator II ⧉ (registered customers only) . A list of routers and Cisco IOS Software releases that support EtherChannel is found under the FEC feature. Load Balancing at 7-9

No.	’740 Patent Claim 10	Cisco EtherChannel
		 **Cisco Express Forwarding Load Balancing Internal Mechanisms** Let's start by breaking down the internal mechanism behind Cisco Express Forwarding load balancing. • Each session (see the table above) is assigned to an active path. • The *session-to-path assignment* is done using a hash function that takes the source and destination IP addresses and, in recent releases of Cisco IOS, a unique hash ID that randomizes the assignment across the end-to-end path. • Active paths are assigned internally to several of 16 hash buckets. The *path-to-bucket assignment* varies with the type of load balancing and the number of active paths. • The result of the hash function is used to pick one of the enabled buckets, and thus which path to use for the session. • For all sessions being forwarded by the router, each active path carries the same number of sessions.

No.	’740 Patent Claim 10	Cisco EtherChannel



The 16 hash buckets are set up depending on the type of load balancing and the number of active paths. The simple case is for an even number of paths. The 16 buckets are evenly filled with the active paths. If 16 isn't divisible by the number of active paths, the last few buckets that represent the remainder are disabled. The following table shows how the hash buckets look for two and three active paths.

Bucket /Paths	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
3	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	x

In the following example, we have three paths to the destination. Notice how Cisco Express Forwarding has removed hash bucket 16 and how the three serial links are assigned evenly to hash buckets 1 through 15.

```
RouterB#show ip cef  192.168.20.0 interface
 192.168.20.0/24, version 64, per-destination sharing
0 packets, 0 bytes
  via 20.20.20.1, Serial2, 0 dependencies
    traffic share 1
    next hop 20.20.20.1, Serial2
    valid adjacency
  via 30.30.30.1, Serial3, 0 dependencies
    traffic share 1
    next hop 30.30.30.1, Serial3
    valid adjacency
  via 10.10.10.1, Serial1, 0 dependencies
    traffic share 1
    next hop 10.10.10.1, Serial1
    valid adjacency

  0 packets, 0 bytes switched through the prefix
  Load distribution: 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 (refcount 1)

!--- The active paths are assigned to hash buckets in a !--- round-robin pattern.
```

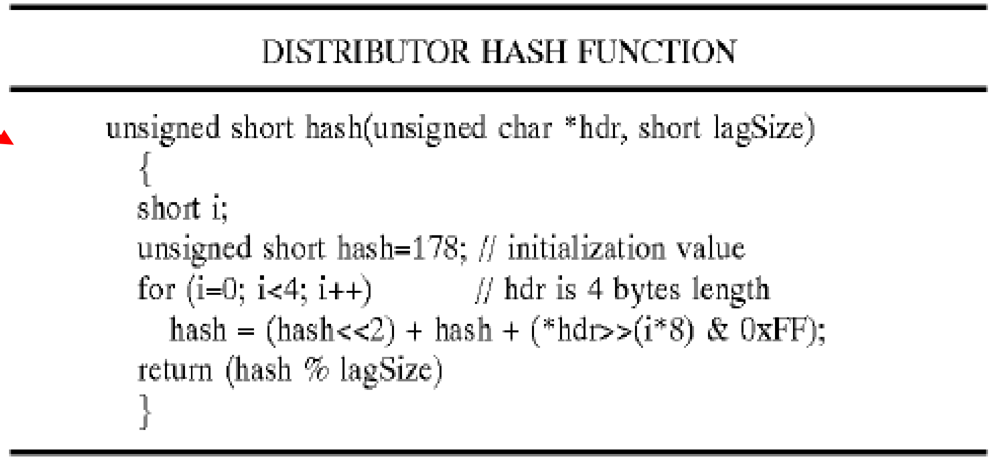
Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Cisco EtherChannel System in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[b] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references.

For example, Bruckman discloses applying a distributor hash function to the frame information which includes determining a number of the plurality of physical links.

Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of

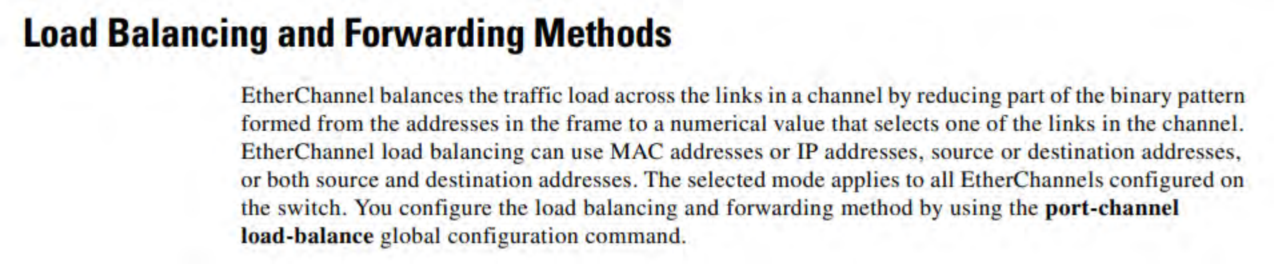
No.	’740 Patent Claim 10	Cisco EtherChannel
		information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”)

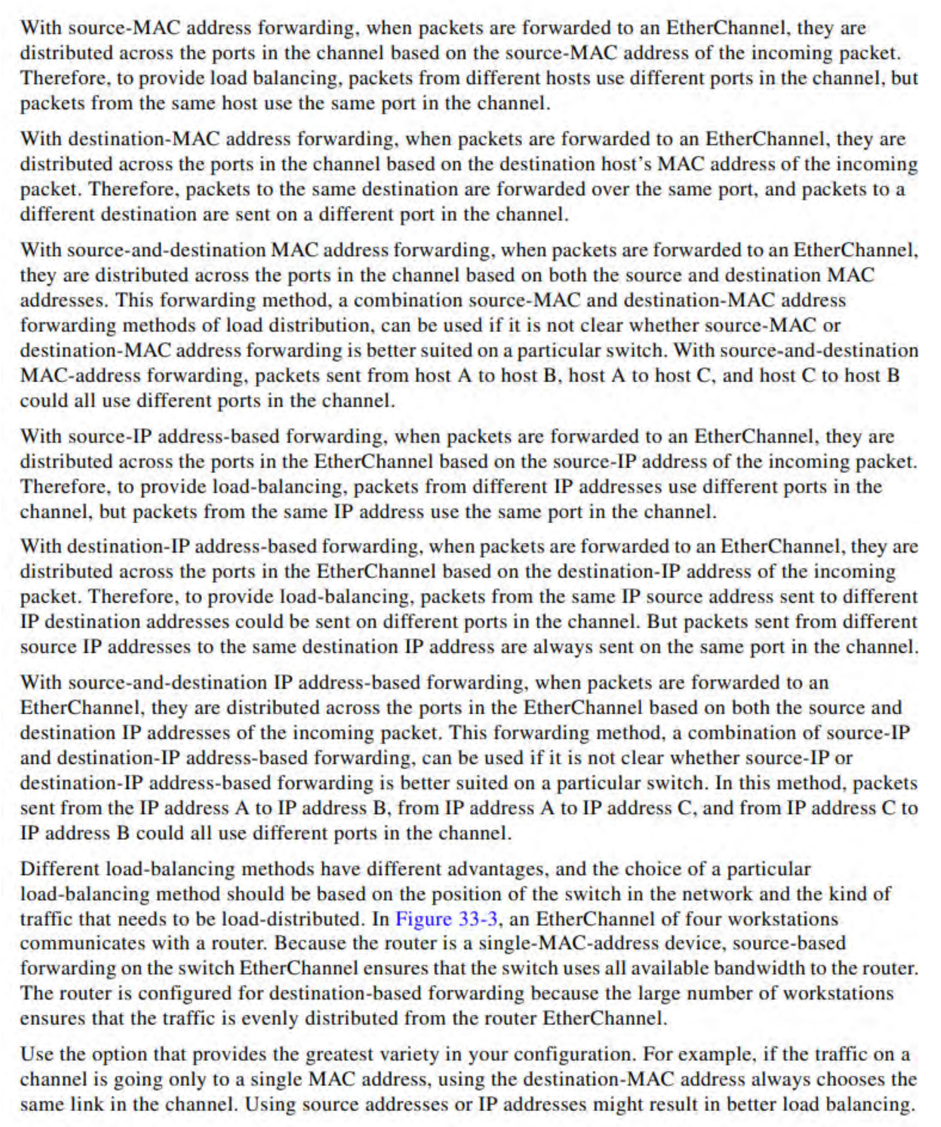
No.	’740 Patent Claim 10	Cisco EtherChannel

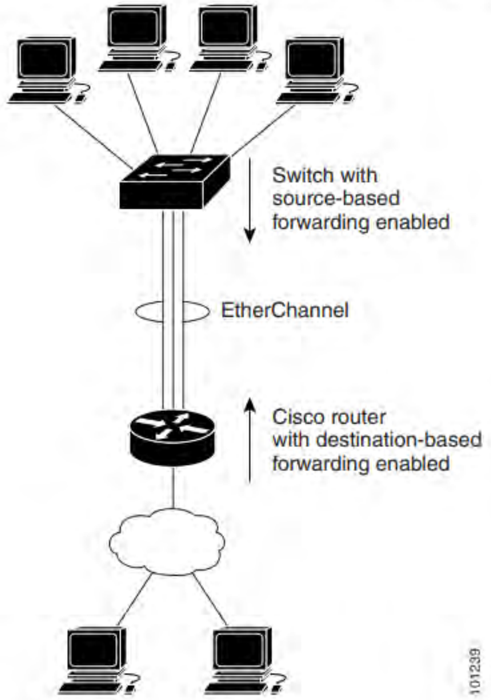
		Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria:
		[0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link.
		[0060] The hash function uses all the specified input data from the frame headers.
		[0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values
		[0062] The hash function generates very different hash values for similar data.”)
		
		Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:
		
		Bruckman at Table 1 (annotated)
		hashing function *“mapping function”*
		
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		    hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}

No.	’740 Patent Claim 10	Cisco EtherChannel
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”) For example, Alexander discloses applying a distributor hash function to the frame information which includes determining a hash key based on packet information. Alexander at 3:1-40 (“The hash function is preferably selected such that suc-cessive application of the hash function to all source and destination addresses expected to be seen by the Ethernet switch will produce a lowest value hash key, a highest value hash key, and a group of hash keys having intermediate values distributed evenly between the lowest and highest values. The distribution table contains a separate port identifier look-up table for each aggregated grouping of outgoing ports. Advantageously, the hash key is an N bit hash key; and, each port identifier look-up table contains 2^N entries occupying 2^N consecutive locations, with each entry being an identifier of a particular one of the physical outgoing ports. Identifiers for particular outgoing ports are retrieved from the distribution table by extracting first and second N bit hash keys which form part of the retrieved destination and source address contexts respectively. The hash keys are combined to form an N bit connection identifier. The port identifier look-up table corresponding to the aggregated grouping represented by the retrieved destination address is selected, and the entry at the table location corresponding to the value of the N bit connection identifier is retrieved. If the address look-up table does not contain a destination address corresponding to the extracted destination address then first and second hash keys are produced by applying a hash function to the extracted source and destination addresses respectively. The hash keys are combined to form an N bit connection identifier. The incoming port on which the packet containing the extracted source address was received is identified. All of the aggregated groupings are scanned to identify all outgoing ports to which packets may be directed from the incoming port on which the packet was received. For each one of those outgoing ports, the port identifier look-up table corresponding to the aggregated grouping containing that outgoing port is selected, the entry at the table location corresponding to

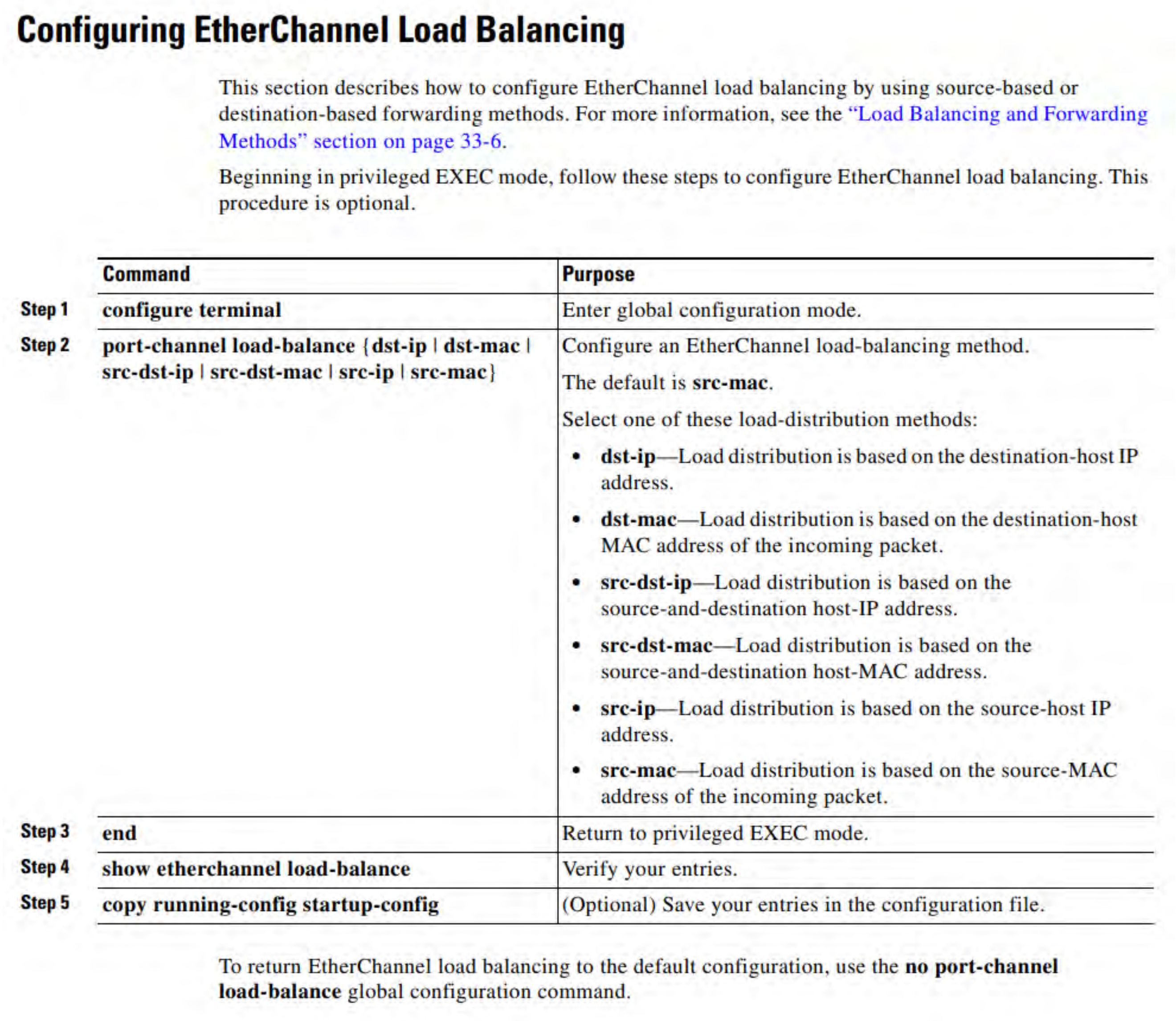
No.	’740 Patent Claim 10	Cisco EtherChannel
		the value of the N bit connection identifier is retrieved, and the received packet is queued for outgoing transmission on the outgoing port corresponding to the retrieved entry.”) Alexander at 5:10-35 (“If a packet arrives bearing a source Ethernet MAC address that was not found in look-up table 12 by address resolution unit 10, learning function 16 is invoked to update look-up table 12 with the new address (i.e. processing branches along the "No" exit from FIG. 2, block 36). Learning function 16 first computes a hash function on the source Ethernet MAC address, generating an N-bit hash key ("partial connection identifier") from the 48-bit MAC address, where N is some small integer in the range of 3 to 8 (FIG. 2, block 38). The physical port on which the packet arrived is then determined. If the physical port is found to be associated with an aggregate group (i.e., it is one of a set of ports that have been bound into a single logical port), then the logical identifier assigned to the aggregate group is also determined. The hash key is then stored into address look-up table 12 in conjunction with the actual Ethernet MAC address and the port identifier (FIG. 2, block 40). The physical port identifier is used if the port is not part of an aggregate group (i.e. if processing branched along the "No" exit from block 30 and through block 32), while the logical identifier is used for ports that have been aggregated (i.e. if processing branched along the "Yes" exit from block 30 and through block 34). The hash key and port identifier are considered to form the "context" for the given MAC address.”) Alexander at 5:36-46 (“The hash function should be selected to ensure an even distribution of hash key values over the range of MAC addresses that are expected to be seen by the Ethernet switch. As a specific example, the EXACT™ Ethernet switch system employs an exclusive-OR based hash function, wherein the 48-bit MAC address is divided into 16-bit blocks, which are then exclusive-ORed together to form a single 16-bit number; the 3 least significant bits (LSBs) of this number are taken to produce a 3-bit hash key. Other schemes such as CRC-based or checksum-based hashes may also be used.”) Alexander at 6:49-65 (“If the context information for the destination address indicates, however, that the target is an aggregate group (i.e. if processing branches along the "Yes" exit from FIG. 2, block 42) then the logical identifier assigned to the aggregate group is retrieved and is used to select the proper look-up table contained within the distribution table data structure. The hash keys (partial connection identifiers) stored into the contexts for the source and destination MAC addresses are obtained from address resolution unit 10 and combined to generate a "connection identifier" with the same number of bits (FIG. 2, block 44). (In the EXACT™ Ethernet switch, a Boolean exclusive-OR operation is used to combine the hash keys without increasing the number of

No.	’740 Patent Claim 10	Cisco EtherChannel
		bits.) This connection identifier is then used to index into the selected look-up table, and finally retrieve an actual physical port index on which the packet must be transmitted (FIG. 2, block 46).”)
10[c]	calculating a modulo of a division operation of the hashing key by the hashing size, and	Cisco EtherChannel System discloses calculating a modulo of a division operation of the hashing key by the hashing size. For example, Cisco EtherChannel System discloses forwarding path determinations by dividing the reduced value of the hash by the number of active paths. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Cisco EtherChannel System is found to not meet this limitation, calculating a modulo of a division operation of the hashing key by the hashing size would have been obvious to a person having ordinary skill in the art, as explained below. Catalyst 3560 Configuration Guide at 33-6 – 33-8  **Load Balancing and Forwarding Methods** EtherChannel balances the traffic load across the links in a channel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel. EtherChannel load balancing can use MAC addresses or IP addresses, source or destination addresses, or both source and destination addresses. The selected mode applies to all EtherChannels configured on the switch. You configure the load balancing and forwarding method by using the **port-channel load-balance** global configuration command.

No.	’740 Patent Claim 10	Cisco EtherChannel
		 With source-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the source-MAC address of the incoming packet. Therefore, to provide load balancing, packets from different hosts use different ports in the channel, but packets from the same host use the same port in the channel. With destination-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the destination host’s MAC address of the incoming packet. Therefore, packets to the same destination are forwarded over the same port, and packets to a different destination are sent on a different port in the channel. With source-and-destination MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on both the source and destination MAC addresses. This forwarding method, a combination source-MAC and destination-MAC address forwarding methods of load distribution, can be used if it is not clear whether source-MAC or destination-MAC address forwarding is better suited on a particular switch. With source-and-destination MAC-address forwarding, packets sent from host A to host B, host A to host C, and host C to host B could all use different ports in the channel. With source-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the source-IP address of the incoming packet. Therefore, to provide load-balancing, packets from different IP addresses use different ports in the channel, but packets from the same IP address use the same port in the channel. With destination-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the destination-IP address of the incoming packet. Therefore, to provide load-balancing, packets from the same IP source address sent to different IP destination addresses could be sent on different ports in the channel. But packets sent from different source IP addresses to the same destination IP address are always sent on the same port in the channel. With source-and-destination IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on both the source and destination IP addresses of the incoming packet. This forwarding method, a combination of source-IP and destination-IP address-based forwarding, can be used if it is not clear whether source-IP or destination-IP address-based forwarding is better suited on a particular switch. In this method, packets sent from the IP address A to IP address B, from IP address A to IP address C, and from IP address C to IP address B could all use different ports in the channel. Different load-balancing methods have different advantages, and the choice of a particular load-balancing method should be based on the position of the switch in the network and the kind of traffic that needs to be load-distributed. In Figure 33-3, an EtherChannel of four workstations communicates with a router. Because the router is a single-MAC-address device, source-based forwarding on the switch EtherChannel ensures that the switch uses all available bandwidth to the router. The router is configured for destination-based forwarding because the large number of workstations ensures that the traffic is evenly distributed from the router EtherChannel. Use the option that provides the greatest variety in your configuration. For example, if the traffic on a channel is going only to a single MAC address, using the destination-MAC address always chooses the same link in the channel. Using source addresses or IP addresses might result in better load balancing.

No.	’740 Patent Claim 10	Cisco EtherChannel
		Figure 33-3 Load Distribution and Forwarding Methods  Switch with source-based forwarding enabled EtherChannel Cisco router with destination-based forwarding enabled 101239 Catalyst 3560 Configuration Guide at 33-16

No.	’740 Patent Claim 10	Cisco EtherChannel



Configuring EtherChannel Load Balancing

This section describes how to configure EtherChannel load balancing by using source-based or destination-based forwarding methods. For more information, see the “Load Balancing and Forwarding Methods” section on page 33-6.

Beginning in privileged EXEC mode, follow these steps to configure EtherChannel load balancing. This procedure is optional.

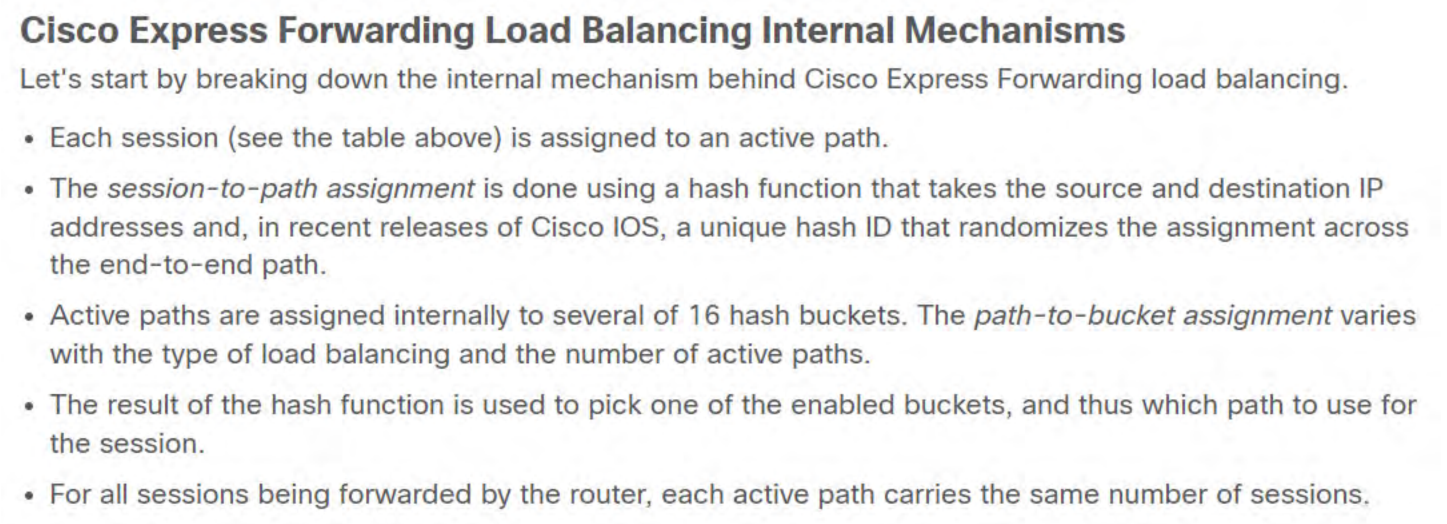
	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**port-channel load-balance** {**dst-ip** | **dst-mac** | **src-dst-ip** | **src-dst-mac** | **src-ip** | **src-mac**}	Configure an EtherChannel load-balancing method. The default is **src-mac**. Select one of these load-distribution methods: • **dst-ip**—Load distribution is based on the destination-host IP address. • **dst-mac**—Load distribution is based on the destination-host MAC address of the incoming packet. • **src-dst-ip**—Load distribution is based on the source-and-destination host-IP address. • **src-dst-mac**—Load distribution is based on the source-and-destination host-MAC address. • **src-ip**—Load distribution is based on the source-host IP address. • **src-mac**—Load distribution is based on the source-MAC address of the incoming packet.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show etherchannel load-balance**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return EtherChannel load balancing to the default configuration, use the **no port-channel load-balance** global configuration command.

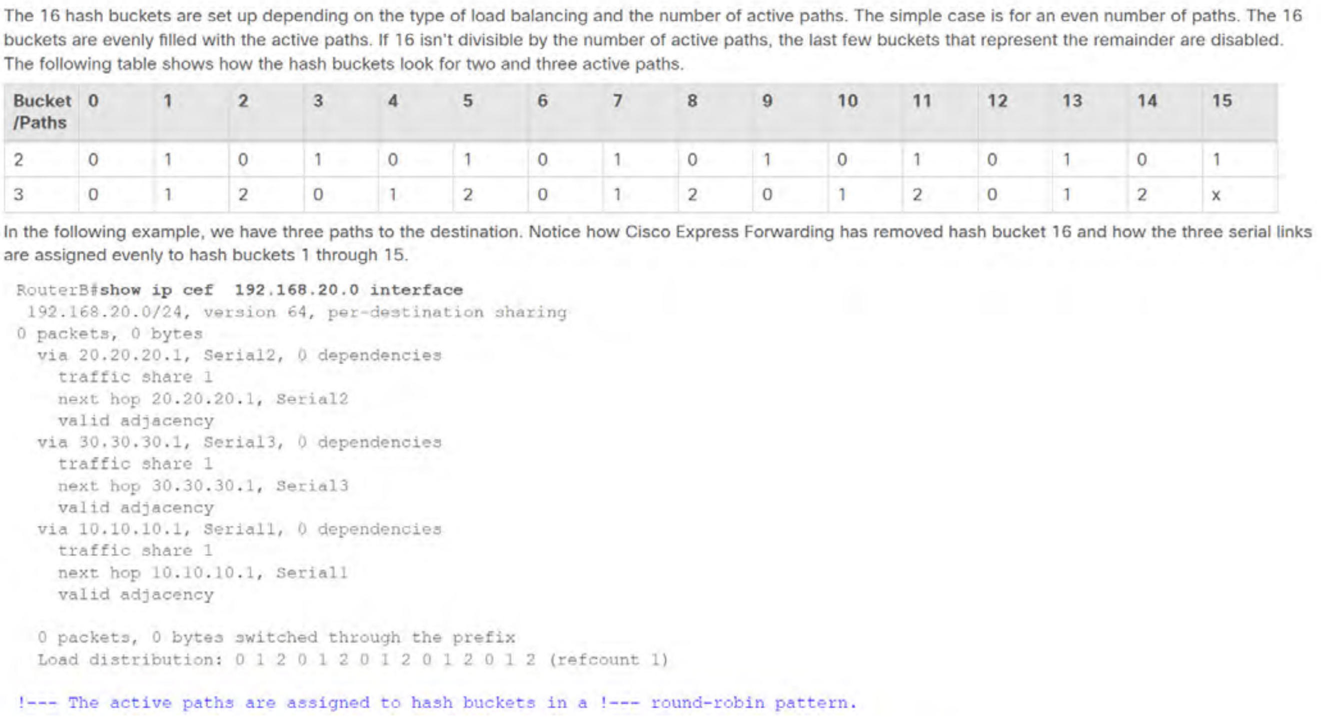
No.	’740 Patent Claim 10	Cisco EtherChannel
		Layer 2 EtherChannel **Catalyst 2950** Catalyst 2950 switches only support 802.1Q trunking and do not support ISL trunking. Catalyst 2950 switches support DTP and PAgP dynamic trunking and channel negotiation with Cisco IOS Software Release 12.1 releases and static modes only with Cisco IOS Software Release 12.0 releases. EtherChannel load balancing can use either source-MAC or destination-MAC address forwarding. You can configure the load balancing method by issuing the **port-channel load-balance** global configuration command. These switches support up to eight switch ports per channel. **Catalyst 6500 That Runs Cisco IOS Software** Catalyst 6500 switches that run Cisco IOS Software support L2 (switchport) and Layer 3 (L3) (routed port) EtherChannel configurations. A Catalyst 6500/6000 series switch supports a maximum of 64 EtherChannels (256 with Cisco IOS Software Release 12.1(2)E and earlier). You can form an EtherChannel with up to eight compatibly configured LAN ports on any module in a Catalyst 6000 series switch, with the exception of Digital Feature Card (DFC)-equipped modules (such as WS-X6816 and so on) which currently allow an L2 channel only using ports on the same DFC module. However, an L3 channel can be configured across different DFC-equipped modules. This limitation has been removed in Catalyst 6500/6000 Cisco IOS Software Release 12.1(11b)EX and later. This document configures an L2 EtherChannel. The Catalyst 6500/6000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP addresses, or Layer 4 (L4) port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. Catalyst 6500/6000 switches support both ISL and 802.1Q trunking encapsulations and DTP. Detailed information on port capabilities is available by issuing the **show interface** *interface_id* **capabilities** command.

No.	’740 Patent Claim 10	Cisco EtherChannel
		Catalyst 4000 That Runs Cisco IOS Software Catalyst 4000 switches that run Cisco IOS Software (with Supervisor Engine III and IV) support L2 (switchport) and L3 (routed port) EtherChannel configurations. A Catalyst 4000 series switch supports a maximum of 64 EtherChannels. You can form an EtherChannel with up to eight compatibly configured Ethernet interfaces on any module, and across modules in a Catalyst 4000 series switch. All interfaces in each EtherChannel must be the same speed and must all be configured as either L2 or L3 interfaces. The Catalyst 4000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP address, or L4 port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. The Catalyst 4000 that runs Cisco IOS Software supports ISL and 802.1Q trunking encapsulations and DTP. ISL is not available on certain modules. For a complete list of such modules, refer to the *Understanding VLAN Trunks* section of Configuring Layer 2 Ethernet Interfaces. In a future software release, detailed information on port capabilities will be available by issuing the **show interface capabilities** command. Currently this command is not available. Configuring EtherChannel at 1 -2 **Introduction** This sample configuration demonstrates how to set up a Layer 3 (L3) EtherChannel, without VLAN trunking, between a Cisco router and a Cisco Catalyst 6500 switch running Cisco IOS® System Software. EtherChannel can be called Fast EtherChannel (FEC) or Gigabit EtherChannel (GEC); the term depends on the speed of the interfaces or ports you use to form the EtherChannel. In this example, two Fast Ethernet ports from a Cisco router and a Catalyst 6500 switch have been bundled into a FEC. Throughout this document, the terms FEC, GEC, port channel, channel, and port group all refer to EtherChannel.

No.	’740 Patent Claim 10	Cisco EtherChannel
		Before you attempt this configuration, ensure that you meet these requirements: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software support both Layer 2 (L2) and L3 EtherChannel, with up to eight compatibly configured Ethernet interfaces on any module. All interfaces in each EtherChannel must be the same speed. All must be configured as either L2 or L3 interfaces. • EtherChannel load balancing can use either MAC addresses, IP addresses, or the TCP port numbers. **Note:** The selected mode applies to all EtherChannels configured on the switch. • Catalyst 6500/6000 Cisco IOS Software Release 12.1E or later and Catalyst 4500/4000 Cisco IOS Software Release 12.1(8a)EW or later. • Cisco routers: • IP traffic distributes over the port channel interface while traffic from other routing protocols sends over a single link. Bridged traffic distributes on the basis of the L3 information in the packet. If the L3 information does not exist in the packet, the traffic sends over the first link. • A wide variety of Cisco routers support EtherChannel. To find a platform or version of code that supports EtherChannel on a Cisco router, use the Cisco Feature Navigator II (registered customers only) . A list of routers and Cisco IOS Software releases that support EtherChannel is found under the FEC feature. Load Balancing at 7-9

No.	’740 Patent Claim 10	Cisco EtherChannel
		 **Cisco Express Forwarding Load Balancing Internal Mechanisms** Let's start by breaking down the internal mechanism behind Cisco Express Forwarding load balancing. • Each session (see the table above) is assigned to an active path. • The *session-to-path assignment* is done using a hash function that takes the source and destination IP addresses and, in recent releases of Cisco IOS, a unique hash ID that randomizes the assignment across the end-to-end path. • Active paths are assigned internally to several of 16 hash buckets. The *path-to-bucket assignment* varies with the type of load balancing and the number of active paths. • The result of the hash function is used to pick one of the enabled buckets, and thus which path to use for the session. • For all sessions being forwarded by the router, each active path carries the same number of sessions.

No.	’740 Patent Claim 10	Cisco EtherChannel



The 16 hash buckets are set up depending on the type of load balancing and the number of active paths. The simple case is for an even number of paths. The 16 buckets are evenly filled with the active paths. If 16 isn't divisible by the number of active paths, the last few buckets that represent the remainder are disabled. The following table shows how the hash buckets look for two and three active paths.

Bucket /Paths	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
3	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	x

In the following example, we have three paths to the destination. Notice how Cisco Express Forwarding has removed hash bucket 16 and how the three serial links are assigned evenly to hash buckets 1 through 15.

```
RouterB#show ip cef  192.168.20.0 interface
 192.168.20.0/24, version 64, per-destination sharing
0 packets, 0 bytes
  via 20.20.20.1, Serial2, 0 dependencies
    traffic share 1
    next hop 20.20.20.1, Serial2
    valid adjacency
  via 30.30.30.1, Serial3, 0 dependencies
    traffic share 1
    next hop 30.30.30.1, Serial3
    valid adjacency
  via 10.10.10.1, Serial1, 0 dependencies
    traffic share 1
    next hop 10.10.10.1, Serial1
    valid adjacency

  0 packets, 0 bytes switched through the prefix
  Load distribution: 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 (refcount 1)

!--- The active paths are assigned to hash buckets in a !--- round-robin pattern.
```

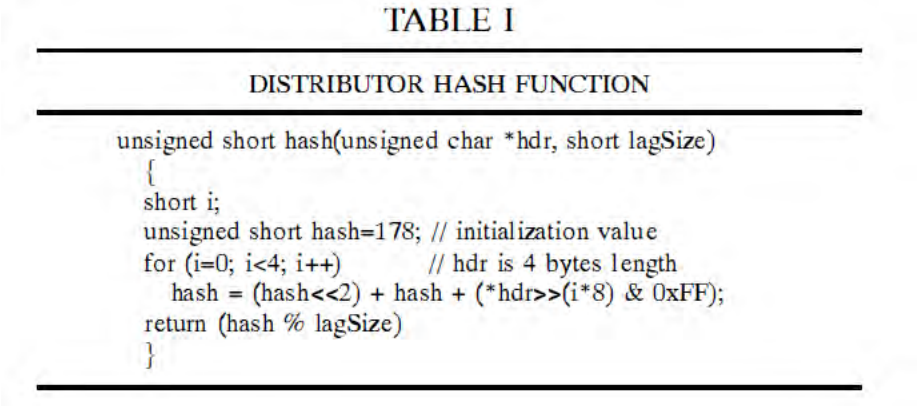
Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Cisco EtherChannel System in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[c] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references.

For example, Bruckman discloses distributing data frames over physical links and traces based on a hash function involving a division operation (%).

Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”)

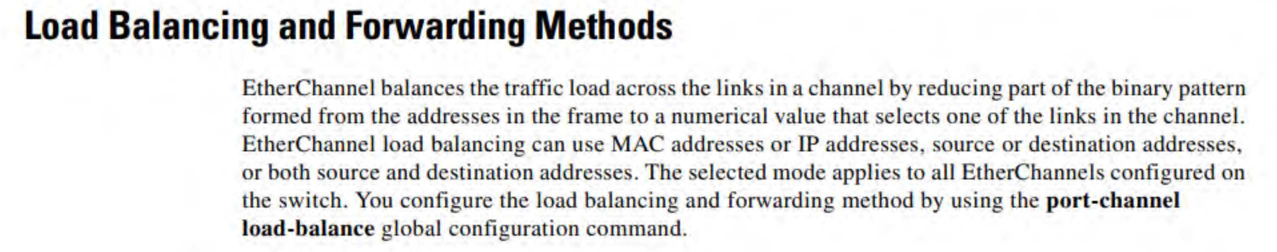
No.	’740 Patent Claim 10	Cisco EtherChannel
		Bruckman at [0025] (“Typically, setting the protection policy includes determining a maximum number of the physical links that may fail while the logical link continues to provide at least the guaranteed bandwidth for the connection. In one embodiment, the guaranteed bandwidth is a bandwidth B, and the plurality of physical links consists of N links, and the maximum number is an integer P, and the link bandwidth allocated to each of the links is no less than B/(N-P). Conveying the data may further include managing the transmission of the data responsively to an actual number X of the physical links that have failed so that the guaranteed bandwidth on each of the links is limited to B/(N-X), X P, and an excess bandwidth on the physical links over the guaranteed bandwidth is available for other connections.”) Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”) Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:

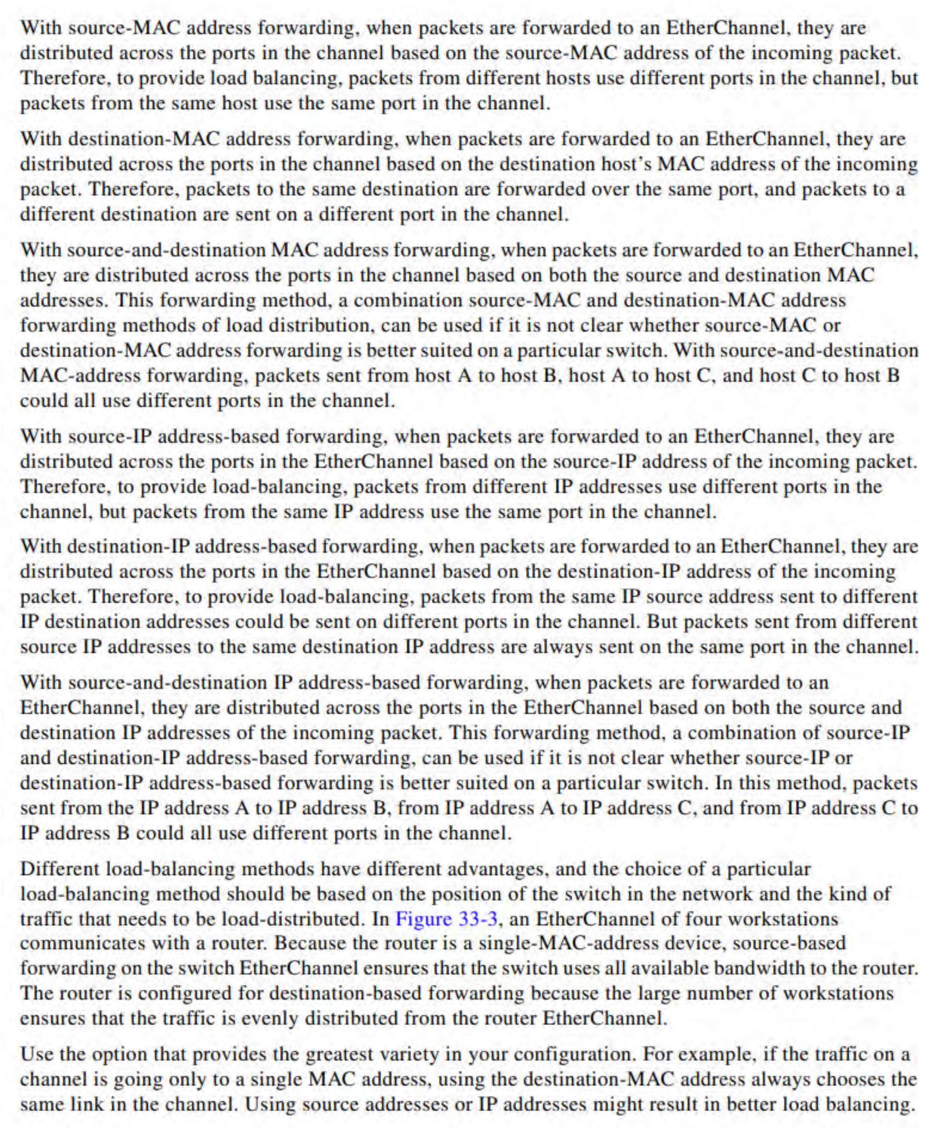
No.	’740 Patent Claim 10	Cisco EtherChannel

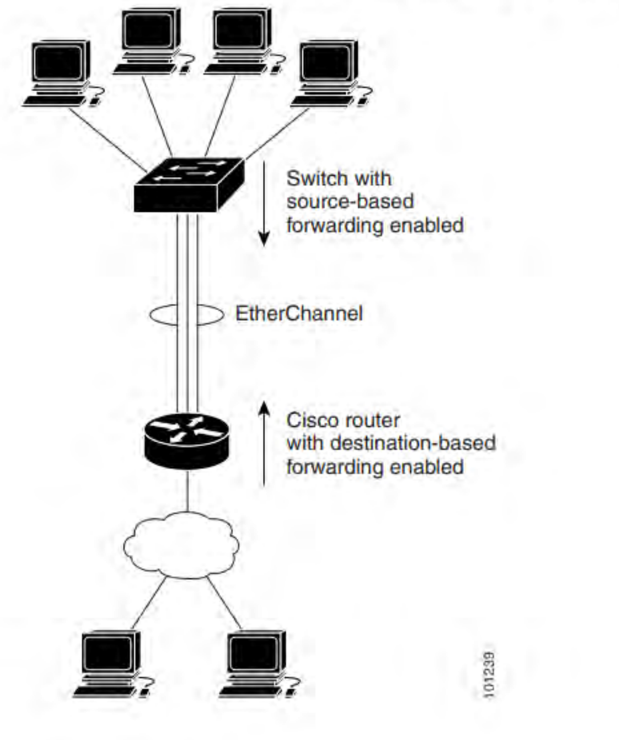
		
		TABLE I
		DISTRIBUTOR HASH FUNCTION
		`unsigned short hash(unsigned char *hdr, short lagSize)`
		`{`
		`short i;`
		`unsigned short hash=178; // initialization value`
		`for (i=0; i<4; i++)          // hdr is 4 bytes length`
		`hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);`
		`return (hash % lagSize)`
		`}`
		”)
		
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		
		Bruckman at [0067] (“A similar problem may arise if there is a failure in a link in an aggregation group or in one of a number of line cards serving the aggregation group. In this case, to maintain the bandwidth allocation B made by CAC 44, each of the remaining links in the group must now carry, on average, B/(N-M) traffic, wherein M is the number of links in the group that are out of service. If only BIN has been allocated to each link, the remaining active links may not have sufficient bandwidth to continue to provide the bandwidth that has been guaranteed to the connections that they are required to carry. A similar problem arises with respect to loading of traces 52. For example, if there is a failure in LC2 or in one of links 30 in group 36 that connect to LC2, the trace connecting the multiplexer 50 in LCl will have to carry a substantially larger share of the bandwidth, or even all of the bandwidth, that is allocated to the connection in ques-tion.”)
		
		Bruckman at [0068] (“FIG. 3 is a flow chart that schematically illustrates a method for dealing with these problems of fluctuating bandwidth requirements, in accordance with an embodiment of the present invention. In order to provide sufficient bandwidth for failure protection, CAC 44 uses a safety margin based on a protection parameter P, which is assigned at a protection setting step 60. P represents the maximum number of links in the

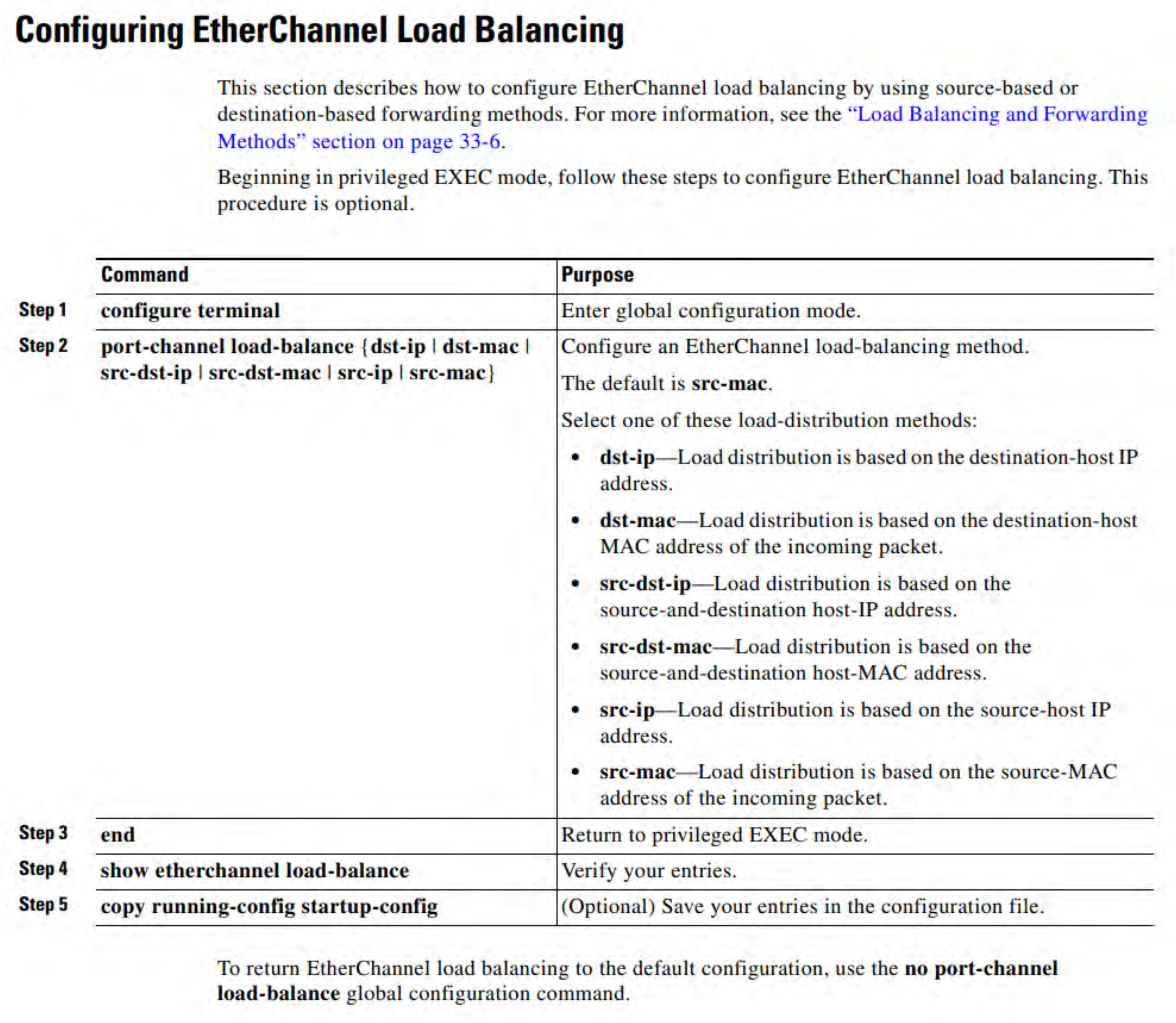
No.	’740 Patent Claim 10	Cisco EtherChannel
		group that can be out of service while still permitting the aggregation group to provide a given connection with the bandwidth that has been guaranteed to the connection. CAC 44 will then allocate at least B/(N-P) bandwidth to each link in the group, so that if P links fail, the group still provides total bandwidth of (N-P)*B/(N-P)= B. Setting P=l is equivalent to 1:N protection, so that the group will be unaffected by failure of a single link. In the example of group 36, shown in FIG. 2, setting P=2 will give both facility and equipment protection, i.e., the group will be unaffected not only by failure of a link, but also by failure of one of line cards 34. In the extreme case, in which P=N-1, CAC 44 will allocate the full bandwidth B on each link in the group.”) As another example, Singh discloses determining a ratio between the number of ingress and egress links and the number of links carrying data to the backplane and using a modulo to correspond to the channel’s link number. Singh at 9:30-43 (“The ratio between the number of line ingress links and the number of links carrying data to the backplane gives the backplane speedup for the system. In this example, there are 10 ingress links into the MS and 20 links (2 backplane channels) carrying that data to the backplane. This gives a backplane speedup of 2x. As another example, with 8 ingress links and 12 backplane links, there is a speedup of 1.5x. It should be noted that in addition to the backplane speedup, there is also an ingress/egress speedup. With 10 ingress links capable of carrying 2 Gbps each of raw data, this presents a 20 Gbps interface to the MS. An OC-192 only has approximately 10 Gbps worth of data. Taking into account cell overhead and cell quantization inefficiencies, there still remains excess capacity in the links.”) Singh at 11:29-38 (“FIG. 9 is a diagram illustrating link to channel assignments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a channel, whether it is a backplane channel or an ingress/egress channel. There is no logical relationship between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.”) Singh at 15:15-39 (“The number of crossbars that are required in a system is dependent on how many links are being used to create the backplane channels. There should be an even number of crossbars and they would be divided evenly across the switch cards. The following equation, for most cases, pro-vides the correct number of crossbars:

No.	’740 Patent Claim 10	Cisco EtherChannel
		# of Crossbars=(# links per ingress channelx# of ingress channels per portx# of port cardsx speedup)/32. For the 8x8 configuration, the # of crossbars should be multiplied by (4x# of iMS)/(# backplane channels per port card). The number of port cards should be rounded up to the nearest supported configuration, i.e. 8, 16, or 32. The speedup in the case of crossbars should be the fractional speedup that is desired. Example to determine the number of arbiters and cross-bars for the following system: 4 channel port cards (40 Gbps) 8 links per channel 16 port cards Speedup=l.5 # of arbiters=(4x2x2)/2=8 # of crossbars=(8x4x16xl.5)/32=24. This would give 3crossbars per arbiter.”) Singh at 16:28-44 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 16x16 and 32x32 is the organization of the switchplane. The port card remains the same. Backplane channels 1 and 2 are used for the backplane connectivity. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 16, correspond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”) Singh at 17:31-49 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 8x8 and 16x16 is the organi-zation of the switchplane. The port card remains the same. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Links 0-7 and 24-31 on the arbiters would not be used and would be powered off. Links 0-7 and 24-31 on the crossbars would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B

No.	’740 Patent Claim 10	Cisco EtherChannel
		on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Backplane channels 1 and 2 are used for the backplane connectivity. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 8, corre-spond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”)
10[d]	selecting the first and second physical links responsively to the modulo.	Cisco EtherChannel System discloses selecting the first and second physical links responsively to the modulo. For example, Cisco EtherChannel System discloses using the result of the division to select the forwarding path of the port and switch fabric interconnection channels over which to send the packet. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Cisco EtherChannel System is found to not meet this limitation, selecting the first and second physical links responsively to the modulo would have been obvious to a person having ordinary skill in the art, as explained below. Catalyst 3560 Configuration Guide at 33-6 – 33-8  **Load Balancing and Forwarding Methods** EtherChannel balances the traffic load across the links in a channel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel. EtherChannel load balancing can use MAC addresses or IP addresses, source or destination addresses, or both source and destination addresses. The selected mode applies to all EtherChannels configured on the switch. You configure the load balancing and forwarding method by using the **port-channel load-balance** global configuration command.

No.	’740 Patent Claim 10	Cisco EtherChannel
		 With source-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the source-MAC address of the incoming packet. Therefore, to provide load balancing, packets from different hosts use different ports in the channel, but packets from the same host use the same port in the channel. With destination-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the destination host’s MAC address of the incoming packet. Therefore, packets to the same destination are forwarded over the same port, and packets to a different destination are sent on a different port in the channel. With source-and-destination MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on both the source and destination MAC addresses. This forwarding method, a combination source-MAC and destination-MAC address forwarding methods of load distribution, can be used if it is not clear whether source-MAC or destination-MAC address forwarding is better suited on a particular switch. With source-and-destination MAC-address forwarding, packets sent from host A to host B, host A to host C, and host C to host B could all use different ports in the channel. With source-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the source-IP address of the incoming packet. Therefore, to provide load-balancing, packets from different IP addresses use different ports in the channel, but packets from the same IP address use the same port in the channel. With destination-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the destination-IP address of the incoming packet. Therefore, to provide load-balancing, packets from the same IP source address sent to different IP destination addresses could be sent on different ports in the channel. But packets sent from different source IP addresses to the same destination IP address are always sent on the same port in the channel. With source-and-destination IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on both the source and destination IP addresses of the incoming packet. This forwarding method, a combination of source-IP and destination-IP address-based forwarding, can be used if it is not clear whether source-IP or destination-IP address-based forwarding is better suited on a particular switch. In this method, packets sent from the IP address A to IP address B, from IP address A to IP address C, and from IP address C to IP address B could all use different ports in the channel. Different load-balancing methods have different advantages, and the choice of a particular load-balancing method should be based on the position of the switch in the network and the kind of traffic that needs to be load-distributed. In Figure 33-3, an EtherChannel of four workstations communicates with a router. Because the router is a single-MAC-address device, source-based forwarding on the switch EtherChannel ensures that the switch uses all available bandwidth to the router. The router is configured for destination-based forwarding because the large number of workstations ensures that the traffic is evenly distributed from the router EtherChannel. Use the option that provides the greatest variety in your configuration. For example, if the traffic on a channel is going only to a single MAC address, using the destination-MAC address always chooses the same link in the channel. Using source addresses or IP addresses might result in better load balancing.

No.	’740 Patent Claim 10	Cisco EtherChannel
		Figure 33-3 Load Distribution and Forwarding Methods  Switch with source-based forwarding enabled EtherChannel Cisco router with destination-based forwarding enabled 101239 Catalyst 3560 Configuration Guide at 33-16

No.	’740 Patent Claim 10	Cisco EtherChannel
		

Configuring EtherChannel Load Balancing

This section describes how to configure EtherChannel load balancing by using source-based or destination-based forwarding methods. For more information, see the “Load Balancing and Forwarding Methods” section on page 33-6.

Beginning in privileged EXEC mode, follow these steps to configure EtherChannel load balancing. This procedure is optional.

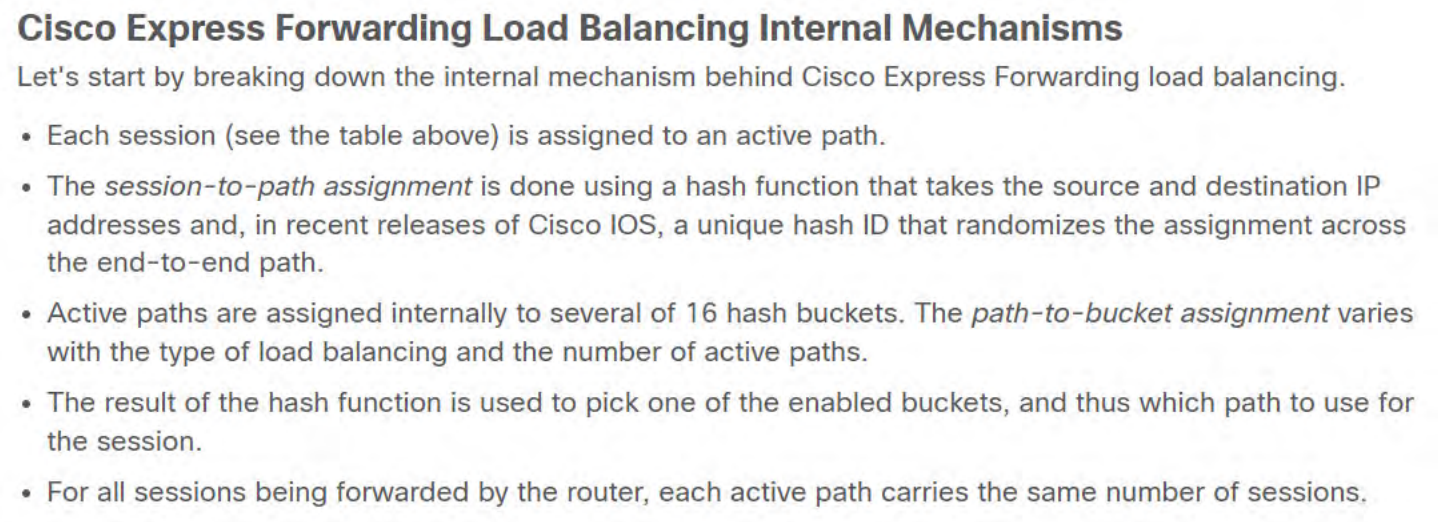
	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**port-channel load-balance** {**dst-ip** | **dst-mac** | **src-dst-ip** | **src-dst-mac** | **src-ip** | **src-mac**}	Configure an EtherChannel load-balancing method. The default is **src-mac**. Select one of these load-distribution methods: • **dst-ip**—Load distribution is based on the destination-host IP address. • **dst-mac**—Load distribution is based on the destination-host MAC address of the incoming packet. • **src-dst-ip**—Load distribution is based on the source-and-destination host-IP address. • **src-dst-mac**—Load distribution is based on the source-and-destination host-MAC address. • **src-ip**—Load distribution is based on the source-host IP address. • **src-mac**—Load distribution is based on the source-MAC address of the incoming packet.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show etherchannel load-balance**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return EtherChannel load balancing to the default configuration, use the **no port-channel load-balance** global configuration command.

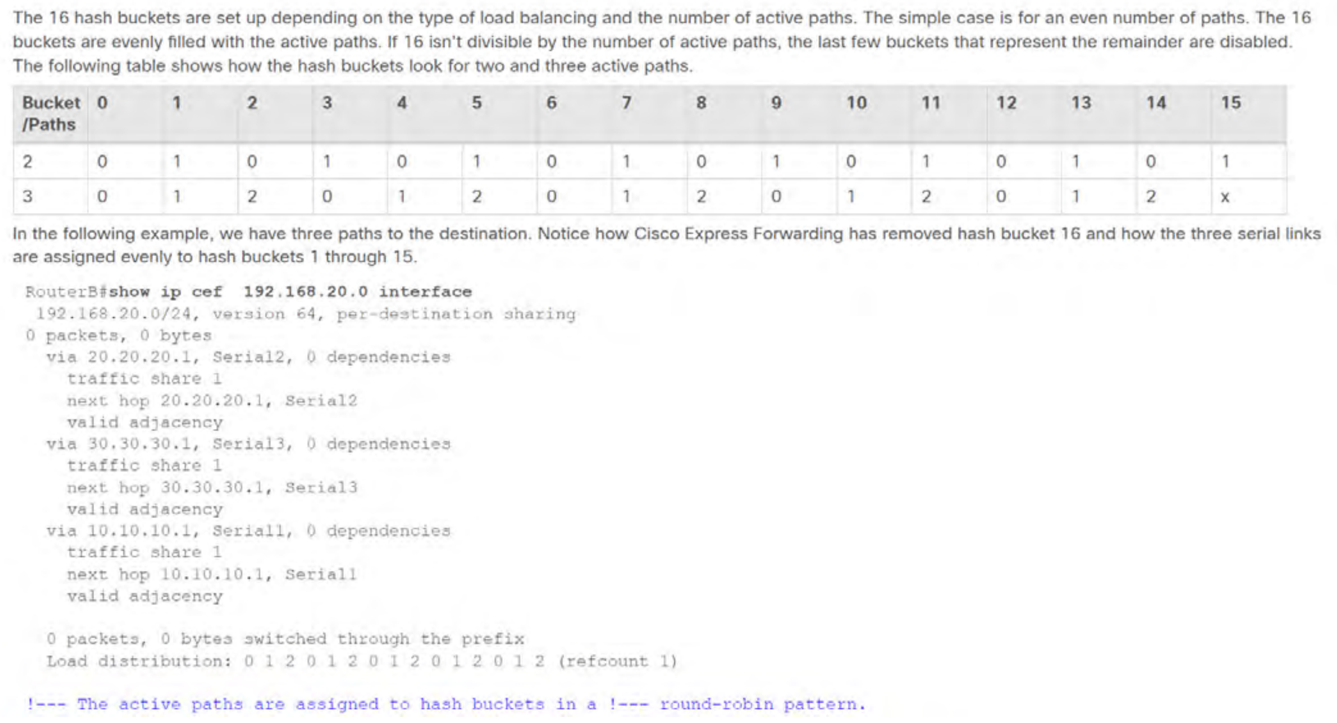
No.	’740 Patent Claim 10	Cisco EtherChannel
		Layer 2 EtherChannel **Catalyst 2950** Catalyst 2950 switches only support 802.1Q trunking and do not support ISL trunking. Catalyst 2950 switches support DTP and PAgP dynamic trunking and channel negotiation with Cisco IOS Software Release 12.1 releases and static modes only with Cisco IOS Software Release 12.0 releases. EtherChannel load balancing can use either source-MAC or destination-MAC address forwarding. You can configure the load balancing method by issuing the **port-channel load-balance** global configuration command. These switches support up to eight switch ports per channel. **Catalyst 6500 That Runs Cisco IOS Software** Catalyst 6500 switches that run Cisco IOS Software support L2 (switchport) and Layer 3 (L3) (routed port) EtherChannel configurations. A Catalyst 6500/6000 series switch supports a maximum of 64 EtherChannels (256 with Cisco IOS Software Release 12.1(2)E and earlier). You can form an EtherChannel with up to eight compatibly configured LAN ports on any module in a Catalyst 6000 series switch, with the exception of Digital Feature Card (DFC)-equipped modules (such as WS-X6816 and so on) which currently allow an L2 channel only using ports on the same DFC module. However, an L3 channel can be configured across different DFC-equipped modules. This limitation has been removed in Catalyst 6500/6000 Cisco IOS Software Release 12.1(11b)EX and later. This document configures an L2 EtherChannel. The Catalyst 6500/6000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP addresses, or Layer 4 (L4) port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. Catalyst 6500/6000 switches support both ISL and 802.1Q trunking encapsulations and DTP. Detailed information on port capabilities is available by issuing the **show interface** *interface_id* **capabilities** command.

No.	’740 Patent Claim 10	Cisco EtherChannel
		Catalyst 4000 That Runs Cisco IOS Software Catalyst 4000 switches that run Cisco IOS Software (with Supervisor Engine III and IV) support L2 (switchport) and L3 (routed port) EtherChannel configurations. A Catalyst 4000 series switch supports a maximum of 64 EtherChannels. You can form an EtherChannel with up to eight compatibly configured Ethernet interfaces on any module, and across modules in a Catalyst 4000 series switch. All interfaces in each EtherChannel must be the same speed and must all be configured as either L2 or L3 interfaces. The Catalyst 4000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP address, or L4 port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. The Catalyst 4000 that runs Cisco IOS Software supports ISL and 802.1Q trunking encapsulations and DTP. ISL is not available on certain modules. For a complete list of such modules, refer to the *Understanding VLAN Trunks* section of Configuring Layer 2 Ethernet Interfaces. In a future software release, detailed information on port capabilities will be available by issuing the **show interface capabilities** command. Currently this command is not available. Configuring EtherChannel at 1 -2 **Introduction** This sample configuration demonstrates how to set up a Layer 3 (L3) EtherChannel, without VLAN trunking, between a Cisco router and a Cisco Catalyst 6500 switch running Cisco IOS® System Software. EtherChannel can be called Fast EtherChannel (FEC) or Gigabit EtherChannel (GEC); the term depends on the speed of the interfaces or ports you use to form the EtherChannel. In this example, two Fast Ethernet ports from a Cisco router and a Catalyst 6500 switch have been bundled into a FEC. Throughout this document, the terms FEC, GEC, port channel, channel, and port group all refer to EtherChannel.

No.	’740 Patent Claim 10	Cisco EtherChannel
		Before you attempt this configuration, ensure that you meet these requirements: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software support both Layer 2 (L2) and L3 EtherChannel, with up to eight compatibly configured Ethernet interfaces on any module. All interfaces in each EtherChannel must be the same speed. All must be configured as either L2 or L3 interfaces. • EtherChannel load balancing can use either MAC addresses, IP addresses, or the TCP port numbers. **Note:** The selected mode applies to all EtherChannels configured on the switch. • Catalyst 6500/6000 Cisco IOS Software Release 12.1E or later and Catalyst 4500/4000 Cisco IOS Software Release 12.1(8a)EW or later. • Cisco routers: • IP traffic distributes over the port channel interface while traffic from other routing protocols sends over a single link. Bridged traffic distributes on the basis of the L3 information in the packet. If the L3 information does not exist in the packet, the traffic sends over the first link. • A wide variety of Cisco routers support EtherChannel. To find a platform or version of code that supports EtherChannel on a Cisco router, use the Cisco Feature Navigator II (registered customers only) . A list of routers and Cisco IOS Software releases that support EtherChannel is found under the FEC feature. Load Balancing at 7-9

No.	’740 Patent Claim 10	Cisco EtherChannel
		 **Cisco Express Forwarding Load Balancing Internal Mechanisms** Let's start by breaking down the internal mechanism behind Cisco Express Forwarding load balancing. • Each session (see the table above) is assigned to an active path. • The *session-to-path assignment* is done using a hash function that takes the source and destination IP addresses and, in recent releases of Cisco IOS, a unique hash ID that randomizes the assignment across the end-to-end path. • Active paths are assigned internally to several of 16 hash buckets. The *path-to-bucket assignment* varies with the type of load balancing and the number of active paths. • The result of the hash function is used to pick one of the enabled buckets, and thus which path to use for the session. • For all sessions being forwarded by the router, each active path carries the same number of sessions.

No.	’740 Patent Claim 10	Cisco EtherChannel



The 16 hash buckets are set up depending on the type of load balancing and the number of active paths. The simple case is for an even number of paths. The 16 buckets are evenly filled with the active paths. If 16 isn't divisible by the number of active paths, the last few buckets that represent the remainder are disabled. The following table shows how the hash buckets look for two and three active paths.

Bucket /Paths	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
3	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	x

In the following example, we have three paths to the destination. Notice how Cisco Express Forwarding has removed hash bucket 16 and how the three serial links are assigned evenly to hash buckets 1 through 15.

```
RouterB#show ip cef  192.168.20.0 interface
 192.168.20.0/24, version 64, per-destination sharing
0 packets, 0 bytes
  via 20.20.20.1, Serial2, 0 dependencies
    traffic share 1
    next hop 20.20.20.1, Serial2
    valid adjacency
  via 30.30.30.1, Serial3, 0 dependencies
    traffic share 1
    next hop 30.30.30.1, Serial3
    valid adjacency
  via 10.10.10.1, Serial1, 0 dependencies
    traffic share 1
    next hop 10.10.10.1, Serial1
    valid adjacency

  0 packets, 0 bytes switched through the prefix
  Load distribution: 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 (refcount 1)

!--- The active paths are assigned to hash buckets in a !--- round-robin pattern.
```

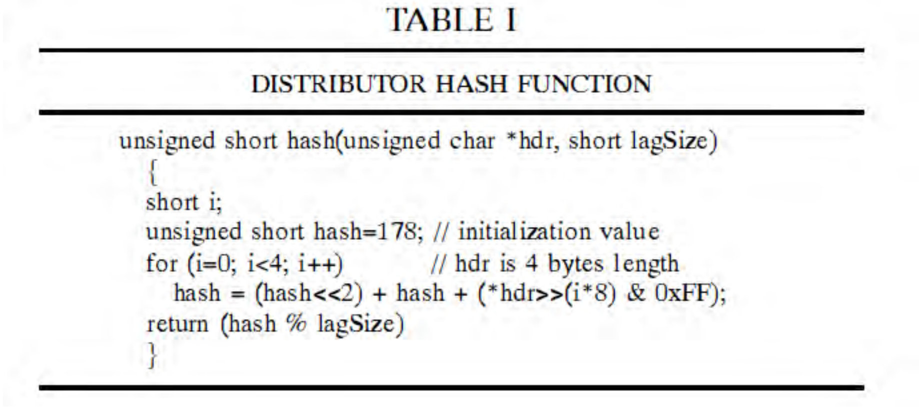
Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Cisco EtherChannel System in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[d] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references.

For example, Bruckman discloses distributing data frames over physical links and traces based on a hash function involving a division operation (%).

Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”)

No.	’740 Patent Claim 10	Cisco EtherChannel
		Bruckman at [0025] (“Typically, setting the protection policy includes determining a maximum number of the physical links that may fail while the logical link continues to provide at least the guaranteed bandwidth for the connection. In one embodiment, the guaranteed bandwidth is a bandwidth B, and the plurality of physical links consists of N links, and the maximum number is an integer P, and the link bandwidth allocated to each of the links is no less than B/(N-P). Conveying the data may further include managing the transmission of the data responsively to an actual number X of the physical links that have failed so that the guaranteed bandwidth on each of the links is limited to B/(N-X), X P, and an excess bandwidth on the physical links over the guaranteed bandwidth is available for other connections.”) Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”) Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:

No.	’740 Patent Claim 10	Cisco EtherChannel

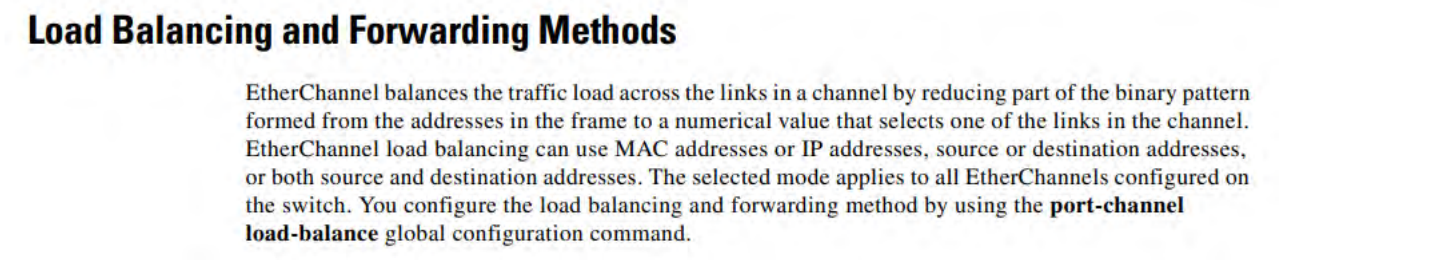
		
		TABLE I
		DISTRIBUTOR HASH FUNCTION
		`unsigned short hash(unsigned char *hdr, short lagSize)`
		`  {`
		`  short i;`
		`  unsigned short hash=178; // initialization value`
		`  for (i=0; i<4; i++)          // hdr is 4 bytes length`
		`    hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);`
		`  return (hash % lagSize)`
		`  }`
		”)
		
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		
		Bruckman at [0067] (“A similar problem may arise if there is a failure in a link in an aggregation group or in one of a number of line cards serving the aggregation group. In this case, to maintain the bandwidth allocation B made by CAC 44, each of the remaining links in the group must now carry, on average, B/(N-M) traffic, wherein M is the number of links in the group that are out of service. If only BIN has been allocated to each link, the remaining active links may not have sufficient bandwidth to continue to provide the bandwidth that has been guaranteed to the connections that they are required to carry. A similar problem arises with respect to loading of traces 52. For example, if there is a failure in LC2 or in one of links 30 in group 36 that connect to LC2, the trace connecting the multiplexer 50 in LCl will have to carry a substantially larger share of the bandwidth, or even all of the bandwidth, that is allocated to the connection in ques-tion.”)
		
		Bruckman at [0068] (“FIG. 3 is a flow chart that schematically illustrates a method for dealing with these problems of fluctuating bandwidth requirements, in accordance with an embodiment of the present invention. In order to provide sufficient bandwidth for failure protection, CAC 44 uses a safety margin based on a protection parameter P, which is assigned at a protection setting step 60. P represents the maximum number of links in the

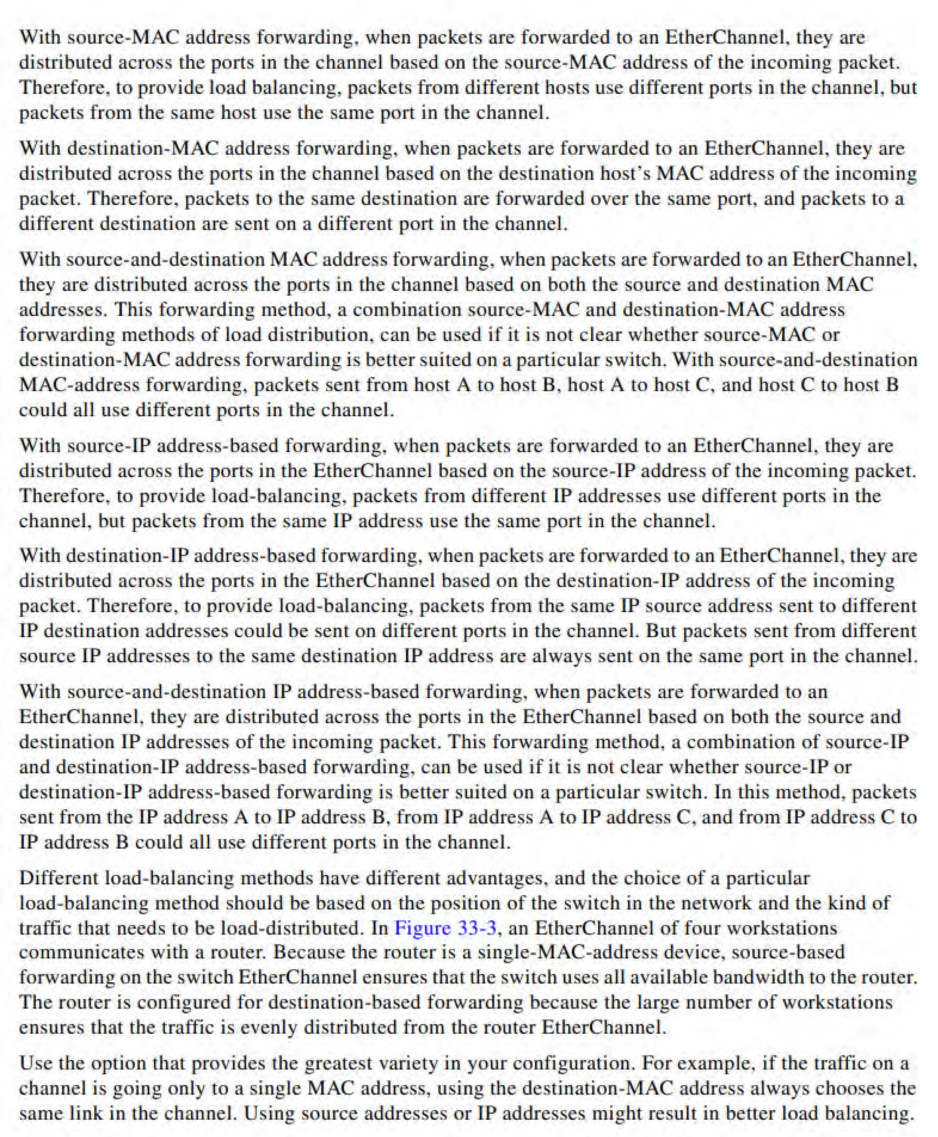
No.	’740 Patent Claim 10	Cisco EtherChannel
		group that can be out of service while still permitting the aggregation group to provide a given connection with the bandwidth that has been guaranteed to the connection. CAC 44 will then allocate at least B/(N-P) bandwidth to each link in the group, so that if P links fail, the group still provides total bandwidth of (N-P)*B/(N-P)= B. Setting P=l is equivalent to 1:N protection, so that the group will be unaffected by failure of a single link. In the example of group 36, shown in FIG. 2, setting P=2 will give both facility and equipment protection, i.e., the group will be unaffected not only by failure of a link, but also by failure of one of line cards 34. In the extreme case, in which P=N-1, CAC 44 will allocate the full bandwidth B on each link in the group.”) As another example, Singh discloses determining a ratio between the number of ingress and egress links and the number of links carrying data to the backplane and using a modulo to correspond to the channel’s link number. Singh at 9:30-43 (“The ratio between the number of line ingress links and the number of links carrying data to the backplane gives the backplane speedup for the system. In this example, there are 10 ingress links into the MS and 20 links (2 backplane channels) carrying that data to the backplane. This gives a backplane speedup of 2x. As another example, with 8 ingress links and 12 backplane links, there is a speedup of 1.5x. It should be noted that in addition to the backplane speedup, there is also an ingress/egress speedup. With 10 ingress links capable of carrying 2 Gbps each of raw data, this presents a 20 Gbps interface to the MS. An OC-192 only has approximately 10 Gbps worth of data. Taking into account cell overhead and cell quantization inefficiencies, there still remains excess capacity in the links.”) Singh at 11:29-38 (“FIG. 9 is a diagram illustrating link to channel assignments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a channel, whether it is a backplane channel or an ingress/egress channel. There is no logical relationship between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.”) Singh at 15:15-39 (“The number of crossbars that are required in a system is dependent on how many links are being used to create the backplane channels. There should be an even number of crossbars and they would be divided evenly across the switch cards. The following equation, for most cases, pro-vides the correct number of crossbars:

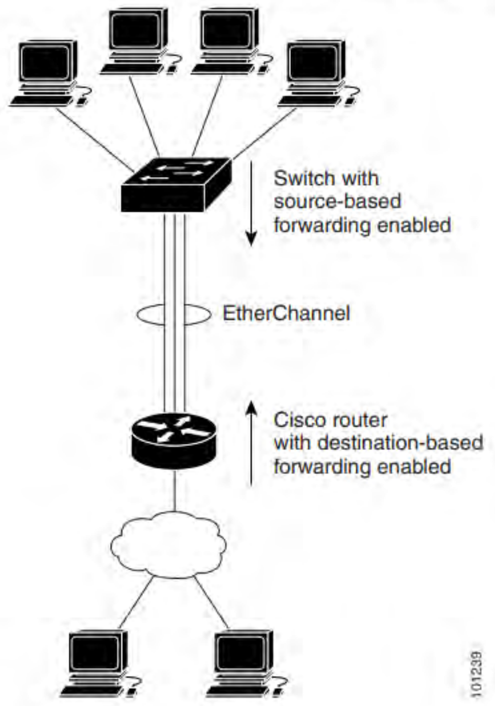
No.	’740 Patent Claim 10	Cisco EtherChannel
		# of Crossbars=(# links per ingress channelx# of ingress channels per portx# of port cardsx speedup)/32. For the 8x8 configuration, the # of crossbars should be multiplied by (4x# of iMS)/(# backplane channels per port card). The number of port cards should be rounded up to the nearest supported configuration, i.e. 8, 16, or 32. The speedup in the case of crossbars should be the fractional speedup that is desired. Example to determine the number of arbiters and cross-bars for the following system: 4 channel port cards (40 Gbps) 8 links per channel 16 port cards Speedup=l.5 # of arbiters=(4x2x2)/2=8 # of crossbars=(8x4x16xl.5)/32=24. This would give 3crossbars per arbiter.”) Singh at 16:28-44 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 16x16 and 32x32 is the organization of the switchplane. The port card remains the same. Backplane channels 1 and 2 are used for the backplane connectivity. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 16, correspond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”) Singh at 17:31-49 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 8x8 and 16x16 is the organi-zation of the switchplane. The port card remains the same. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Links 0-7 and 24-31 on the arbiters would not be used and would be powered off. Links 0-7 and 24-31 on the crossbars would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B

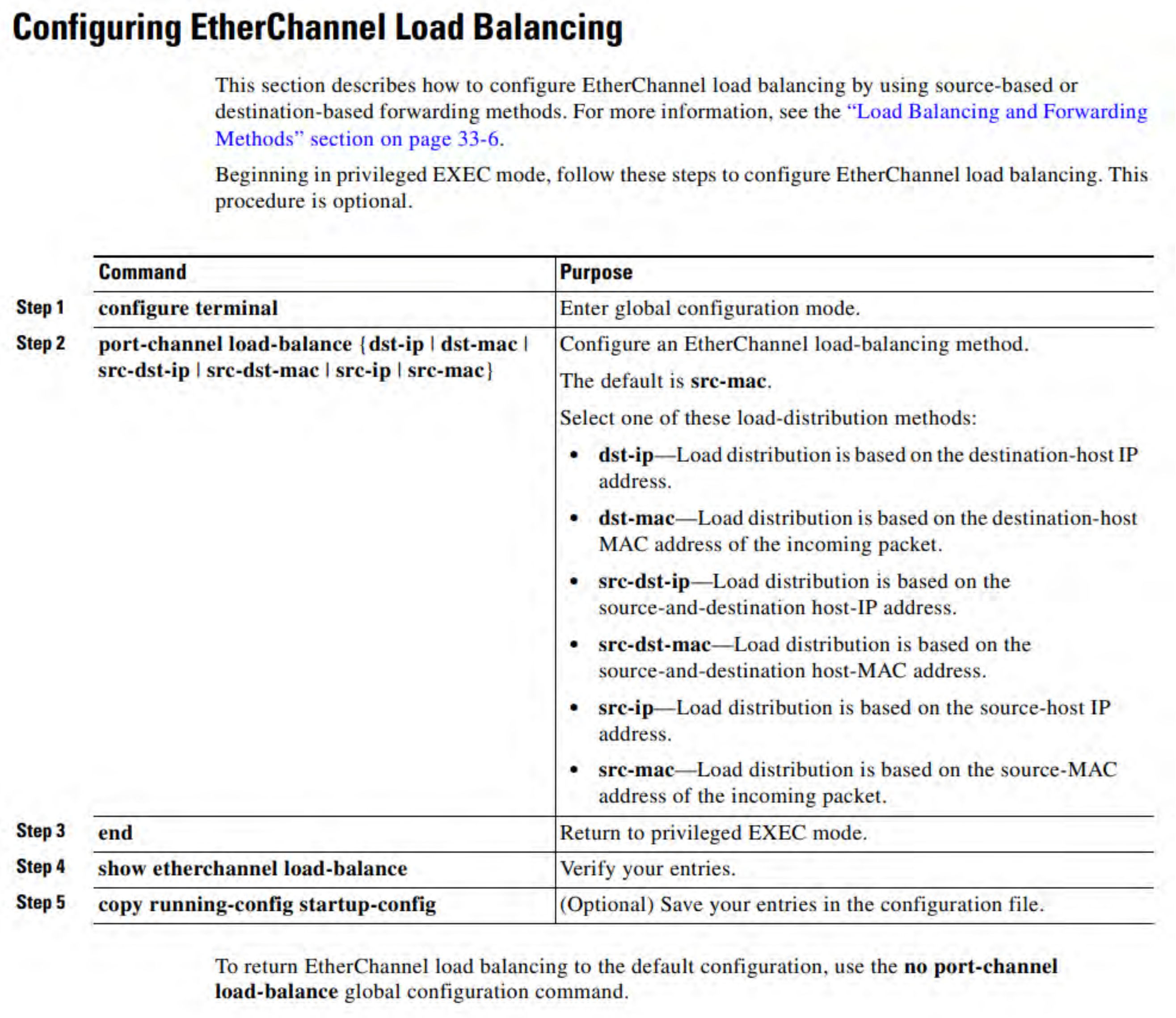
No.	’740 Patent Claim 10	Cisco EtherChannel
		on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Backplane channels 1 and 2 are used for the backplane connectivity. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 8, corre-spond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”)

No.	’740 Patent Claim 11	Cisco EtherChannel
11	The method according to claim 10, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a	Cisco EtherChannel System discloses the method according to claim 10, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo. For example, Cisco EtherChannel System discloses using the result of the division to select the forwarding path of the port and switch fabric interconnection channels over which to send the packet. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Cisco EtherChannel System is found to not meet this limitation, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo would have been obvious to a person having ordinary skill in the art, as explained below. *See supra* Claim 10 Catalyst 3560 Configuration Guide at 33-6 – 33-8

No.	’740 Patent Claim 11	Cisco EtherChannel
	binary representation of the modulo.	 **Load Balancing and Forwarding Methods** EtherChannel balances the traffic load across the links in a channel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel. EtherChannel load balancing can use MAC addresses or IP addresses, source or destination addresses, or both source and destination addresses. The selected mode applies to all EtherChannels configured on the switch. You configure the load balancing and forwarding method by using the **port-channel load-balance** global configuration command.

No.	’740 Patent Claim 11	Cisco EtherChannel
		 With source-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the source-MAC address of the incoming packet. Therefore, to provide load balancing, packets from different hosts use different ports in the channel, but packets from the same host use the same port in the channel. With destination-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the destination host’s MAC address of the incoming packet. Therefore, packets to the same destination are forwarded over the same port, and packets to a different destination are sent on a different port in the channel. With source-and-destination MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on both the source and destination MAC addresses. This forwarding method, a combination source-MAC and destination-MAC address forwarding methods of load distribution, can be used if it is not clear whether source-MAC or destination-MAC address forwarding is better suited on a particular switch. With source-and-destination MAC-address forwarding, packets sent from host A to host B, host A to host C, and host C to host B could all use different ports in the channel. With source-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the source-IP address of the incoming packet. Therefore, to provide load-balancing, packets from different IP addresses use different ports in the channel, but packets from the same IP address use the same port in the channel. With destination-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the destination-IP address of the incoming packet. Therefore, to provide load-balancing, packets from the same IP source address sent to different IP destination addresses could be sent on different ports in the channel. But packets sent from different source IP addresses to the same destination IP address are always sent on the same port in the channel. With source-and-destination IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on both the source and destination IP addresses of the incoming packet. This forwarding method, a combination of source-IP and destination-IP address-based forwarding, can be used if it is not clear whether source-IP or destination-IP address-based forwarding is better suited on a particular switch. In this method, packets sent from the IP address A to IP address B, from IP address A to IP address C, and from IP address C to IP address B could all use different ports in the channel. Different load-balancing methods have different advantages, and the choice of a particular load-balancing method should be based on the position of the switch in the network and the kind of traffic that needs to be load-distributed. In Figure 33-3, an EtherChannel of four workstations communicates with a router. Because the router is a single-MAC-address device, source-based forwarding on the switch EtherChannel ensures that the switch uses all available bandwidth to the router. The router is configured for destination-based forwarding because the large number of workstations ensures that the traffic is evenly distributed from the router EtherChannel. Use the option that provides the greatest variety in your configuration. For example, if the traffic on a channel is going only to a single MAC address, using the destination-MAC address always chooses the same link in the channel. Using source addresses or IP addresses might result in better load balancing.

No.	’740 Patent Claim 11	Cisco EtherChannel
		Figure 33-3 Load Distribution and Forwarding Methods  Switch with source-based forwarding enabled EtherChannel Cisco router with destination-based forwarding enabled 101239 Catalyst 3560 Configuration Guide at 33-16

No.	’740 Patent Claim 11	Cisco EtherChannel
		

Configuring EtherChannel Load Balancing

This section describes how to configure EtherChannel load balancing by using source-based or destination-based forwarding methods. For more information, see the “Load Balancing and Forwarding Methods” section on page 33-6.

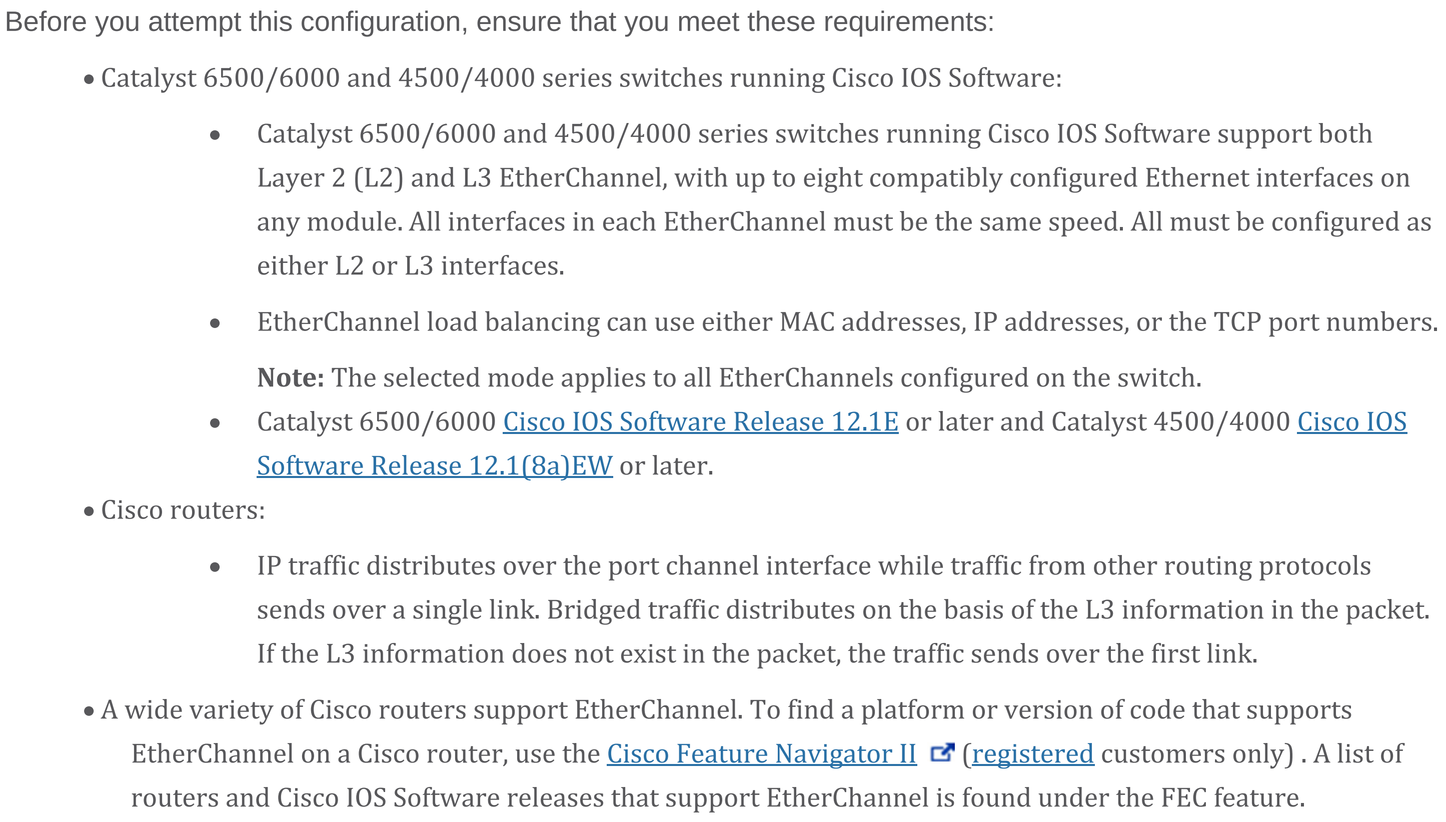
Beginning in privileged EXEC mode, follow these steps to configure EtherChannel load balancing. This procedure is optional.

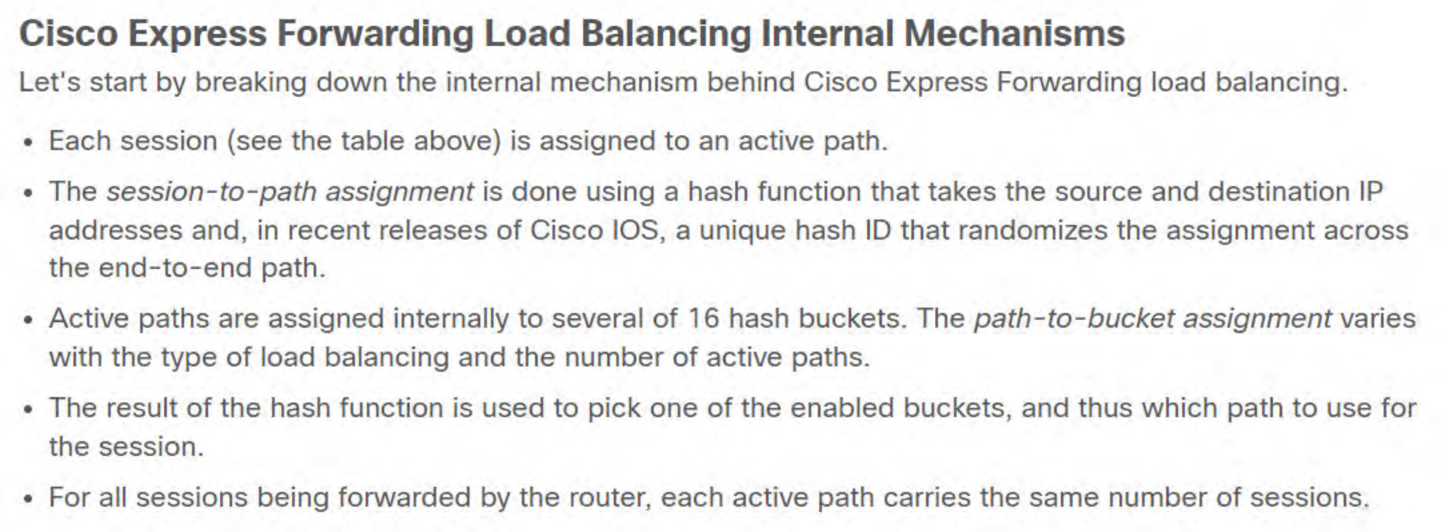
	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**port-channel load-balance** {**dst-ip** | **dst-mac** | **src-dst-ip** | **src-dst-mac** | **src-ip** | **src-mac**}	Configure an EtherChannel load-balancing method. The default is **src-mac**. Select one of these load-distribution methods: • **dst-ip**—Load distribution is based on the destination-host IP address. • **dst-mac**—Load distribution is based on the destination-host MAC address of the incoming packet. • **src-dst-ip**—Load distribution is based on the source-and-destination host-IP address. • **src-dst-mac**—Load distribution is based on the source-and-destination host-MAC address. • **src-ip**—Load distribution is based on the source-host IP address. • **src-mac**—Load distribution is based on the source-MAC address of the incoming packet.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show etherchannel load-balance**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return EtherChannel load balancing to the default configuration, use the **no port-channel load-balance** global configuration command.

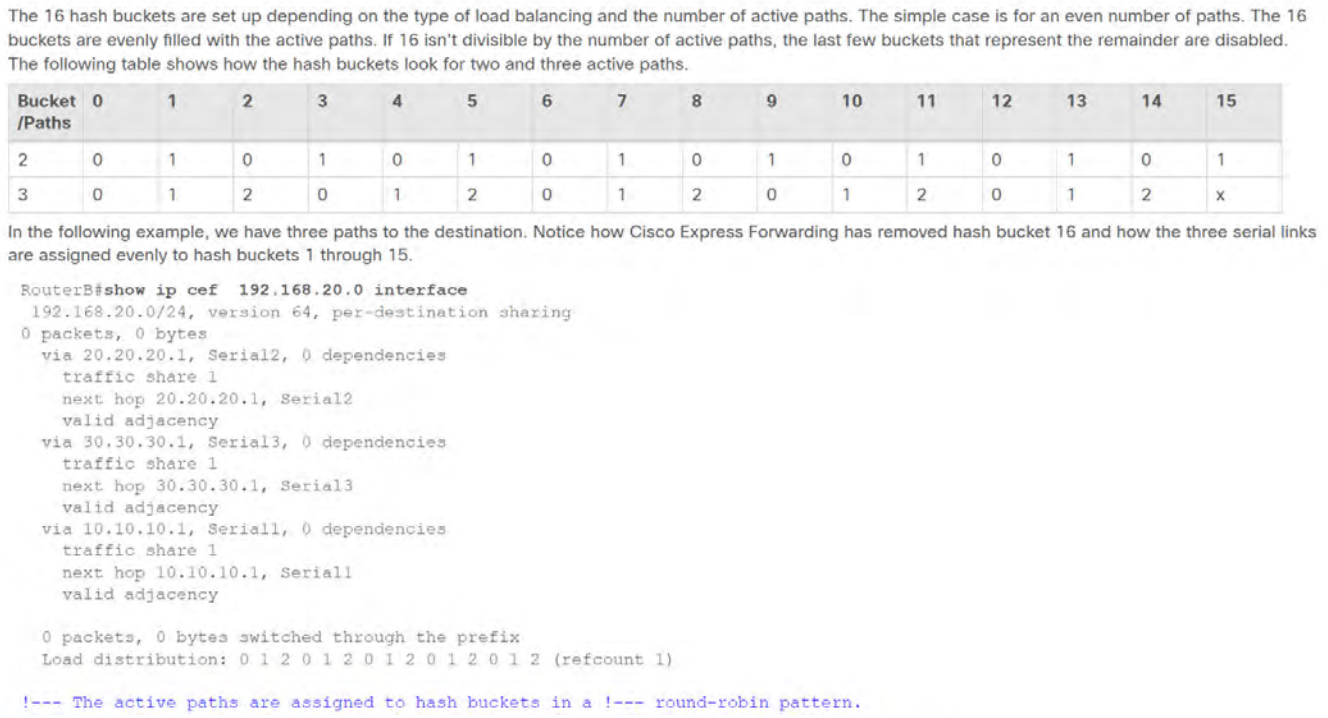
No.	’740 Patent Claim 11	Cisco EtherChannel
		Cisco EtherChannel at Layer 2 EtherChannel **Catalyst 2950** Catalyst 2950 switches only support 802.1Q trunking and do not support ISL trunking. Catalyst 2950 switches support DTP and PAgP dynamic trunking and channel negotiation with Cisco IOS Software Release 12.1 releases and static modes only with Cisco IOS Software Release 12.0 releases. EtherChannel load balancing can use either source-MAC or destination-MAC address forwarding. You can configure the load balancing method by issuing the **port-channel load-balance** global configuration command. These switches support up to eight switch ports per channel. **Catalyst 6500 That Runs Cisco IOS Software** Catalyst 6500 switches that run Cisco IOS Software support L2 (switchport) and Layer 3 (L3) (routed port) EtherChannel configurations. A Catalyst 6500/6000 series switch supports a maximum of 64 EtherChannels (256 with Cisco IOS Software Release 12.1(2)E and earlier). You can form an EtherChannel with up to eight compatibly configured LAN ports on any module in a Catalyst 6000 series switch, with the exception of Digital Feature Card (DFC)-equipped modules (such as WS-X6816 and so on) which currently allow an L2 channel only using ports on the same DFC module. However, an L3 channel can be configured across different DFC-equipped modules. This limitation has been removed in Catalyst 6500/6000 Cisco IOS Software Release 12.1(11b)EX and later. This document configures an L2 EtherChannel. The Catalyst 6500/6000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP addresses, or Layer 4 (L4) port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. Catalyst 6500/6000 switches support both ISL and 802.1Q trunking encapsulations and DTP. Detailed information on port capabilities is available by issuing the **show interface** *interface_id* **capabilities** command.

No.	’740 Patent Claim 11	Cisco EtherChannel
		Catalyst 4000 That Runs Cisco IOS Software Catalyst 4000 switches that run Cisco IOS Software (with Supervisor Engine III and IV) support L2 (switchport) and L3 (routed port) EtherChannel configurations. A Catalyst 4000 series switch supports a maximum of 64 EtherChannels. You can form an EtherChannel with up to eight compatibly configured Ethernet interfaces on any module, and across modules in a Catalyst 4000 series switch. All interfaces in each EtherChannel must be the same speed and must all be configured as either L2 or L3 interfaces. The Catalyst 4000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP address, or L4 port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. The Catalyst 4000 that runs Cisco IOS Software supports ISL and 802.1Q trunking encapsulations and DTP. ISL is not available on certain modules. For a complete list of such modules, refer to the *Understanding VLAN Trunks* section of Configuring Layer 2 Ethernet Interfaces. In a future software release, detailed information on port capabilities will be available by issuing the **show interface capabilities** command. Currently this command is not available. Configuring EtherChannel at 1 -2 **Introduction** This sample configuration demonstrates how to set up a Layer 3 (L3) EtherChannel, without VLAN trunking, between a Cisco router and a Cisco Catalyst 6500 switch running Cisco IOS® System Software. EtherChannel can be called Fast EtherChannel (FEC) or Gigabit EtherChannel (GEC); the term depends on the speed of the interfaces or ports you use to form the EtherChannel. In this example, two Fast Ethernet ports from a Cisco router and a Catalyst 6500 switch have been bundled into a FEC. Throughout this document, the terms FEC, GEC, port channel, channel, and port group all refer to EtherChannel.

No.	’740 Patent Claim 11	Cisco EtherChannel
		 Before you attempt this configuration, ensure that you meet these requirements: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software support both Layer 2 (L2) and L3 EtherChannel, with up to eight compatibly configured Ethernet interfaces on any module. All interfaces in each EtherChannel must be the same speed. All must be configured as either L2 or L3 interfaces. • EtherChannel load balancing can use either MAC addresses, IP addresses, or the TCP port numbers. **Note:** The selected mode applies to all EtherChannels configured on the switch. • Catalyst 6500/6000 Cisco IOS Software Release 12.1E or later and Catalyst 4500/4000 Cisco IOS Software Release 12.1(8a)EW or later. • Cisco routers: • IP traffic distributes over the port channel interface while traffic from other routing protocols sends over a single link. Bridged traffic distributes on the basis of the L3 information in the packet. If the L3 information does not exist in the packet, the traffic sends over the first link. • A wide variety of Cisco routers support EtherChannel. To find a platform or version of code that supports EtherChannel on a Cisco router, use the Cisco Feature Navigator II (registered customers only) . A list of routers and Cisco IOS Software releases that support EtherChannel is found under the FEC feature. Load Balancing at 7-9

No.	’740 Patent Claim 11	Cisco EtherChannel
		 **Cisco Express Forwarding Load Balancing Internal Mechanisms** Let's start by breaking down the internal mechanism behind Cisco Express Forwarding load balancing. • Each session (see the table above) is assigned to an active path. • The *session-to-path assignment* is done using a hash function that takes the source and destination IP addresses and, in recent releases of Cisco IOS, a unique hash ID that randomizes the assignment across the end-to-end path. • Active paths are assigned internally to several of 16 hash buckets. The *path-to-bucket assignment* varies with the type of load balancing and the number of active paths. • The result of the hash function is used to pick one of the enabled buckets, and thus which path to use for the session. • For all sessions being forwarded by the router, each active path carries the same number of sessions.

No.	’740 Patent Claim 11	Cisco EtherChannel



The 16 hash buckets are set up depending on the type of load balancing and the number of active paths. The simple case is for an even number of paths. The 16 buckets are evenly filled with the active paths. If 16 isn't divisible by the number of active paths, the last few buckets that represent the remainder are disabled. The following table shows how the hash buckets look for two and three active paths.

Bucket /Paths	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
3	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	x

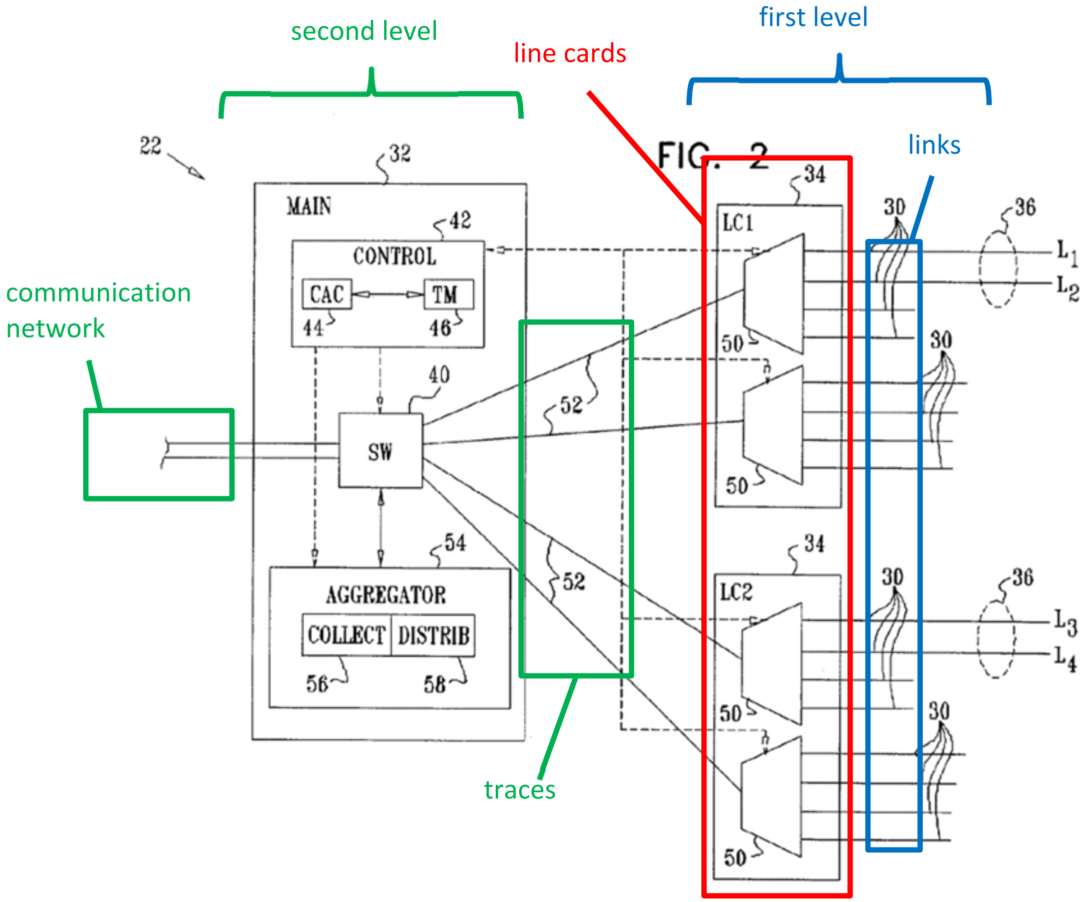
In the following example, we have three paths to the destination. Notice how Cisco Express Forwarding has removed hash bucket 16 and how the three serial links are assigned evenly to hash buckets 1 through 15.

```
RouterB#show ip cef  192.168.20.0 interface
 192.168.20.0/24, version 64, per-destination sharing
0 packets, 0 bytes
  via 20.20.20.1, Serial2, 0 dependencies
    traffic share 1
    next hop 20.20.20.1, Serial2
    valid adjacency
  via 30.30.30.1, Serial3, 0 dependencies
    traffic share 1
    next hop 30.30.30.1, Serial3
    valid adjacency
  via 10.10.10.1, Serial1, 0 dependencies
    traffic share 1
    next hop 10.10.10.1, Serial1
    valid adjacency

  0 packets, 0 bytes switched through the prefix
  Load distribution: 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 (refcount 1)

!--- The active paths are assigned to hash buckets in a !--- round-robin pattern.
```

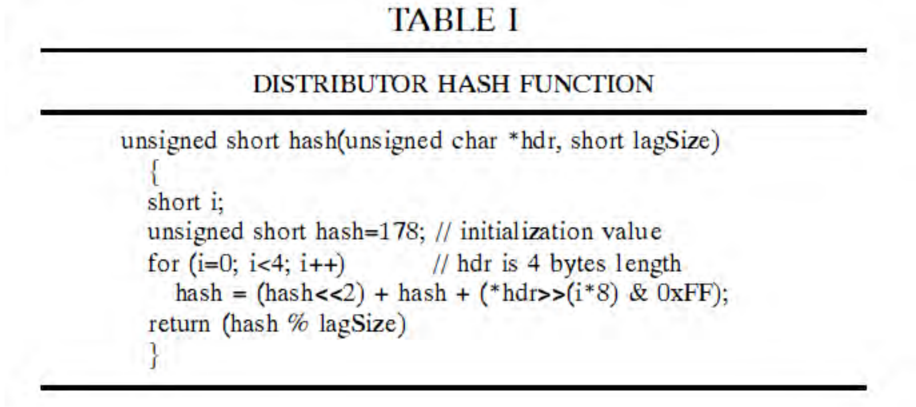
Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Cisco EtherChannel System in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 11 of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references.

For example, Bruckman discloses distributing data frames over physical links and traces based on a division operation in the hash function involving specific byte lengths of the frame information.

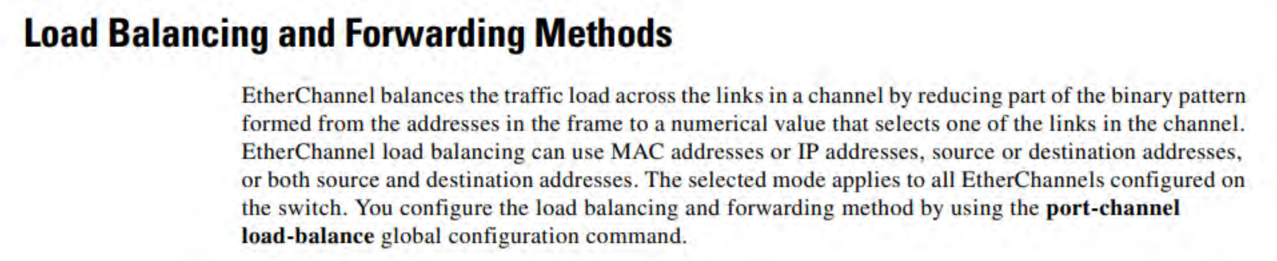
Bruckman at Figure 2 (annotated)

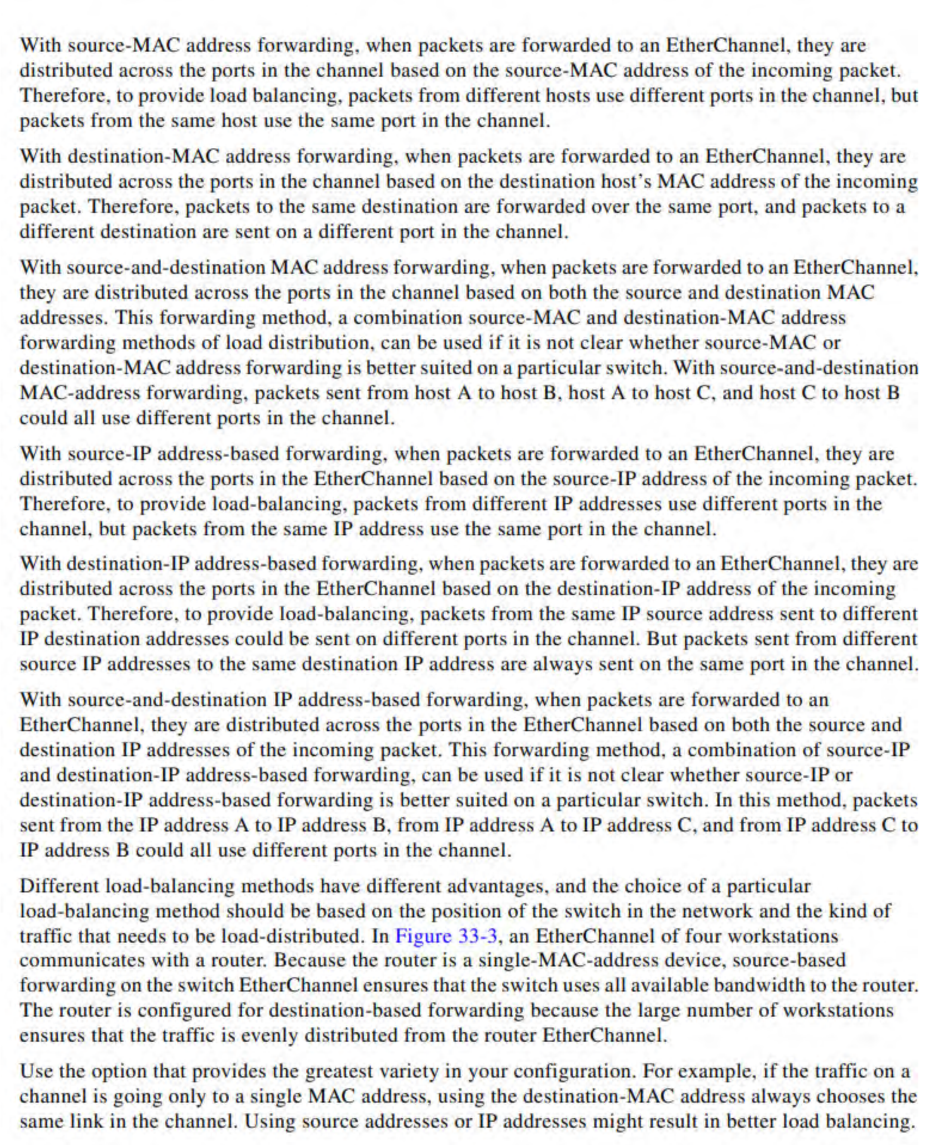
No.	’740 Patent Claim 11	Cisco EtherChannel
		 second level line cards first level links communication network traces 22 32 MAIN 42 CONTROL CAC TM 44 46 40 SW 52 52 54 AGGREGATOR COLLECT DISTRIB 56 58 34 LC1 50 50 34 LC2 50 50 30 30 30 30 36 36 L_1 L_2 L_3 L_4 FIG. 2 Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:

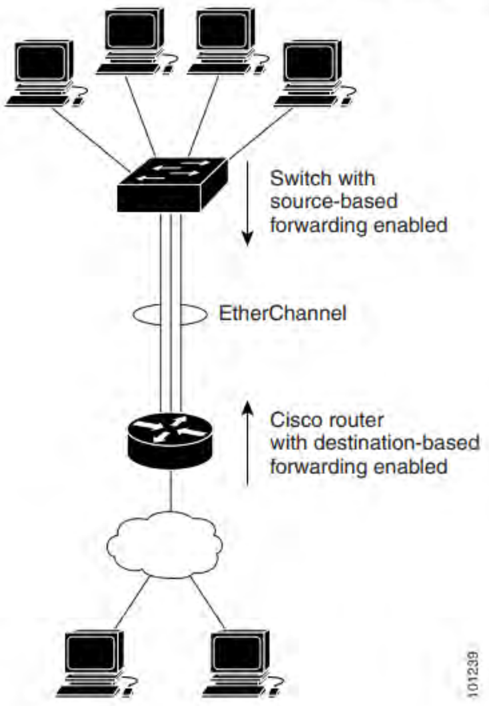
No.	’740 Patent Claim 11	Cisco EtherChannel

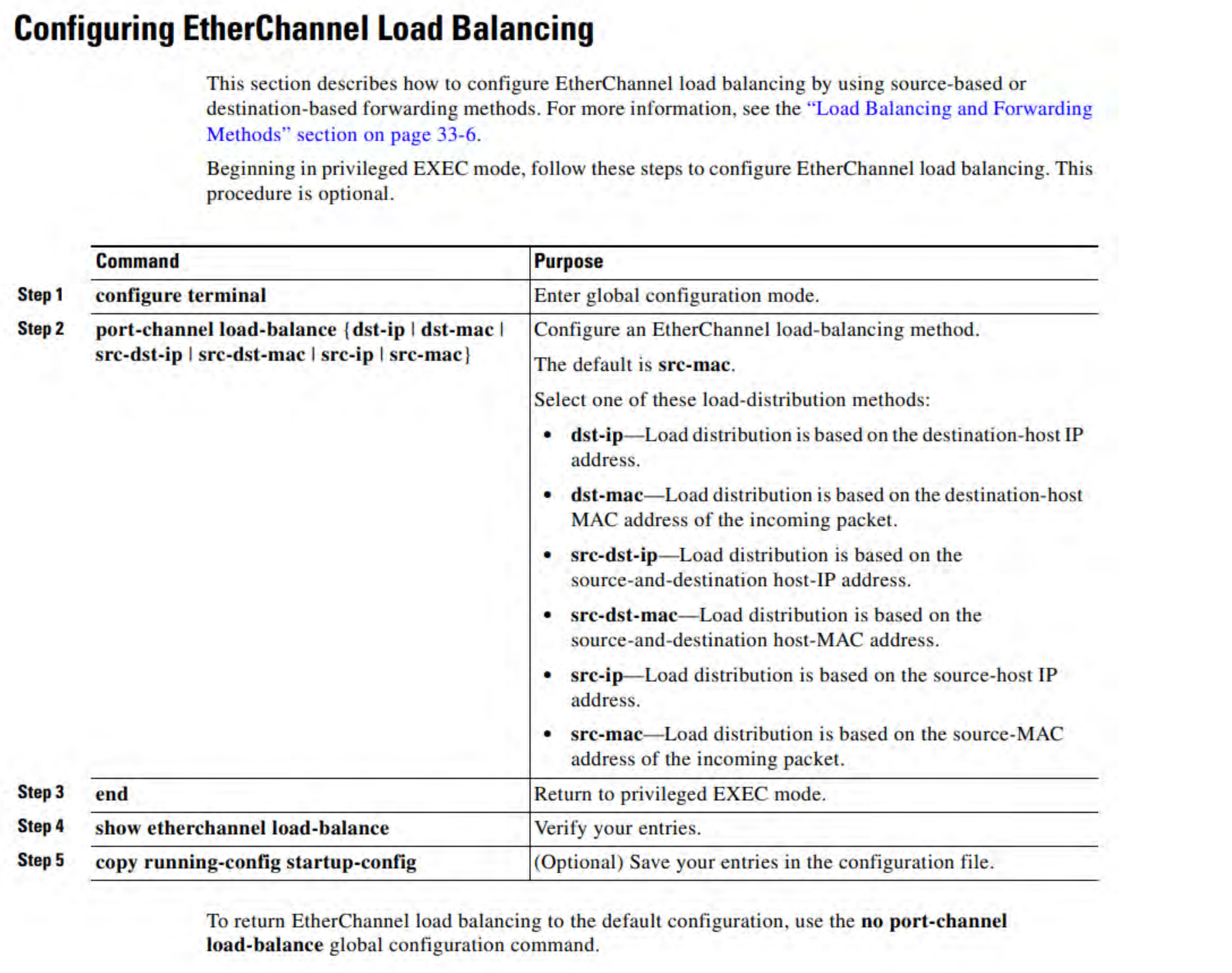
		
		TABLE I
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		”)
		
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		
		As another example, Solomon discloses using a subset of bits to encode for the selected physical port, involving specific byte lengths of the frame information.
		
		Solomon at [0054] (“Having selected a physical port, RSVP-TE proces-sor 30 of switch A now generates a suitable MPLS label, at a label generation step 64. The preceding node upstream of switch A will subsequently attach this MPLS label to all MPLS packets transmitted through tunnel 28 to switch A. The label is assigned, in conjunction with the mapping function of mapper 34, so as to ensure that all MPLS packets carrying this label are switched through the physical port that was selected for this tunnel at step 62. For this purpose, RSVP-TE processor 30 of switch A dedicates a sub-set of the bits of MPLS label 52 to encode the serial number of the selected physical port. For example, the four least-signifi-cant bits of MPLS label 52 may be used for encoding the selected port number. This configuration is suitable for representing LAG groups having up to 16 physical ports (N<16). The remaining bits of MPLS label 52 may be chosen at random or using any suitable method known in the art.”)

No.	’740 Patent Claim 11	Cisco EtherChannel
		Solomon at [0056] (“Mapper 34 of switch A maps the received packets belonging to tunnel 28 to the selected physical Ethernet port at a mapping step 70. For this purpose, mapper 34 extracts the MPLS label from each received packet and decodes the selected physical port number from the dedicated sub-set of bits, such as the four LSB, as described in step 64 above. The decoded value is used for mapping the packet to the selected physical port, which was allocated by the CAC processor at step 62 above. In the four-bit example described above, the mapping function may be written explicitly as: Selected port number=((MPLS label) and (0x0000F)), wherein "and" denotes the "bitwise and" operator.”)

No.	’740 Patent Claim 12	Cisco EtherChannel
12	The method according to claim 1, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a	Cisco EtherChannel System discloses the method according to claim 1, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port. For example, Cisco EtherChannel System discloses packet information including header fields, source and destination MAC addresses, source and destination IP addresses, port information, and other packet information. *See supra* Claim 1 Catalyst 3560 Configuration Guide at 33-6 – 33-8  **Load Balancing and Forwarding Methods** EtherChannel balances the traffic load across the links in a channel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel. EtherChannel load balancing can use MAC addresses or IP addresses, source or destination addresses, or both source and destination addresses. The selected mode applies to all EtherChannels configured on the switch. You configure the load balancing and forwarding method by using the **port-channel load-balance** global configuration command.

No.	’740 Patent Claim 12	Cisco EtherChannel
	destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port.	 With source-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the source-MAC address of the incoming packet. Therefore, to provide load balancing, packets from different hosts use different ports in the channel, but packets from the same host use the same port in the channel. With destination-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the destination host’s MAC address of the incoming packet. Therefore, packets to the same destination are forwarded over the same port, and packets to a different destination are sent on a different port in the channel. With source-and-destination MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on both the source and destination MAC addresses. This forwarding method, a combination source-MAC and destination-MAC address forwarding methods of load distribution, can be used if it is not clear whether source-MAC or destination-MAC address forwarding is better suited on a particular switch. With source-and-destination MAC-address forwarding, packets sent from host A to host B, host A to host C, and host C to host B could all use different ports in the channel. With source-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the source-IP address of the incoming packet. Therefore, to provide load-balancing, packets from different IP addresses use different ports in the channel, but packets from the same IP address use the same port in the channel. With destination-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the destination-IP address of the incoming packet. Therefore, to provide load-balancing, packets from the same IP source address sent to different IP destination addresses could be sent on different ports in the channel. But packets sent from different source IP addresses to the same destination IP address are always sent on the same port in the channel. With source-and-destination IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on both the source and destination IP addresses of the incoming packet. This forwarding method, a combination of source-IP and destination-IP address-based forwarding, can be used if it is not clear whether source-IP or destination-IP address-based forwarding is better suited on a particular switch. In this method, packets sent from the IP address A to IP address B, from IP address A to IP address C, and from IP address C to IP address B could all use different ports in the channel. Different load-balancing methods have different advantages, and the choice of a particular load-balancing method should be based on the position of the switch in the network and the kind of traffic that needs to be load-distributed. In Figure 33-3, an EtherChannel of four workstations communicates with a router. Because the router is a single-MAC-address device, source-based forwarding on the switch EtherChannel ensures that the switch uses all available bandwidth to the router. The router is configured for destination-based forwarding because the large number of workstations ensures that the traffic is evenly distributed from the router EtherChannel. Use the option that provides the greatest variety in your configuration. For example, if the traffic on a channel is going only to a single MAC address, using the destination-MAC address always chooses the same link in the channel. Using source addresses or IP addresses might result in better load balancing.

No.	’740 Patent Claim 12	Cisco EtherChannel
		Figure 33-3 Load Distribution and Forwarding Methods  Switch with source-based forwarding enabled EtherChannel Cisco router with destination-based forwarding enabled 101239 Catalyst 3560 Configuration Guide at 33-16

No.	’740 Patent Claim 12	Cisco EtherChannel
		

Configuring EtherChannel Load Balancing

This section describes how to configure EtherChannel load balancing by using source-based or destination-based forwarding methods. For more information, see the “Load Balancing and Forwarding Methods” section on page 33-6.

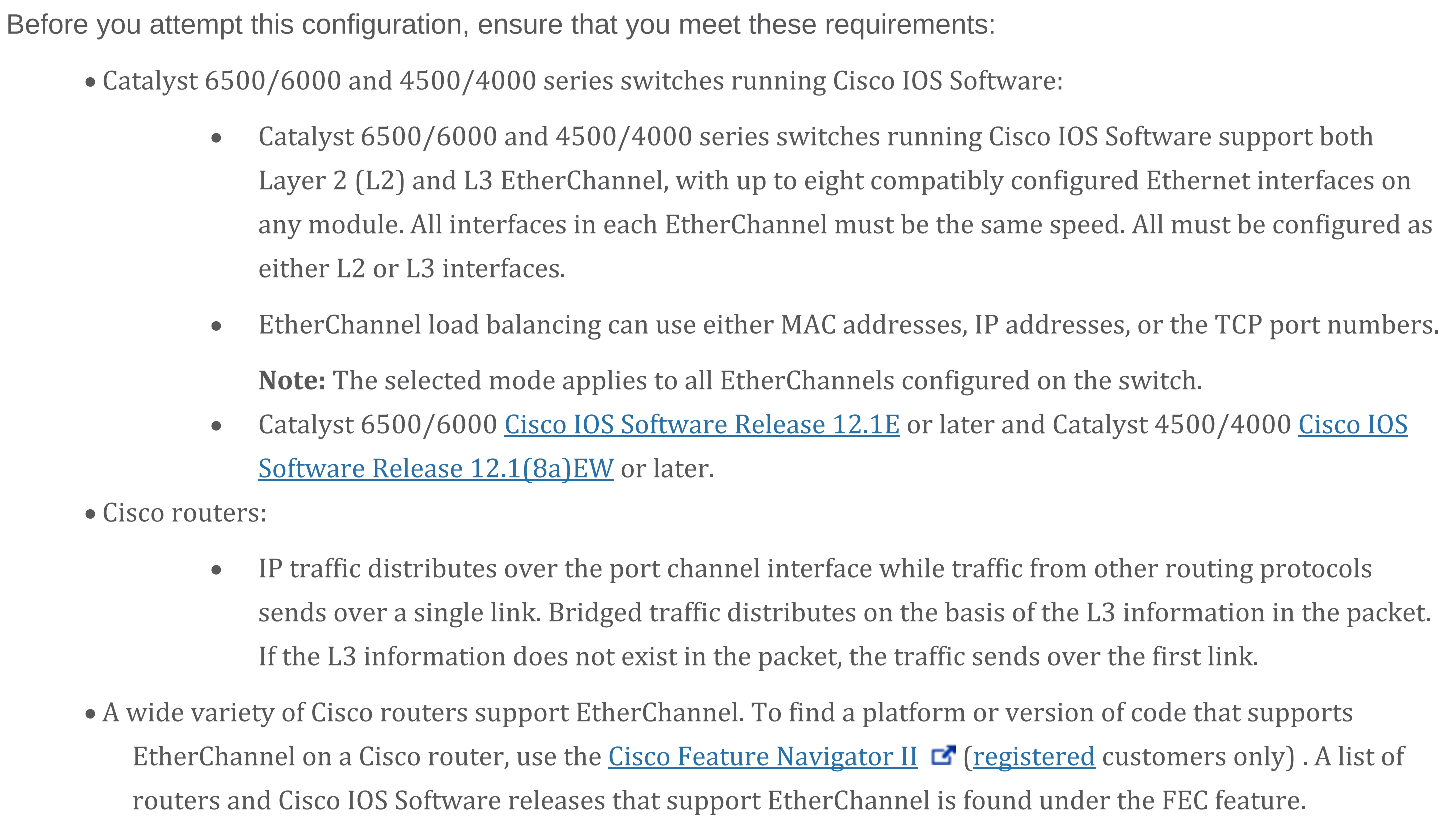
Beginning in privileged EXEC mode, follow these steps to configure EtherChannel load balancing. This procedure is optional.

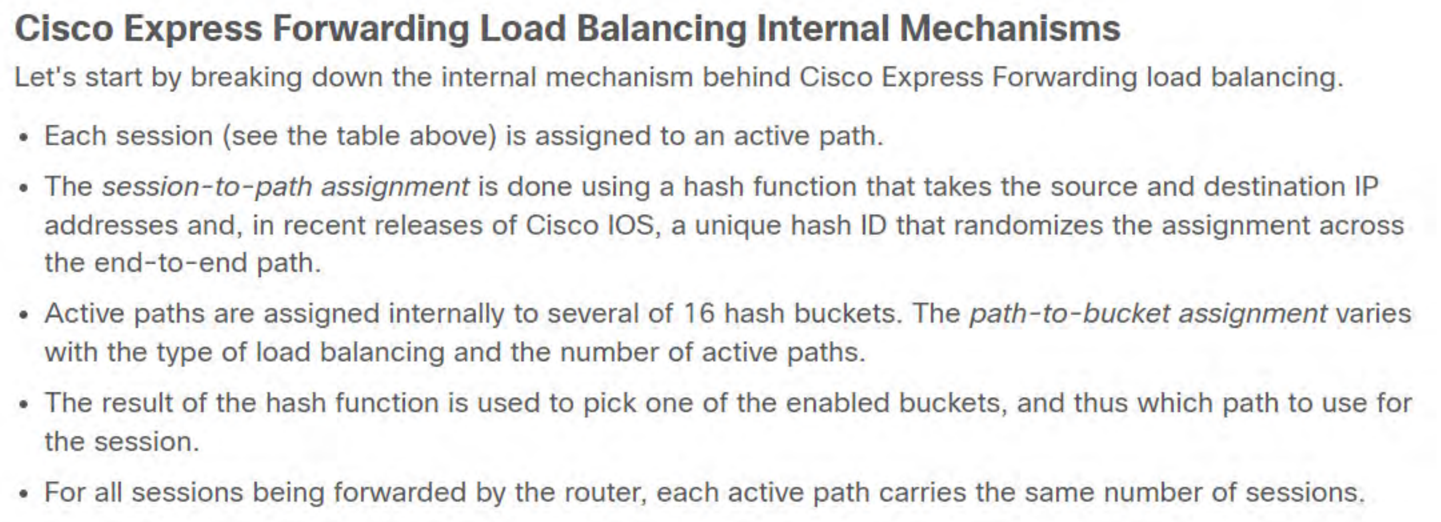
	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**port-channel load-balance** {**dst-ip** | **dst-mac** | **src-dst-ip** | **src-dst-mac** | **src-ip** | **src-mac**}	Configure an EtherChannel load-balancing method. The default is **src-mac**. Select one of these load-distribution methods: • **dst-ip**—Load distribution is based on the destination-host IP address. • **dst-mac**—Load distribution is based on the destination-host MAC address of the incoming packet. • **src-dst-ip**—Load distribution is based on the source-and-destination host-IP address. • **src-dst-mac**—Load distribution is based on the source-and-destination host-MAC address. • **src-ip**—Load distribution is based on the source-host IP address. • **src-mac**—Load distribution is based on the source-MAC address of the incoming packet.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show etherchannel load-balance**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

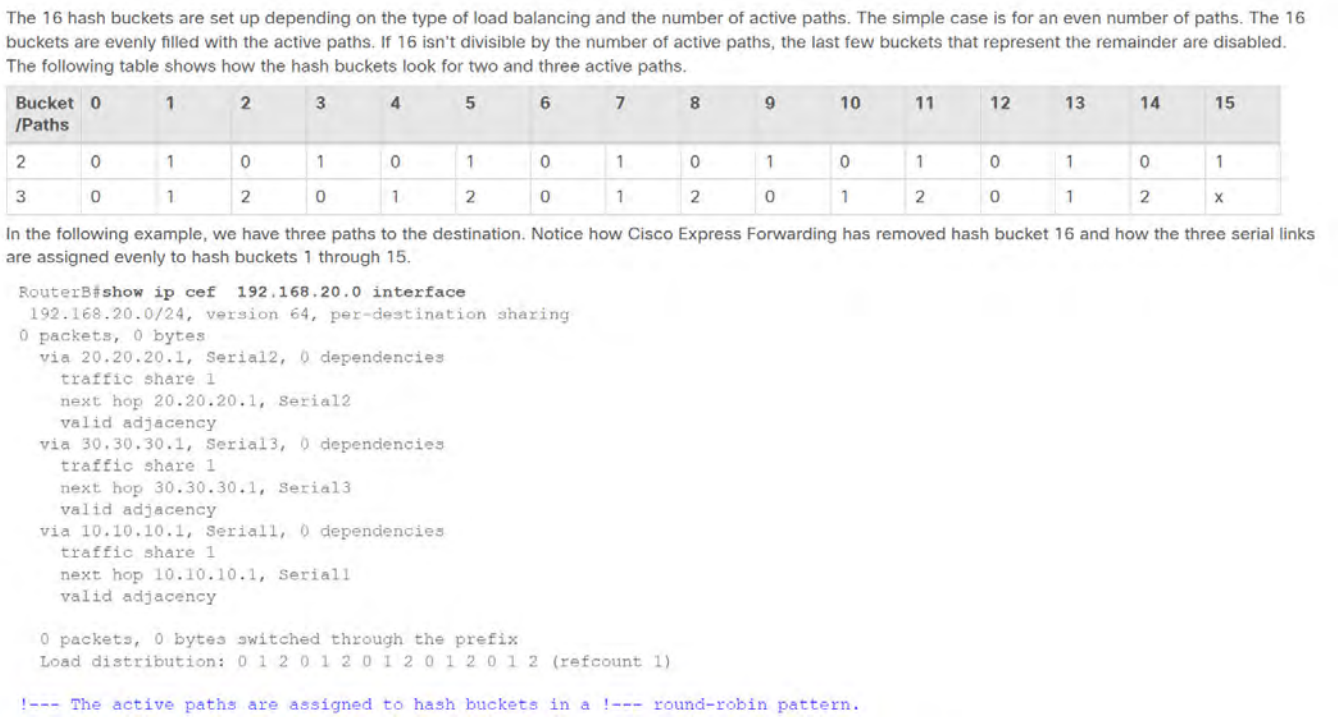
To return EtherChannel load balancing to the default configuration, use the **no port-channel load-balance** global configuration command.

No.	’740 Patent Claim 12	Cisco EtherChannel
		Layer 2 EtherChannel **Catalyst 2950** Catalyst 2950 switches only support 802.1Q trunking and do not support ISL trunking. Catalyst 2950 switches support DTP and PAgP dynamic trunking and channel negotiation with Cisco IOS Software Release 12.1 releases and static modes only with Cisco IOS Software Release 12.0 releases. EtherChannel load balancing can use either source-MAC or destination-MAC address forwarding. You can configure the load balancing method by issuing the **port-channel load-balance** global configuration command. These switches support up to eight switch ports per channel. **Catalyst 6500 That Runs Cisco IOS Software** Catalyst 6500 switches that run Cisco IOS Software support L2 (switchport) and Layer 3 (L3) (routed port) EtherChannel configurations. A Catalyst 6500/6000 series switch supports a maximum of 64 EtherChannels (256 with Cisco IOS Software Release 12.1(2)E and earlier). You can form an EtherChannel with up to eight compatibly configured LAN ports on any module in a Catalyst 6000 series switch, with the exception of Digital Feature Card (DFC)-equipped modules (such as WS-X6816 and so on) which currently allow an L2 channel only using ports on the same DFC module. However, an L3 channel can be configured across different DFC-equipped modules. This limitation has been removed in Catalyst 6500/6000 Cisco IOS Software Release 12.1(11b)EX and later. This document configures an L2 EtherChannel. The Catalyst 6500/6000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP addresses, or Layer 4 (L4) port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. Catalyst 6500/6000 switches support both ISL and 802.1Q trunking encapsulations and DTP. Detailed information on port capabilities is available by issuing the **show interface** *interface_id* **capabilities** command.

No.	’740 Patent Claim 12	Cisco EtherChannel
		Catalyst 4000 That Runs Cisco IOS Software Catalyst 4000 switches that run Cisco IOS Software (with Supervisor Engine III and IV) support L2 (switchport) and L3 (routed port) EtherChannel configurations. A Catalyst 4000 series switch supports a maximum of 64 EtherChannels. You can form an EtherChannel with up to eight compatibly configured Ethernet interfaces on any module, and across modules in a Catalyst 4000 series switch. All interfaces in each EtherChannel must be the same speed and must all be configured as either L2 or L3 interfaces. The Catalyst 4000 that runs Cisco IOS Software allows you to configure EtherChannel load balancing to use MAC addresses, IP address, or L4 port information in any source, destination, and source-destination combination by issuing the **port-channel load-balance** global configuration command. The default is to use a hash function between source and destination IP addresses. The Catalyst 4000 that runs Cisco IOS Software supports ISL and 802.1Q trunking encapsulations and DTP. ISL is not available on certain modules. For a complete list of such modules, refer to the *Understanding VLAN Trunks* section of Configuring Layer 2 Ethernet Interfaces. In a future software release, detailed information on port capabilities will be available by issuing the **show interface capabilities** command. Currently this command is not available. Configuring EtherChannel at 1 -2 **Introduction** This sample configuration demonstrates how to set up a Layer 3 (L3) EtherChannel, without VLAN trunking, between a Cisco router and a Cisco Catalyst 6500 switch running Cisco IOS® System Software. EtherChannel can be called Fast EtherChannel (FEC) or Gigabit EtherChannel (GEC); the term depends on the speed of the interfaces or ports you use to form the EtherChannel. In this example, two Fast Ethernet ports from a Cisco router and a Catalyst 6500 switch have been bundled into a FEC. Throughout this document, the terms FEC, GEC, port channel, channel, and port group all refer to EtherChannel.

No.	’740 Patent Claim 12	Cisco EtherChannel
		 Before you attempt this configuration, ensure that you meet these requirements: - Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software: - Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software support both Layer 2 (L2) and L3 EtherChannel, with up to eight compatibly configured Ethernet interfaces on any module. All interfaces in each EtherChannel must be the same speed. All must be configured as either L2 or L3 interfaces. - EtherChannel load balancing can use either MAC addresses, IP addresses, or the TCP port numbers. **Note:** The selected mode applies to all EtherChannels configured on the switch. - Catalyst 6500/6000 Cisco IOS Software Release 12.1E or later and Catalyst 4500/4000 Cisco IOS Software Release 12.1(8a)EW or later. - Cisco routers: - IP traffic distributes over the port channel interface while traffic from other routing protocols sends over a single link. Bridged traffic distributes on the basis of the L3 information in the packet. If the L3 information does not exist in the packet, the traffic sends over the first link. - A wide variety of Cisco routers support EtherChannel. To find a platform or version of code that supports EtherChannel on a Cisco router, use the Cisco Feature Navigator II (registered customers only) . A list of routers and Cisco IOS Software releases that support EtherChannel is found under the FEC feature. Load Balancing at 7-9

No.	’740 Patent Claim 12	Cisco EtherChannel
		 **Cisco Express Forwarding Load Balancing Internal Mechanisms** Let's start by breaking down the internal mechanism behind Cisco Express Forwarding load balancing. • Each session (see the table above) is assigned to an active path. • The *session-to-path assignment* is done using a hash function that takes the source and destination IP addresses and, in recent releases of Cisco IOS, a unique hash ID that randomizes the assignment across the end-to-end path. • Active paths are assigned internally to several of 16 hash buckets. The *path-to-bucket assignment* varies with the type of load balancing and the number of active paths. • The result of the hash function is used to pick one of the enabled buckets, and thus which path to use for the session. • For all sessions being forwarded by the router, each active path carries the same number of sessions.

<table>
<tr><th>No.</th><th>’740 Patent Claim 12</th><th>Cisco EtherChannel</th></tr>
<tr><td></td><td></td><td>


The 16 hash buckets are set up depending on the type of load balancing and the number of active paths. The simple case is for an even number of paths. The 16 buckets are evenly filled with the active paths. If 16 isn't divisible by the number of active paths, the last few buckets that represent the remainder are disabled. The following table shows how the hash buckets look for two and three active paths.

Bucket /Paths	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
3	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	x

In the following example, we have three paths to the destination. Notice how Cisco Express Forwarding has removed hash bucket 16 and how the three serial links are assigned evenly to hash buckets 1 through 15.

```
RouterB#show ip cef  192.168.20.0 interface
 192.168.20.0/24, version 64, per-destination sharing
0 packets, 0 bytes
  via 20.20.20.1, Serial2, 0 dependencies
    traffic share 1
    next hop 20.20.20.1, Serial2
    valid adjacency
  via 30.30.30.1, Serial3, 0 dependencies
    traffic share 1
    next hop 30.30.30.1, Serial3
    valid adjacency
  via 10.10.10.1, Serial1, 0 dependencies
    traffic share 1
    next hop 10.10.10.1, Serial1
    valid adjacency

  0 packets, 0 bytes switched through the prefix
  Load distribution: 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 (refcount 1)

!--- The active paths are assigned to hash buckets in a !--- round-robin pattern.
```
</td></tr>
</table>

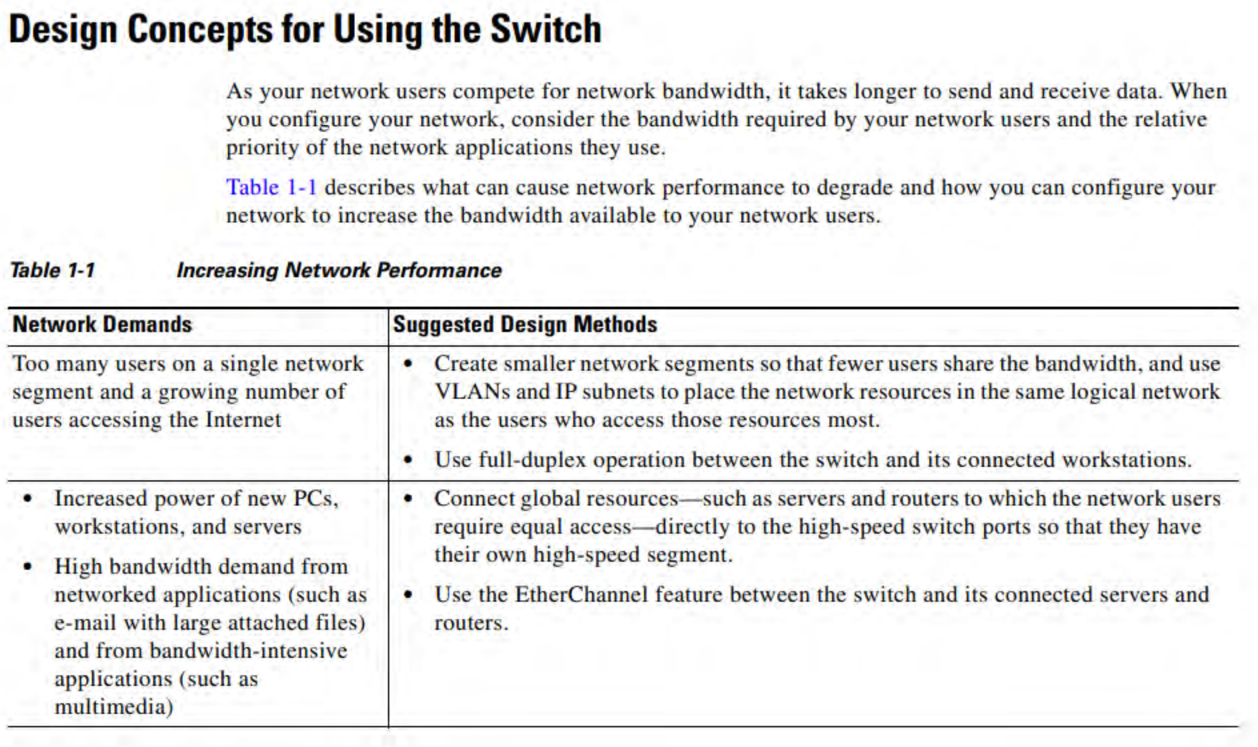
No.	’740 Patent Claim 13	Cisco EtherChannel
13[preamble]	A method for communication, comprising:	Cisco EtherChannel System discloses a method for communication. *See supra at* 1[preamble].
13[a]	coupling a network node to one or more interface	Cisco EtherChannel System discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].

No.	’740 Patent Claim 13	Cisco EtherChannel
	modules using a first group of first physical links arranged in parallel;	
13[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Cisco EtherChannel System discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra at* 1[c].
13[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	Cisco EtherChannel System discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra at* 1[e].
13[d]	selecting, in a single computation based on at least one of the frame attributes, a first physical	Cisco EtherChannel System discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. *See supra at* 1[f].

No.	’740 Patent Claim 13	Cisco EtherChannel
	link out of the first group and a second physical link out of the second group; and	
13[e]	sending the data frame over the selected first and second physical links,	Cisco EtherChannel System discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].
13[f]	coupling the network node to the one or more interface modules and	Cisco EtherChannel System discloses coupling the network node to the one or more interface modules. *See supra at* 1[a].
13[g]	coupling each of the one or more interface modules to the communication network comprising	Cisco EtherChannel System discloses coupling each of the one or more interface modules to the communication network. *See supra at* 1[c].
13[h]	specifying bandwidth requirements comprising at least one of a committed information rate	Cisco EtherChannel System discloses specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node. For example, Cisco EtherChannel System discloses system bandwidth considerations, including total available system bandwidth, involved in connecting interface modules or line cards to the switch fabric and larger network. A person of ordinary skill in the art would understand that these

No.	’740 Patent Claim 13	Cisco EtherChannel
	(CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node, and	considerations could include at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Cisco EtherChannel System is found to not meet this limitation, coupling each of the one or more interface modules to the communication network comprising specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node would have been obvious to a person having ordinary skill in the art, as explained below. Cisco Catalyst 6500 Data Sheet at 2 (“Software features such as Network-Based Application Recognition (NBAR) enhance network management and control of bandwidth utilization.”) Cisco Catalyst 6500 Data Sheet at 13 (“**Switch Fabric Modules** Designed to support distributed forwarding for interface modules with that capability, the Cisco Catalyst 6500 Series Switch Fabric Module (SFM or SFM2), in combination with the Cisco Catalyst 6000 Multilayer Switch Feature Card (MSFC2) and Cisco distributed forwarding cards (DFCs) on interface modules, increases available system bandwidth from 32 to 256 Gbps. The SFM or SFM2 supports the Cisco Catalyst 6500 CEF256 and dCEF256 interface modules. Designed to support new interface modules with 720-Gbps forwarding capabilities, the switch fabric onboard the Cisco Catalyst 6500 Series Supervisor Engine 720 increases available bandwidth to 720 Gbps and provides packet-forwarding rates up to 400 mpps. By using automatic sensing and negotiation, the switch fabric is fully interoperable with the 8- and 16-Gbps switch-fabric interconnections used by the CEF256 and dCEF256 interface modules. When a CEF256 or dCEF256 interface module is detected, the switch fabric will automatically connect those modules by offering 8 to 16 Gbps of bandwidth to each module, as applicable.”) Catalyst 3560 Configuration Guide at 1-14

No.	’740 Patent Claim 13	Cisco EtherChannel



Design Concepts for Using the Switch

As your network users compete for network bandwidth, it takes longer to send and receive data. When you configure your network, consider the bandwidth required by your network users and the relative priority of the network applications they use.

Table 1-1 describes what can cause network performance to degrade and how you can configure your network to increase the bandwidth available to your network users.

Table 1-1 Increasing Network Performance

Network Demands	Suggested Design Methods
Too many users on a single network segment and a growing number of users accessing the Internet	• Create smaller network segments so that fewer users share the bandwidth, and use VLANs and IP subnets to place the network resources in the same logical network as the users who access those resources most. • Use full-duplex operation between the switch and its connected workstations.
• Increased power of new PCs, workstations, and servers • High bandwidth demand from networked applications (such as e-mail with large attached files) and from bandwidth-intensive applications (such as multimedia)	• Connect global resources—such as servers and routers to which the network users require equal access—directly to the high-speed switch ports so that they have their own high-speed segment. • Use the EtherChannel feature between the switch and its connected servers and routers.

Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Cisco EtherChannel System in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 13[h] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references.

For example, Bruckman discloses specifying certain committed, excess, and guaranteed bandwidths, including CIR, EIR, and PIR, respectively.

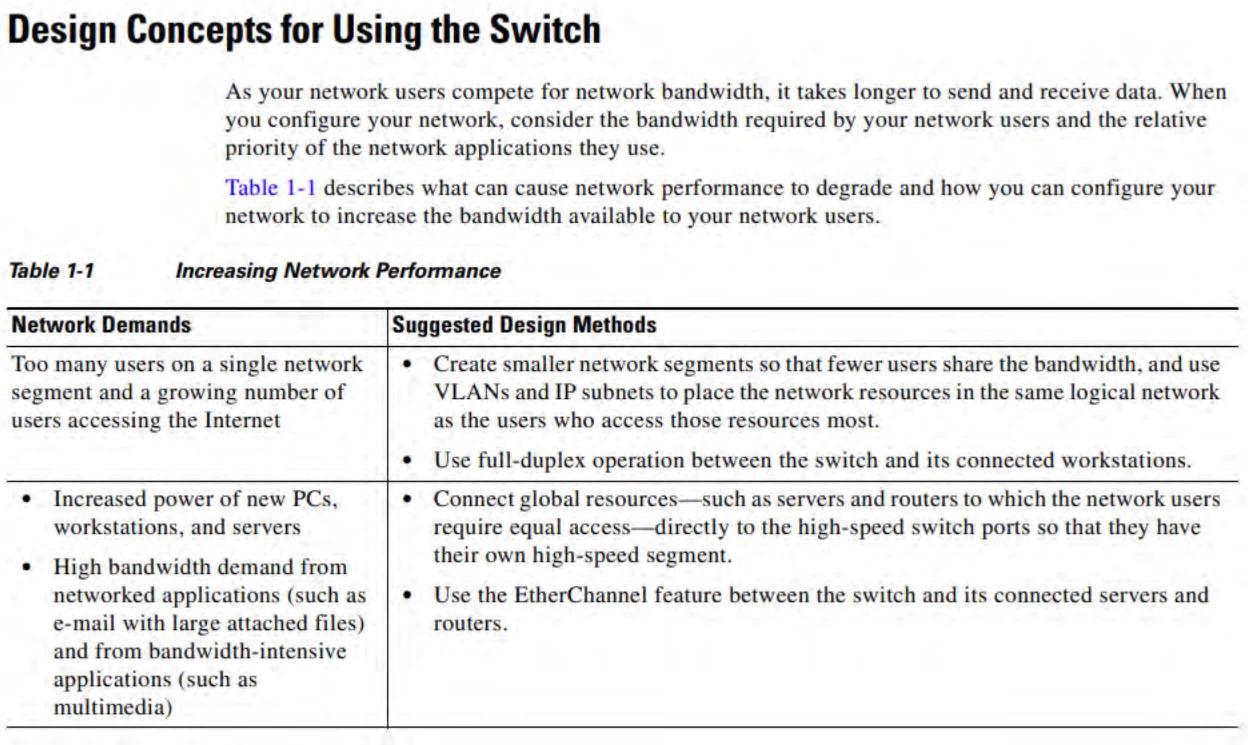
No.	’740 Patent Claim 13	Cisco EtherChannel
		Bruckman at [0013] (“Service level agreements between network service providers and customers commonly specify a certain com-mitted bandwidth, or committed information rate (CIR), which the service provider guarantees to provide to the customer at all times, regardless of bandwidth stress on the network. Additionally or alternatively, the agreement may specify an excess bandwidth, which is available to the customer when network traffic permits. The excess band-width is typically used by customers for lower-priority services, which do not require committed bandwidth. The network service provider may guarantee the customer a certain minimum excess bandwidth, or excess information rate (EIR), in order to avoid starvation of such services in case of bandwidth stress. In general, the bandwidth guaran-teed by a service provider, referred to as the peak informa-tion rate (PIR), may include either CIR, or EIR, or both CIR and EIR (in which case PIR=CIR+EIR). The term "guaran-teed bandwidth," as used in the context of the present patent application and in the claims, includes all these types of guaranteed bandwidth.”) As another example, Solomon discloses a service property of a guaranteed bandwidth, sometimes denoted as CIR-Committed Information Rate and PIR-Peak Information Rate. Solomon at [0023] (“In another embodiment, establishing the path includes receiving an indication of a requested service property of the flow, and selecting the port includes assign-ing the port to the flow so as to comply with the requested service property. In a disclosed embodiment, the requested service property includes at least one of a guaranteed bandwidth, a peak bandwidth and a class-of-service. Addi-tionally or alternatively, assigning the port includes selecting the port having a maximum available bandwidth out of the plurality of aggregated ports. Further additionally or alter-natively, assigning the port includes selecting the port hav-ing a minimum available bandwidth out of the plurality of aggregated ports, which is still greater than or equal to the guaranteed bandwidth.”) Solomon at [0050] (“The method of FIG. 3 begins when the preceding node asks to establish a part of tunnel 28 (comprising one or more hops) for sending MPLS packets to MPLS/LAG switch 26 A. The preceding node requests and then receives the MPLS label, which it will subsequently attach to all packets that are sent to MPLS/LAG switch 26 labeledA. The preceding node sends downstream an RSVP-TE PATH mes-sage augmented with a LABEL_REQUEST object, as defined by RSVP-

No.	’740 Patent Claim 13	Cisco EtherChannel
		TE, to MPLS/LAG switch A, at a label requesting step 60. The PATH message typically comprises information regarding service properties that are requested for tunnel 28. The service properties may comprise a guar-anteed bandwidth (sometimes denoted CIR-Committed Information Rate) and a peak bandwidth (sometimes denoted PIR-Peak Information Rate), as well as a requested CoS (Class of Service-a measure of packet priority).”)
13[i]	allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements.	Cisco EtherChannel System discloses allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements. For examples, Cisco EtherChannel System discloses load balancing and distributing packets in response to bandwidth considerations. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Cisco EtherChannel System is found to not meet this limitation, allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements would have been obvious to a person having ordinary skill in the art, as explained below. Cisco Catalyst 6500 Data Sheet at 2 (“Software features such as Network-Based Application Recognition (NBAR) enhance network management and control of bandwidth utilization.”) Cisco Catalyst 6500 Data Sheet at 13 (“**Switch Fabric Modules** Designed to support distributed forwarding for interface modules with that capability, the Cisco Catalyst 6500 Series Switch Fabric Module (SFM or SFM2), in combination with the Cisco Catalyst 6000 Multilayer Switch Feature Card (MSFC2) and Cisco distributed forwarding cards (DFCs) on interface modules, increases available system bandwidth from 32 to 256 Gbps. The SFM or SFM2 supports the Cisco Catalyst 6500 CEF256 and dCEF256 interface modules. Designed to support new interface modules with 720-Gbps forwarding capabilities, the switch fabric onboard the Cisco Catalyst 6500 Series Supervisor Engine 720 increases available bandwidth to 720 Gbps and provides packet-forwarding rates up to 400 mpps. By using automatic sensing and negotiation, the switch fabric is fully interoperable with the 8- and 16-Gbps switch-fabric

<table>
<tr><th>No.</th><th>’740 Patent Claim 13</th><th>Cisco EtherChannel</th></tr>
<tr><td></td><td></td><td>

interconnections used by the CEF256 and dCEF256 interface modules. When a CEF256 or dCEF256 interface module is detected, the switch fabric will automatically connect those modules by offering 8 to 16 Gbps of bandwidth to each module, as applicable.”)

Catalyst 3560 Configuration Guide at 1-14



Design Concepts for Using the Switch

As your network users compete for network bandwidth, it takes longer to send and receive data. When you configure your network, consider the bandwidth required by your network users and the relative priority of the network applications they use.

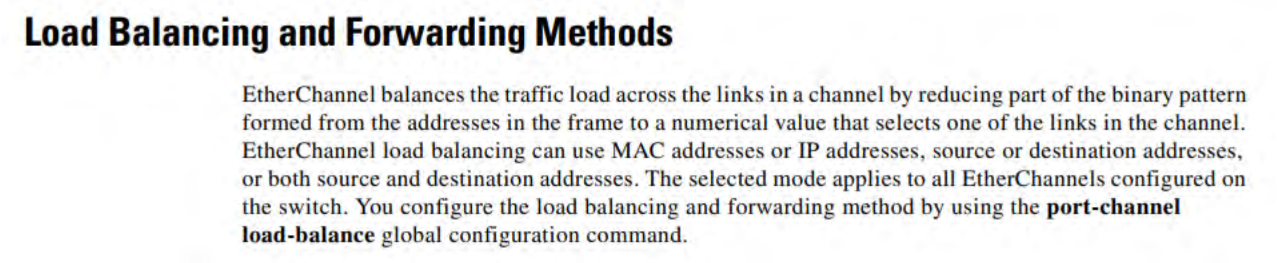
Table 1-1 describes what can cause network performance to degrade and how you can configure your network to increase the bandwidth available to your network users.

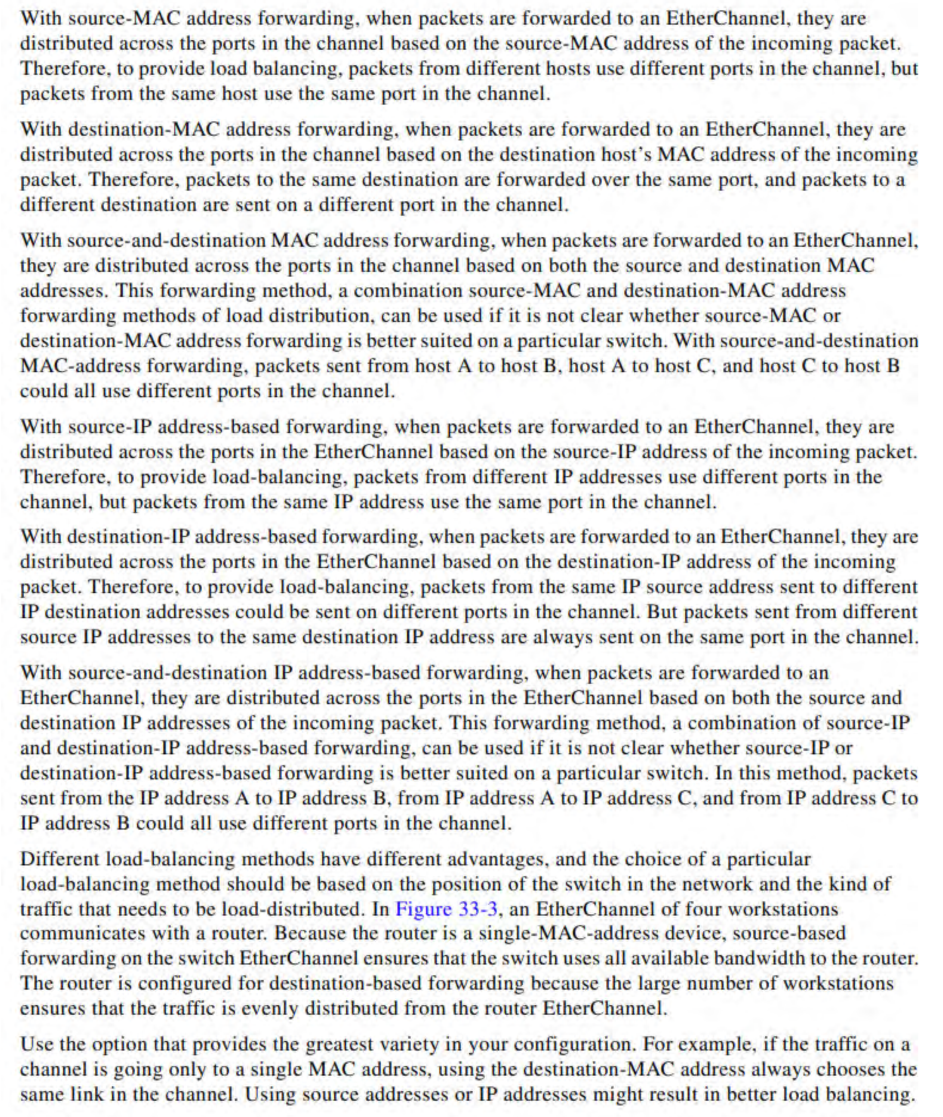
Table 1-1 Increasing Network Performance

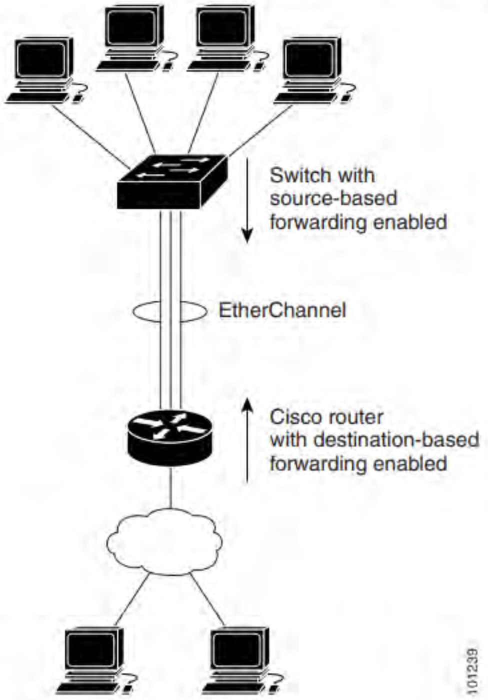
<table>
<tr><th>Network Demands</th><th>Suggested Design Methods</th></tr>
<tr><td>Too many users on a single network segment and a growing number of users accessing the Internet</td><td>• Create smaller network segments so that fewer users share the bandwidth, and use VLANs and IP subnets to place the network resources in the same logical network as the users who access those resources most.
• Use full-duplex operation between the switch and its connected workstations.</td></tr>
<tr><td>• Increased power of new PCs, workstations, and servers
• High bandwidth demand from networked applications (such as e-mail with large attached files) and from bandwidth-intensive applications (such as multimedia)</td><td>• Connect global resources—such as servers and routers to which the network users require equal access—directly to the high-speed switch ports so that they have their own high-speed segment.
• Use the EtherChannel feature between the switch and its connected servers and routers.</td></tr>
</table>

Catalyst 3560 Configuration Guide at 33-6 – 33-8

</td></tr>
</table>

No.	’740 Patent Claim 13	Cisco EtherChannel
		 **Load Balancing and Forwarding Methods** EtherChannel balances the traffic load across the links in a channel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel. EtherChannel load balancing can use MAC addresses or IP addresses, source or destination addresses, or both source and destination addresses. The selected mode applies to all EtherChannels configured on the switch. You configure the load balancing and forwarding method by using the **port-channel load-balance** global configuration command.

No.	’740 Patent Claim 13	Cisco EtherChannel
		 With source-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the source-MAC address of the incoming packet. Therefore, to provide load balancing, packets from different hosts use different ports in the channel, but packets from the same host use the same port in the channel. With destination-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the destination host’s MAC address of the incoming packet. Therefore, packets to the same destination are forwarded over the same port, and packets to a different destination are sent on a different port in the channel. With source-and-destination MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on both the source and destination MAC addresses. This forwarding method, a combination source-MAC and destination-MAC address forwarding methods of load distribution, can be used if it is not clear whether source-MAC or destination-MAC address forwarding is better suited on a particular switch. With source-and-destination MAC-address forwarding, packets sent from host A to host B, host A to host C, and host C to host B could all use different ports in the channel. With source-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the source-IP address of the incoming packet. Therefore, to provide load-balancing, packets from different IP addresses use different ports in the channel, but packets from the same IP address use the same port in the channel. With destination-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the destination-IP address of the incoming packet. Therefore, to provide load-balancing, packets from the same IP source address sent to different IP destination addresses could be sent on different ports in the channel. But packets sent from different source IP addresses to the same destination IP address are always sent on the same port in the channel. With source-and-destination IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on both the source and destination IP addresses of the incoming packet. This forwarding method, a combination of source-IP and destination-IP address-based forwarding, can be used if it is not clear whether source-IP or destination-IP address-based forwarding is better suited on a particular switch. In this method, packets sent from the IP address A to IP address B, from IP address A to IP address C, and from IP address C to IP address B could all use different ports in the channel. Different load-balancing methods have different advantages, and the choice of a particular load-balancing method should be based on the position of the switch in the network and the kind of traffic that needs to be load-distributed. In Figure 33-3, an EtherChannel of four workstations communicates with a router. Because the router is a single-MAC-address device, source-based forwarding on the switch EtherChannel ensures that the switch uses all available bandwidth to the router. The router is configured for destination-based forwarding because the large number of workstations ensures that the traffic is evenly distributed from the router EtherChannel. Use the option that provides the greatest variety in your configuration. For example, if the traffic on a channel is going only to a single MAC address, using the destination-MAC address always chooses the same link in the channel. Using source addresses or IP addresses might result in better load balancing.

No.	’740 Patent Claim 13	Cisco EtherChannel
		Figure 33-3 Load Distribution and Forwarding Methods  Switch with source-based forwarding enabled EtherChannel Cisco router with destination-based forwarding enabled 101239 Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Cisco EtherChannel System in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 13[i] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For examples, Bruckman discloses allocating bandwidth over the physical links and traces based on bandwidth requirements, margins, and fluctuations.

No.	’740 Patent Claim 13	Cisco EtherChannel
		Bruckman at [0013] (“Service level agreements between network service providers and customers commonly specify a certain com-mitted bandwidth, or committed information rate (CIR), which the service provider guarantees to provide to the customer at all times, regardless of bandwidth stress on the network. Additionally or alternatively, the agreement may specify an excess bandwidth, which is available to the customer when network traffic permits. The excess band-width is typically used by customers for lower-priority services, which do not require committed bandwidth. The network service provider may guarantee the customer a certain minimum excess bandwidth, or excess information rate (EIR), in order to avoid starvation of such services in case of bandwidth stress. In general, the bandwidth guaran-teed by a service provider, referred to as the peak informa-tion rate (PIR), may include either CIR, or EIR, or both CIR and EIR (in which case PIR=CIR+EIR). The term "guaran-teed bandwidth," as used in the context of the present patent application and in the claims, includes all these types of guaranteed bandwidth.”) As another example, Solomon discloses selecting the path to transmit the flow to comply with the service properties of the flow. Solomon at [0023] (“In another embodiment, establishing the path includes receiving an indication of a requested service property of the flow, and selecting the port includes assign-ing the port to the flow so as to comply with the requested service property. In a disclosed embodiment, the requested service property includes at least one of a guaranteed bandwidth, a peak bandwidth and a class-of-service. Addi-tionally or alternatively, assigning the port includes selecting the port having a maximum available bandwidth out of the plurality of aggregated ports. Further additionally or alter-natively, assigning the port includes selecting the port hav-ing a minimum available bandwidth out of the plurality of aggregated ports, which is still greater than or equal to the guaranteed bandwidth.”) Solomon at [0050] (“The method of FIG. 3 begins when the preceding node asks to establish a part of tunnel 28 (comprising one or more hops) for sending MPLS packets to MPLS/LAG switch 26 A. The preceding node requests and then receives the MPLS label, which it will subsequently attach to all packets that are sent to MPLS/LAG switch 26 labeledA. The preceding node sends downstream

No.	’740 Patent Claim 13	Cisco EtherChannel
		an RSVP-TE PATH mes-sage augmented with a LABEL_REQUEST object, as defined by RSVP-TE, to MPLS/LAG switch A, at a label requesting step 60. The PATH message typically comprises information regarding service properties that are requested for tunnel 28. The service properties may comprise a guar-anteed bandwidth (sometimes denoted CIR-Committed Information Rate) and a peak bandwidth (sometimes denoted PIR-Peak Information Rate), as well as a requested CoS (Class of Service-a measure of packet priority).”)

No.	’740 Patent Claim 14	Cisco EtherChannel
14[preamble]	A method for connecting user ports to a communication network, comprising:	Cisco EtherChannel System discloses a method for connecting user ports to a communication network. For example, Cisco EtherChannel System discloses connecting Ethernet ports to a network. Catalyst 3500 Installation Guide at Table 1-1

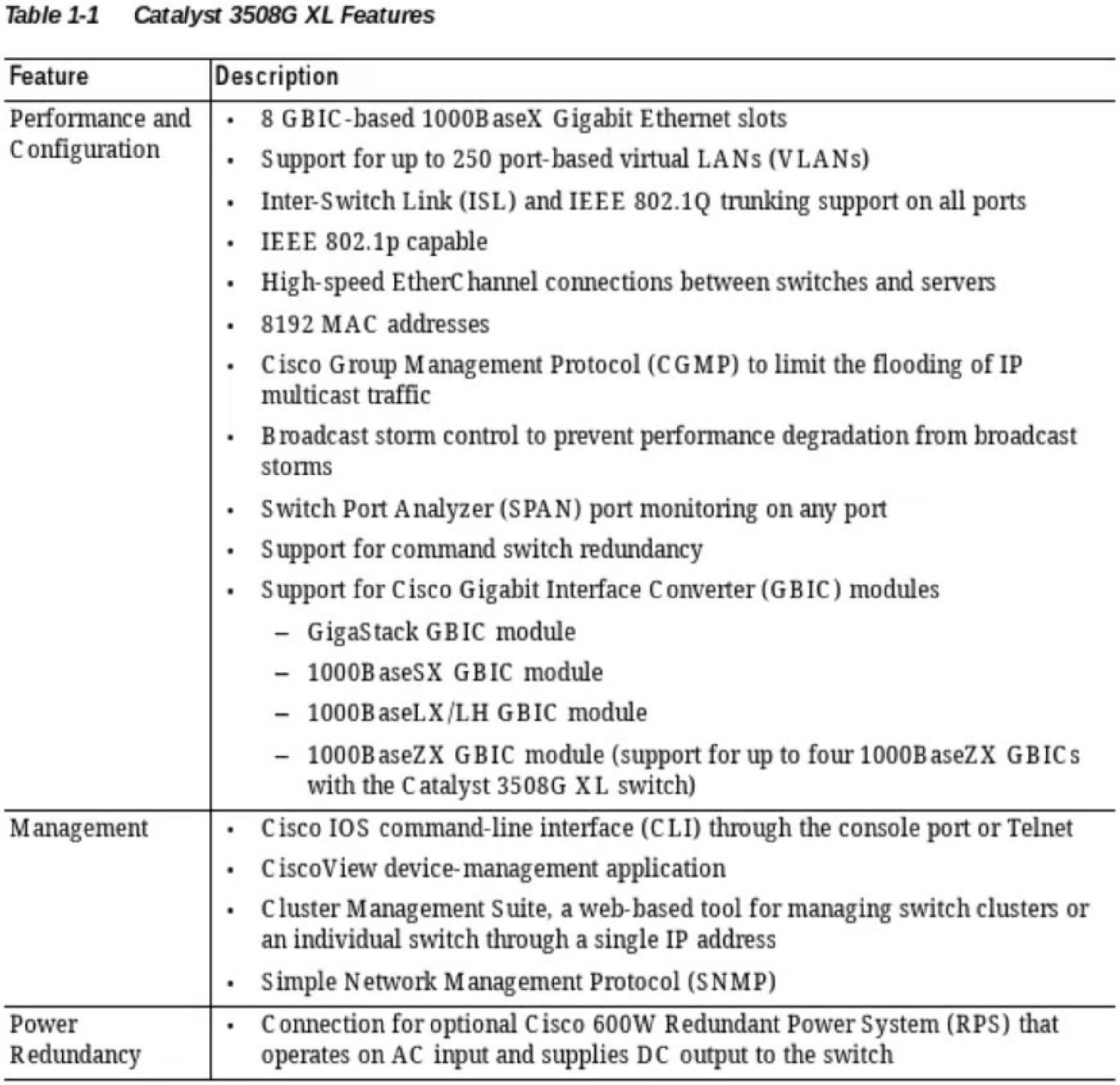
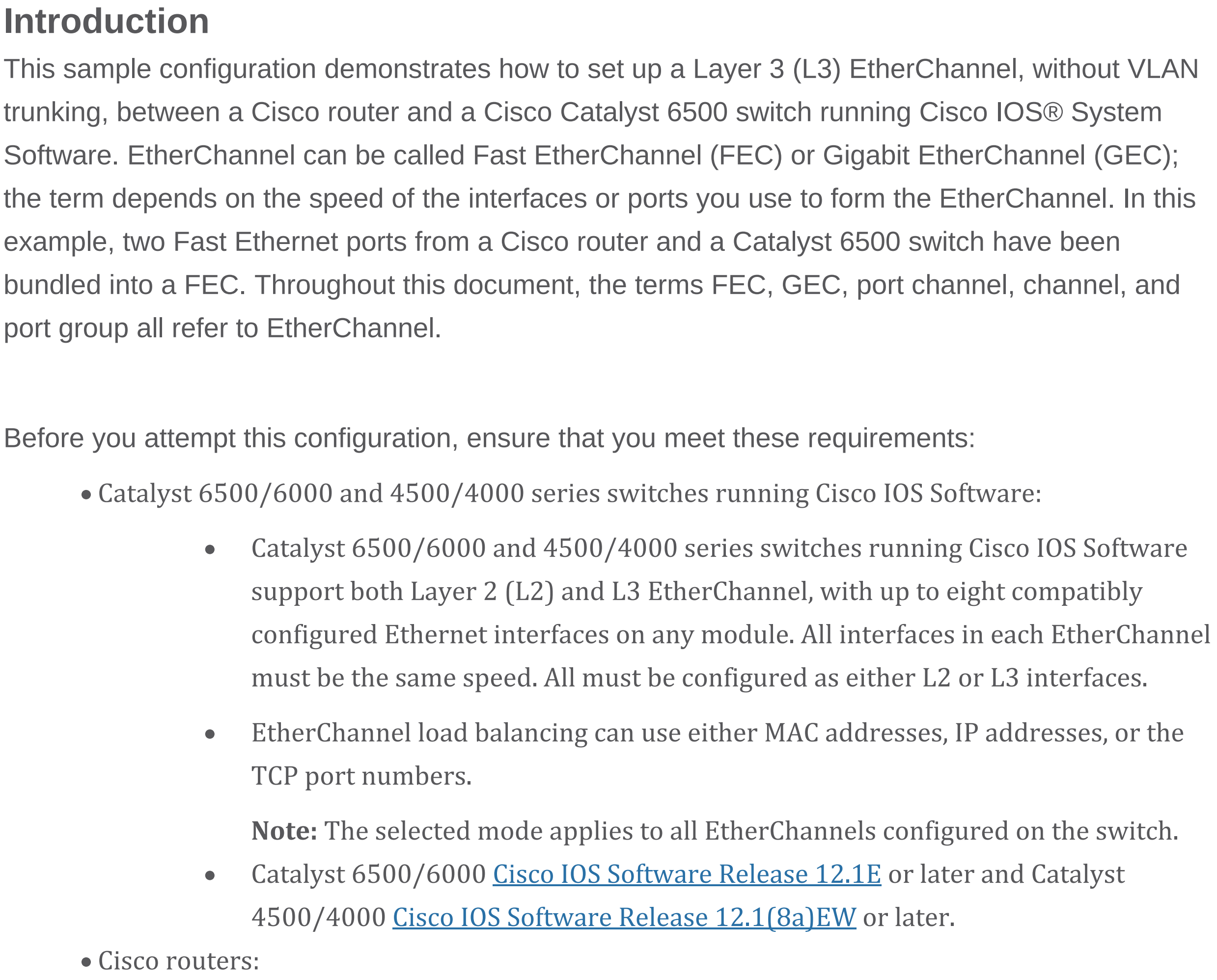
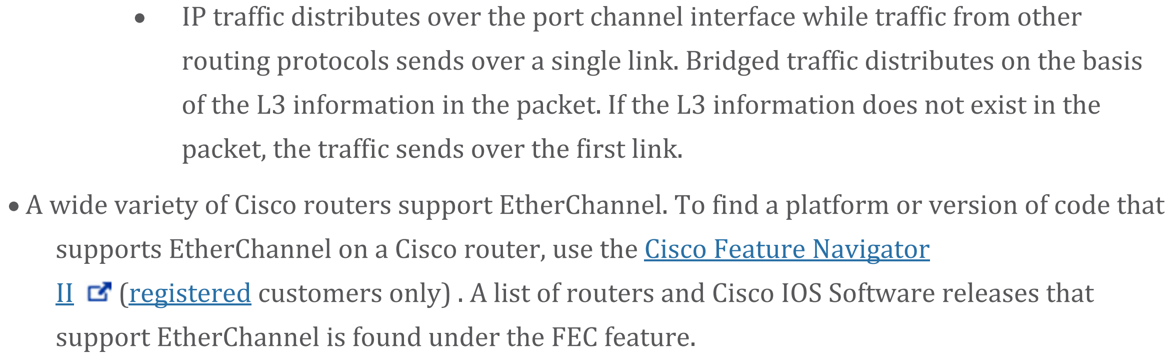
No.	’740 Patent Claim 14	Cisco EtherChannel
		

Table 1-1 Catalyst 3508G XL Features

Feature	Description
Performance and Configuration	• 8 GBIC-based 1000BaseX Gigabit Ethernet slots • Support for up to 250 port-based virtual LANs (VLANs) • Inter-Switch Link (ISL) and IEEE 802.1Q trunking support on all ports • IEEE 802.1p capable • High-speed EtherChannel connections between switches and servers • 8192 MAC addresses • Cisco Group Management Protocol (CGMP) to limit the flooding of IP multicast traffic • Broadcast storm control to prevent performance degradation from broadcast storms • Switch Port Analyzer (SPAN) port monitoring on any port • Support for command switch redundancy • Support for Cisco Gigabit Interface Converter (GBIC) modules – GigaStack GBIC module – 1000BaseSX GBIC module – 1000BaseLX/LH GBIC module – 1000BaseZX GBIC module (support for up to four 1000BaseZX GBICs with the Catalyst 3508G XL switch)
Management	• Cisco IOS command-line interface (CLI) through the console port or Telnet • CiscoView device-management application • Cluster Management Suite, a web-based tool for managing switch clusters or an individual switch through a single IP address • Simple Network Management Protocol (SNMP)
Power Redundancy	• Connection for optional Cisco 600W Redundant Power System (RPS) that operates on AC input and supplies DC output to the switch

No.	’740 Patent Claim 14	Cisco EtherChannel
		Cisco EtherChannel at Configuring EtherChannel at 1 -2  **Introduction** This sample configuration demonstrates how to set up a Layer 3 (L3) EtherChannel, without VLAN trunking, between a Cisco router and a Cisco Catalyst 6500 switch running Cisco IOS® System Software. EtherChannel can be called Fast EtherChannel (FEC) or Gigabit EtherChannel (GEC); the term depends on the speed of the interfaces or ports you use to form the EtherChannel. In this example, two Fast Ethernet ports from a Cisco router and a Catalyst 6500 switch have been bundled into a FEC. Throughout this document, the terms FEC, GEC, port channel, channel, and port group all refer to EtherChannel. Before you attempt this configuration, ensure that you meet these requirements: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software support both Layer 2 (L2) and L3 EtherChannel, with up to eight compatibly configured Ethernet interfaces on any module. All interfaces in each EtherChannel must be the same speed. All must be configured as either L2 or L3 interfaces. • EtherChannel load balancing can use either MAC addresses, IP addresses, or the TCP port numbers. **Note:** The selected mode applies to all EtherChannels configured on the switch. • Catalyst 6500/6000 Cisco IOS Software Release 12.1E or later and Catalyst 4500/4000 Cisco IOS Software Release 12.1(8a)EW or later. • Cisco routers:

No.	’740 Patent Claim 14	Cisco EtherChannel
		 - IP traffic distributes over the port channel interface while traffic from other routing protocols sends over a single link. Bridged traffic distributes on the basis of the L3 information in the packet. If the L3 information does not exist in the packet, the traffic sends over the first link. - A wide variety of Cisco routers support EtherChannel. To find a platform or version of code that supports EtherChannel on a Cisco router, use the Cisco Feature Navigator II (registered customers only) . A list of routers and Cisco IOS Software releases that support EtherChannel is found under the FEC feature.
14[a]	coupling the user ports to one or more user interface modules;	Cisco EtherChannel System discloses coupling the user ports to one or more user interface modules. *See supra at* 1[a].
14[b]	coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel,	Cisco EtherChannel System discloses coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel. *See supra at* 1[c], 3.
14[c]	at least one of said backplane traces being bi-directional and operative to communicate in	Cisco EtherChannel System discloses at least one of said backplane traces being bi-directional and operative to communicate in both an upstream direction and a downstream direction. *See supra at* 14[b], 1[d].

No.	’740 Patent Claim 14	Cisco EtherChannel
	both an upstream direction and a downstream direction;	
14[d]	receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes;	Cisco EtherChannel System discloses receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes. *See supra at* 14[a], 1[e].
14[e]	for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces; and	Cisco EtherChannel System discloses for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces. *See supra at* 14[b], 1[f].
14[f]	sending the data frame over the selected backplane trace;	Cisco EtherChannel System discloses sending the data frame over the selected backplane trace. *See supra at* 14[e], 1[g].
14[g]	said sending comprising communicating along said at least	Cisco EtherChannel System discloses said sending comprising communicating along said at least one of said backplane traces. *See supra at* 14[f], 1[h].

No.	’740 Patent Claim 14	Cisco EtherChannel
	one of said backplane traces.	

No.	’740 Patent Claim 15	Cisco EtherChannel
15[preamble]	A method for connecting user ports to a communication network, comprising:	Cisco EtherChannel System discloses a method for connecting user ports to a communication network. *See supra at* 14[preamble].
15[a]	coupling the user ports to one or more user interface modules;	Cisco EtherChannel System discloses coupling the user ports to one or more user interface modules. *See supra at* 14[a].
15[b]	coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel;	Cisco EtherChannel System discloses coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel. *See supra at* 14[b].
15[c]	receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes;	Cisco EtherChannel System discloses receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes. *See supra at* 14[d].

No.	’740 Patent Claim 15	Cisco EtherChannel
15[d]	for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces; and	Cisco EtherChannel System discloses for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces. *See supra at* 14[e].
15[e]	sending the data frame over the selected backplane trace,	Cisco EtherChannel System discloses sending the data frame over the selected backplane trace. *See supra at* 14[f].
15[f]	at least some of the backplane traces being aggregated into an Ethernet link aggregation (LAG) group.	Cisco EtherChannel System discloses at least some of the backplane traces being aggregated into an Ethernet link aggregation (LAG) group. *See supra at* 15[e], 4[f], 3.

No.	’740 Patent Claim 16	Cisco EtherChannel
16	The method according to claim 14, wherein selecting the backplane trace comprises applying a hashing function to the at least one of the frame attributes.	Cisco EtherChannel System discloses the method according to claim 14, wherein selecting the backplane trace comprises applying a hashing function to the at least one of the frame attributes. *See supra* at 14, 9, 8.

No.	’740 Patent Claim 17	Cisco EtherChannel
17[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Cisco EtherChannel System discloses apparatus for connecting a network node with a communication network. For example, Cisco EtherChannel System discloses a system for connecting a phone, router, switch, or server with a larger communication network. Catalyst 3500 Installation Guide at Table 1-1

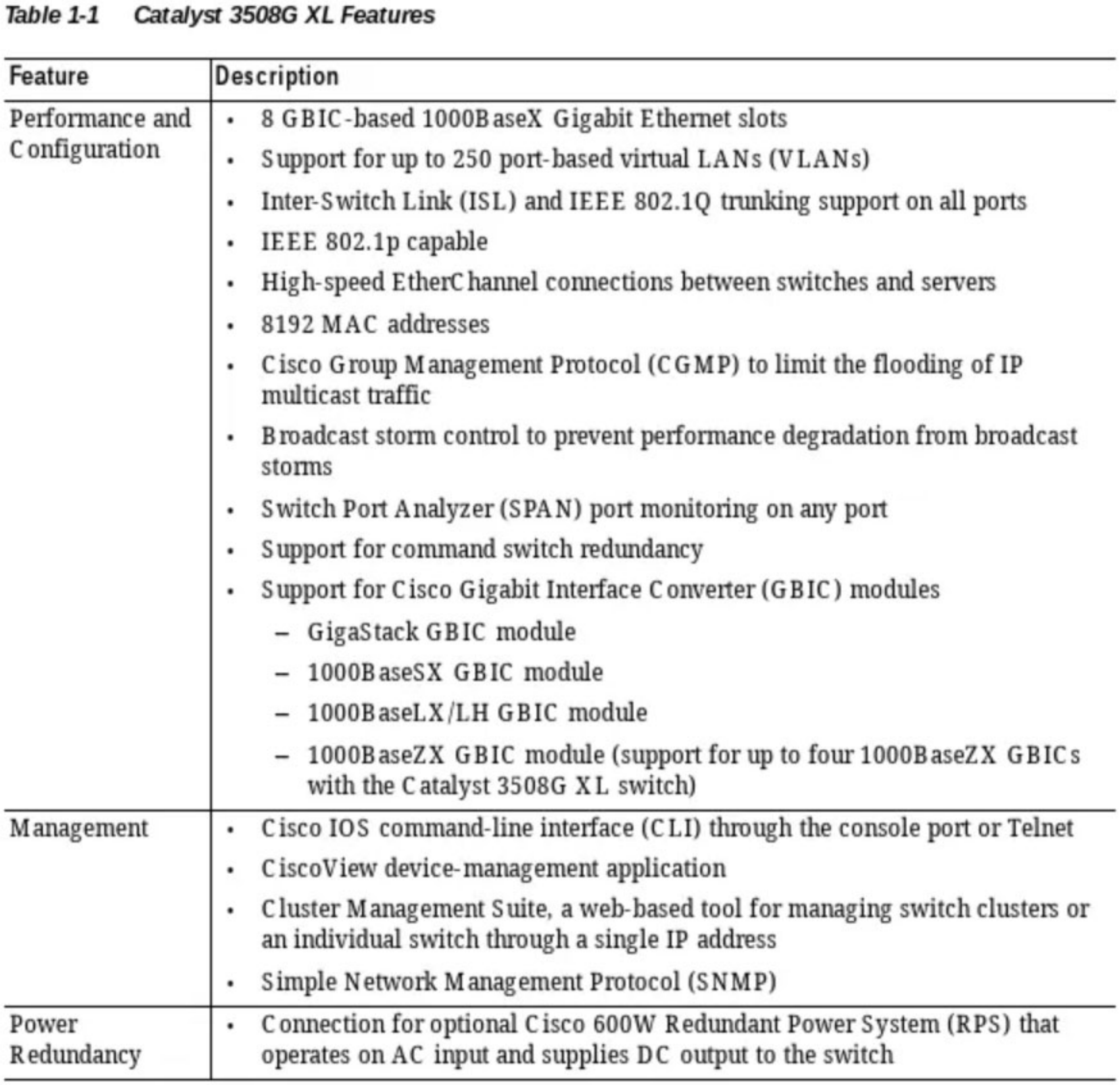
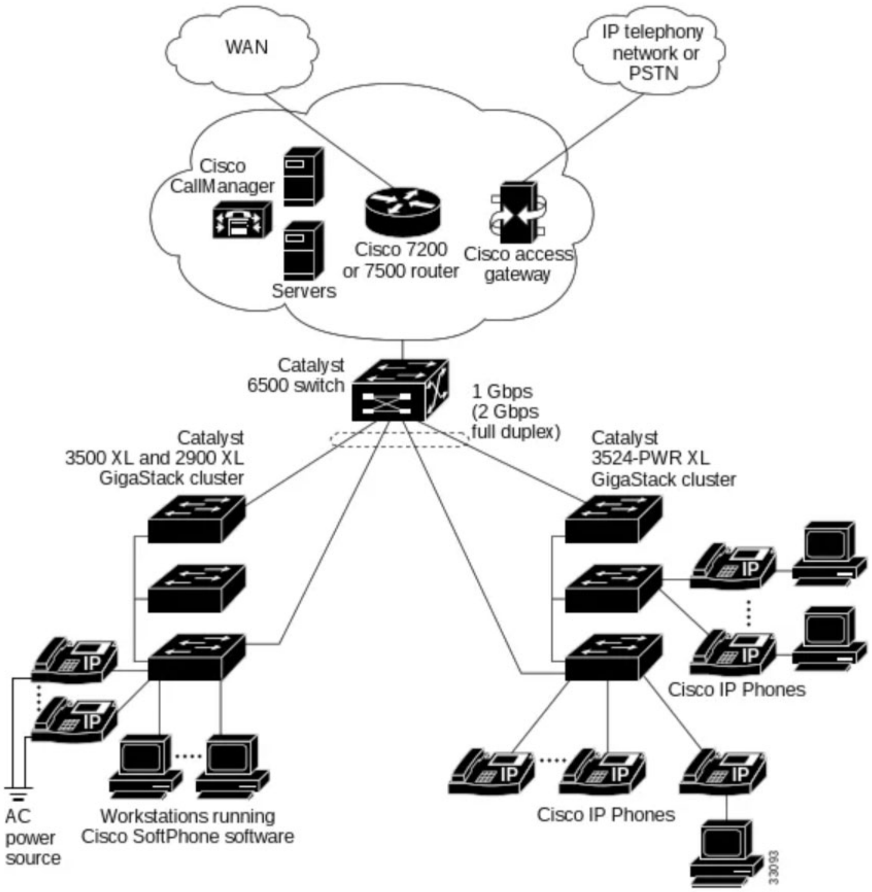
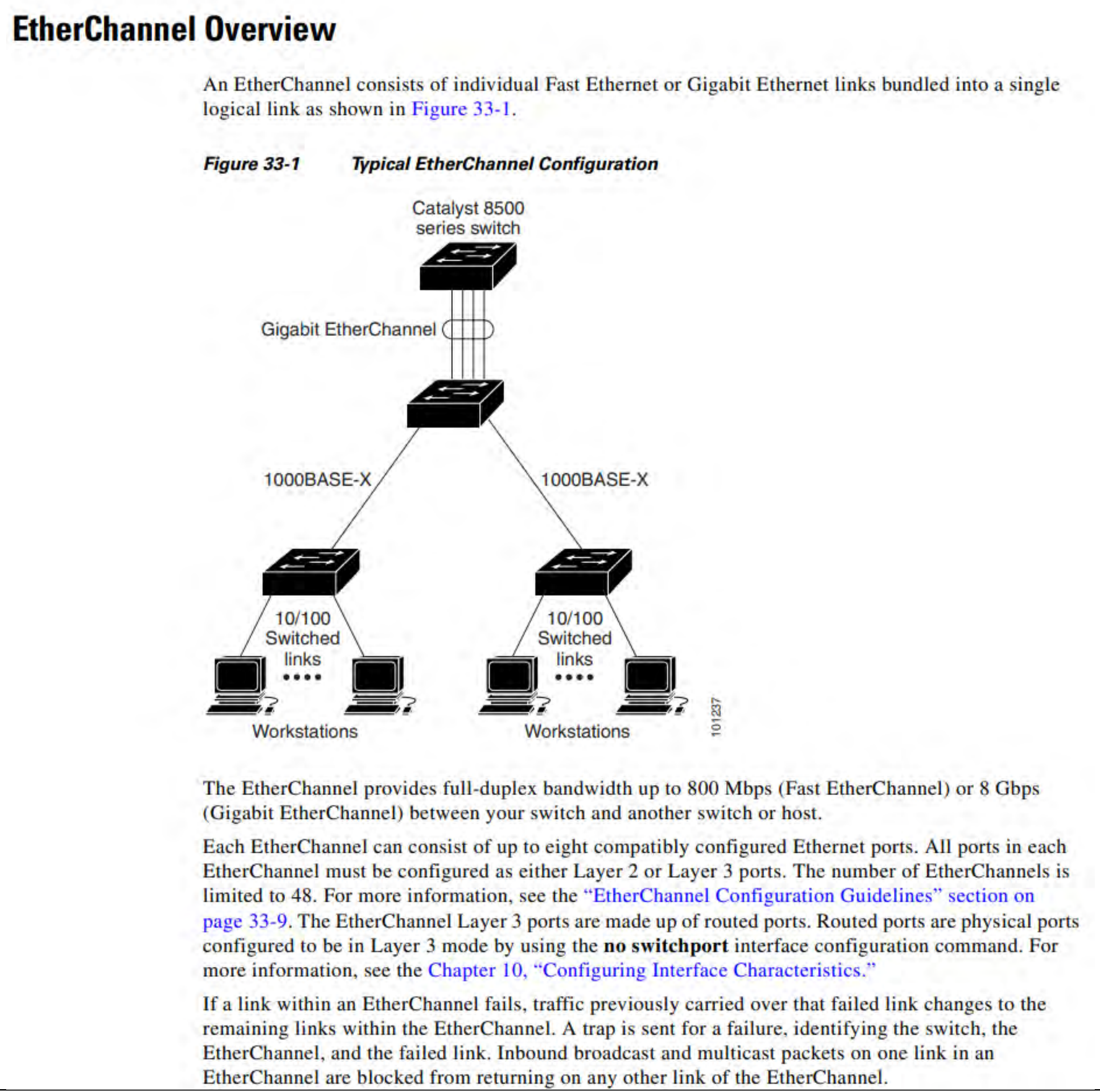
No.	’740 Patent Claim 17	Cisco EtherChannel
		

Table 1-1 Catalyst 3508G XL Features

Feature	Description
Performance and Configuration	• 8 GBIC-based 1000BaseX Gigabit Ethernet slots • Support for up to 250 port-based virtual LANs (VLANs) • Inter-Switch Link (ISL) and IEEE 802.1Q trunking support on all ports • IEEE 802.1p capable • High-speed EtherChannel connections between switches and servers • 8192 MAC addresses • Cisco Group Management Protocol (CGMP) to limit the flooding of IP multicast traffic • Broadcast storm control to prevent performance degradation from broadcast storms • Switch Port Analyzer (SPAN) port monitoring on any port • Support for command switch redundancy • Support for Cisco Gigabit Interface Converter (GBIC) modules – GigaStack GBIC module – 1000BaseSX GBIC module – 1000BaseLX/LH GBIC module – 1000BaseZX GBIC module (support for up to four 1000BaseZX GBICs with the Catalyst 3508G XL switch)
Management	• Cisco IOS command-line interface (CLI) through the console port or Telnet • CiscoView device-management application • Cluster Management Suite, a web-based tool for managing switch clusters or an individual switch through a single IP address • Simple Network Management Protocol (SNMP)
Power Redundancy	• Connection for optional Cisco 600W Redundant Power System (RPS) that operates on AC input and supplies DC output to the switch

No.	’740 Patent Claim 17	Cisco EtherChannel
		Catalyst 3500 Installation Guide at Figure 1-24 *Figure 1-24 Large Campus Configuration*  WAN IP telephony network or PSTN Cisco CallManager Servers Cisco 7200 or 7500 router Cisco access gateway Catalyst 6500 switch 1 Gbps (2 Gbps full duplex) Catalyst 3500 XL and 2900 XL GigaStack cluster Catalyst 3524-PWR XL GigaStack cluster Cisco IP Phones IP AC power source Workstations running Cisco SoftPhone software Cisco IP Phones 33093 Catalyst 3560 Configuration Guide at 33-2

<table>
<tr><th>No.</th><th>’740 Patent Claim 17</th><th>Cisco EtherChannel</th></tr>
<tr><td></td><td></td><td>


EtherChannel Overview

An EtherChannel consists of individual Fast Ethernet or Gigabit Ethernet links bundled into a single logical link as shown in Figure 33-1.

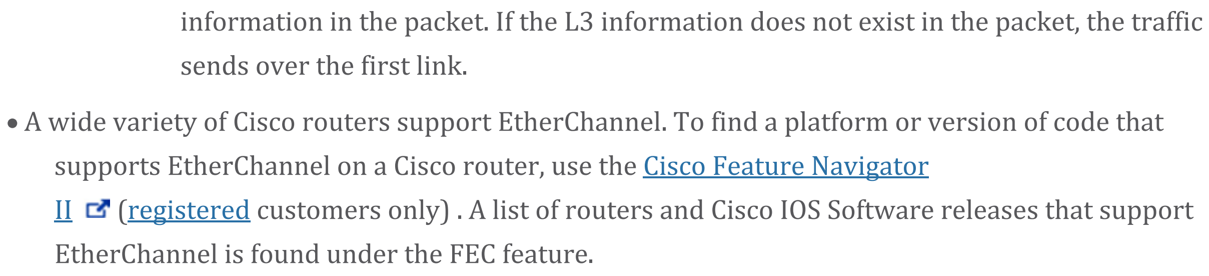
Figure 33-1 Typical EtherChannel Configuration

The EtherChannel provides full-duplex bandwidth up to 800 Mbps (Fast EtherChannel) or 8 Gbps (Gigabit EtherChannel) between your switch and another switch or host.

Each EtherChannel can consist of up to eight compatibly configured Ethernet ports. All ports in each EtherChannel must be configured as either Layer 2 or Layer 3 ports. The number of EtherChannels is limited to 48. For more information, see the “EtherChannel Configuration Guidelines” section on page 33-9. The EtherChannel Layer 3 ports are made up of routed ports. Routed ports are physical ports configured to be in Layer 3 mode by using the **no switchport** interface configuration command. For more information, see the Chapter 10, “Configuring Interface Characteristics.”

If a link within an EtherChannel fails, traffic previously carried over that failed link changes to the remaining links within the EtherChannel. A trap is sent for a failure, identifying the switch, the EtherChannel, and the failed link. Inbound broadcast and multicast packets on one link in an EtherChannel are blocked from returning on any other link of the EtherChannel.
</td></tr>
</table>

No.	’740 Patent Claim 17	Cisco EtherChannel
		Configuring EtherChannel at 1 -2 **Introduction** This sample configuration demonstrates how to set up a Layer 3 (L3) EtherChannel, without VLAN trunking, between a Cisco router and a Cisco Catalyst 6500 switch running Cisco IOS® System Software. EtherChannel can be called Fast EtherChannel (FEC) or Gigabit EtherChannel (GEC); the term depends on the speed of the interfaces or ports you use to form the EtherChannel. In this example, two Fast Ethernet ports from a Cisco router and a Catalyst 6500 switch have been bundled into a FEC. Throughout this document, the terms FEC, GEC, port channel, channel, and port group all refer to EtherChannel. Before you attempt this configuration, ensure that you meet these requirements: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software support both Layer 2 (L2) and L3 EtherChannel, with up to eight compatibly configured Ethernet interfaces on any module. All interfaces in each EtherChannel must be the same speed. All must be configured as either L2 or L3 interfaces. • EtherChannel load balancing can use either MAC addresses, IP addresses, or the TCP port numbers. **Note:** The selected mode applies to all EtherChannels configured on the switch. • Catalyst 6500/6000 Cisco IOS Software Release 12.1E or later and Catalyst 4500/4000 Cisco IOS Software Release 12.1(8a)EW or later. • Cisco routers: • IP traffic distributes over the port channel interface while traffic from other routing protocols sends over a single link. Bridged traffic distributes on the basis of the L3

No.	’740 Patent Claim 17	Cisco EtherChannel
		 information in the packet. If the L3 information does not exist in the packet, the traffic sends over the first link. • A wide variety of Cisco routers support EtherChannel. To find a platform or version of code that supports EtherChannel on a Cisco router, use the Cisco Feature Navigator II (registered customers only) . A list of routers and Cisco IOS Software releases that support EtherChannel is found under the FEC feature.
17[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network,	Cisco EtherChannel System discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. For example, Cisco EtherChannel System discloses interface modules or line cards, which process and forward data packets with specific packet information between a phone, router, switch, or server and the larger communication network. Cisco Catalyst 6500 Data Sheet at 14 (“Using the same ASIC engine design as the central PFCx, distributed forwarding cards (DFCs) located on the interface modules forward packets between two ports, directly or across the switch fabric, without involving the supervisor engine. With the DFC, each interface module has a dedicated forwarding engine complete with the full forwarding tables. Distributed Cisco Express Forwarding (Figure 6) works like this: • As in standard Cisco Express Forwarding, the central PFCx located on the supervisor engine and the DFC engines located on the interface modules are loaded with the same Cisco Express Forwarding information derived from the forwarding table before any user traffic arrives at the switch. • As a packet arrives at an interface module, its DFC engine inspects the packet and uses the information in the Cisco Express Forwarding table (including Layer 2, Layer 3, ACLs, and QoS) to make a completely hardware-based forwarding decision for that packet. • The Distributed Cisco Express Forwarding engine manages all hardware-based forwarding for traffic on that module, including Layer 2 and Layer 3 forwarding, ACLs, QoS policing and marking, and NetFlow.

No.	’740 Patent Claim 17	Cisco EtherChannel
		• Because the DFCs make all the switching decisions locally, the central PFCx can dedicate more hardware-forwarding resources to any modules not equipped with a DFC.”) Catalyst 3500 Installation Guide at Table 1-1

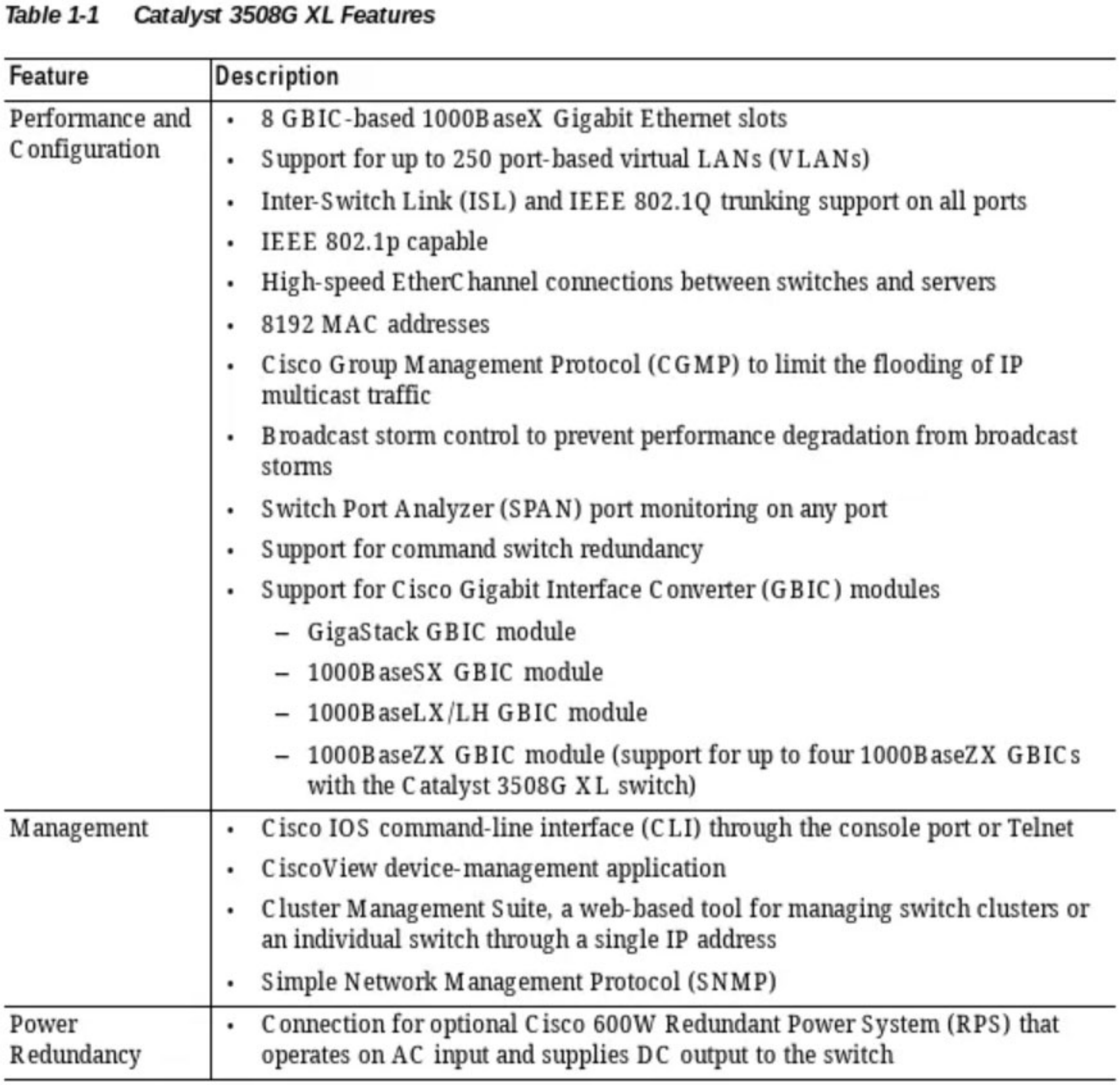
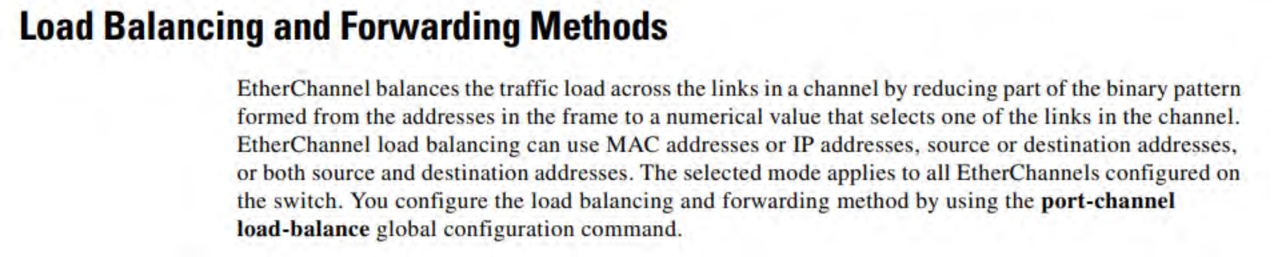
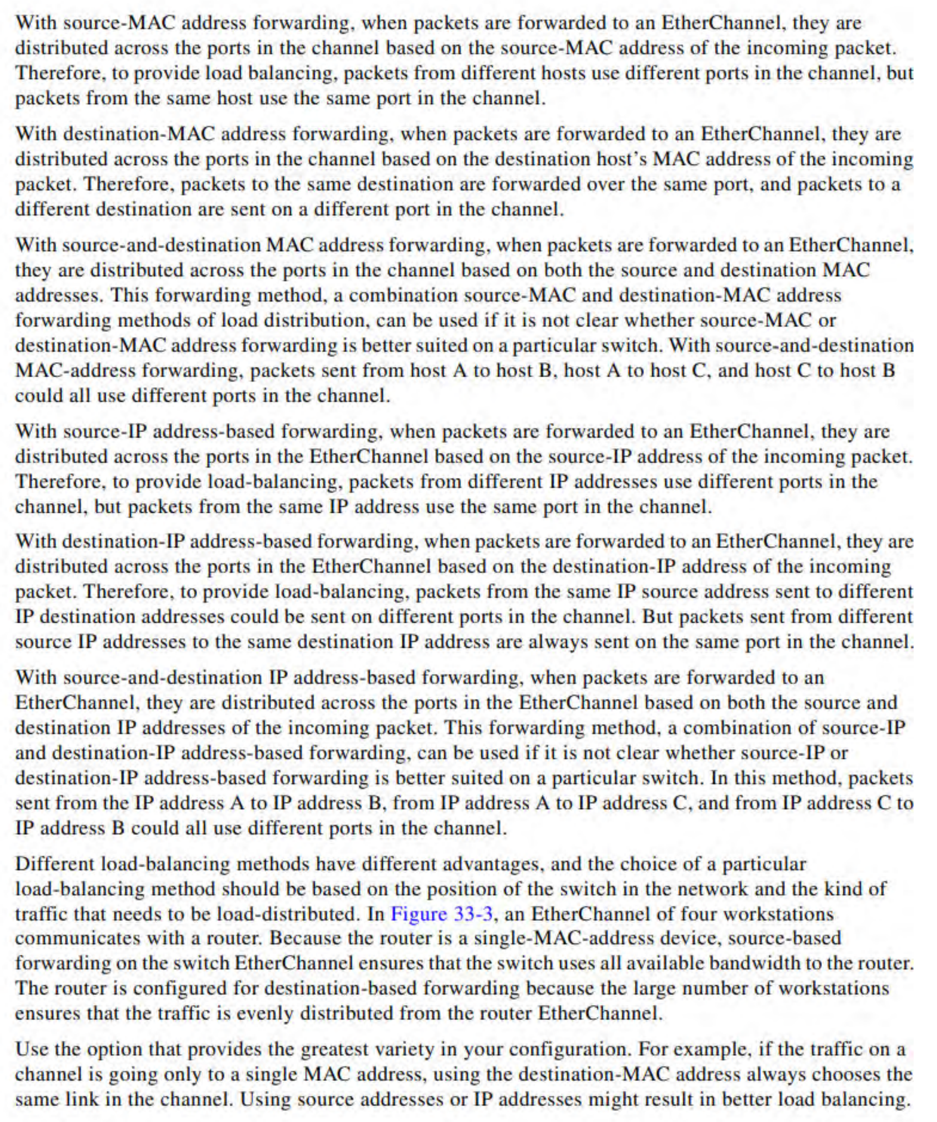
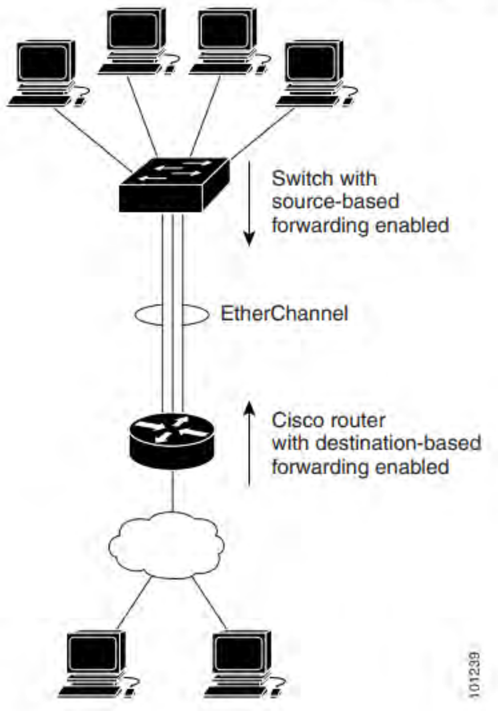
<table>
<tr><th>No.</th><th>’740 Patent Claim 17</th><th>Cisco EtherChannel</th></tr>
<tr><td></td><td></td><td>


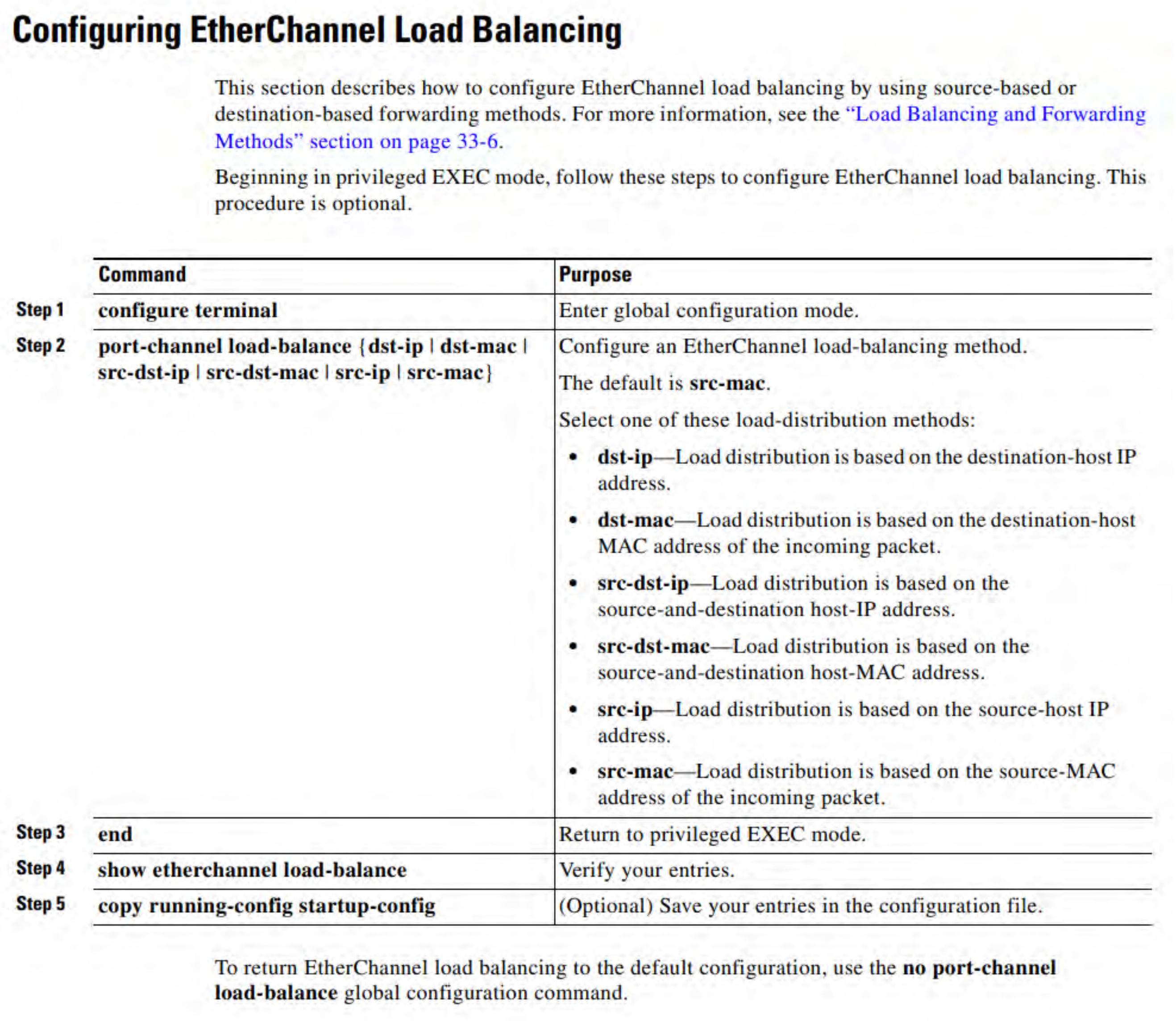
Table 1-1 Catalyst 3508G XL Features

<table>
<tr><th>Feature</th><th>Description</th></tr>
<tr><td>Performance and Configuration</td><td>• 8 GBIC-based 1000BaseX Gigabit Ethernet slots
• Support for up to 250 port-based virtual LANs (VLANs)
• Inter-Switch Link (ISL) and IEEE 802.1Q trunking support on all ports
• IEEE 802.1p capable
• High-speed EtherChannel connections between switches and servers
• 8192 MAC addresses
• Cisco Group Management Protocol (CGMP) to limit the flooding of IP multicast traffic
• Broadcast storm control to prevent performance degradation from broadcast storms
• Switch Port Analyzer (SPAN) port monitoring on any port
• Support for command switch redundancy
• Support for Cisco Gigabit Interface Converter (GBIC) modules
– GigaStack GBIC module
– 1000BaseSX GBIC module
– 1000BaseLX/LH GBIC module
– 1000BaseZX GBIC module (support for up to four 1000BaseZX GBICs with the Catalyst 3508G XL switch)</td></tr>
<tr><td>Management</td><td>• Cisco IOS command-line interface (CLI) through the console port or Telnet
• CiscoView device-management application
• Cluster Management Suite, a web-based tool for managing switch clusters or an individual switch through a single IP address
• Simple Network Management Protocol (SNMP)</td></tr>
<tr><td>Power Redundancy</td><td>• Connection for optional Cisco 600W Redundant Power System (RPS) that operates on AC input and supplies DC output to the switch</td></tr>
</table>
</td></tr>
</table>

No.	’740 Patent Claim 17	Cisco EtherChannel
		Catalyst 3560 Configuration Guide at 33-6 – 33-8  **Load Balancing and Forwarding Methods** EtherChannel balances the traffic load across the links in a channel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the channel. EtherChannel load balancing can use MAC addresses or IP addresses, source or destination addresses, or both source and destination addresses. The selected mode applies to all EtherChannels configured on the switch. You configure the load balancing and forwarding method by using the **port-channel load-balance** global configuration command.

No.	’740 Patent Claim 17	Cisco EtherChannel
		 With source-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the source-MAC address of the incoming packet. Therefore, to provide load balancing, packets from different hosts use different ports in the channel, but packets from the same host use the same port in the channel. With destination-MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on the destination host’s MAC address of the incoming packet. Therefore, packets to the same destination are forwarded over the same port, and packets to a different destination are sent on a different port in the channel. With source-and-destination MAC address forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the channel based on both the source and destination MAC addresses. This forwarding method, a combination source-MAC and destination-MAC address forwarding methods of load distribution, can be used if it is not clear whether source-MAC or destination-MAC address forwarding is better suited on a particular switch. With source-and-destination MAC-address forwarding, packets sent from host A to host B, host A to host C, and host C to host B could all use different ports in the channel. With source-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the source-IP address of the incoming packet. Therefore, to provide load-balancing, packets from different IP addresses use different ports in the channel, but packets from the same IP address use the same port in the channel. With destination-IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on the destination-IP address of the incoming packet. Therefore, to provide load-balancing, packets from the same IP source address sent to different IP destination addresses could be sent on different ports in the channel. But packets sent from different source IP addresses to the same destination IP address are always sent on the same port in the channel. With source-and-destination IP address-based forwarding, when packets are forwarded to an EtherChannel, they are distributed across the ports in the EtherChannel based on both the source and destination IP addresses of the incoming packet. This forwarding method, a combination of source-IP and destination-IP address-based forwarding, can be used if it is not clear whether source-IP or destination-IP address-based forwarding is better suited on a particular switch. In this method, packets sent from the IP address A to IP address B, from IP address A to IP address C, and from IP address C to IP address B could all use different ports in the channel. Different load-balancing methods have different advantages, and the choice of a particular load-balancing method should be based on the position of the switch in the network and the kind of traffic that needs to be load-distributed. In Figure 33-3, an EtherChannel of four workstations communicates with a router. Because the router is a single-MAC-address device, source-based forwarding on the switch EtherChannel ensures that the switch uses all available bandwidth to the router. The router is configured for destination-based forwarding because the large number of workstations ensures that the traffic is evenly distributed from the router EtherChannel. Use the option that provides the greatest variety in your configuration. For example, if the traffic on a channel is going only to a single MAC address, using the destination-MAC address always chooses the same link in the channel. Using source addresses or IP addresses might result in better load balancing.

No.	’740 Patent Claim 17	Cisco EtherChannel
		Figure 33-3 Load Distribution and Forwarding Methods  Switch with source-based forwarding enabled EtherChannel Cisco router with destination-based forwarding enabled 101239 Catalyst 3560 Configuration Guide at 33-16

No.	’740 Patent Claim 17	Cisco EtherChannel
		

Configuring EtherChannel Load Balancing

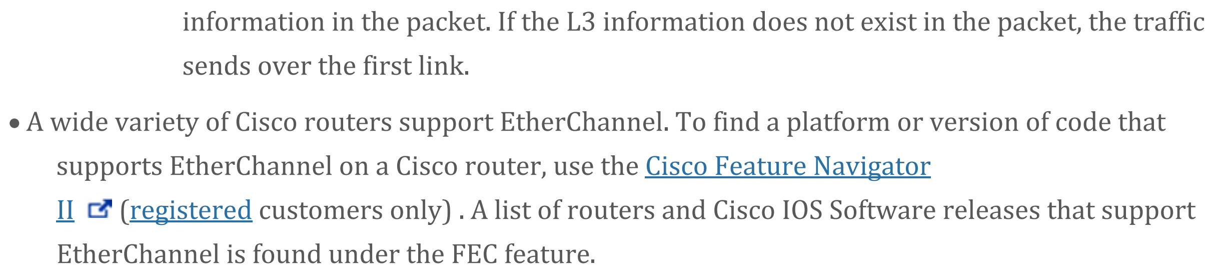
This section describes how to configure EtherChannel load balancing by using source-based or destination-based forwarding methods. For more information, see the “Load Balancing and Forwarding Methods” section on page 33-6.

Beginning in privileged EXEC mode, follow these steps to configure EtherChannel load balancing. This procedure is optional.

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**port-channel load-balance** {**dst-ip** | **dst-mac** | **src-dst-ip** | **src-dst-mac** | **src-ip** | **src-mac**}	Configure an EtherChannel load-balancing method. The default is **src-mac**. Select one of these load-distribution methods: • **dst-ip**—Load distribution is based on the destination-host IP address. • **dst-mac**—Load distribution is based on the destination-host MAC address of the incoming packet. • **src-dst-ip**—Load distribution is based on the source-and-destination host-IP address. • **src-dst-mac**—Load distribution is based on the source-and-destination host-MAC address. • **src-ip**—Load distribution is based on the source-host IP address. • **src-mac**—Load distribution is based on the source-MAC address of the incoming packet.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show etherchannel load-balance**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To return EtherChannel load balancing to the default configuration, use the **no port-channel load-balance** global configuration command.

No.	’740 Patent Claim 17	Cisco EtherChannel
		Configuring EtherChannel at 1 -2 **Introduction** This sample configuration demonstrates how to set up a Layer 3 (L3) EtherChannel, without VLAN trunking, between a Cisco router and a Cisco Catalyst 6500 switch running Cisco IOS® System Software. EtherChannel can be called Fast EtherChannel (FEC) or Gigabit EtherChannel (GEC); the term depends on the speed of the interfaces or ports you use to form the EtherChannel. In this example, two Fast Ethernet ports from a Cisco router and a Catalyst 6500 switch have been bundled into a FEC. Throughout this document, the terms FEC, GEC, port channel, channel, and port group all refer to EtherChannel. Before you attempt this configuration, ensure that you meet these requirements: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software: • Catalyst 6500/6000 and 4500/4000 series switches running Cisco IOS Software support both Layer 2 (L2) and L3 EtherChannel, with up to eight compatibly configured Ethernet interfaces on any module. All interfaces in each EtherChannel must be the same speed. All must be configured as either L2 or L3 interfaces. • EtherChannel load balancing can use either MAC addresses, IP addresses, or the TCP port numbers. **Note:** The selected mode applies to all EtherChannels configured on the switch. • Catalyst 6500/6000 Cisco IOS Software Release 12.1E or later and Catalyst 4500/4000 Cisco IOS Software Release 12.1(8a)EW or later. • Cisco routers: • IP traffic distributes over the port channel interface while traffic from other routing protocols sends over a single link. Bridged traffic distributes on the basis of the L3

No.	’740 Patent Claim 17	Cisco EtherChannel
		 information in the packet. If the L3 information does not exist in the packet, the traffic sends over the first link. • A wide variety of Cisco routers support EtherChannel. To find a platform or version of code that supports EtherChannel on a Cisco router, use the Cisco Feature Navigator II (registered customers only) . A list of routers and Cisco IOS Software releases that support EtherChannel is found under the FEC feature.
17[b]	at least one of said interface modules being operative to communicate in both an upstream direction and a downstream direction;	Cisco EtherChannel System discloses at least one of said interface modules being operative to communicate in both an upstream direction and a downstream direction. For example, Cisco EtherChannel System discloses fully duplexed interface modules or line cards capable of distributing traffic in both upstream and downstream directions. Cisco Catalyst 6500 Data Sheet at 14 (“Using the same ASIC engine design as the central PFCx, distributed forwarding cards (DFCs) located on the interface modules forward packets between two ports, directly or across the switch fabric, without involving the supervisor engine. With the DFC, each interface module has a dedicated forwarding engine complete with the full forwarding tables. Distributed Cisco Express Forwarding (Figure 6) works like this: • As in standard Cisco Express Forwarding, the central PFCx located on the supervisor engine and the DFC engines located on the interface modules are loaded with the same Cisco Express Forwarding information derived from the forwarding table before any user traffic arrives at the switch. • As a packet arrives at an interface module, its DFC engine inspects the packet and uses the information in the Cisco Express Forwarding table (including Layer 2, Layer 3, ACLs, and QoS) to make a completely hardware-based forwarding decision for that packet. • The Distributed Cisco Express Forwarding engine manages all hardware-based forwarding for traffic on that module, including Layer 2 and Layer 3 forwarding, ACLs, QoS policing and marking, and NetFlow. • Because the DFCs make all the switching decisions locally, the central PFCx can dedicate more hardware-forwarding resources to any modules not equipped with a DFC.”)

No.	’740 Patent Claim 17	Cisco EtherChannel
		Catalyst 3500 Installation Guide at Table 1-1

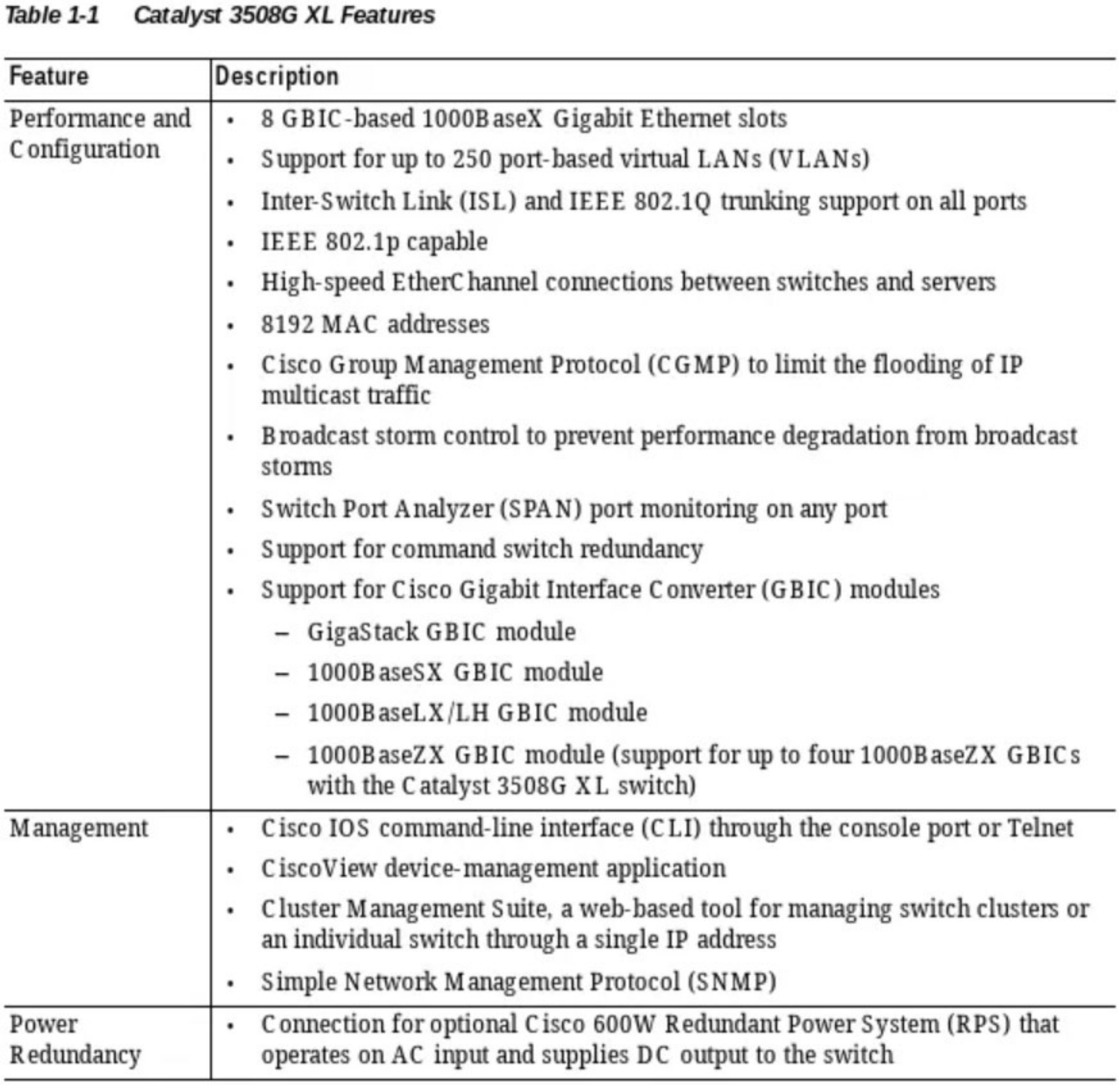
No.	’740 Patent Claim 17	Cisco EtherChannel
		

Table 1-1 Catalyst 3508G XL Features

Feature	Description
Performance and Configuration	• 8 GBIC-based 1000BaseX Gigabit Ethernet slots • Support for up to 250 port-based virtual LANs (VLANs) • Inter-Switch Link (ISL) and IEEE 802.1Q trunking support on all ports • IEEE 802.1p capable • High-speed EtherChannel connections between switches and servers • 8192 MAC addresses • Cisco Group Management Protocol (CGMP) to limit the flooding of IP multicast traffic • Broadcast storm control to prevent performance degradation from broadcast storms • Switch Port Analyzer (SPAN) port monitoring on any port • Support for command switch redundancy • Support for Cisco Gigabit Interface Converter (GBIC) modules – GigaStack GBIC module – 1000BaseSX GBIC module – 1000BaseLX/LH GBIC module – 1000BaseZX GBIC module (support for up to four 1000BaseZX GBICs with the Catalyst 3508G XL switch)
Management	• Cisco IOS command-line interface (CLI) through the console port or Telnet • CiscoView device-management application • Cluster Management Suite, a web-based tool for managing switch clusters or an individual switch through a single IP address • Simple Network Management Protocol (SNMP)
Power Redundancy	• Connection for optional Cisco 600W Redundant Power System (RPS) that operates on AC input and supplies DC output to the switch

<table>
<tr><th>No.</th><th>’740 Patent Claim 17</th><th>Cisco EtherChannel</th></tr>
<tr><td></td><td></td><td>Catalyst 3500 Installation Guide at 1-2

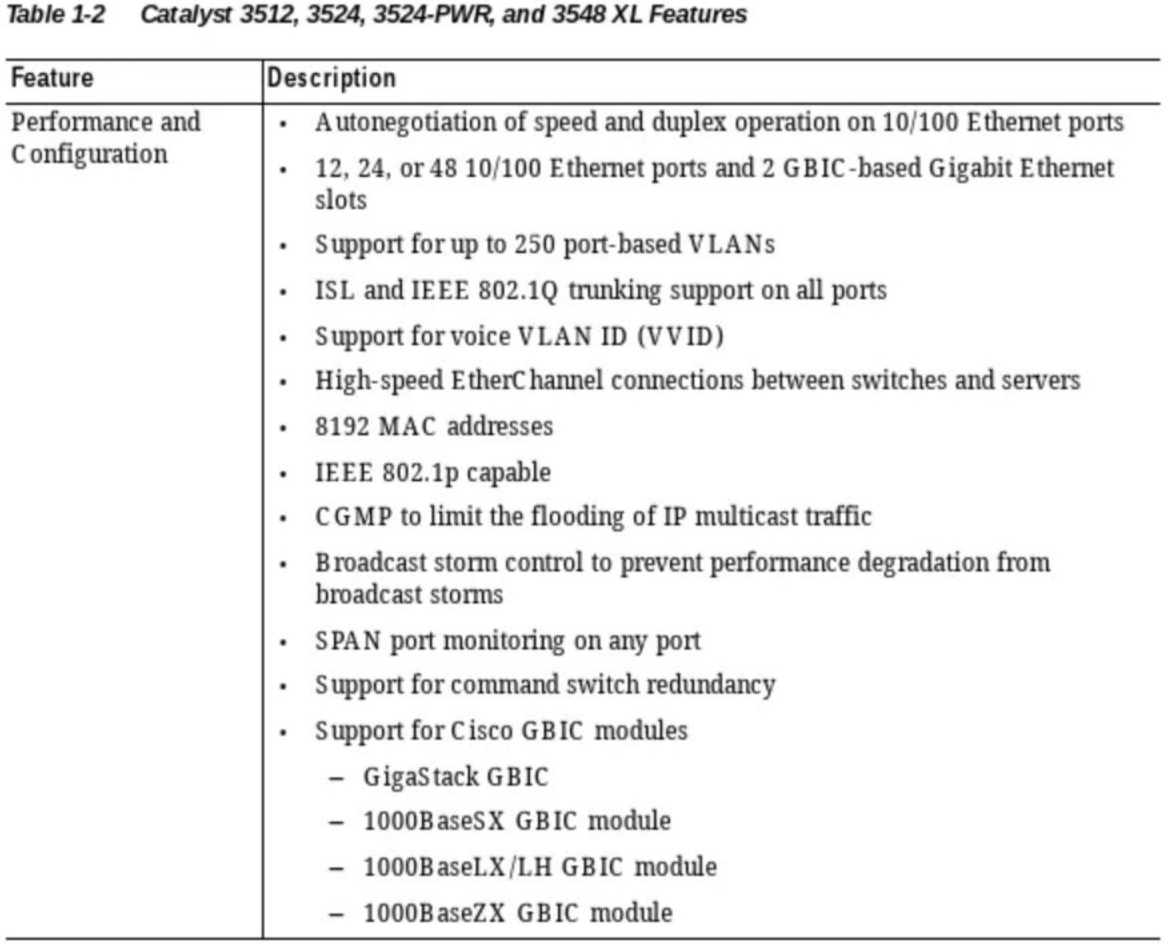
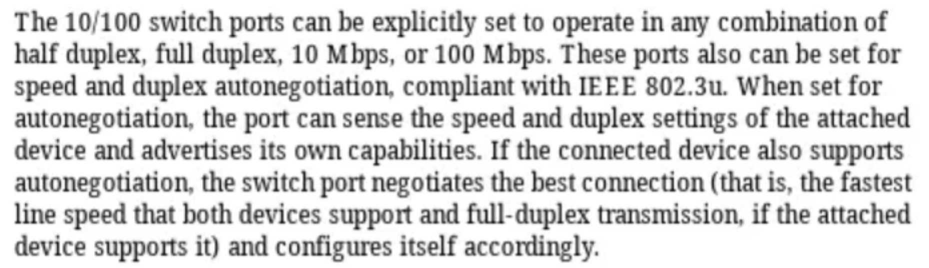



Table 1-2 Catalyst 3512, 3524, 3524-PWR, and 3548 XL Features
<table>
<tr><th>Feature</th><th>Description</th></tr>
<tr><td>Performance and Configuration</td><td>• Autonegotiation of speed and duplex operation on 10/100 Ethernet ports
• 12, 24, or 48 10/100 Ethernet ports and 2 GBIC-based Gigabit Ethernet slots
• Support for up to 250 port-based VLANs
• ISL and IEEE 802.1Q trunking support on all ports
• Support for voice VLAN ID (VVID)
• High-speed EtherChannel connections between switches and servers
• 8192 MAC addresses
• IEEE 802.1p capable
• CGMP to limit the flooding of IP multicast traffic
• Broadcast storm control to prevent performance degradation from broadcast storms
• SPAN port monitoring on any port
• Support for command switch redundancy
• Support for Cisco GBIC modules
– GigaStack GBIC
– 1000BaseSX GBIC module
– 1000BaseLX/LH GBIC module
– 1000BaseZX GBIC module</td></tr>
</table>
Catalyst 3500 Installation Guide at 1-8</td></tr>
</table>

No.	’740 Patent Claim 17	Cisco EtherChannel
		 The 10/100 switch ports can be explicitly set to operate in any combination of half duplex, full duplex, 10 Mbps, or 100 Mbps. These ports also can be set for speed and duplex autonegotiation, compliant with IEEE 802.3u. When set for autonegotiation, the port can sense the speed and duplex settings of the attached device and advertises its own capabilities. If the connected device also supports autonegotiation, the switch port negotiates the best connection (that is, the fastest line speed that both devices support and full-duplex transmission, if the attached device supports it) and configures itself accordingly.
17[c]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Cisco EtherChannel System discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 1[a].
17[d]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Cisco EtherChannel System discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 1[c].

No.	’740 Patent Claim 17	Cisco EtherChannel
17[e]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame;	Cisco EtherChannel System discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 1[f].
17[f]	at least one of said first physical links and at least one of said second	Cisco EtherChannel System discloses at least one of said first physical links and at least one of said second links being bi-directional links operative to communicate in both said upstream direction and said downstream direction. *See supra at* 1[b], 1[d].

No.	’740 Patent Claim 17	Cisco EtherChannel
	links being bi-directional links operative to communicate in both said upstream direction and said downstream direction.	

No.	’740 Patent Claim 18	Cisco EtherChannel
18[a]	The apparatus according to claim 17, and comprising a backplane to which the one or more interface modules are coupled,	Cisco EtherChannel System discloses the apparatus according to claim 17, and comprising a backplane to which the one or more interface modules are coupled. *See supra at* 3, 17.
18[b]	wherein the second physical links comprise backplane traces formed on the backplane.	Cisco EtherChannel System discloses wherein the second physical links comprise backplane traces formed on the backplane. *See supra at* 3, 17.

No.	’740 Patent Claim 19	Cisco EtherChannel
19[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Cisco EtherChannel System discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
19[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	Cisco EtherChannel System discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].
19[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Cisco EtherChannel System discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].
19[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Cisco EtherChannel System discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].
19[d]	a control module, which is arranged to	Cisco EtherChannel System discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single

No.	’740 Patent Claim 19	Cisco EtherChannel
	select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,	computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].
19[e]	at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group.	Cisco EtherChannel System discloses at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group. *See supra at* 4[f].

No.	’740 Patent Claim 20	Cisco EtherChannel
20[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Cisco EtherChannel System discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].

No.	**’740 Patent Claim 20**	**Cisco EtherChannel**
20[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	Cisco EtherChannel System discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].
20[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Cisco EtherChannel System discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].
20[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Cisco EtherChannel System discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].
20[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a	Cisco EtherChannel System discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].

No.	’740 Patent Claim 20	Cisco EtherChannel
	single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,	
20[e]	two or more of the first physical links being aggregated into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node.	Cisco EtherChannel System discloses two or more of the first physical links being aggregated into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node. *See supra at* 19[e], 5[f].

No.	’740 Patent Claim 21	Cisco EtherChannel
21	The apparatus according to claim 17, and comprising a multiplexer, which is arranged to perform at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node.	Cisco EtherChannel System discloses the apparatus according to claim 17, and comprising a multiplexer, which is arranged to perform at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node. *See supra at* 6, 17.

No.	’740 Patent Claim 22	Cisco EtherChannel
22	The apparatus according to claim 17, wherein the control module is arranged to balance a frame data rate among at least some of the first and second physical links.	Cisco EtherChannel System discloses the apparatus according to claim 17, wherein the control module is arranged to balance a frame data rate among at least some of the first and second physical links. *See supra at* 7, 17.

No.	’740 Patent Claim 23	Cisco EtherChannel
23	The apparatus according to claim 17, wherein the control module is arranged to apply a mapping function to the at least one of the frame attributes so as to select the first and second physical links.	Cisco EtherChannel System discloses the apparatus according to claim 17, wherein the control module is arranged to apply a mapping function to the at least one of the frame attributes so as to select the first and second physical links. *See supra at* 8, 17.

No.	’740 Patent Claim 24	Cisco EtherChannel
24	The apparatus according to claim 23, wherein the mapping function comprises a hashing function.	Cisco EtherChannel System discloses the apparatus according to claim 23, wherein the mapping function comprises a hashing function. *See supra at* 9, 23.

No.	’740 Patent Claim 25	Cisco EtherChannel
25[a]	The apparatus according to claim 24, wherein the control module is arranged to determine a hashing size responsively to a number of at least some of the first and second physical links,	Cisco EtherChannel System discloses the apparatus according to claim 24, wherein the control module is arranged to determine a hashing size responsively to a number of at least some of the first and second physical links. *See supra at* 10[a], 24.
25[b]	to apply the hashing function to the at least one of the frame attributes to produce a hashing key,	Cisco EtherChannel System discloses to apply the hashing function to the at least one of the frame attributes to produce a hashing key. *See supra at* 10[b].
25[c]	to calculate a modulo of a division operation of the hashing key by the hashing size, and	Cisco EtherChannel System discloses to calculate a modulo of a division operation of the hashing key by the hashing size. *See supra at* 10[c].
25[d]	to select the first and second physical links responsively to the modulo.	Cisco EtherChannel System discloses to select the first and second physical links responsively to the modulo. *See supra at* 10[d].

No.	’740 Patent Claim 26	Cisco EtherChannel
26	The apparatus according to claim 25, wherein the control module is arranged to select the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo.	Cisco EtherChannel System discloses the apparatus according to claim 25, wherein the control module is arranged to select the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo. *See supra at* 11, 25.

No.	’740 Patent Claim 27	Cisco EtherChannel
27	The apparatus according to claim 17, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port.	Cisco EtherChannel System discloses the apparatus according to claim 17, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port. *See supra at* 12, 17.

No.	’740 Patent Claim 28	Cisco EtherChannel
28[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Cisco EtherChannel System discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].

No.	’740 Patent Claim 28	Cisco EtherChannel
28[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	Cisco EtherChannel System discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].
28[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Cisco EtherChannel System discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].
28[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Cisco EtherChannel System discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].
28[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a	Cisco EtherChannel System discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].

No.	’740 Patent Claim 28	Cisco EtherChannel
	single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,	
28[e]	the communication network being arranged to provide a communication service to the network node,	Cisco EtherChannel System discloses the communication network being arranged to provide a communication service to the network node. *See supra at* 2[b].
28[f]	the service having specified bandwidth requirements comprising at least one of a committed information rate (CR), a peak information rate (PIR) and an excess information rate (EIR), and	Cisco EtherChannel System discloses the service having specified bandwidth requirements comprising at least one of a committed information rate (CR), a peak information rate (PIR) and an excess information rate (EIR). *See supra at* 13[i].
28[g]	the first and second groups of physical links being dimensioned to	Cisco EtherChannel System discloses the first and second groups of physical links being dimensioned to provide an allocated bandwidth for the communication service responsively to the bandwidth requirements.

No.	’740 Patent Claim 28	Cisco EtherChannel
	provide an allocated bandwidth for the communication service responsively to the bandwidth requirements.	*See supra at* 13[j].

No.	’740 Patent Claim 29	Cisco EtherChannel
29[preamble]	Apparatus for connecting user ports to a communication network, comprising:	Cisco EtherChannel System discloses apparatus for connecting user ports to a communication network. *See supra at* 17[preamble], 14[preamble].
29[a]	one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network,	Cisco EtherChannel System discloses one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network. *See supra at* 17[a], 14[a].
29[b]	at least one of said user interface modules being bi-directional and operative to communicate in both an upstream direction	Cisco EtherChannel System discloses at least one of said user interface modules being bi-directional and operative to communicate in both an upstream direction and a downstream direction. *See supra at* 17[b], 14[c].

No.	’740 Patent Claim 29	Cisco EtherChannel
	and a downstream direction;	
29[c]	a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network,	Cisco EtherChannel System discloses a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network. *See supra at* 14[b]-[e].
29[d]	at least one of said backplane traces being bi-directional and operative to communicate in both said upstream direction and said downstream direction; and	Cisco EtherChannel System discloses at least one of said backplane traces being bi-directional and operative to communicate in both said upstream direction and said downstream direction. *See supra at* 14[c], 17[b].
29[e]	a control module, which is arranged to select, for each data frame, responsively to at least one of the	Cisco EtherChannel System discloses a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame. *See supra at* 14[e], 17[e].

No.	’740 Patent Claim 29	Cisco EtherChannel
	frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame.	

No.	’740 Patent Claim 30	Cisco EtherChannel
30[preamble]	Apparatus for connecting user ports to a communication network, comprising:	Cisco EtherChannel System discloses apparatus for connecting user ports to a communication network. *See supra at* 29[preamble].
30[a]	one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network;	Cisco EtherChannel System discloses one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network. *See supra at* 29[a].
30[b]	a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in	Cisco EtherChannel System discloses a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network. *See supra at* 29[c].

No.	’740 Patent Claim 30	Cisco EtherChannel
	parallel so as to transfer the data frames between the one or more user interface modules and the communication network;	
30[c]	a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame;	Cisco EtherChannel System discloses a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame. *See supra at* 29[e].
30[d]	at least some of the backplane traces are aggregated into an Ethernet link aggregation (LAG) group.	Cisco EtherChannel System discloses at least some of the backplane traces are aggregated into an Ethernet link aggregation (LAG) group. *See supra at* 4[f], 15[f].

No.	’740 Patent Claim 31	Cisco EtherChannel
31	The apparatus according to claim 29, wherein the control module is arranged to apply a hashing function to the at least one of the frame attributes so as to select the backplane trace.	Cisco EtherChannel System discloses the apparatus according to claim 29, wherein the control module is arranged to apply a hashing function to the at least one of the frame attributes so as to select the backplane trace. *See supra at* 16, 29, 30[c].

EXHIBIT C-4
Defendant's Preliminary Invalidity Contentions
Orckit Corporation v. Cisco Systems, Inc., 2:22-cv-00276-JRG-RSP

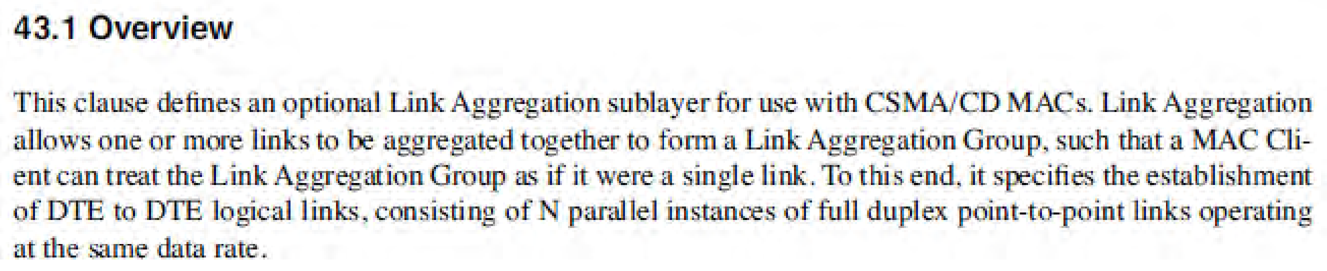
Chart for U.S. Patent 7,545,740 ("the '740 Patent")
IEEE Standard 802.3, 2002 ("IEEE 802.3")

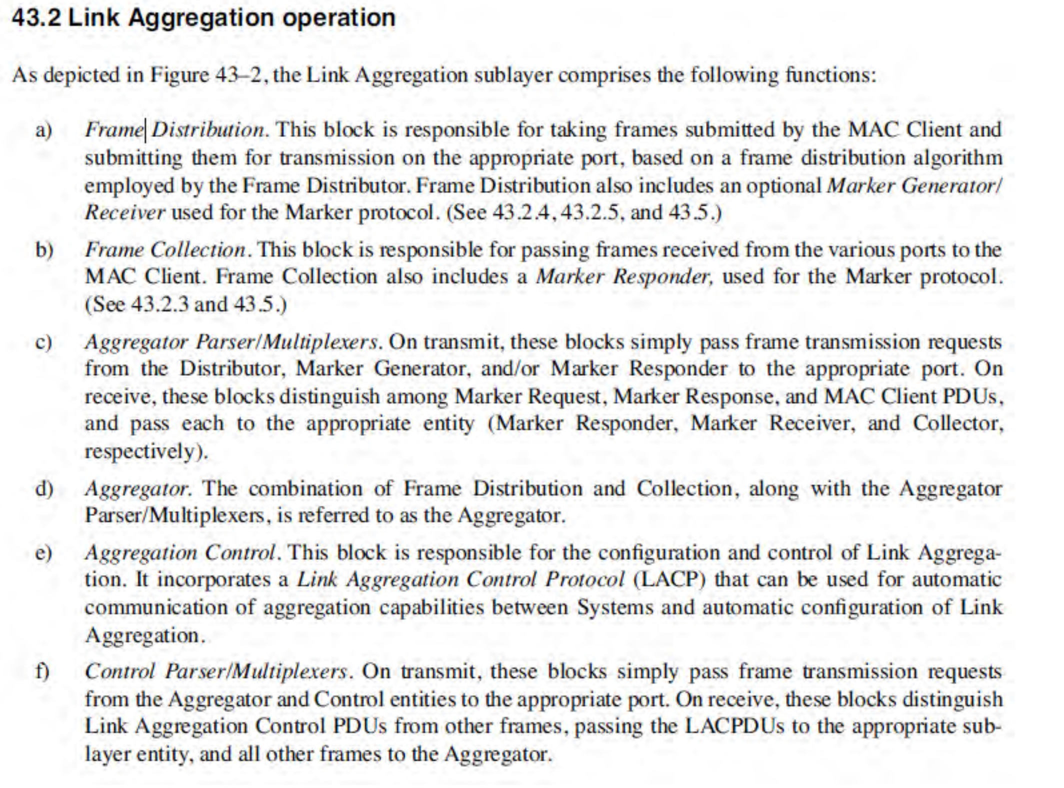
As shown in the chart below, all Asserted Claims of the '740 Patent are invalid under (1) 35 U.S.C. §§ 102 (a), (b), and (g) because IEEE 802.3 meets each element of those claims, and/or (2) 35 U.S.C. § 103 because IEEE 802.3 renders those claims obvious either alone, or in combination with the knowledge of a person having ordinary skill in the art, and in further combination with the references specifically identified below and in the following claim chart and/or one or more references identified in Defendant's Preliminary Invalidity Contentions. The following quotations and diagrams come from IEEE 802.3 titled "Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications", which was published on March 8, 2002.

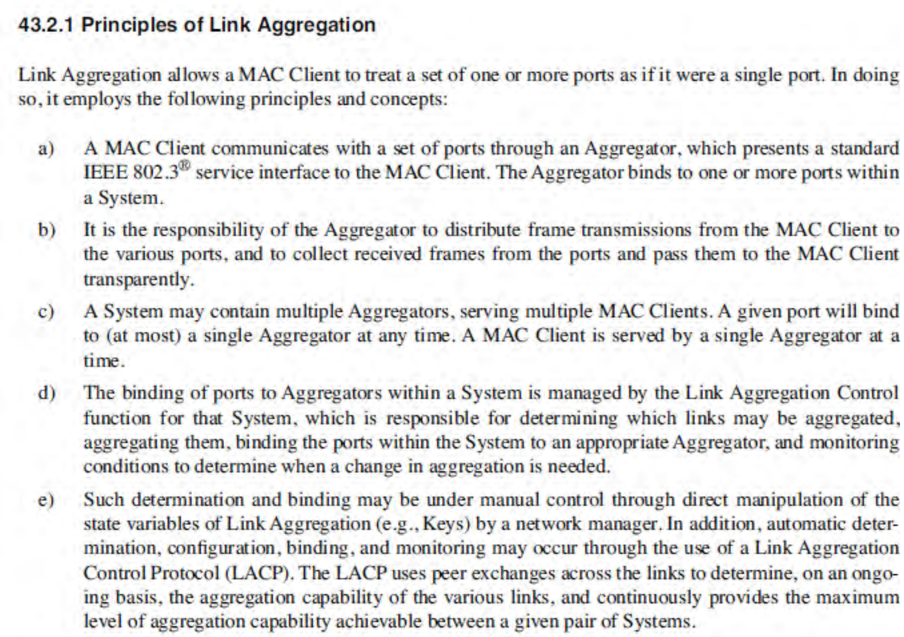
Motivations to combine the disclosures in IEEE 802.3 with disclosures in other publications known in the art, as explained in this chart, include at least the similarity in subject matter between the references to the extent they concern methods of data communication systems, and specifically to methods and systems for link aggregation in a data communication network. Insofar as the references cite other patents or publications, or suggest additional changes, one of ordinary skill in the art would look beyond a single reference to other references in the field.

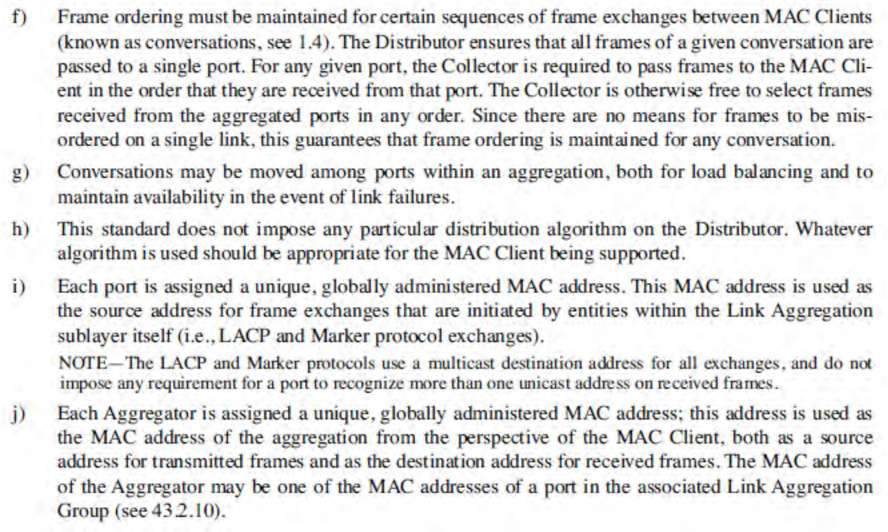
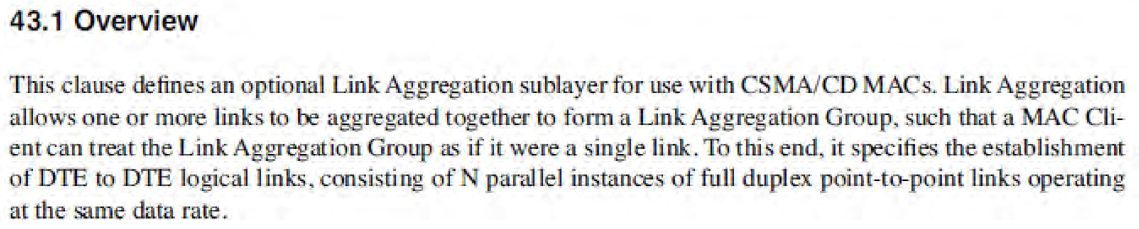
These invalidity contentions are based on Defendant's present understanding of the asserted claims, and Orckit's apparent construction of the claims in its November 3, 2022 Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1, and Orckit's January 19, 2023 First Amended Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1 (Orckit's "Infringement Disclosures"), which is deficient at least insofar as it fails to cite any documents or identify accused structures, acts, or materials in the Accused Products with particularity. Defendant does not agree with Orckit's application of the claims, or that the claims satisfy the requirements of 35 U.S.C. § 112. Defendant's contentions herein are not, and should in no way be seen as, admissions or adoptions as to any particular claim scope or construction, or as any admission that any particular element is met by any accused product in any particular way. Defendant objects to any attempt to imply claim construction from this chart. Defendant's prior art invalidity contentions are made in a variety of alternatives and do not represent Defendant's agreement or view as to the meaning, definiteness, written description support for, or enablement of any claim contained therein.

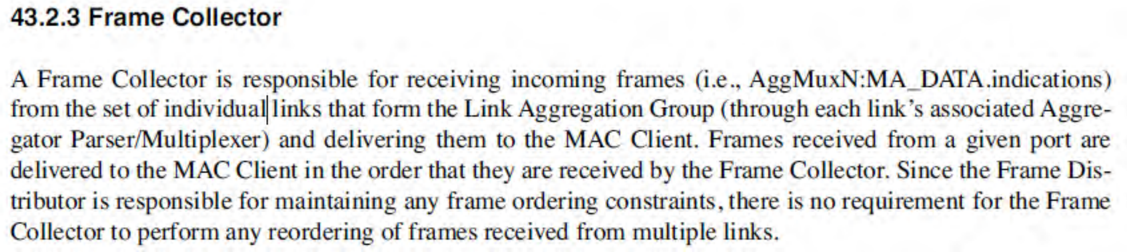
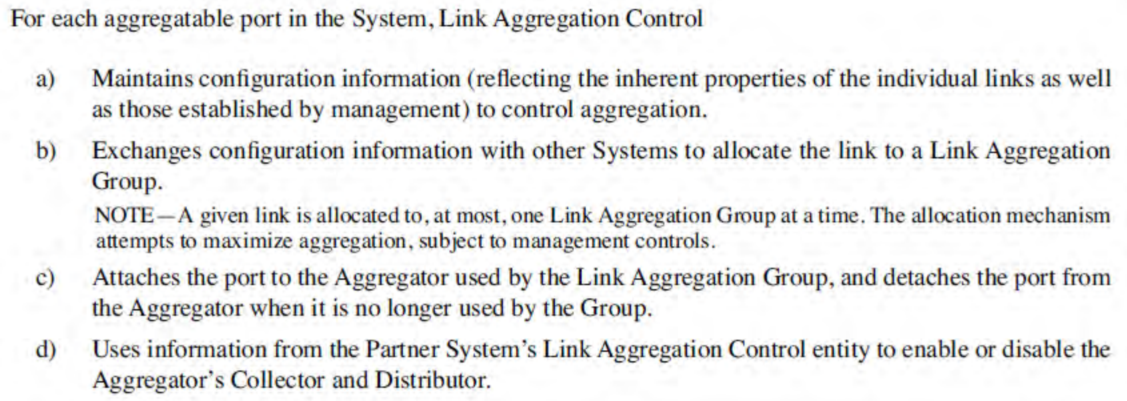
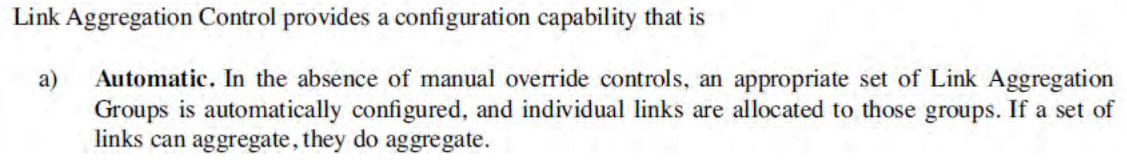
The following contentions are subject to revision and amendment pursuant to Federal Rule of Civil Procedure 26(e), the Local Rules, and the Orders of record in this matter subject to further investigation and discovery regarding the prior art and the Court's construction of the claims at issue.

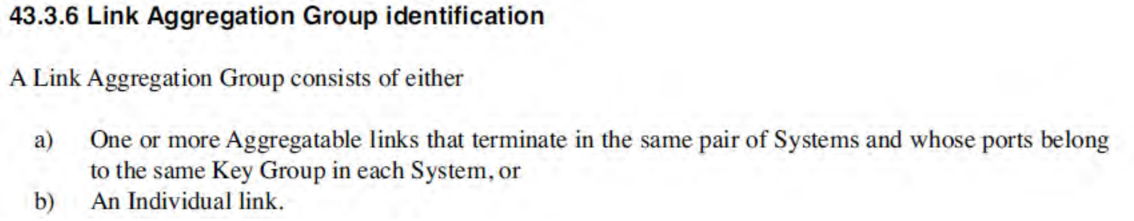
No.	'740 Patent Claim 1	IEEE 802.3
1[preamble]	A method for communication, comprising:	IEEE 802.3 discloses a method for communication. For example, IEEE 802.3 describes a protocol for collecting and distributing data frames with a MAC Client. IEEE 802.3 at 1465  43.1 Overview This clause defines an optional Link Aggregation sublayer for use with CSMA/CD MACs. Link Aggregation allows one or more links to be aggregated together to form a Link Aggregation Group, such that a MAC Client can treat the Link Aggregation Group as if it were a single link. To this end, it specifies the establishment of DTE to DTE logical links, consisting of N parallel instances of full duplex point-to-point links operating at the same data rate. IEEE 802.3 at 1468

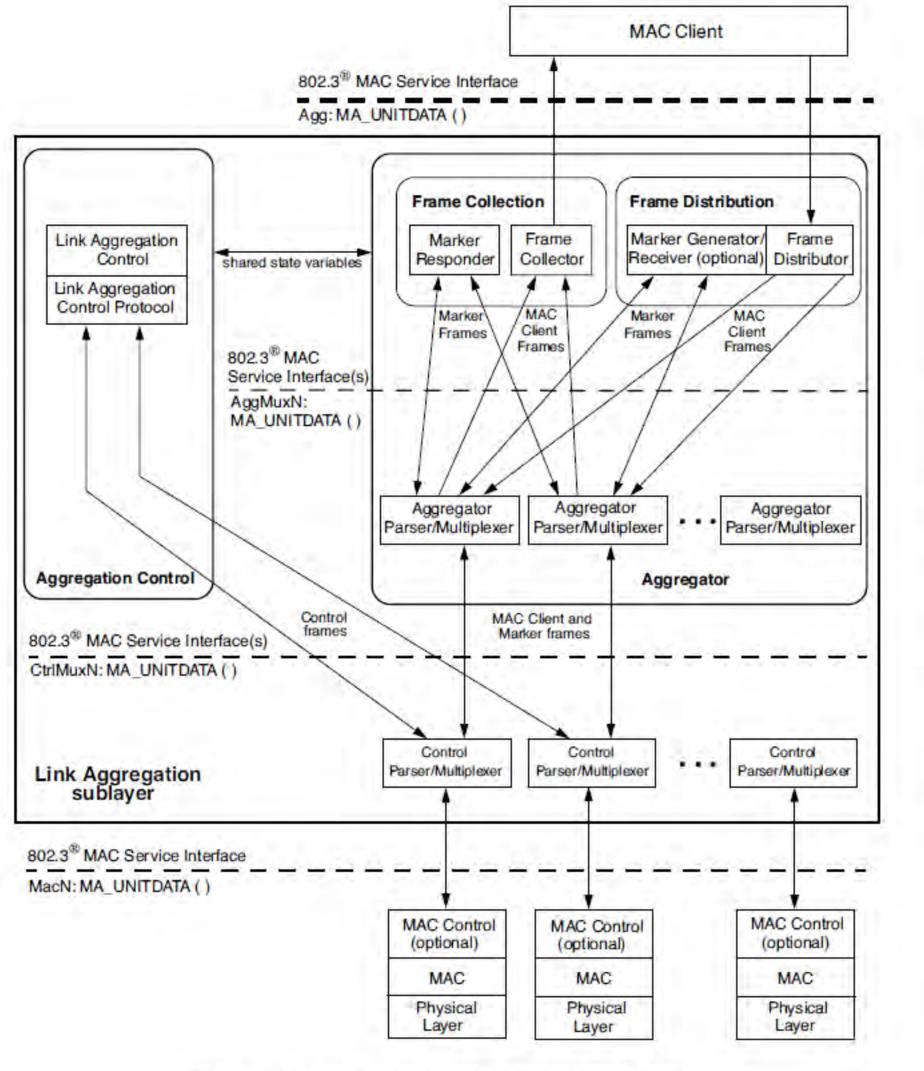
No.	’740 Patent Claim 1	IEEE 802.3
		 **43.2 Link Aggregation operation** As depicted in Figure 43–2, the Link Aggregation sublayer comprises the following functions: a) *Frame Distribution*. This block is responsible for taking frames submitted by the MAC Client and submitting them for transmission on the appropriate port, based on a frame distribution algorithm employed by the Frame Distributor. Frame Distribution also includes an optional *Marker Generator/Receiver* used for the Marker protocol. (See 43.2.4, 43.2.5, and 43.5.) b) *Frame Collection*. This block is responsible for passing frames received from the various ports to the MAC Client. Frame Collection also includes a *Marker Responder,* used for the Marker protocol. (See 43.2.3 and 43.5.) c) *Aggregator Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Distributor, Marker Generator, and/or Marker Responder to the appropriate port. On receive, these blocks distinguish among Marker Request, Marker Response, and MAC Client PDUs, and pass each to the appropriate entity (Marker Responder, Marker Receiver, and Collector, respectively). d) *Aggregator*. The combination of Frame Distribution and Collection, along with the Aggregator Parser/Multiplexers, is referred to as the Aggregator. e) *Aggregation Control*. This block is responsible for the configuration and control of Link Aggregation. It incorporates a *Link Aggregation Control Protocol* (LACP) that can be used for automatic communication of aggregation capabilities between Systems and automatic configuration of Link Aggregation. f) *Control Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Aggregator and Control entities to the appropriate port. On receive, these blocks distinguish Link Aggregation Control PDUs from other frames, passing the LACPDUs to the appropriate sublayer entity, and all other frames to the Aggregator. IEEE 802.3 at 1468-69

No.	’740 Patent Claim 1	IEEE 802.3
		 **43.2.1 Principles of Link Aggregation** Link Aggregation allows a MAC Client to treat a set of one or more ports as if it were a single port. In doing so, it employs the following principles and concepts: a) A MAC Client communicates with a set of ports through an Aggregator, which presents a standard IEEE 802.3® service interface to the MAC Client. The Aggregator binds to one or more ports within a System. b) It is the responsibility of the Aggregator to distribute frame transmissions from the MAC Client to the various ports, and to collect received frames from the ports and pass them to the MAC Client transparently. c) A System may contain multiple Aggregators, serving multiple MAC Clients. A given port will bind to (at most) a single Aggregator at any time. A MAC Client is served by a single Aggregator at a time. d) The binding of ports to Aggregators within a System is managed by the Link Aggregation Control function for that System, which is responsible for determining which links may be aggregated, aggregating them, binding the ports within the System to an appropriate Aggregator, and monitoring conditions to determine when a change in aggregation is needed. e) Such determination and binding may be under manual control through direct manipulation of the state variables of Link Aggregation (e.g., Keys) by a network manager. In addition, automatic determination, configuration, binding, and monitoring may occur through the use of a Link Aggregation Control Protocol (LACP). The LACP uses peer exchanges across the links to determine, on an ongoing basis, the aggregation capability of the various links, and continuously provides the maximum level of aggregation capability achievable between a given pair of Systems.

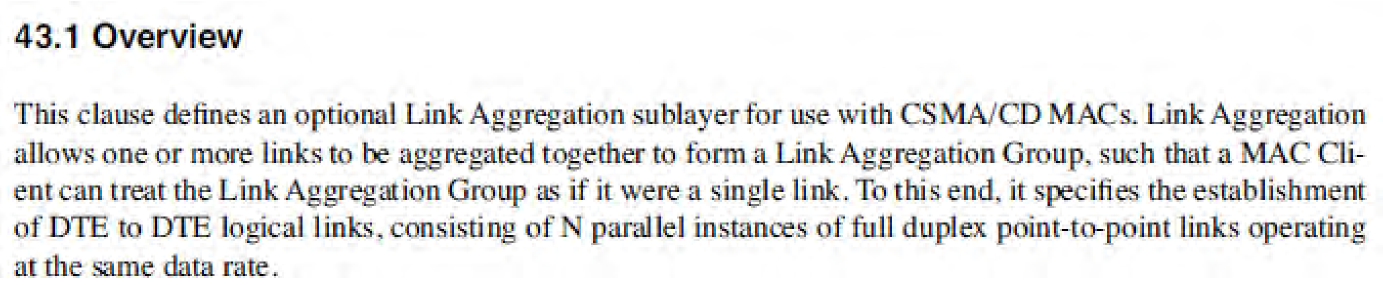
No.	’740 Patent Claim 1	IEEE 802.3
		 f) Frame ordering must be maintained for certain sequences of frame exchanges between MAC Clients (known as conversations, see 1.4). The Distributor ensures that all frames of a given conversation are passed to a single port. For any given port, the Collector is required to pass frames to the MAC Client in the order that they are received from that port. The Collector is otherwise free to select frames received from the aggregated ports in any order. Since there are no means for frames to be misordered on a single link, this guarantees that frame ordering is maintained for any conversation. g) Conversations may be moved among ports within an aggregation, both for load balancing and to maintain availability in the event of link failures. h) This standard does not impose any particular distribution algorithm on the Distributor. Whatever algorithm is used should be appropriate for the MAC Client being supported. i) Each port is assigned a unique, globally administered MAC address. This MAC address is used as the source address for frame exchanges that are initiated by entities within the Link Aggregation sublayer itself (i.e., LACP and Marker protocol exchanges). NOTE—The LACP and Marker protocols use a multicast destination address for all exchanges, and do not impose any requirement for a port to recognize more than one unicast address on received frames. j) Each Aggregator is assigned a unique, globally administered MAC address; this address is used as the MAC address of the aggregation from the perspective of the MAC Client, both as a source address for transmitted frames and as the destination address for received frames. The MAC address of the Aggregator may be one of the MAC addresses of a port in the associated Link Aggregation Group (see 43.2.10).
1[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel,	IEEE 802.3 discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. For example, IEEE 802.3 discloses a protocol for connecting a MAC Client to an aggregator via parallel links. IEEE 802.3 at 1465  43.1 Overview This clause defines an optional Link Aggregation sublayer for use with CSMA/CD MACs. Link Aggregation allows one or more links to be aggregated together to form a Link Aggregation Group, such that a MAC Client can treat the Link Aggregation Group as if it were a single link. To this end, it specifies the establishment of DTE to DTE logical links, consisting of N parallel instances of full duplex point-to-point links operating at the same data rate.

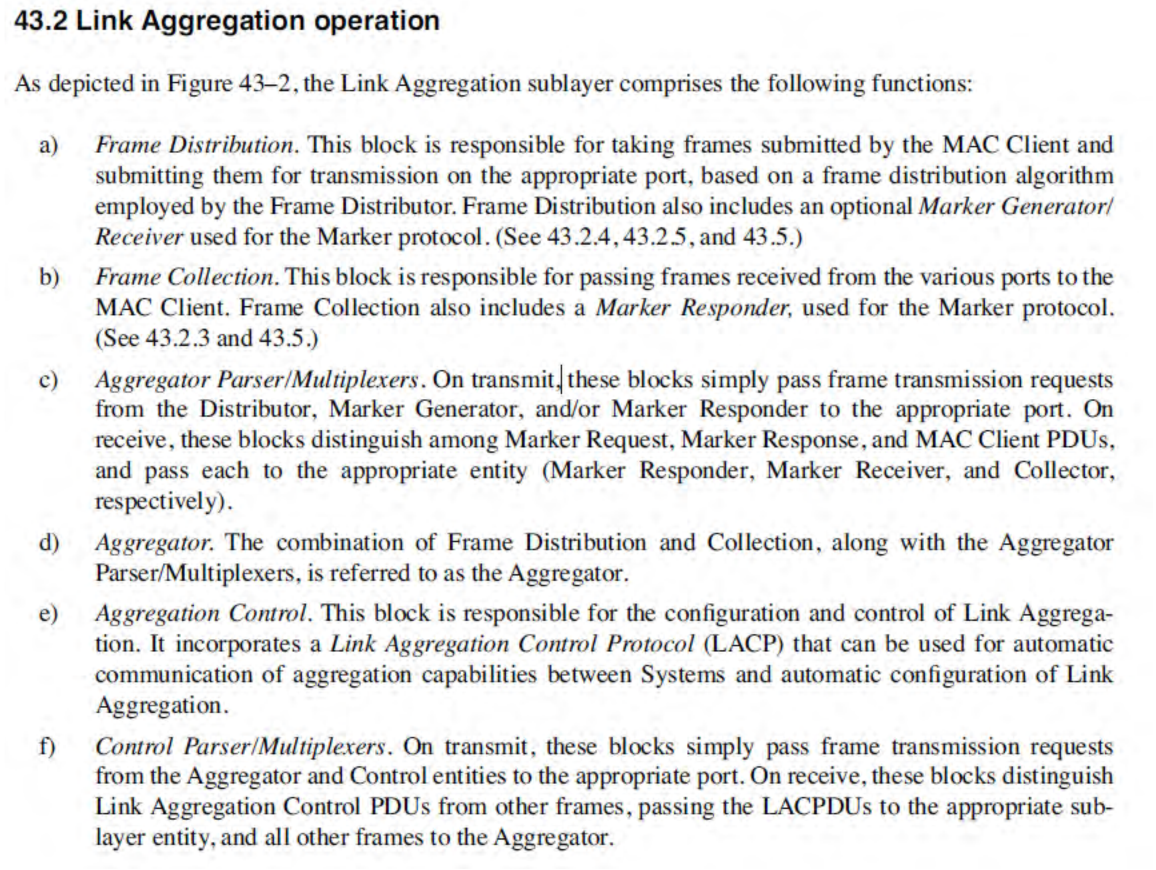
No.	’740 Patent Claim 1	IEEE 802.3
		IEEE 802.3 at 1470  **43.2.3 Frame Collector** A Frame Collector is responsible for receiving incoming frames (i.e., AggMuxN:MA_DATA.indications) from the set of individual links that form the Link Aggregation Group (through each link’s associated Aggregator Parser/Multiplexer) and delivering them to the MAC Client. Frames received from a given port are delivered to the MAC Client in the order that they are received by the Frame Collector. Since the Frame Distributor is responsible for maintaining any frame ordering constraints, there is no requirement for the Frame Collector to perform any reordering of frames received from multiple links. IEEE 802.3 at 1478  For each aggregatable port in the System, Link Aggregation Control a) Maintains configuration information (reflecting the inherent properties of the individual links as well as those established by management) to control aggregation. b) Exchanges configuration information with other Systems to allocate the link to a Link Aggregation Group. NOTE—A given link is allocated to, at most, one Link Aggregation Group at a time. The allocation mechanism attempts to maximize aggregation, subject to management controls. c) Attaches the port to the Aggregator used by the Link Aggregation Group, and detaches the port from the Aggregator when it is no longer used by the Group. d) Uses information from the Partner System’s Link Aggregation Control entity to enable or disable the Aggregator’s Collector and Distributor. IEEE 802.3 at 1478  Link Aggregation Control provides a configuration capability that is a) **Automatic.** In the absence of manual override controls, an appropriate set of Link Aggregation Groups is automatically configured, and individual links are allocated to those groups. If a set of links can aggregate, they do aggregate. IEEE 802.3 at 1481

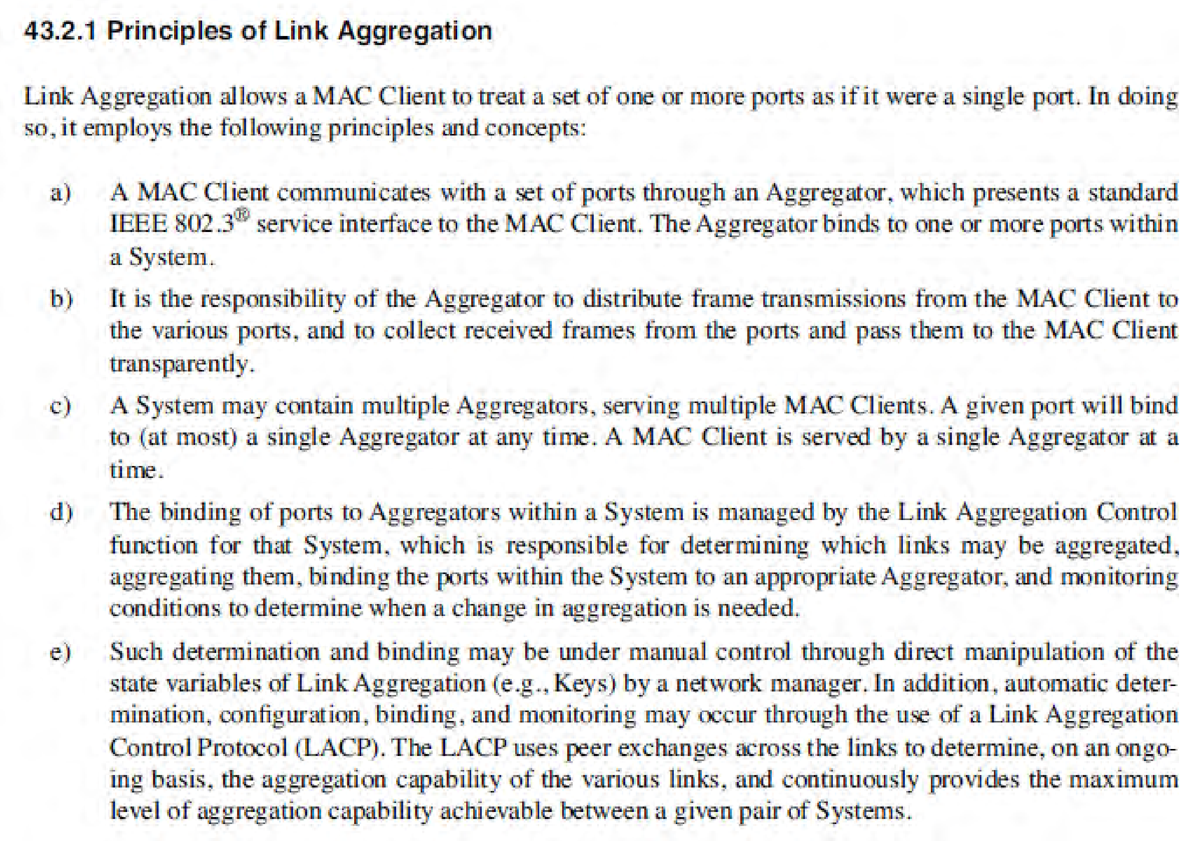
No.	’740 Patent Claim 1	IEEE 802.3
		 **43.3.6 Link Aggregation Group identification** A Link Aggregation Group consists of either a) One or more Aggregatable links that terminate in the same pair of Systems and whose ports belong to the same Key Group in each System, or b) An Individual link. IEEE 802.3 at Figure 43-2

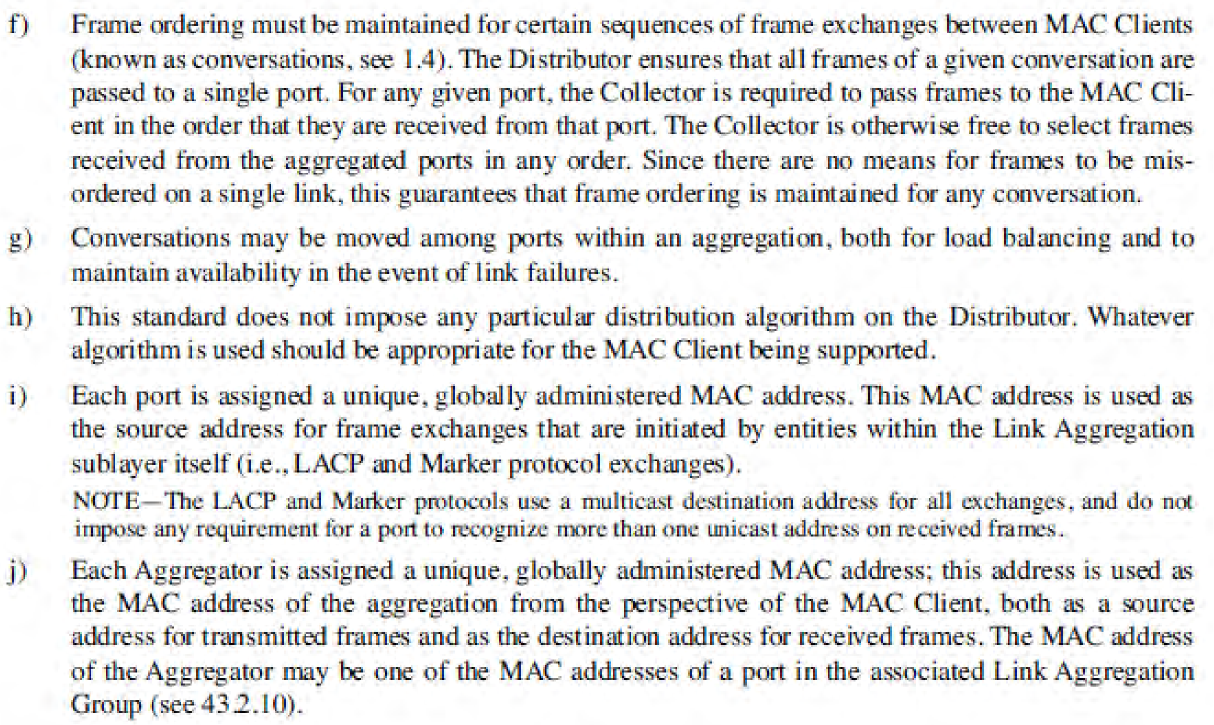
No.	’740 Patent Claim 1	IEEE 802.3
		 **Figure 43–2—Link Aggregation sublayer block diagram**

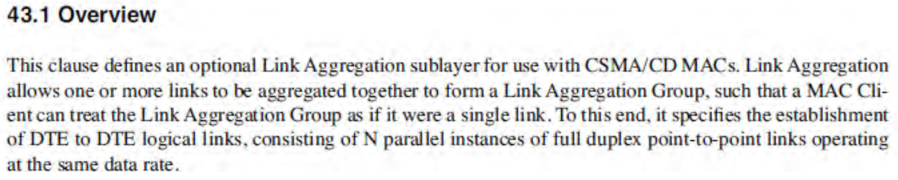
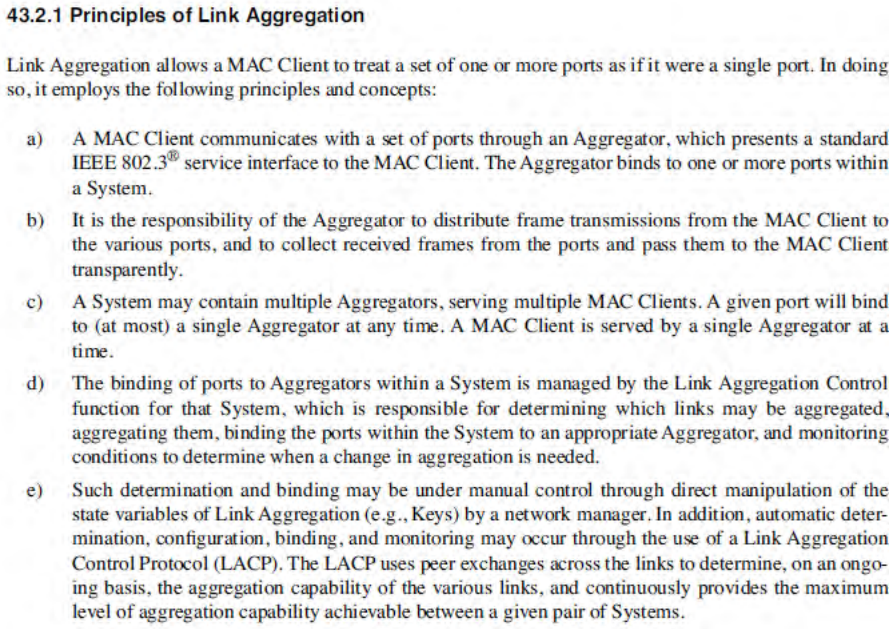
MAC Client
802.3® MAC Service Interface
Agg: MA_UNITDATA ( )
Link Aggregation Control
Link Aggregation Control Protocol
shared state variables
Frame Collection
Marker Responder
Frame Collector
Frame Distribution
Marker Generator/ Receiver (optional)
Frame Distributor
Marker Frames
MAC Client Frames
Marker Frames
MAC Client Frames
802.3® MAC Service Interface(s)
AggMuxN: MA_UNITDATA ( )
Aggregator Parser/Multiplexer
Aggregator Parser/Multiplexer
Aggregator Parser/Multiplexer
Aggregation Control
Aggregator
Control frames
MAC Client and Marker frames
802.3® MAC Service Interface(s)
CtrlMuxN: MA_UNITDATA ( )
Control Parser/Multiplexer
Control Parser/Multiplexer
Control Parser/Multiplexer
Link Aggregation sublayer
802.3® MAC Service Interface
MacN: MA_UNITDATA ( )
MAC Control (optional)
MAC
Physical Layer
MAC Control (optional)
MAC
Physical Layer
MAC Control (optional)
MAC
Physical Layer

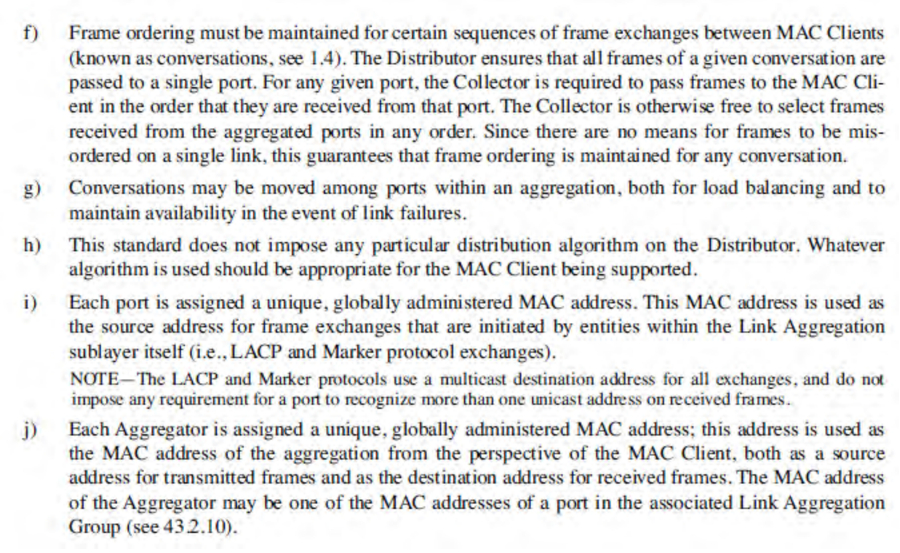
No.	’740 Patent Claim 1	IEEE 802.3
1[b]	at least one of said first physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction	IEEE 802.3 discloses at least one of said first physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction. For example, IEEE 802.3 discloses parallel instances of full duplex point-to-point links, which means exchanging network traffic in both directions. IEEE 802.3 at 1465  **43.1 Overview** This clause defines an optional Link Aggregation sublayer for use with CSMA/CD MACs. Link Aggregation allows one or more links to be aggregated together to form a Link Aggregation Group, such that a MAC Client can treat the Link Aggregation Group as if it were a single link. To this end, it specifies the establishment of DTE to DTE logical links, consisting of N parallel instances of full duplex point-to-point links operating at the same data rate. IEEE 802.3 at 1468

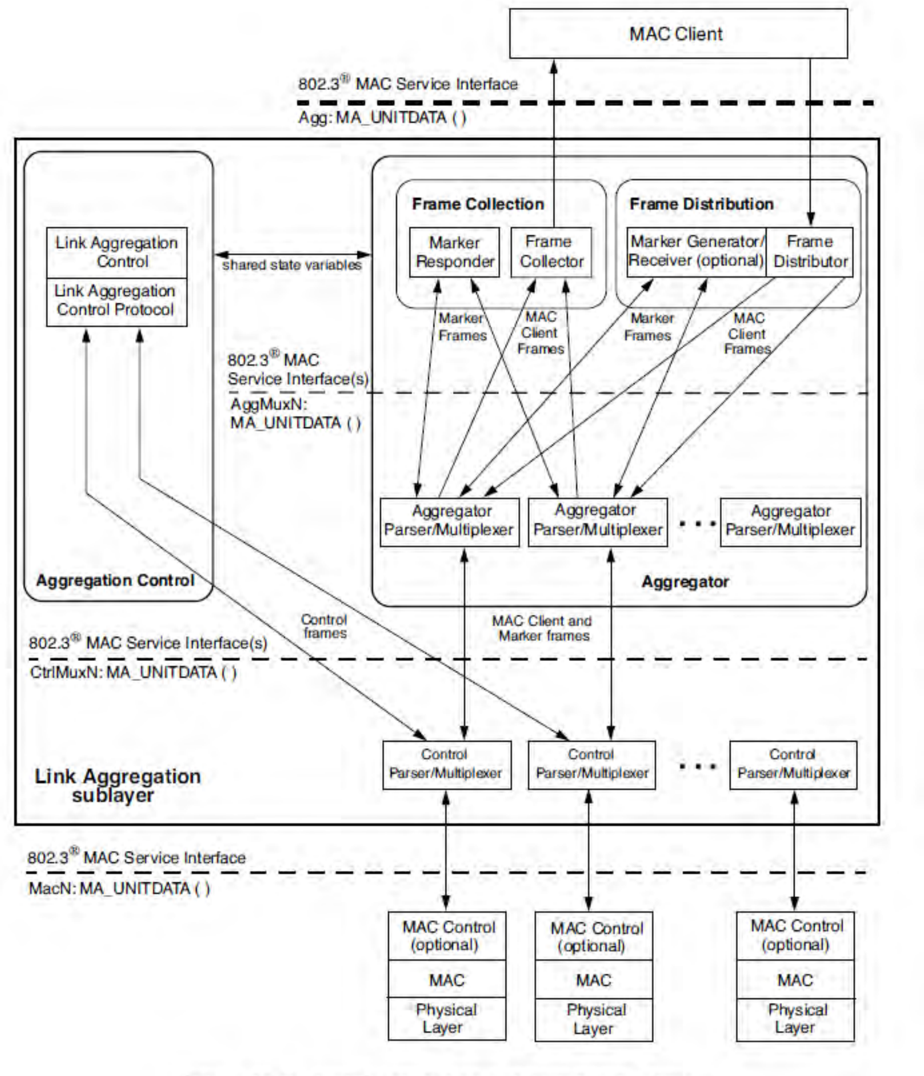
No.	’740 Patent Claim 1	IEEE 802.3
		 43.2 Link Aggregation operation As depicted in Figure 43–2, the Link Aggregation sublayer comprises the following functions: a) *Frame Distribution*. This block is responsible for taking frames submitted by the MAC Client and submitting them for transmission on the appropriate port, based on a frame distribution algorithm employed by the Frame Distributor. Frame Distribution also includes an optional *Marker Generator/Receiver* used for the Marker protocol. (See 43.2.4, 43.2.5, and 43.5.) b) *Frame Collection*. This block is responsible for passing frames received from the various ports to the MAC Client. Frame Collection also includes a *Marker Responder*, used for the Marker protocol. (See 43.2.3 and 43.5.) c) *Aggregator Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Distributor, Marker Generator, and/or Marker Responder to the appropriate port. On receive, these blocks distinguish among Marker Request, Marker Response, and MAC Client PDUs, and pass each to the appropriate entity (Marker Responder, Marker Receiver, and Collector, respectively). d) *Aggregator*. The combination of Frame Distribution and Collection, along with the Aggregator Parser/Multiplexers, is referred to as the Aggregator. e) *Aggregation Control*. This block is responsible for the configuration and control of Link Aggregation. It incorporates a *Link Aggregation Control Protocol* (LACP) that can be used for automatic communication of aggregation capabilities between Systems and automatic configuration of Link Aggregation. f) *Control Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Aggregator and Control entities to the appropriate port. On receive, these blocks distinguish Link Aggregation Control PDUs from other frames, passing the LACPDUs to the appropriate sublayer entity, and all other frames to the Aggregator. IEEE 802.3 at 1468-69

No.	’740 Patent Claim 1	IEEE 802.3
		 **43.2.1 Principles of Link Aggregation** Link Aggregation allows a MAC Client to treat a set of one or more ports as if it were a single port. In doing so, it employs the following principles and concepts: a) A MAC Client communicates with a set of ports through an Aggregator, which presents a standard IEEE 802.3® service interface to the MAC Client. The Aggregator binds to one or more ports within a System. b) It is the responsibility of the Aggregator to distribute frame transmissions from the MAC Client to the various ports, and to collect received frames from the ports and pass them to the MAC Client transparently. c) A System may contain multiple Aggregators, serving multiple MAC Clients. A given port will bind to (at most) a single Aggregator at any time. A MAC Client is served by a single Aggregator at a time. d) The binding of ports to Aggregators within a System is managed by the Link Aggregation Control function for that System, which is responsible for determining which links may be aggregated, aggregating them, binding the ports within the System to an appropriate Aggregator, and monitoring conditions to determine when a change in aggregation is needed. e) Such determination and binding may be under manual control through direct manipulation of the state variables of Link Aggregation (e.g., Keys) by a network manager. In addition, automatic determination, configuration, binding, and monitoring may occur through the use of a Link Aggregation Control Protocol (LACP). The LACP uses peer exchanges across the links to determine, on an ongoing basis, the aggregation capability of the various links, and continuously provides the maximum level of aggregation capability achievable between a given pair of Systems.

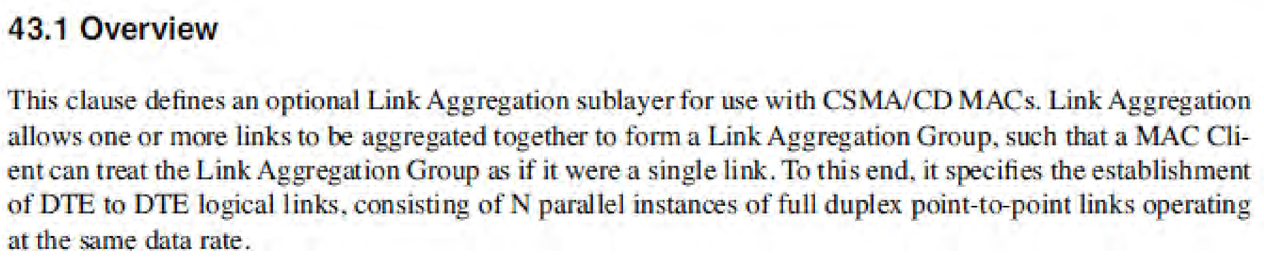
No.	’740 Patent Claim 1	IEEE 802.3
		 f) Frame ordering must be maintained for certain sequences of frame exchanges between MAC Clients (known as conversations, see 1.4). The Distributor ensures that all frames of a given conversation are passed to a single port. For any given port, the Collector is required to pass frames to the MAC Client in the order that they are received from that port. The Collector is otherwise free to select frames received from the aggregated ports in any order. Since there are no means for frames to be misordered on a single link, this guarantees that frame ordering is maintained for any conversation. g) Conversations may be moved among ports within an aggregation, both for load balancing and to maintain availability in the event of link failures. h) This standard does not impose any particular distribution algorithm on the Distributor. Whatever algorithm is used should be appropriate for the MAC Client being supported. i) Each port is assigned a unique, globally administered MAC address. This MAC address is used as the source address for frame exchanges that are initiated by entities within the Link Aggregation sublayer itself (i.e., LACP and Marker protocol exchanges). NOTE—The LACP and Marker protocols use a multicast destination address for all exchanges, and do not impose any requirement for a port to recognize more than one unicast address on received frames. j) Each Aggregator is assigned a unique, globally administered MAC address; this address is used as the MAC address of the aggregation from the perspective of the MAC Client, both as a source address for transmitted frames and as the destination address for received frames. The MAC address of the Aggregator may be one of the MAC addresses of a port in the associated Link Aggregation Group (see 43.2.10).
1[c]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel,	IEEE 802.3 discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. For example, IEEE 802.3 discloses connecting the aggregator to a network via full duplex point-to-point links. A person of ordinary skill in the art would understand from IEEE 802.3 that there are both connections between the aggregator and control parser/multiplexer. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the IEEE 802.3 is found to not meet this limitation, coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel would have been obvious to a person having ordinary skill in the art, as explained below.

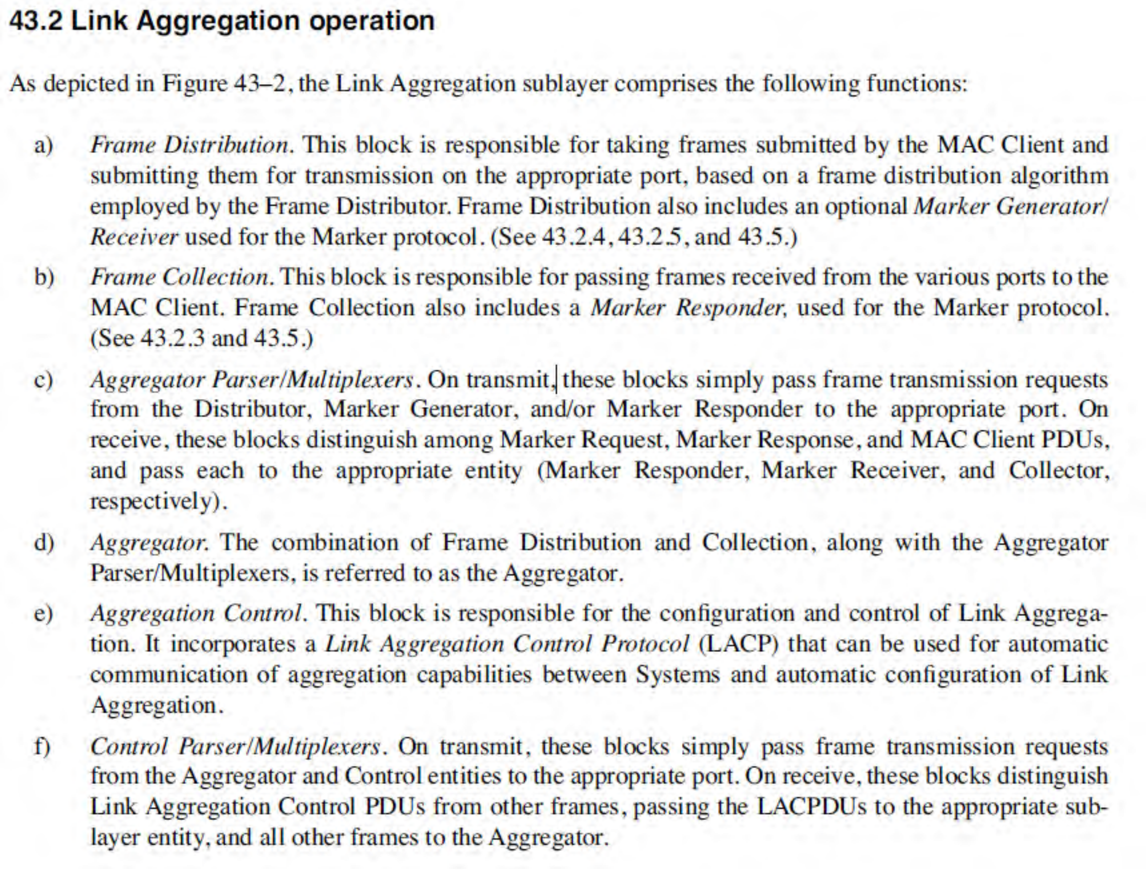
No.	’740 Patent Claim 1	IEEE 802.3
		IEEE 802.3 at 1465  **43.1 Overview** This clause defines an optional Link Aggregation sublayer for use with CSMA/CD MACs. Link Aggregation allows one or more links to be aggregated together to form a Link Aggregation Group, such that a MAC Client can treat the Link Aggregation Group as if it were a single link. To this end, it specifies the establishment of DTE to DTE logical links, consisting of N parallel instances of full duplex point-to-point links operating at the same data rate. IEEE 802.3 at 1468-69  **43.2.1 Principles of Link Aggregation** Link Aggregation allows a MAC Client to treat a set of one or more ports as if it were a single port. In doing so, it employs the following principles and concepts: a) A MAC Client communicates with a set of ports through an Aggregator, which presents a standard IEEE 802.3® service interface to the MAC Client. The Aggregator binds to one or more ports within a System. b) It is the responsibility of the Aggregator to distribute frame transmissions from the MAC Client to the various ports, and to collect received frames from the ports and pass them to the MAC Client transparently. c) A System may contain multiple Aggregators, serving multiple MAC Clients. A given port will bind to (at most) a single Aggregator at any time. A MAC Client is served by a single Aggregator at a time. d) The binding of ports to Aggregators within a System is managed by the Link Aggregation Control function for that System, which is responsible for determining which links may be aggregated, aggregating them, binding the ports within the System to an appropriate Aggregator, and monitoring conditions to determine when a change in aggregation is needed. e) Such determination and binding may be under manual control through direct manipulation of the state variables of Link Aggregation (e.g., Keys) by a network manager. In addition, automatic determination, configuration, binding, and monitoring may occur through the use of a Link Aggregation Control Protocol (LACP). The LACP uses peer exchanges across the links to determine, on an ongoing basis, the aggregation capability of the various links, and continuously provides the maximum level of aggregation capability achievable between a given pair of Systems.

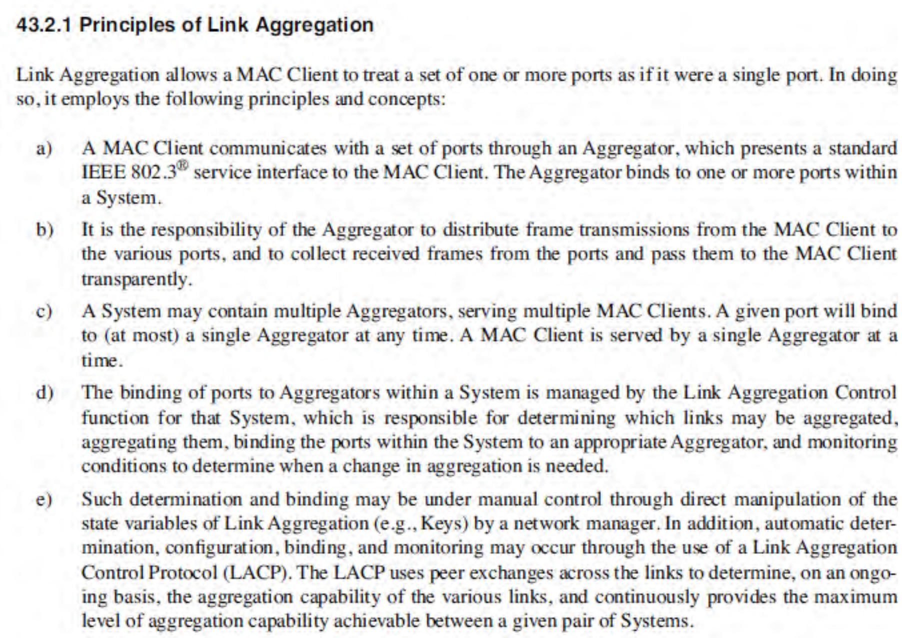
No.	’740 Patent Claim 1	IEEE 802.3
		 f) Frame ordering must be maintained for certain sequences of frame exchanges between MAC Clients (known as conversations, see 1.4). The Distributor ensures that all frames of a given conversation are passed to a single port. For any given port, the Collector is required to pass frames to the MAC Client in the order that they are received from that port. The Collector is otherwise free to select frames received from the aggregated ports in any order. Since there are no means for frames to be misordered on a single link, this guarantees that frame ordering is maintained for any conversation. g) Conversations may be moved among ports within an aggregation, both for load balancing and to maintain availability in the event of link failures. h) This standard does not impose any particular distribution algorithm on the Distributor. Whatever algorithm is used should be appropriate for the MAC Client being supported. i) Each port is assigned a unique, globally administered MAC address. This MAC address is used as the source address for frame exchanges that are initiated by entities within the Link Aggregation sublayer itself (i.e., LACP and Marker protocol exchanges). NOTE—The LACP and Marker protocols use a multicast destination address for all exchanges, and do not impose any requirement for a port to recognize more than one unicast address on received frames. j) Each Aggregator is assigned a unique, globally administered MAC address; this address is used as the MAC address of the aggregation from the perspective of the MAC Client, both as a source address for transmitted frames and as the destination address for received frames. The MAC address of the Aggregator may be one of the MAC addresses of a port in the associated Link Aggregation Group (see 43.2.10). IEEE 802.3 at Figure 43-2

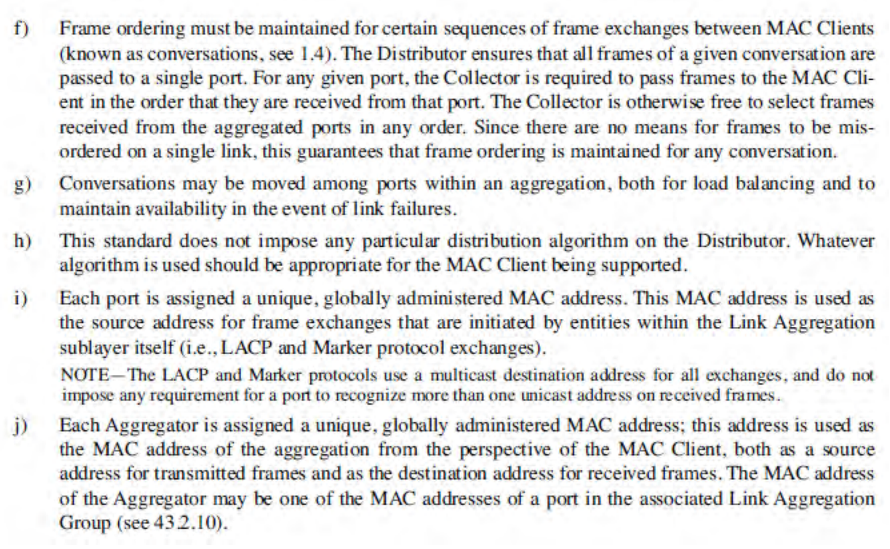
No.	’740 Patent Claim 1	IEEE 802.3
		 MAC Client 802.3® MAC Service Interface Agg: MA_UNITDATA ( ) Frame Collection Marker Responder Frame Collector Frame Distribution Marker Generator/ Receiver (optional) Frame Distributor Link Aggregation Control Link Aggregation Control Protocol shared state variables Marker Frames MAC Client Frames Marker Frames MAC Client Frames 802.3® MAC Service Interface(s) AggMuxN: MA_UNITDATA ( ) Aggregator Parser/Multiplexer Aggregator Parser/Multiplexer Aggregator Parser/Multiplexer Aggregation Control Aggregator Control frames MAC Client and Marker frames 802.3® MAC Service Interface(s) CtrlMuxN: MA_UNITDATA ( ) Control Parser/Multiplexer Control Parser/Multiplexer Control Parser/Multiplexer Link Aggregation sublayer 802.3® MAC Service Interface MacN: MA_UNITDATA ( ) MAC Control (optional) MAC Physical Layer MAC Control (optional) MAC Physical Layer MAC Control (optional) MAC Physical Layer **Figure 43–2—Link Aggregation sublayer block diagram**

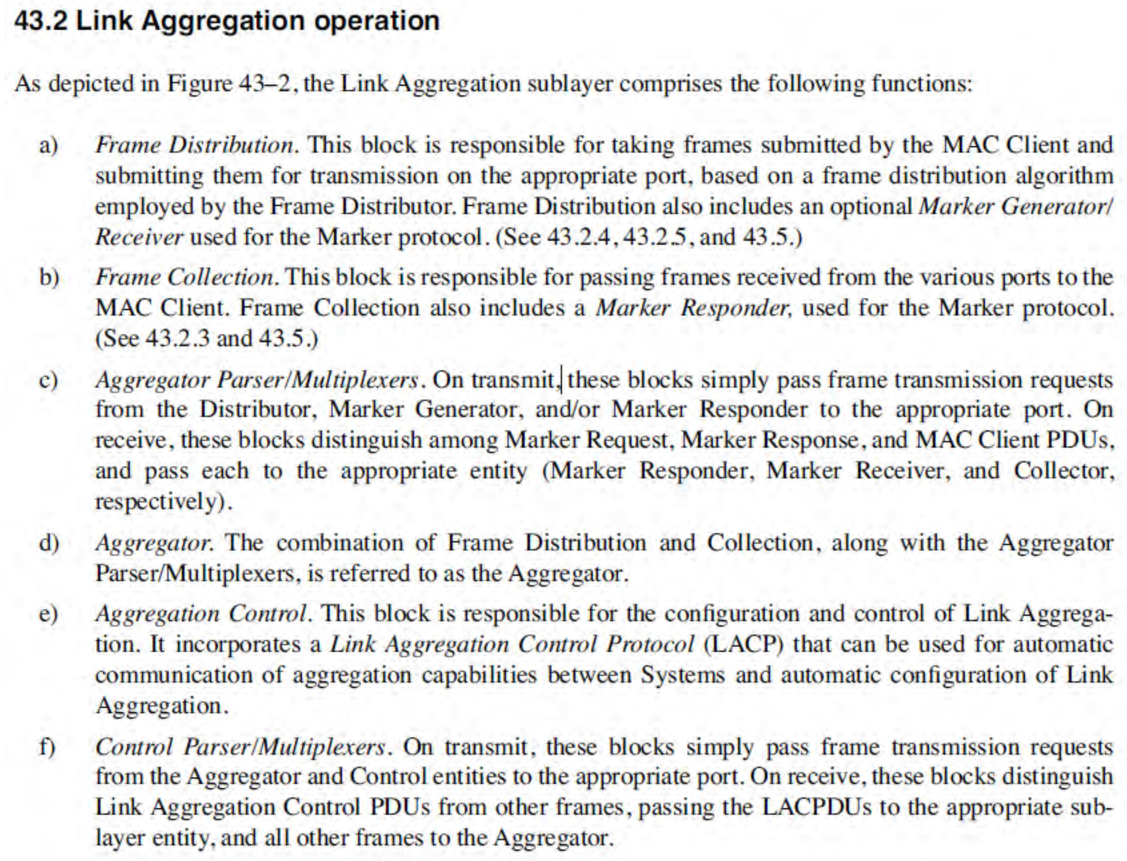
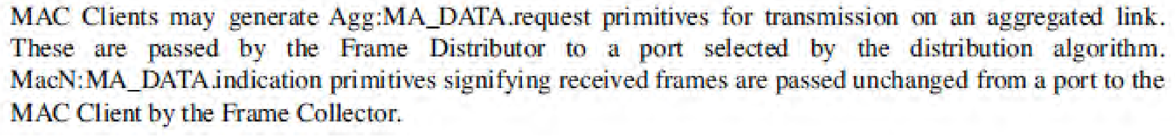
No.	’740 Patent Claim 1	IEEE 802.3
		Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, IEEE 802.3 in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 1[c] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Ghosh discloses connecting the line cards, i.e., interface modules, to the active supervisor via the backplane using parallel interface circuitry. Ghosh at [0059] (“Line cards 803, 805, and 807 can communicate with an active supervisor 811 through interface circuitry 883, 885, and 887 and the backplane 815. According to various embodiments, each line card includes a plurality of ports that can act as either input ports or output ports for communication with external fibre channel network entities 851 and 853. The backplane 815 can provide a communi-cations channel for all traffic between line cards and super- visors. Individual line cards 803 and 807 can also be coupled to external fibre channel network entities 851 and 853 through fibre channel ports 843 and 847.”) For example, Bruckman discloses connecting the line cards to the network using traces comprising a backplane. Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”)

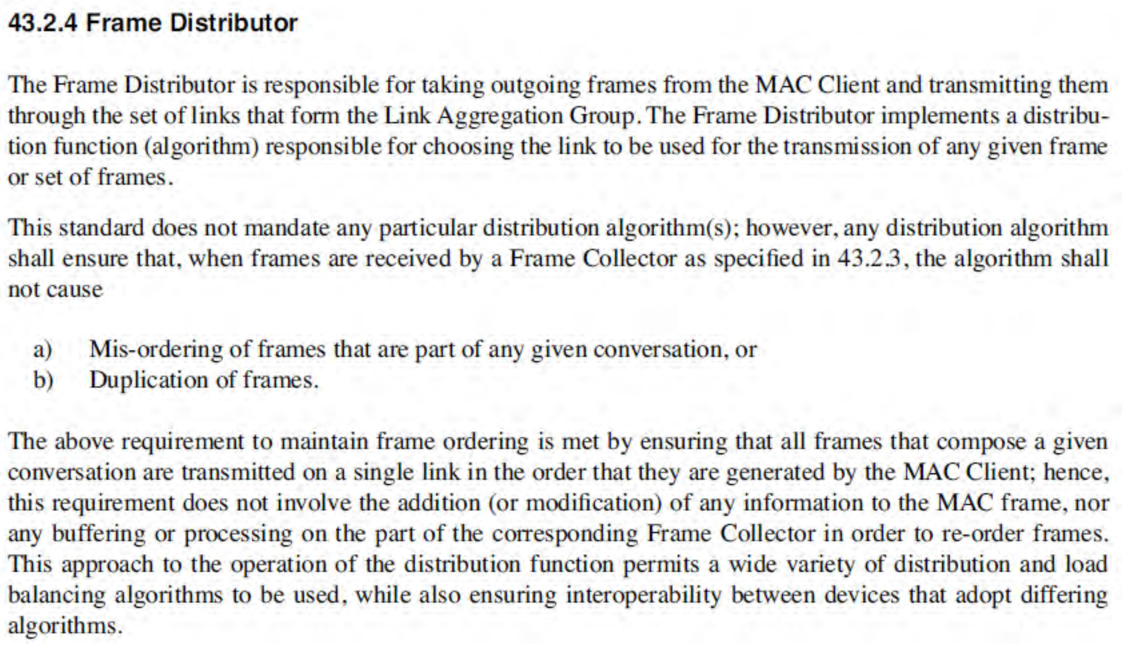
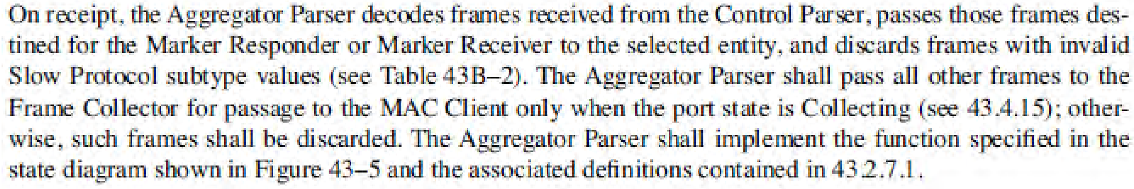
No.	’740 Patent Claim 1	IEEE 802.3
		For example, Basso discloses coupling the blades to the communication network via link connections to the switch fabric. Basso at [0009] (“A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more ports. These ports may be connected to another one or more network devices. The switch fabric may be configured to direct incoming packets of data to particular blades where one or more of the network proces-sors in the recipient blade may be configured to process the received packets.”)
1[d]	at least one of said second physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction;	IEEE 802.3 discloses at least one of said second physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction. For example, IEEE 802.3 discloses parallel instances of full duplex point-to-point links. As described above in limitation 1[c], IEEE 802.3 further discloses a second group of physical links. IEEE 802.3 at 1465  **43.1 Overview** This clause defines an optional Link Aggregation sublayer for use with CSMA/CD MACs. Link Aggregation allows one or more links to be aggregated together to form a Link Aggregation Group, such that a MAC Client can treat the Link Aggregation Group as if it were a single link. To this end, it specifies the establishment of DTE to DTE logical links, consisting of N parallel instances of full duplex point-to-point links operating at the same data rate. IEEE 802.3 at 1468

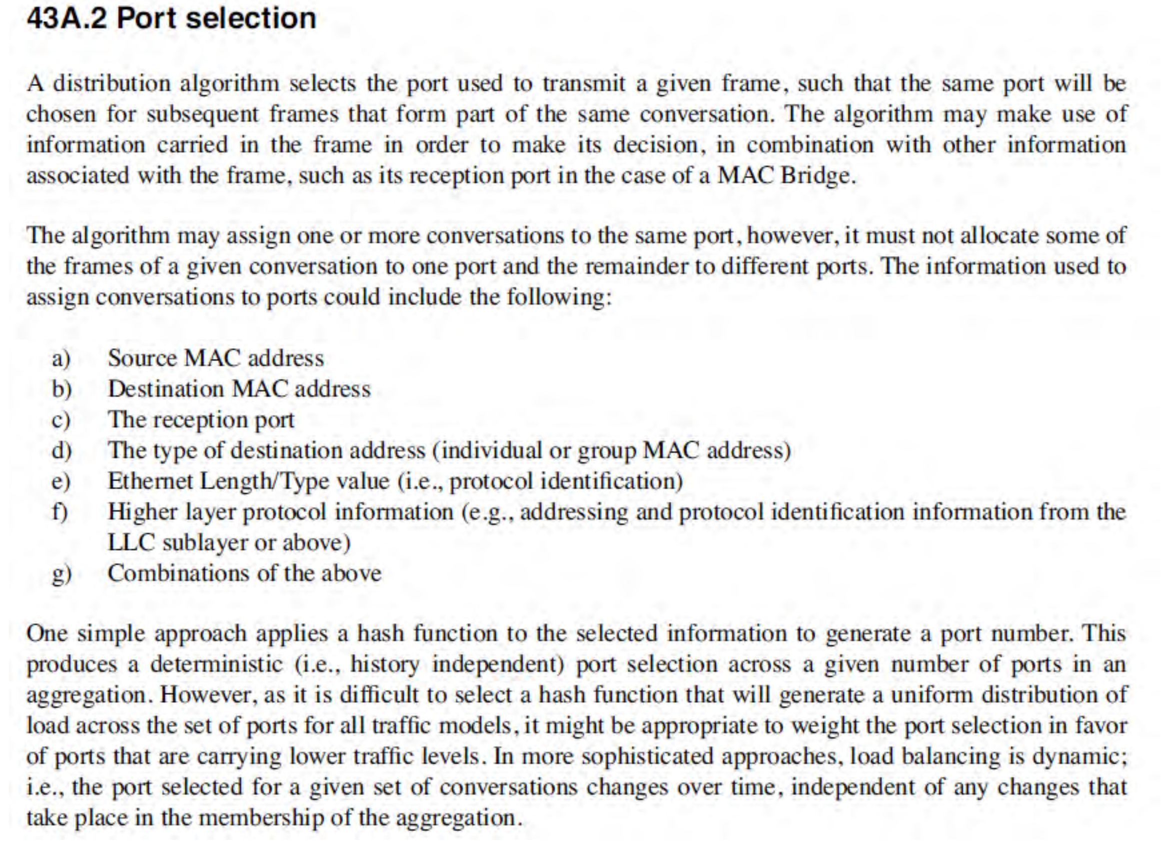
No.	’740 Patent Claim 1	IEEE 802.3
		 **43.2 Link Aggregation operation** As depicted in Figure 43–2, the Link Aggregation sublayer comprises the following functions: a) *Frame Distribution*. This block is responsible for taking frames submitted by the MAC Client and submitting them for transmission on the appropriate port, based on a frame distribution algorithm employed by the Frame Distributor. Frame Distribution also includes an optional *Marker Generator/ Receiver* used for the Marker protocol. (See 43.2.4, 43.2.5, and 43.5.) b) *Frame Collection*. This block is responsible for passing frames received from the various ports to the MAC Client. Frame Collection also includes a *Marker Responder,* used for the Marker protocol. (See 43.2.3 and 43.5.) c) *Aggregator Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Distributor, Marker Generator, and/or Marker Responder to the appropriate port. On receive, these blocks distinguish among Marker Request, Marker Response, and MAC Client PDUs, and pass each to the appropriate entity (Marker Responder, Marker Receiver, and Collector, respectively). d) *Aggregator*. The combination of Frame Distribution and Collection, along with the Aggregator Parser/Multiplexers, is referred to as the Aggregator. e) *Aggregation Control*. This block is responsible for the configuration and control of Link Aggregation. It incorporates a *Link Aggregation Control Protocol* (LACP) that can be used for automatic communication of aggregation capabilities between Systems and automatic configuration of Link Aggregation. f) *Control Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Aggregator and Control entities to the appropriate port. On receive, these blocks distinguish Link Aggregation Control PDUs from other frames, passing the LACPDUs to the appropriate sublayer entity, and all other frames to the Aggregator. IEEE 802.3 at 1468-69

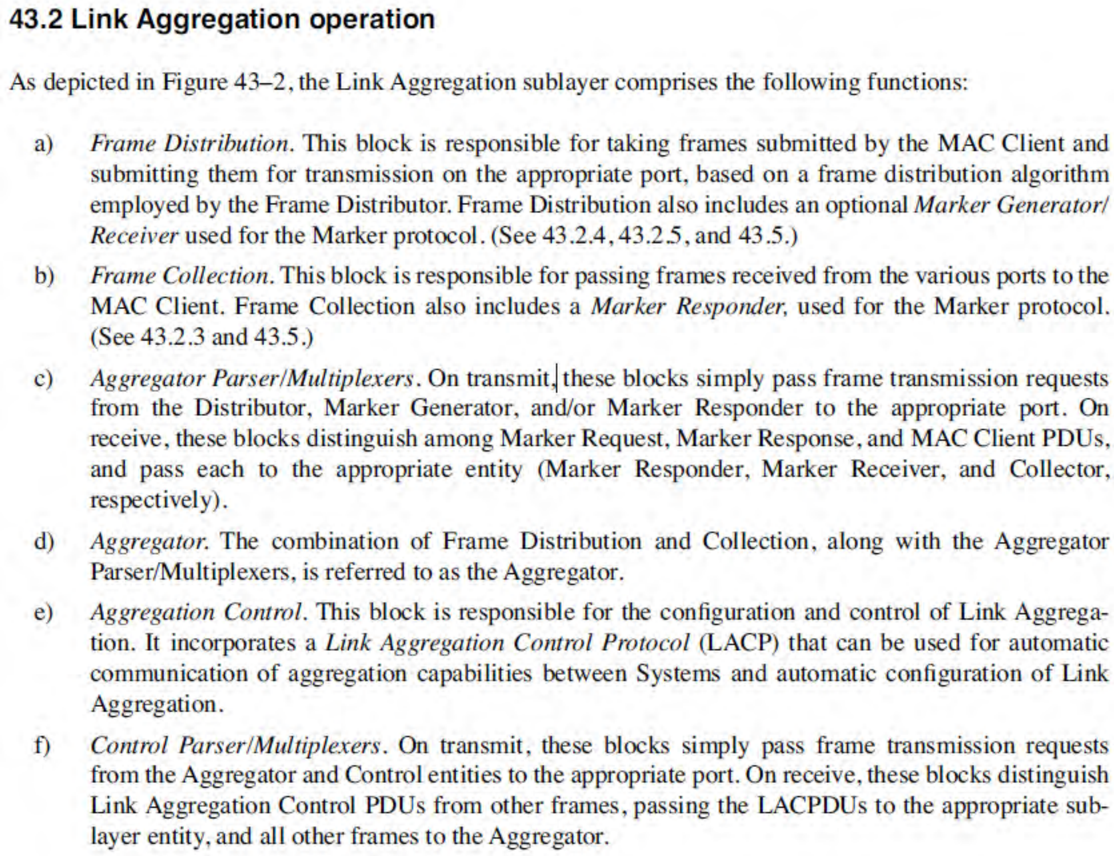
No.	’740 Patent Claim 1	IEEE 802.3
		 **43.2.1 Principles of Link Aggregation** Link Aggregation allows a MAC Client to treat a set of one or more ports as if it were a single port. In doing so, it employs the following principles and concepts: a) A MAC Client communicates with a set of ports through an Aggregator, which presents a standard IEEE 802.3® service interface to the MAC Client. The Aggregator binds to one or more ports within a System. b) It is the responsibility of the Aggregator to distribute frame transmissions from the MAC Client to the various ports, and to collect received frames from the ports and pass them to the MAC Client transparently. c) A System may contain multiple Aggregators, serving multiple MAC Clients. A given port will bind to (at most) a single Aggregator at any time. A MAC Client is served by a single Aggregator at a time. d) The binding of ports to Aggregators within a System is managed by the Link Aggregation Control function for that System, which is responsible for determining which links may be aggregated, aggregating them, binding the ports within the System to an appropriate Aggregator, and monitoring conditions to determine when a change in aggregation is needed. e) Such determination and binding may be under manual control through direct manipulation of the state variables of Link Aggregation (e.g., Keys) by a network manager. In addition, automatic determination, configuration, binding, and monitoring may occur through the use of a Link Aggregation Control Protocol (LACP). The LACP uses peer exchanges across the links to determine, on an ongoing basis, the aggregation capability of the various links, and continuously provides the maximum level of aggregation capability achievable between a given pair of Systems.

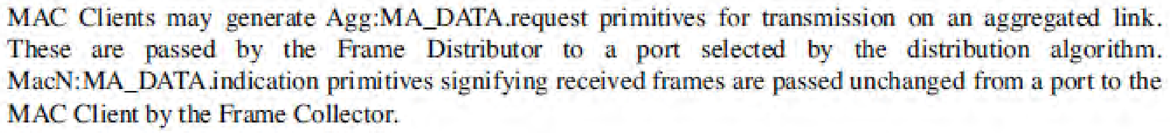
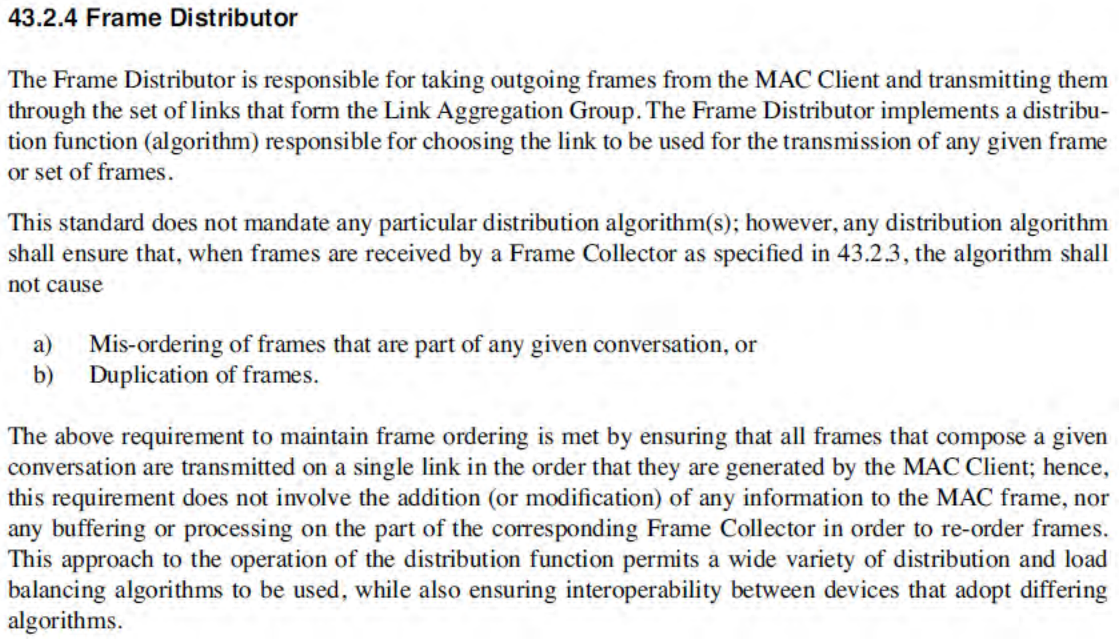
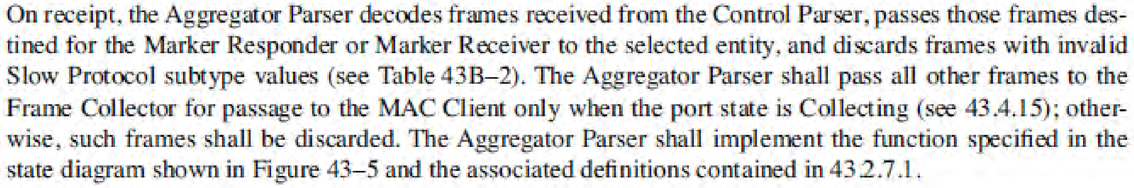
No.	’740 Patent Claim 1	IEEE 802.3
		 f) Frame ordering must be maintained for certain sequences of frame exchanges between MAC Clients (known as conversations, see 1.4). The Distributor ensures that all frames of a given conversation are passed to a single port. For any given port, the Collector is required to pass frames to the MAC Client in the order that they are received from that port. The Collector is otherwise free to select frames received from the aggregated ports in any order. Since there are no means for frames to be misordered on a single link, this guarantees that frame ordering is maintained for any conversation. g) Conversations may be moved among ports within an aggregation, both for load balancing and to maintain availability in the event of link failures. h) This standard does not impose any particular distribution algorithm on the Distributor. Whatever algorithm is used should be appropriate for the MAC Client being supported. i) Each port is assigned a unique, globally administered MAC address. This MAC address is used as the source address for frame exchanges that are initiated by entities within the Link Aggregation sublayer itself (i.e., LACP and Marker protocol exchanges). NOTE—The LACP and Marker protocols use a multicast destination address for all exchanges, and do not impose any requirement for a port to recognize more than one unicast address on received frames. j) Each Aggregator is assigned a unique, globally administered MAC address; this address is used as the MAC address of the aggregation from the perspective of the MAC Client, both as a source address for transmitted frames and as the destination address for received frames. The MAC address of the Aggregator may be one of the MAC addresses of a port in the associated Link Aggregation Group (see 43.2.10).
1[e]	receiving a data frame having frame attributes sent between the communication network and the network node:	IEEE 802.3 discloses receiving a data frame having frame attributes sent between the communication network and the network node. For example, IEEE 802.3 discloses receiving a data frame sent between the network and MAC Client having information used to distribute frames. IEEE 802.3 at 1468

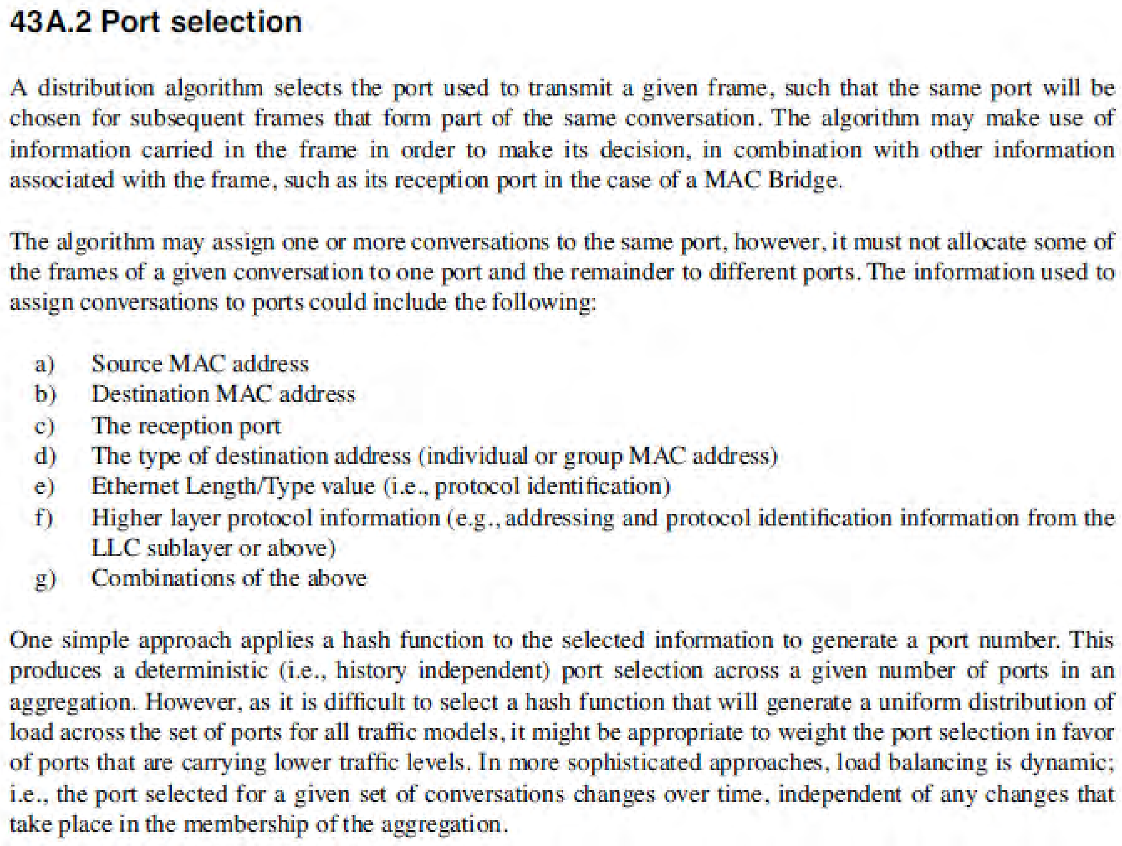
No.	’740 Patent Claim 1	IEEE 802.3
		 **43.2 Link Aggregation operation** As depicted in Figure 43–2, the Link Aggregation sublayer comprises the following functions: a) *Frame Distribution*. This block is responsible for taking frames submitted by the MAC Client and submitting them for transmission on the appropriate port, based on a frame distribution algorithm employed by the Frame Distributor. Frame Distribution also includes an optional *Marker Generator/Receiver* used for the Marker protocol. (See 43.2.4, 43.2.5, and 43.5.) b) *Frame Collection*. This block is responsible for passing frames received from the various ports to the MAC Client. Frame Collection also includes a *Marker Responder*, used for the Marker protocol. (See 43.2.3 and 43.5.) c) *Aggregator Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Distributor, Marker Generator, and/or Marker Responder to the appropriate port. On receive, these blocks distinguish among Marker Request, Marker Response, and MAC Client PDUs, and pass each to the appropriate entity (Marker Responder, Marker Receiver, and Collector, respectively). d) *Aggregator*. The combination of Frame Distribution and Collection, along with the Aggregator Parser/Multiplexers, is referred to as the Aggregator. e) *Aggregation Control*. This block is responsible for the configuration and control of Link Aggregation. It incorporates a *Link Aggregation Control Protocol* (LACP) that can be used for automatic communication of aggregation capabilities between Systems and automatic configuration of Link Aggregation. f) *Control Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Aggregator and Control entities to the appropriate port. On receive, these blocks distinguish Link Aggregation Control PDUs from other frames, passing the LACPDUs to the appropriate sublayer entity, and all other frames to the Aggregator. IEEE 802.3 at 1469  MAC Clients may generate Agg:MA_DATA.request primitives for transmission on an aggregated link. These are passed by the Frame Distributor to a port selected by the distribution algorithm. MacN:MA_DATA.indication primitives signifying received frames are passed unchanged from a port to the MAC Client by the Frame Collector. IEEE 802.3 at 1471

No.	’740 Patent Claim 1	IEEE 802.3
		 **43.2.4 Frame Distributor** The Frame Distributor is responsible for taking outgoing frames from the MAC Client and transmitting them through the set of links that form the Link Aggregation Group. The Frame Distributor implements a distribution function (algorithm) responsible for choosing the link to be used for the transmission of any given frame or set of frames. This standard does not mandate any particular distribution algorithm(s); however, any distribution algorithm shall ensure that, when frames are received by a Frame Collector as specified in 43.2.3, the algorithm shall not cause a) Mis-ordering of frames that are part of any given conversation, or b) Duplication of frames. The above requirement to maintain frame ordering is met by ensuring that all frames that compose a given conversation are transmitted on a single link in the order that they are generated by the MAC Client; hence, this requirement does not involve the addition (or modification) of any information to the MAC frame, nor any buffering or processing on the part of the corresponding Frame Collector in order to re-order frames. This approach to the operation of the distribution function permits a wide variety of distribution and load balancing algorithms to be used, while also ensuring interoperability between devices that adopt differing algorithms. IEEE 802.3 at 1473  On receipt, the Aggregator Parser decodes frames received from the Control Parser, passes those frames destined for the Marker Responder or Marker Receiver to the selected entity, and discards frames with invalid Slow Protocol subtype values (see Table 43B–2). The Aggregator Parser shall pass all other frames to the Frame Collector for passage to the MAC Client only when the port state is Collecting (see 43.4.15); otherwise, such frames shall be discarded. The Aggregator Parser shall implement the function specified in the state diagram shown in Figure 43–5 and the associated definitions contained in 43.2.7.1. IEEE 802.3 at 1549

No.	’740 Patent Claim 1	IEEE 802.3
		 43A.2 Port selection A distribution algorithm selects the port used to transmit a given frame, such that the same port will be chosen for subsequent frames that form part of the same conversation. The algorithm may make use of information carried in the frame in order to make its decision, in combination with other information associated with the frame, such as its reception port in the case of a MAC Bridge. The algorithm may assign one or more conversations to the same port, however, it must not allocate some of the frames of a given conversation to one port and the remainder to different ports. The information used to assign conversations to ports could include the following: a) Source MAC address b) Destination MAC address c) The reception port d) The type of destination address (individual or group MAC address) e) Ethernet Length/Type value (i.e., protocol identification) f) Higher layer protocol information (e.g., addressing and protocol identification information from the LLC sublayer or above) g) Combinations of the above One simple approach applies a hash function to the selected information to generate a port number. This produces a deterministic (i.e., history independent) port selection across a given number of ports in an aggregation. However, as it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models, it might be appropriate to weight the port selection in favor of ports that are carrying lower traffic levels. In more sophisticated approaches, load balancing is dynamic; i.e., the port selected for a given set of conversations changes over time, independent of any changes that take place in the membership of the aggregation.
1[f]:	selecting, in a single computation based on at least one of the frame	IEEE 802.3 discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. For example, IEEE 802.3 discloses a distribution algorithm that selects physical links over which to send data frames based on specific frame information. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the IEEE 802.3 is found to not

No.	’740 Patent Claim 1	IEEE 802.3
	attributes, a first physical link out of the first group and a second physical link out of the second group; and	meet this limitation, selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group would have been obvious to a person having ordinary skill in the art, as explained below. IEEE 802.3 at 1468  **43.2 Link Aggregation operation** As depicted in Figure 43–2, the Link Aggregation sublayer comprises the following functions: a) *Frame Distribution*. This block is responsible for taking frames submitted by the MAC Client and submitting them for transmission on the appropriate port, based on a frame distribution algorithm employed by the Frame Distributor. Frame Distribution also includes an optional *Marker Generator/Receiver* used for the Marker protocol. (See 43.2.4, 43.2.5, and 43.5.) b) *Frame Collection*. This block is responsible for passing frames received from the various ports to the MAC Client. Frame Collection also includes a *Marker Responder*, used for the Marker protocol. (See 43.2.3 and 43.5.) c) *Aggregator Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Distributor, Marker Generator, and/or Marker Responder to the appropriate port. On receive, these blocks distinguish among Marker Request, Marker Response, and MAC Client PDUs, and pass each to the appropriate entity (Marker Responder, Marker Receiver, and Collector, respectively). d) *Aggregator*. The combination of Frame Distribution and Collection, along with the Aggregator Parser/Multiplexers, is referred to as the Aggregator. e) *Aggregation Control*. This block is responsible for the configuration and control of Link Aggregation. It incorporates a *Link Aggregation Control Protocol* (LACP) that can be used for automatic communication of aggregation capabilities between Systems and automatic configuration of Link Aggregation. f) *Control Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Aggregator and Control entities to the appropriate port. On receive, these blocks distinguish Link Aggregation Control PDUs from other frames, passing the LACPDUs to the appropriate sublayer entity, and all other frames to the Aggregator. IEEE 802.3 at 1469

No.	’740 Patent Claim 1	IEEE 802.3
		 MAC Clients may generate Agg:MA_DATA.request primitives for transmission on an aggregated link. These are passed by the Frame Distributor to a port selected by the distribution algorithm. MacN:MA_DATA.indication primitives signifying received frames are passed unchanged from a port to the MAC Client by the Frame Collector. IEEE 802.3 at 1471  **43.2.4 Frame Distributor** The Frame Distributor is responsible for taking outgoing frames from the MAC Client and transmitting them through the set of links that form the Link Aggregation Group. The Frame Distributor implements a distribution function (algorithm) responsible for choosing the link to be used for the transmission of any given frame or set of frames. This standard does not mandate any particular distribution algorithm(s); however, any distribution algorithm shall ensure that, when frames are received by a Frame Collector as specified in 43.2.3, the algorithm shall not cause a) Mis-ordering of frames that are part of any given conversation, or b) Duplication of frames. The above requirement to maintain frame ordering is met by ensuring that all frames that compose a given conversation are transmitted on a single link in the order that they are generated by the MAC Client; hence, this requirement does not involve the addition (or modification) of any information to the MAC frame, nor any buffering or processing on the part of the corresponding Frame Collector in order to re-order frames. This approach to the operation of the distribution function permits a wide variety of distribution and load balancing algorithms to be used, while also ensuring interoperability between devices that adopt differing algorithms. IEEE 802.3 at 1473  On receipt, the Aggregator Parser decodes frames received from the Control Parser, passes those frames destined for the Marker Responder or Marker Receiver to the selected entity, and discards frames with invalid Slow Protocol subtype values (see Table 43B–2). The Aggregator Parser shall pass all other frames to the Frame Collector for passage to the MAC Client only when the port state is Collecting (see 43.4.15); otherwise, such frames shall be discarded. The Aggregator Parser shall implement the function specified in the state diagram shown in Figure 43–5 and the associated definitions contained in 43.2.7.1.

No.	’740 Patent Claim 1	IEEE 802.3
		IEEE 802.3 at 1549  **43A.2 Port selection** A distribution algorithm selects the port used to transmit a given frame, such that the same port will be chosen for subsequent frames that form part of the same conversation. The algorithm may make use of information carried in the frame in order to make its decision, in combination with other information associated with the frame, such as its reception port in the case of a MAC Bridge. The algorithm may assign one or more conversations to the same port, however, it must not allocate some of the frames of a given conversation to one port and the remainder to different ports. The information used to assign conversations to ports could include the following: a) Source MAC address b) Destination MAC address c) The reception port d) The type of destination address (individual or group MAC address) e) Ethernet Length/Type value (i.e., protocol identification) f) Higher layer protocol information (e.g., addressing and protocol identification information from the LLC sublayer or above) g) Combinations of the above One simple approach applies a hash function to the selected information to generate a port number. This produces a deterministic (i.e., history independent) port selection across a given number of ports in an aggregation. However, as it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models, it might be appropriate to weight the port selection in favor of ports that are carrying lower traffic levels. In more sophisticated approaches, load balancing is dynamic; i.e., the port selected for a given set of conversations changes over time, independent of any changes that take place in the membership of the aggregation. Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, IEEE 802.3 in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 1[f] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references.

No.	’740 Patent Claim 1	IEEE 802.3
		For example, Basso discloses using a hash function and index table to select for blade/port combinations over which to send the packet over a user port and a switch fabric link. Basso further discloses that this selection is performed based upon packet information. Basso at [0010] (“Upon a network processor receiving a packet of data, the network processor may index into a table, commonly referred to as a forwarding table, to determine the table associated with a particular logical interface as well as the next destination address. The forwarding table may comprise a plurality of entries where each entry may comprise information indicating a particular table associated with a particular logical interface as well as the next destination address. Each logical interface may be associated with a table storing a plurality of entries containing blade/ port combinations as discussed further below. In one embodiment, an entry may be indexed in the forwarding table using a destination address in the received packet header.”) Basso at [0011] (“A hash function may then be performed on the received packet to generate a hash value. In one embodiment, a hash function may be performed on the source and destination address in the packet header to generate a hash value.”) Basso at [0012] (“The hash value generated may be used to index into the table associated with a particular logical interface. Upon indexing into the table associated with the logical interface, an appropriate blade/port combination may be identified to transmit the received packet of data. In one embodiment, a blade/port combination may be selected in the indexed entry of the table associated with the logical interface by using a portion of the bits of the hashed value. The received packet may then be transmitted through the identified blade/port combination to the next destination (next destination previously identified by the next destination address in the forwarding table).”) Basso at [0035] (“By logically grouping a plurality of ports 404 coupled to a particular network device into a logical interface 405, network processor 403 may be configured to transmit processed packets to that particular network device via any blade 402/port 404 combination grouped in that logical interface 405. For example, referring to FIG. 4, ports 404A-404I are physically connected to router 104B. If ports 404A-404I were logically grouped into logical interface 405, then a particular network processor 403, e.g., network

No.	’740 Patent Claim 1	IEEE 802.3
		processor 403A, may be configured to transmit processed packets that are determined to be transmitted to router 104B through any of ports 404A-404I in blades 402A-C, respectively. Network processor 403, e.g., network processor 403A, may be configured to transmit the processed packets to router 104B through ports 404, e.g., ports 404D-I, not in its blade 402, e.g., blade 402A, by forwarding the processed packets to switch fabric 401 which may then direct the processed packets to another appropriate physical blade 402/port 404 combination. Network processor 403, e.g., network processor 403A, may further be configured to transmit the processed packets to router 104B through any ports 404, e.g., ports 404A-C, in its blade 402, e.g., blade402A, instead of just one physical port 404 in its blade 402, e.g., blade 402A. A more detailed description of routing packets implementing logical interface(s) 405 is provided below in FIG. 5.”) Basso at [0040] (“In step 502, network processor 403, e.g., network processor 403A, may receive a packet of data from switch fabric 401. Upon receiving the packet of data, network processor 403, in step 503, may index into a table, commonly referred to as a forwarding table, to determine the table associated with a particular logical interface 405 as well as the next destination address, i.e., the next hop address. The forwarding table may comprise a plurality of entries where each entry may comprise information indicating a particular table associated with a particular logical interface 405 as well as the next destination address. Each logical interface 405 may be associated with a table storing a plurality of entries containing blade 402/port 404 combinations as discussed further below. In one embodiment, an entry may be indexed in the forwarding table using a destination address in the received packet header. It is noted that an entry may be indexed in the forwarding table usingother means and that such means would be recognized by an artisan of ordinary skill in the art. It is further noted that embodiments implementing such means would fall within the scope of the present invention.”) Basso at [0041] (“In step 504, a hash function may be performed on the received packet to generate a hash value. In one embodiment, a hash function may be performed on the source and destination address in the packet header to generate a hash value. It is noted that in other embodiments a hash function may be performed on different fields, e.g., port, type of service, in the received packet to generate a hash value.”) Basso at [0042] (“In step 505, the hash value generated in step 504 may be used to index into the table associated with a particular logical interface 405 determined in step 503. Upon indexing into the table

<table>
<tr><th>No.</th><th>’740 Patent Claim 1</th><th>IEEE 802.3</th></tr>
<tr><td></td><td></td><td>associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be identified in step 506 to transmit the received packet of data as explained below.”)

Basso at [0043] (“As stated above, the table associated with a particular logical interface 405 may comprise a plurality of entries where each entry may comprise a threshold value associated with a particular blade 402/port 404 combination. The threshold value may represent a percentage of the total number of packets received by router 104A that may be transmitted through the blade 402/port 404 combination associated with that threshold value. In one embodiment, the threshold value may be updated periodically by a user, e.g., system administrator, in control of router 104, e.g., router 104A. For example, the threshold value, e.g., twenty percent of the number of packets received by router 104A, associated with a particular blade 402/port 404 combination may be updated by lowering the threshold value by one percent during each update. An example of an entry of the table associated with a particular logical interface 405 is shown in Table 1 below:

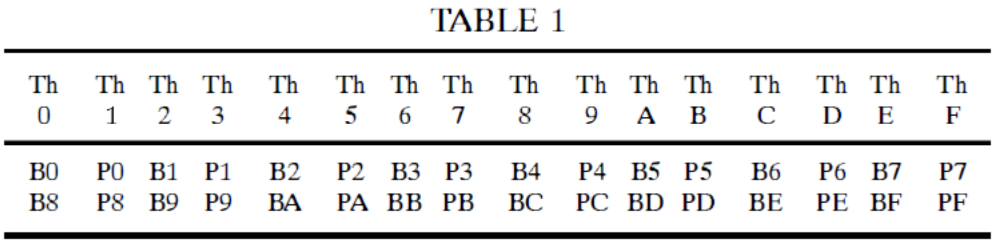
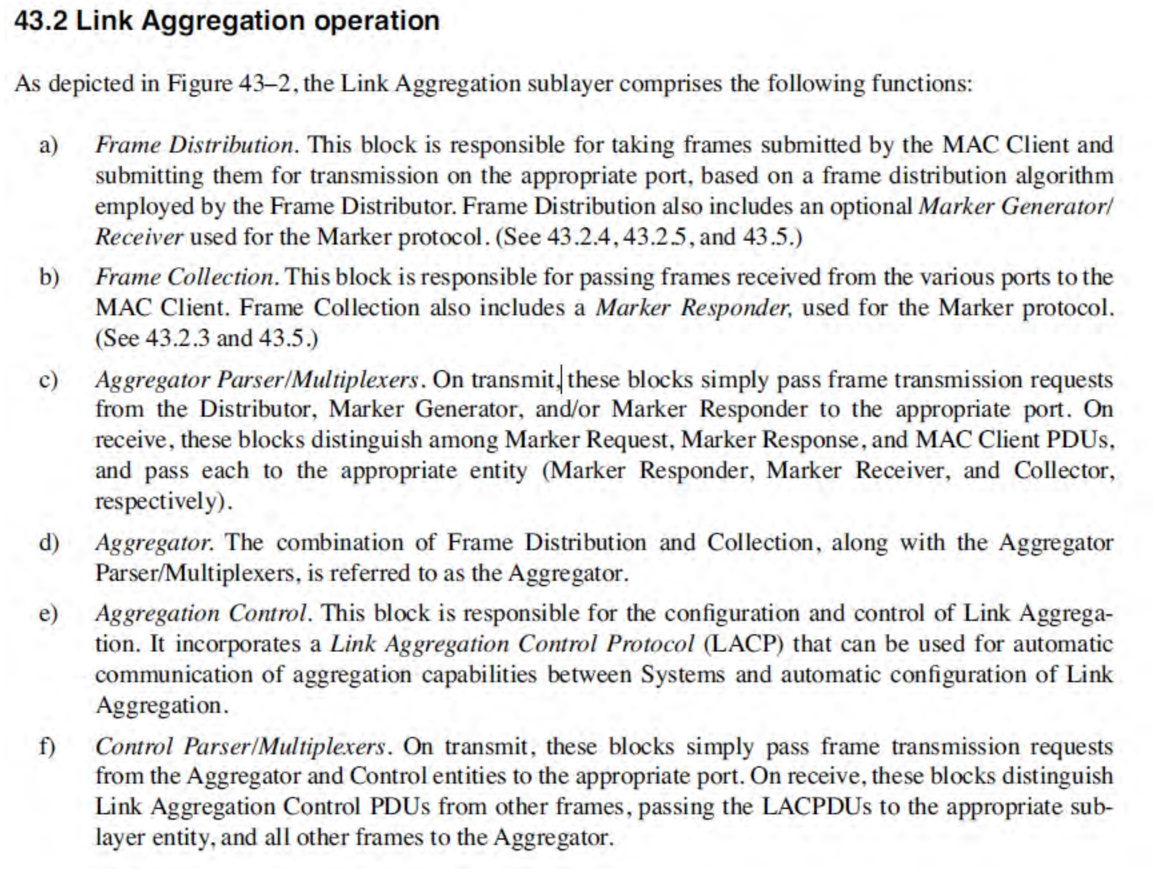


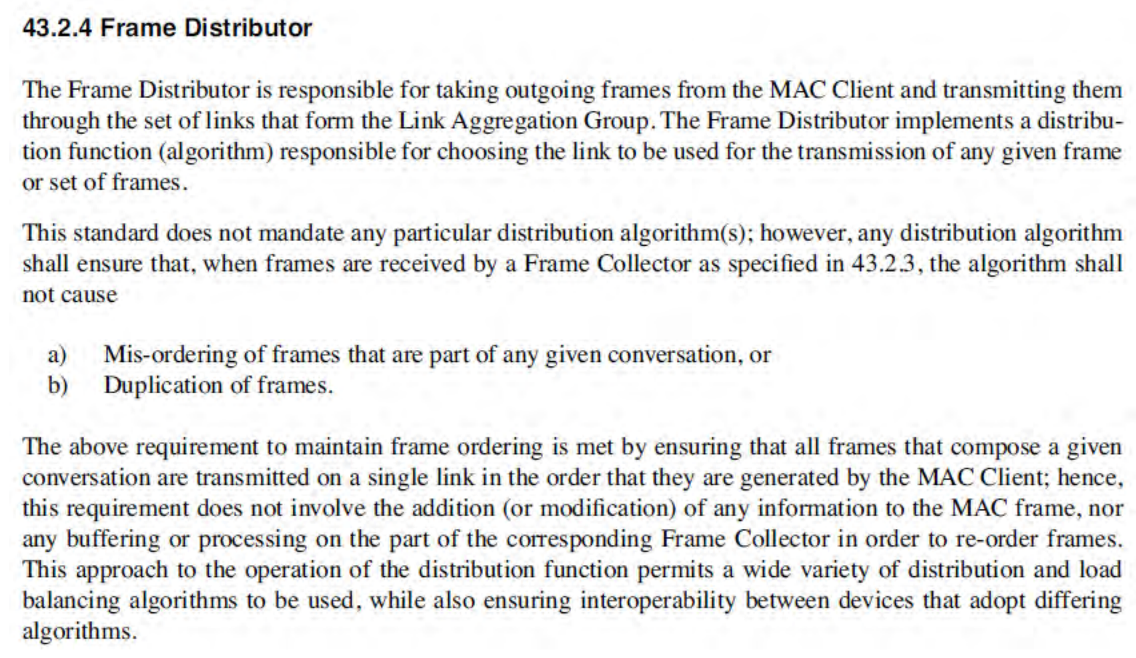
TABLE 1
<table>
<tr><th>Th
0</th><th>Th
1</th><th>Th
2</th><th>Th
3</th><th>Th
4</th><th>Th
5</th><th>Th
6</th><th>Th
7</th><th>Th
8</th><th>Th
9</th><th>Th
A</th><th>Th
B</th><th>Th
C</th><th>Th
D</th><th>Th
E</th><th>Th
F</th></tr>
<tr><td>B0
B8</td><td>P0
P8</td><td>B1
B9</td><td>P1
P9</td><td>B2
BA</td><td>P2
PA</td><td>B3
BB</td><td>P3
PB</td><td>B4
BC</td><td>P4
PC</td><td>B5
BD</td><td>P5
PD</td><td>B6
BE</td><td>P6
PE</td><td>B7
BF</td><td>P7
PF</td></tr>
</table>

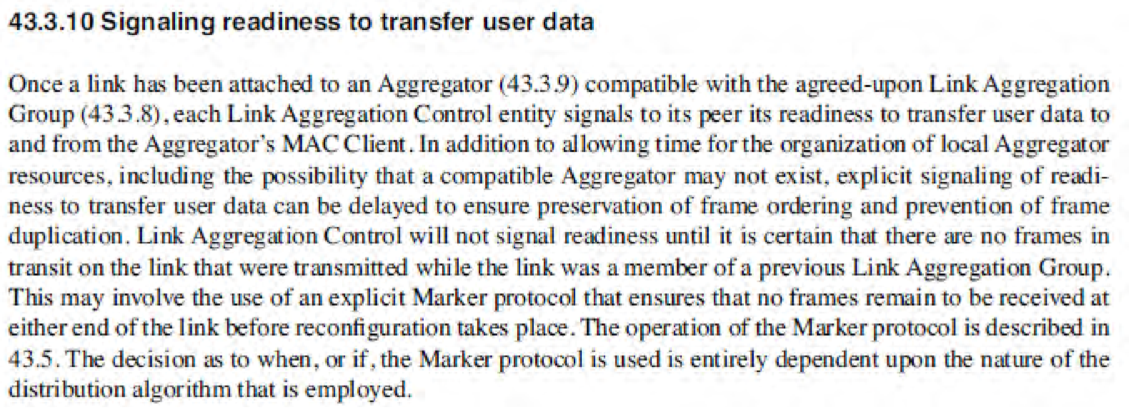
Basso at [0044] (“Table 1 above illustrates an exemplary entry in the table associated with a particular logical interface 405. Each entry may comprise a plurality of threshold values (16 threshold values in exemplary Table 1) where each threshold value is associated with a particular blade 402/port 404 combination. For example, threshold value (Th0) is associated with blade B0/port P0 combination where blade BO may refer to a particular blade 402, e.g., blade 402B, and port PO may refer to a particular port 404, e.g., port 404E. Threshold value (Thl) is associated with blade Bl/port Pl combination and so forth. As stated above, each threshold value may represent a percentage of the total number of packets received by router 104A that may be transmitted through the blade 402/port 404 combination associated with that threshold value. For example, threshold value (Th0) may represent a percentage of the total number of packets received by router 104A that</td></tr>
</table>

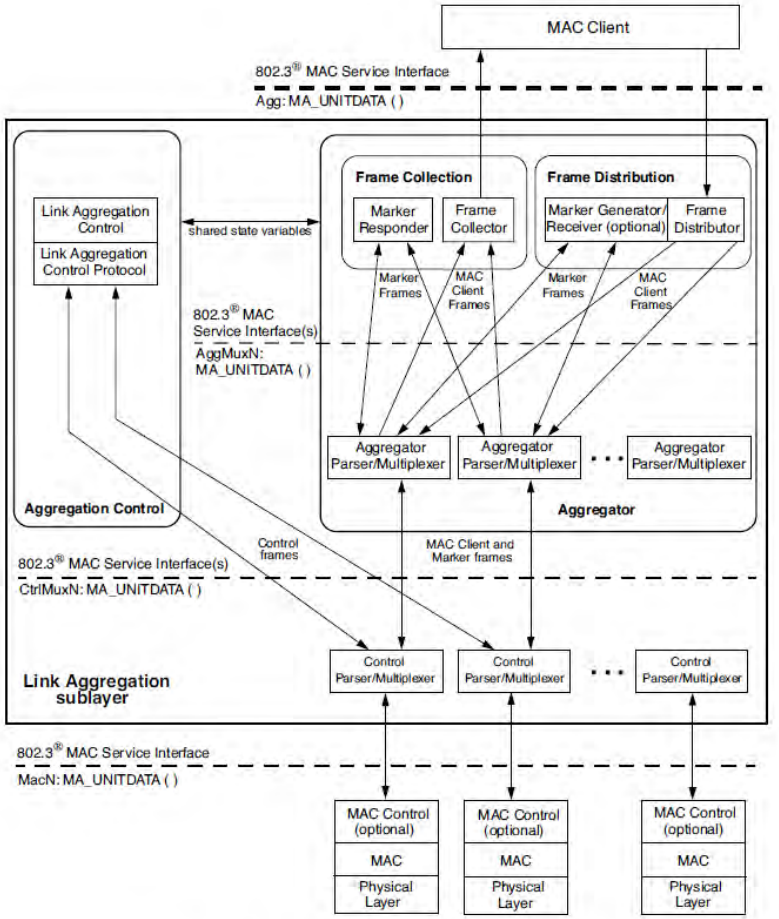
No.	’740 Patent Claim 1	IEEE 802.3
		may be transmitted through port PO in blade BO. If port PO refers to port 404D and blade BO refers to blade 402B, then if Th0 has a value of twenty percent, a maximum of twenty percent of the total packets received by router 104A may be transmitted through port 404D in blade 402B.”) Basso at [0045] (“As stated above, upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be identified in step 506 to transmit the received packet of data. In one embodiment, the hash value generated in step 504 may be used to select a particular threshold value and hence a blade 402/port 404 combination associated with the selected threshold value. In one embodiment, a portion of the bits of the hash value, e.g., most significant bits, may be used to select a particular threshold value in the entry indexed in step 505. For example, referring to Table 1, since there are 16 different threshold values in each entry of the table associated with logical interface 405, only four bits of the hash value generated in step 504 may be used to select a threshold value. Upon selecting a threshold value, the blade 402/port 404 combination associated with the selected threshold value may be used to transmit the received packet.”) As another example, Wiher discloses using cell header information to at each node to select and route the ATM data cell over a data link and selected backplane. Wiher at 3:43-65 (“In general, in another aspect, the invention features an apparatus for communicating data cells between a data link and a backplane. The apparatus includes transceiver circuitry to transmit and receive data cells over a data link and a plurality of backplane interfaces each including at least one cell signal terminal. Each of the backplane interface is coupled to a backplane interconnection circuit. Each backplane interconnection circuit transmits and receives cells over the cell signal terminals of its associated backplane interface. The apparatus also includes de-multiplexing circuitry coupling the transceiver circuitry to each of the backplane interconnection circuits. The de-multiplexing circuitry receives a data cell from the transceiver circuitry, select a backplane interconnection circuit associated with the data cell, and provide the data cell to the selected backplane interconnection circuit for transmission over the cell signal terminals of the associated backplane interface. The apparatus also includes multiplexing circuitry coupling the plurality of backplane interconnection circuits to the transceiver circuitry. The multiplexing circuitry

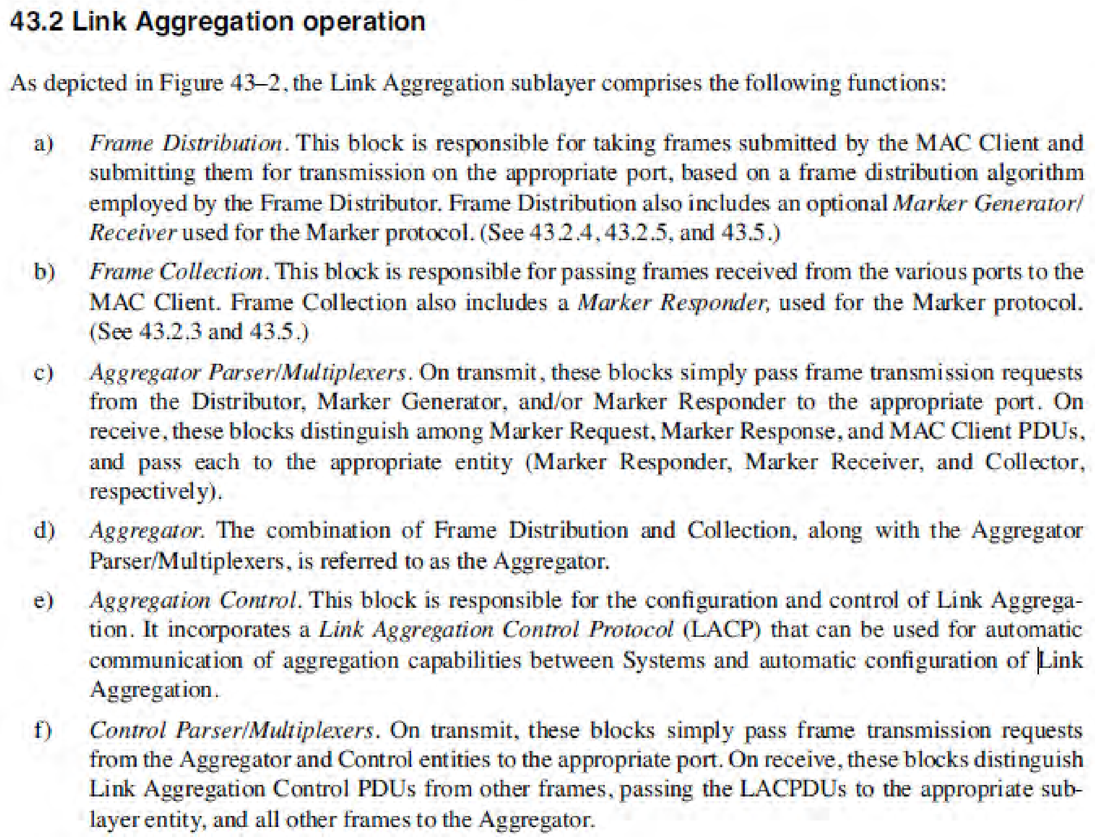
No.	’740 Patent Claim 1	IEEE 802.3
		receives data cells from each of the backplane interconnection circuits and provide the received data cells to the transceiver circuitry.”) Wiher at 3:66-4:22 (“Implementations of the invention may include one or more of the following features. The backplane interconnection circuits may independently receive and transmit data cells over the plurality of backplane interfaces. The de-multiplexing circuitry may select a backplane interface based on data in the header field of the data cell. The apparatus may include header translation circuitry to alter header data in cells sent between the plurality of backplane interfaces and the transceiver circuitry. Each of the plurality of backplane interfaces may include separate terminals to receive cells and separate terminals to transmit cells. The terminals to transmit cells may include a first and second control terminal and at least one outgoing cell data terminal. A backplane interface's backplane interconnection circuitry may accepts a signal on the first control terminal as indicating that a cell may be sent over the interface, asserts a 15 signal on the second control terminal to indicate that a cell is being transmitted, and transmits data bits of the cell on the outgoing cell data terminal. Each backplane interface may include a single outgoing cell data terminal and each bit of the cell may be serially transmitted over the single outgoing cell data terminal. Each backplane interface may include multiple outgoing cell data terminals and bits of the cell may be sent in parallel over the eight outgoing cell data terminals.”)
1[g]	sending the data frame over the selected first and second physical links,	IEEE 802.3 discloses sending the data frame over the selected first and second physical links. For example, IEEE 802.3 discloses transmitting and receiving data frames over the identified links determined by the distribution algorithm. IEEE 802.3 at 1468

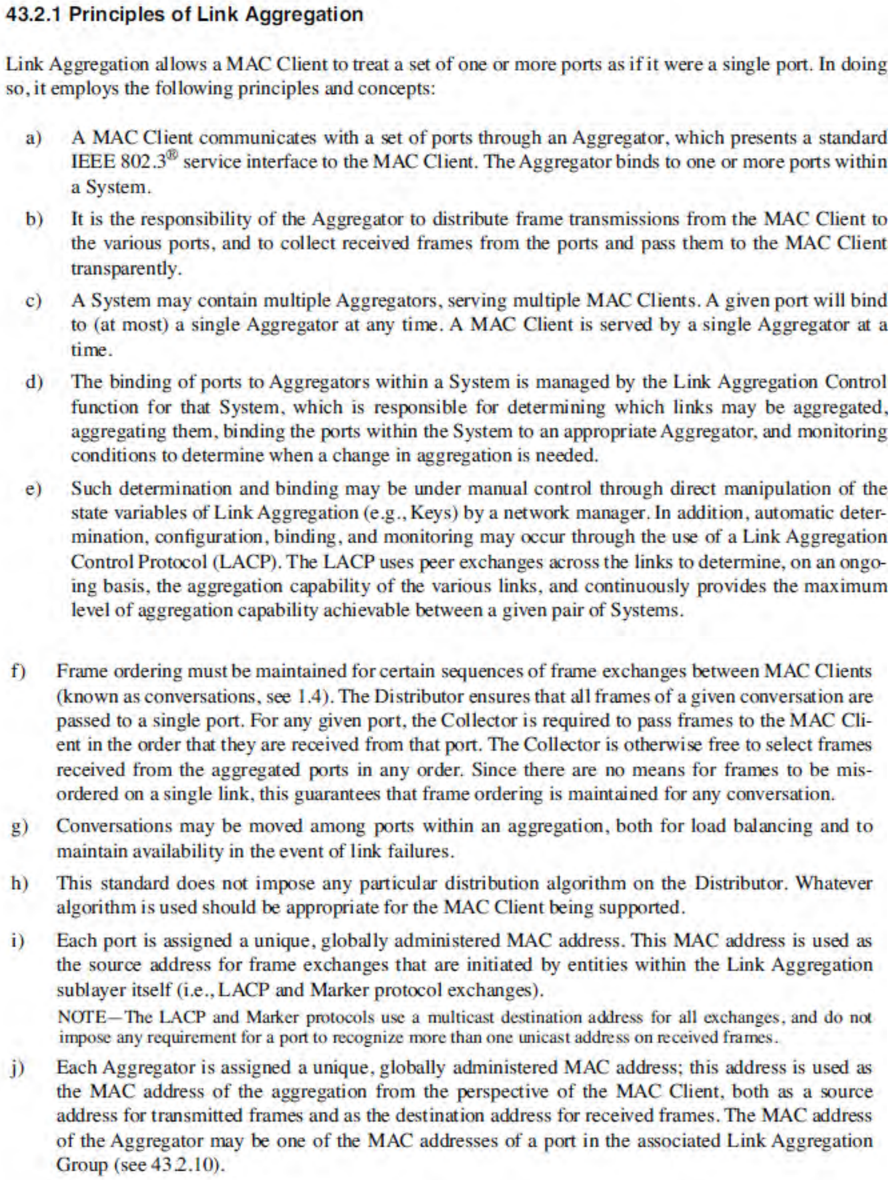
No.	’740 Patent Claim 1	IEEE 802.3
		 **43.2 Link Aggregation operation** As depicted in Figure 43–2, the Link Aggregation sublayer comprises the following functions: a) *Frame Distribution*. This block is responsible for taking frames submitted by the MAC Client and submitting them for transmission on the appropriate port, based on a frame distribution algorithm employed by the Frame Distributor. Frame Distribution also includes an optional *Marker Generator/Receiver* used for the Marker protocol. (See 43.2.4, 43.2.5, and 43.5.) b) *Frame Collection*. This block is responsible for passing frames received from the various ports to the MAC Client. Frame Collection also includes a *Marker Responder*, used for the Marker protocol. (See 43.2.3 and 43.5.) c) *Aggregator Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Distributor, Marker Generator, and/or Marker Responder to the appropriate port. On receive, these blocks distinguish among Marker Request, Marker Response, and MAC Client PDUs, and pass each to the appropriate entity (Marker Responder, Marker Receiver, and Collector, respectively). d) *Aggregator*. The combination of Frame Distribution and Collection, along with the Aggregator Parser/Multiplexers, is referred to as the Aggregator. e) *Aggregation Control*. This block is responsible for the configuration and control of Link Aggregation. It incorporates a *Link Aggregation Control Protocol* (LACP) that can be used for automatic communication of aggregation capabilities between Systems and automatic configuration of Link Aggregation. f) *Control Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Aggregator and Control entities to the appropriate port. On receive, these blocks distinguish Link Aggregation Control PDUs from other frames, passing the LACPDUs to the appropriate sublayer entity, and all other frames to the Aggregator. IEEE 802.3 at 1471

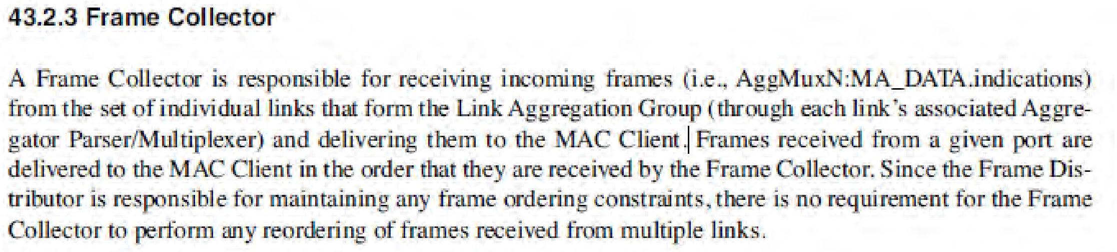
No.	’740 Patent Claim 1	IEEE 802.3
		 **43.2.4 Frame Distributor** The Frame Distributor is responsible for taking outgoing frames from the MAC Client and transmitting them through the set of links that form the Link Aggregation Group. The Frame Distributor implements a distribution function (algorithm) responsible for choosing the link to be used for the transmission of any given frame or set of frames. This standard does not mandate any particular distribution algorithm(s); however, any distribution algorithm shall ensure that, when frames are received by a Frame Collector as specified in 43.2.3, the algorithm shall not cause a) Mis-ordering of frames that are part of any given conversation, or b) Duplication of frames. The above requirement to maintain frame ordering is met by ensuring that all frames that compose a given conversation are transmitted on a single link in the order that they are generated by the MAC Client; hence, this requirement does not involve the addition (or modification) of any information to the MAC frame, nor any buffering or processing on the part of the corresponding Frame Collector in order to re-order frames. This approach to the operation of the distribution function permits a wide variety of distribution and load balancing algorithms to be used, while also ensuring interoperability between devices that adopt differing algorithms.
1[h]	said sending comprising communicating along at least one of said bi-directional links.	IEEE 802.3 discloses said sending comprising communicating along at least one of said bi-directional links. *See supra at* 1[b], 1[d], 1[g].

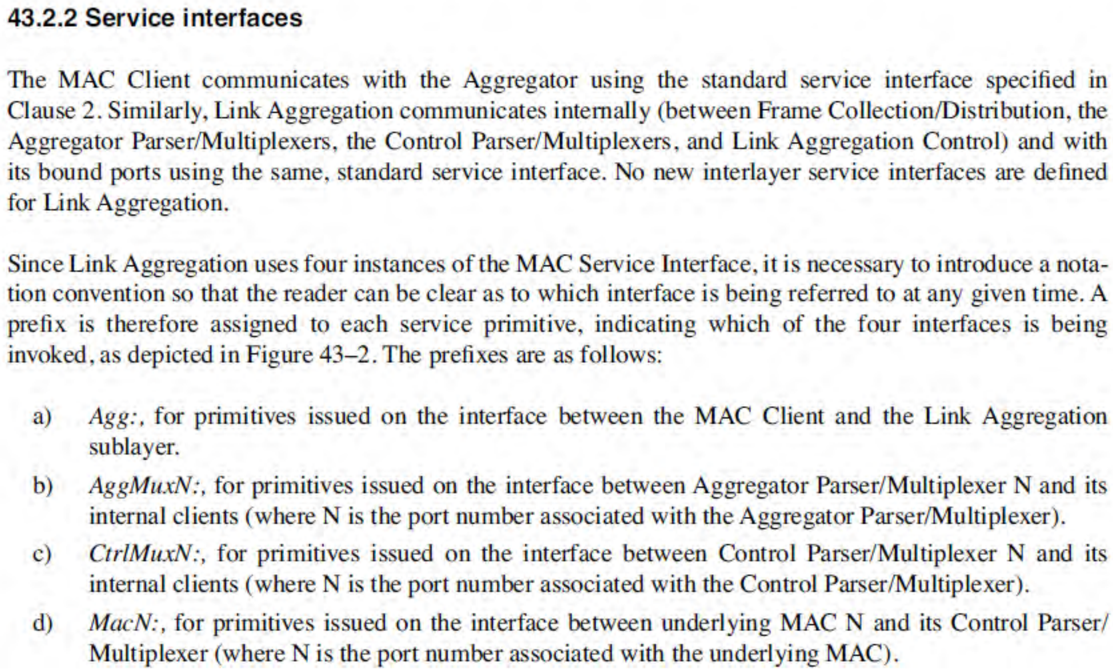
No.	’740 Patent Claim 2	IEEE 802.3
2[a]	The method according to claim 1, wherein the network node comprises a user node, and	IEEE 802.3 discloses the method according to claim 1, wherein the network node comprises a user node. For example, IEEE 802.3 discloses a MAC client from which data frames are transmitted and received. IEEE 802.3 at 1484  **43.3.10 Signaling readiness to transfer user data** Once a link has been attached to an Aggregator (43.3.9) compatible with the agreed-upon Link Aggregation Group (43.3.8), each Link Aggregation Control entity signals to its peer its readiness to transfer user data to and from the Aggregator’s MAC Client. In addition to allowing time for the organization of local Aggregator resources, including the possibility that a compatible Aggregator may not exist, explicit signaling of readiness to transfer user data can be delayed to ensure preservation of frame ordering and prevention of frame duplication. Link Aggregation Control will not signal readiness until it is certain that there are no frames in transit on the link that were transmitted while the link was a member of a previous Link Aggregation Group. This may involve the use of an explicit Marker protocol that ensures that no frames remain to be received at either end of the link before reconfiguration takes place. The operation of the Marker protocol is described in 43.5. The decision as to when, or if, the Marker protocol is used is entirely dependent upon the nature of the distribution algorithm that is employed. IEEE 802.3 at Figure 43-2

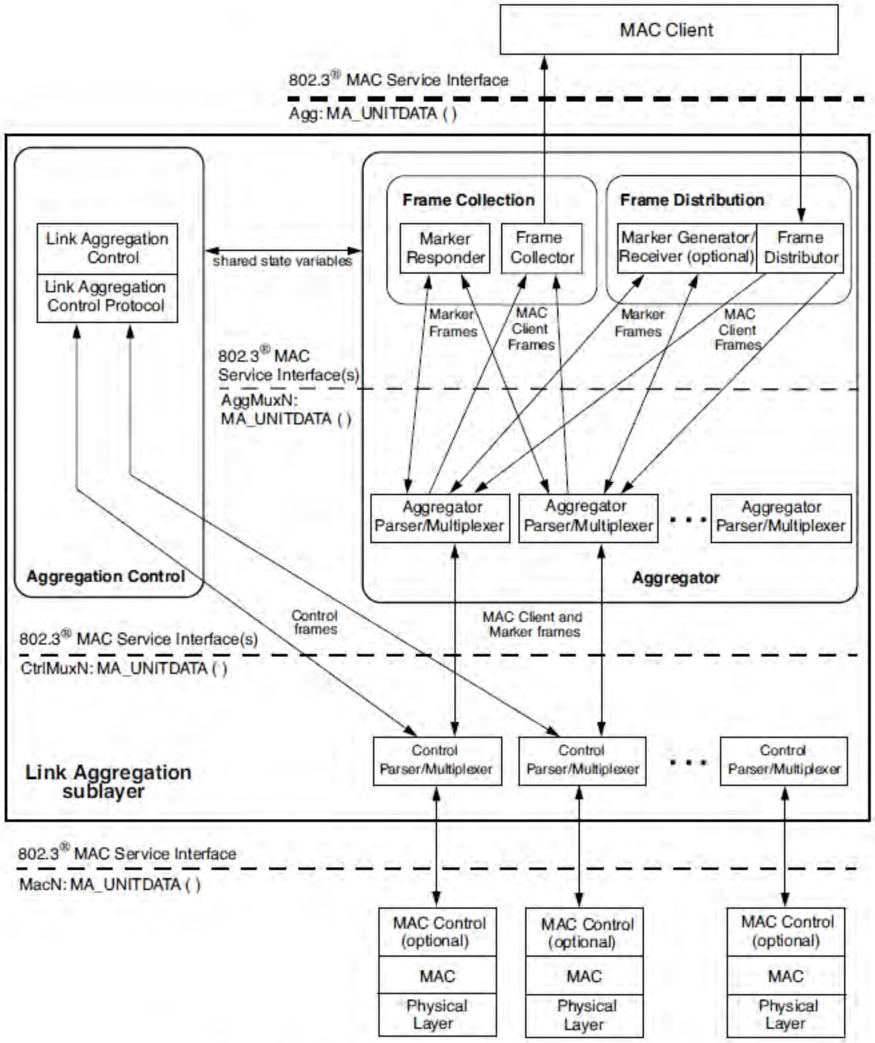
No.	’740 Patent Claim 2	IEEE 802.3
		 MAC Client 802.3® MAC Service Interface Agg: MA_UNITDATA ( ) Link Aggregation Control Link Aggregation Control Protocol shared state variables Frame Collection Marker Responder Frame Collector Frame Distribution Marker Generator/ Receiver (optional) Frame Distributor Marker Frames MAC Client Frames Marker Frames MAC Client Frames 802.3® MAC Service Interface(s) AggMuxN: MA_UNITDATA ( ) Aggregator Parser/Multiplexer Aggregator Parser/Multiplexer Aggregator Parser/Multiplexer Aggregation Control Aggregator Control frames MAC Client and Marker frames 802.3® MAC Service Interface(s) CtrlMuxN: MA_UNITDATA ( ) Link Aggregation sublayer Control Parser/Multiplexer Control Parser/Multiplexer Control Parser/Multiplexer 802.3® MAC Service Interface MacN: MA_UNITDATA ( ) MAC Control (optional) MAC Physical Layer MAC Control (optional) MAC Physical Layer MAC Control (optional) MAC Physical Layer Figure 43–2—Link Aggregation sublayer block diagram IEEE 802.3 at 1468

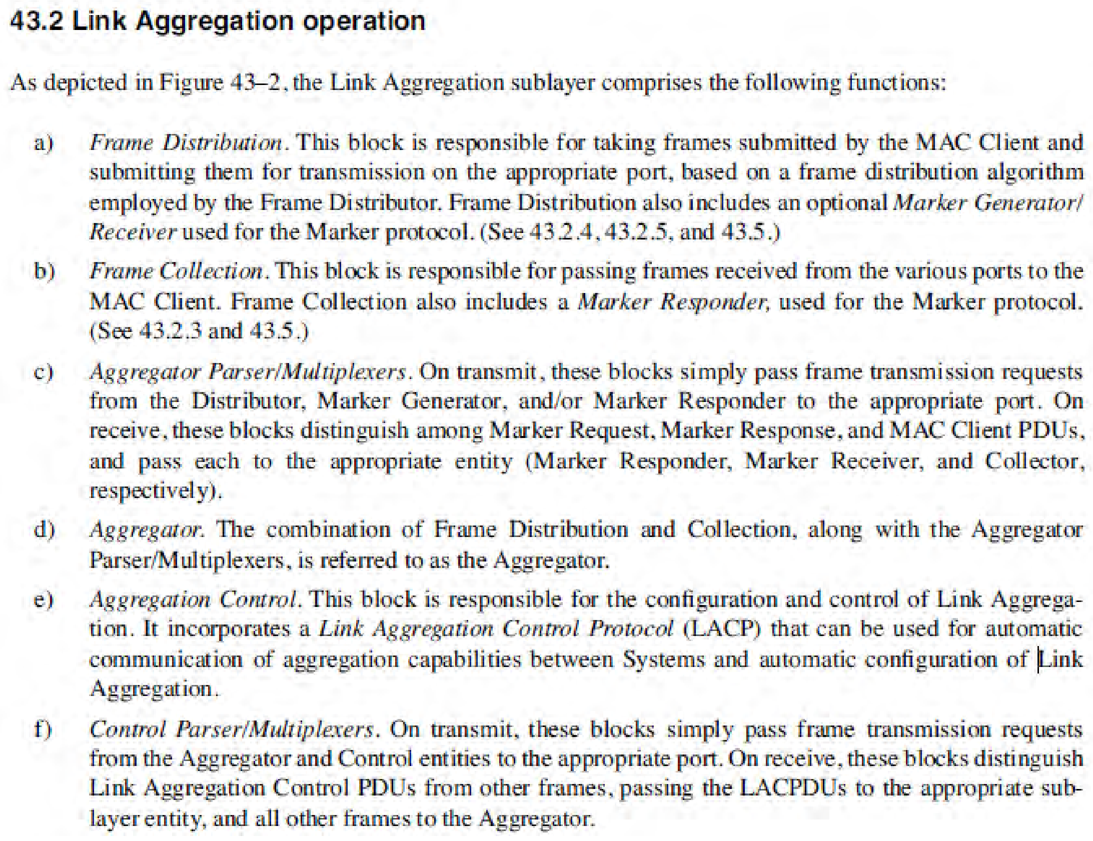
No.	’740 Patent Claim 2	IEEE 802.3
		 **43.2 Link Aggregation operation** As depicted in Figure 43–2, the Link Aggregation sublayer comprises the following functions: a) *Frame Distribution*. This block is responsible for taking frames submitted by the MAC Client and submitting them for transmission on the appropriate port, based on a frame distribution algorithm employed by the Frame Distributor. Frame Distribution also includes an optional *Marker Generator/ Receiver* used for the Marker protocol. (See 43.2.4, 43.2.5, and 43.5.) b) *Frame Collection*. This block is responsible for passing frames received from the various ports to the MAC Client. Frame Collection also includes a *Marker Responder,* used for the Marker protocol. (See 43.2.3 and 43.5.) c) *Aggregator Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Distributor, Marker Generator, and/or Marker Responder to the appropriate port. On receive, these blocks distinguish among Marker Request, Marker Response, and MAC Client PDUs, and pass each to the appropriate entity (Marker Responder, Marker Receiver, and Collector, respectively). d) *Aggregator.* The combination of Frame Distribution and Collection, along with the Aggregator Parser/Multiplexers, is referred to as the Aggregator. e) *Aggregation Control*. This block is responsible for the configuration and control of Link Aggregation. It incorporates a *Link Aggregation Control Protocol* (LACP) that can be used for automatic communication of aggregation capabilities between Systems and automatic configuration of Link Aggregation. f) *Control Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Aggregator and Control entities to the appropriate port. On receive, these blocks distinguish Link Aggregation Control PDUs from other frames, passing the LACPDUs to the appropriate sublayer entity, and all other frames to the Aggregator. IEEE 802.3 at 1468-69

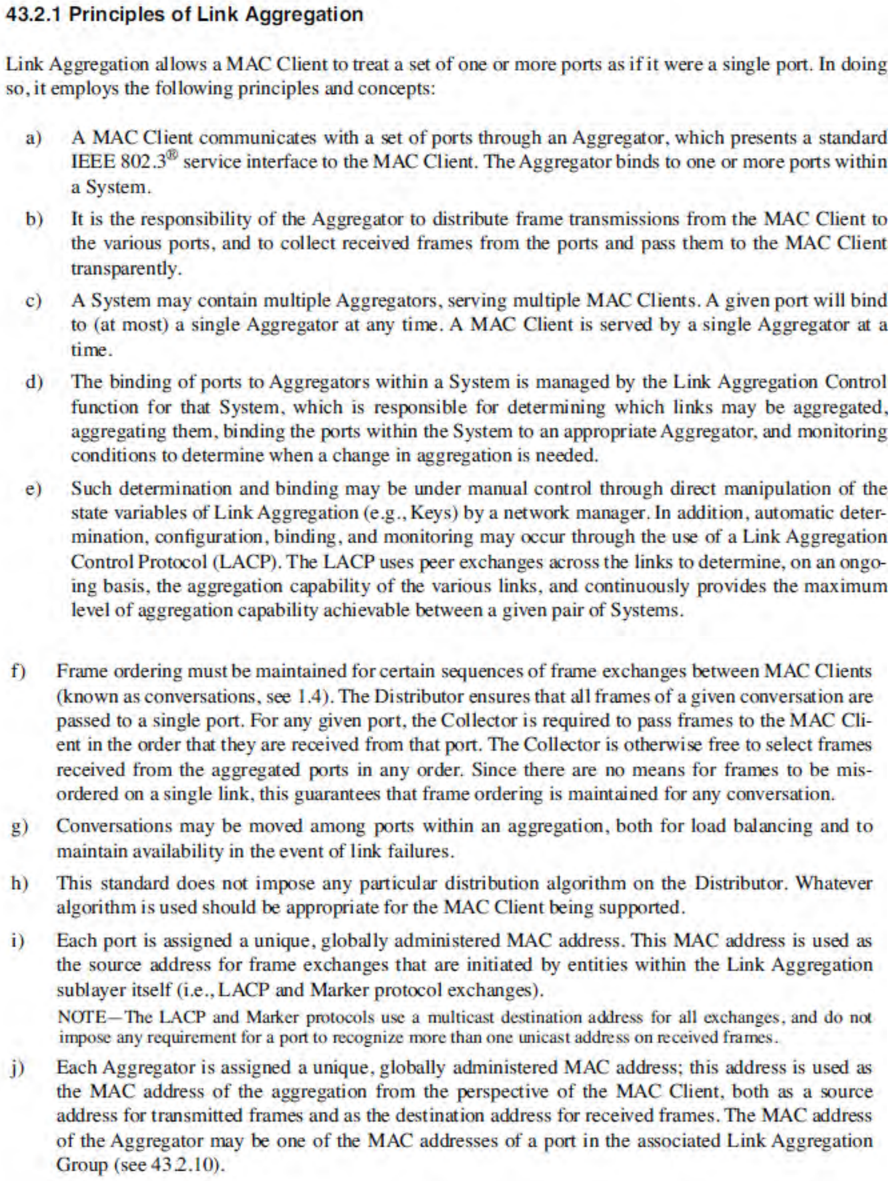
No.	’740 Patent Claim 2	IEEE 802.3
		 **43.2.1 Principles of Link Aggregation** Link Aggregation allows a MAC Client to treat a set of one or more ports as if it were a single port. In doing so, it employs the following principles and concepts: a) A MAC Client communicates with a set of ports through an Aggregator, which presents a standard IEEE 802.3® service interface to the MAC Client. The Aggregator binds to one or more ports within a System. b) It is the responsibility of the Aggregator to distribute frame transmissions from the MAC Client to the various ports, and to collect received frames from the ports and pass them to the MAC Client transparently. c) A System may contain multiple Aggregators, serving multiple MAC Clients. A given port will bind to (at most) a single Aggregator at any time. A MAC Client is served by a single Aggregator at a time. d) The binding of ports to Aggregators within a System is managed by the Link Aggregation Control function for that System, which is responsible for determining which links may be aggregated, aggregating them, binding the ports within the System to an appropriate Aggregator, and monitoring conditions to determine when a change in aggregation is needed. e) Such determination and binding may be under manual control through direct manipulation of the state variables of Link Aggregation (e.g., Keys) by a network manager. In addition, automatic determination, configuration, binding, and monitoring may occur through the use of a Link Aggregation Control Protocol (LACP). The LACP uses peer exchanges across the links to determine, on an ongoing basis, the aggregation capability of the various links, and continuously provides the maximum level of aggregation capability achievable between a given pair of Systems. f) Frame ordering must be maintained for certain sequences of frame exchanges between MAC Clients (known as conversations, see 1.4). The Distributor ensures that all frames of a given conversation are passed to a single port. For any given port, the Collector is required to pass frames to the MAC Client in the order that they are received from that port. The Collector is otherwise free to select frames received from the aggregated ports in any order. Since there are no means for frames to be misordered on a single link, this guarantees that frame ordering is maintained for any conversation. g) Conversations may be moved among ports within an aggregation, both for load balancing and to maintain availability in the event of link failures. h) This standard does not impose any particular distribution algorithm on the Distributor. Whatever algorithm is used should be appropriate for the MAC Client being supported. i) Each port is assigned a unique, globally administered MAC address. This MAC address is used as the source address for frame exchanges that are initiated by entities within the Link Aggregation sublayer itself (i.e., LACP and Marker protocol exchanges). NOTE—The LACP and Marker protocols use a multicast destination address for all exchanges, and do not impose any requirement for a port to recognize more than one unicast address on received frames. j) Each Aggregator is assigned a unique, globally administered MAC address; this address is used as the MAC address of the aggregation from the perspective of the MAC Client, both as a source address for transmitted frames and as the destination address for received frames. The MAC address of the Aggregator may be one of the MAC addresses of a port in the associated Link Aggregation Group (see 43.2.10).

No.	’740 Patent Claim 2	IEEE 802.3
		IEEE 802.3 at 1470  **43.2.3 Frame Collector** A Frame Collector is responsible for receiving incoming frames (i.e., AggMuxN:MA_DATA.indications) from the set of individual links that form the Link Aggregation Group (through each link’s associated Aggregator Parser/Multiplexer) and delivering them to the MAC Client. Frames received from a given port are delivered to the MAC Client in the order that they are received by the Frame Collector. Since the Frame Distributor is responsible for maintaining any frame ordering constraints, there is no requirement for the Frame Collector to perform any reordering of frames received from multiple links.
2[b]	wherein sending the data frame comprises establishing a communication service between the user node and the communication network.	IEEE 802.3 discloses wherein sending the data frame comprises establishing a communication service between the user node and the communication network. For example, IEEE 802.3 discloses a standard IEEE 802.3 service interface to the MAC client. IEEE 802.3 at 1469

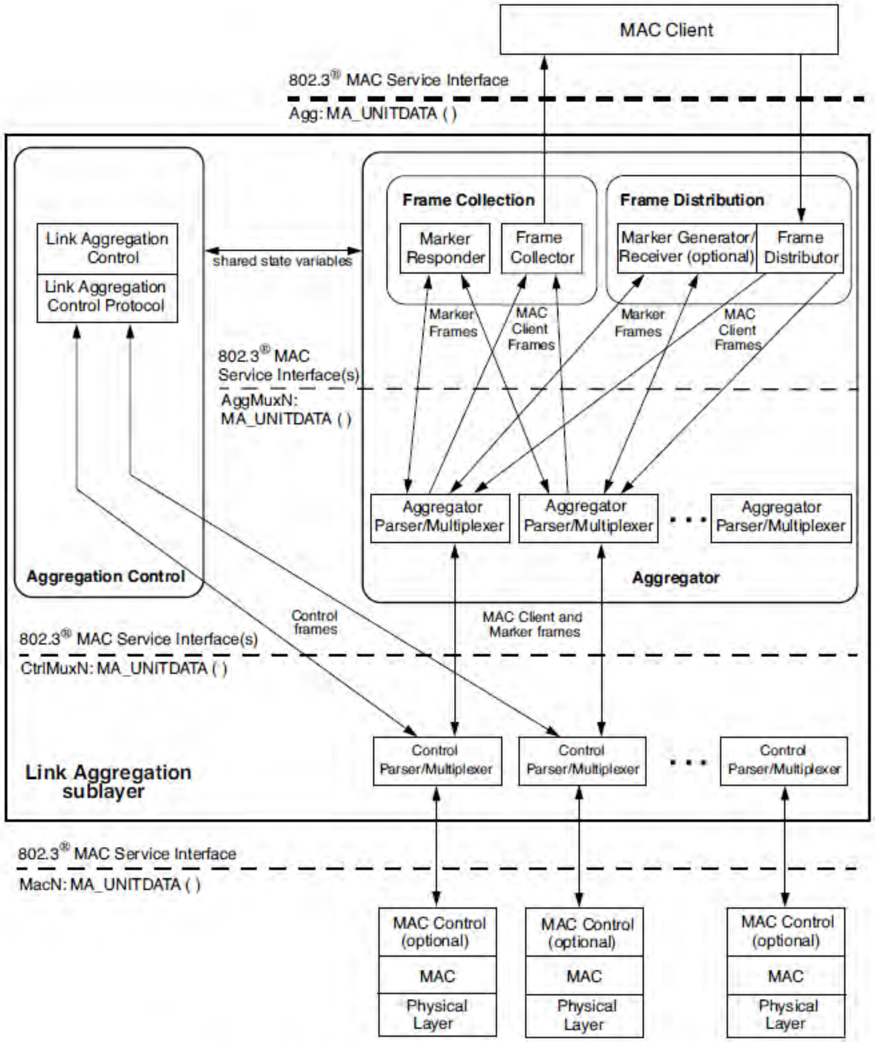
No.	’740 Patent Claim 2	IEEE 802.3
		 **43.2.2 Service interfaces** The MAC Client communicates with the Aggregator using the standard service interface specified in Clause 2. Similarly, Link Aggregation communicates internally (between Frame Collection/Distribution, the Aggregator Parser/Multiplexers, the Control Parser/Multiplexers, and Link Aggregation Control) and with its bound ports using the same, standard service interface. No new interlayer service interfaces are defined for Link Aggregation. Since Link Aggregation uses four instances of the MAC Service Interface, it is necessary to introduce a notation convention so that the reader can be clear as to which interface is being referred to at any given time. A prefix is therefore assigned to each service primitive, indicating which of the four interfaces is being invoked, as depicted in Figure 43–2. The prefixes are as follows: a) *Agg:*, for primitives issued on the interface between the MAC Client and the Link Aggregation sublayer. b) *AggMuxN:*, for primitives issued on the interface between Aggregator Parser/Multiplexer N and its internal clients (where N is the port number associated with the Aggregator Parser/Multiplexer). c) *CtrlMuxN:*, for primitives issued on the interface between Control Parser/Multiplexer N and its internal clients (where N is the port number associated with the Control Parser/Multiplexer). d) *MacN:*, for primitives issued on the interface between underlying MAC N and its Control Parser/ Multiplexer (where N is the port number associated with the underlying MAC). IEEE 802.3 at Figure 43-2

No.	’740 Patent Claim 2	IEEE 802.3
		 MAC Client 802.3® MAC Service Interface Agg: MA_UNITDATA ( ) Frame Collection Marker Responder Frame Collector Frame Distribution Marker Generator/ Receiver (optional) Frame Distributor Link Aggregation Control Link Aggregation Control Protocol shared state variables Marker Frames MAC Client Frames Marker Frames MAC Client Frames 802.3® MAC Service Interface(s) AggMuxN: MA_UNITDATA ( ) Aggregator Parser/Multiplexer Aggregator Parser/Multiplexer Aggregator Parser/Multiplexer Aggregation Control Aggregator Control frames MAC Client and Marker frames 802.3® MAC Service Interface(s) CtrlMuxN: MA_UNITDATA ( ) Link Aggregation sublayer Control Parser/Multiplexer Control Parser/Multiplexer Control Parser/Multiplexer 802.3® MAC Service Interface MacN: MA_UNITDATA ( ) MAC Control (optional) MAC Physical Layer MAC Control (optional) MAC Physical Layer MAC Control (optional) MAC Physical Layer **Figure 43–2—Link Aggregation sublayer block diagram**

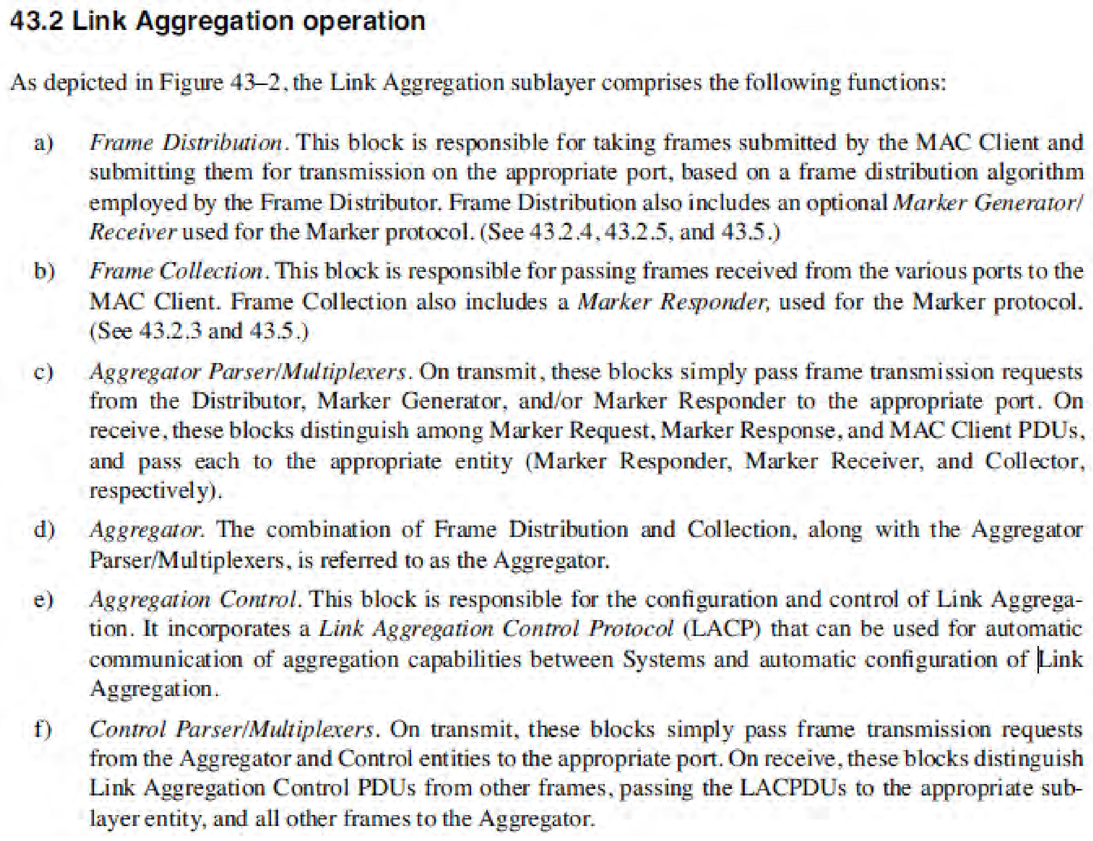
No.	’740 Patent Claim 2	IEEE 802.3
		IEEE 802.3 at 1468  **43.2 Link Aggregation operation** As depicted in Figure 43–2, the Link Aggregation sublayer comprises the following functions: a) *Frame Distribution*. This block is responsible for taking frames submitted by the MAC Client and submitting them for transmission on the appropriate port, based on a frame distribution algorithm employed by the Frame Distributor. Frame Distribution also includes an optional *Marker Generator/ Receiver* used for the Marker protocol. (See 43.2.4, 43.2.5, and 43.5.) b) *Frame Collection*. This block is responsible for passing frames received from the various ports to the MAC Client. Frame Collection also includes a *Marker Responder,* used for the Marker protocol. (See 43.2.3 and 43.5.) c) *Aggregator Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Distributor, Marker Generator, and/or Marker Responder to the appropriate port. On receive, these blocks distinguish among Marker Request, Marker Response, and MAC Client PDUs, and pass each to the appropriate entity (Marker Responder, Marker Receiver, and Collector, respectively). d) *Aggregator*. The combination of Frame Distribution and Collection, along with the Aggregator Parser/Multiplexers, is referred to as the Aggregator. e) *Aggregation Control*. This block is responsible for the configuration and control of Link Aggregation. It incorporates a *Link Aggregation Control Protocol* (LACP) that can be used for automatic communication of aggregation capabilities between Systems and automatic configuration of Link Aggregation. f) *Control Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Aggregator and Control entities to the appropriate port. On receive, these blocks distinguish Link Aggregation Control PDUs from other frames, passing the LACPDUs to the appropriate sublayer entity, and all other frames to the Aggregator. IEEE 802.3 at 1468-69

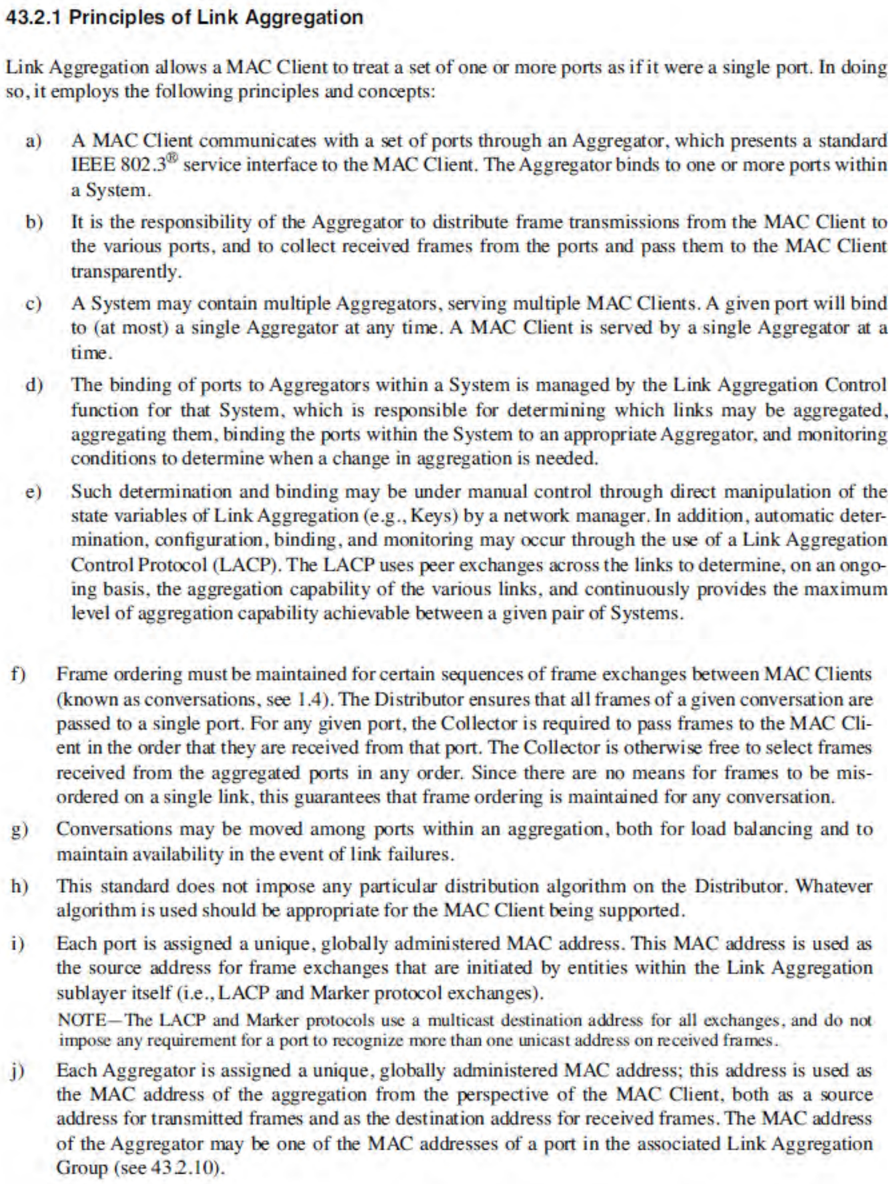
No.	’740 Patent Claim 2	IEEE 802.3
		 **43.2.1 Principles of Link Aggregation** Link Aggregation allows a MAC Client to treat a set of one or more ports as if it were a single port. In doing so, it employs the following principles and concepts: a) A MAC Client communicates with a set of ports through an Aggregator, which presents a standard IEEE 802.3® service interface to the MAC Client. The Aggregator binds to one or more ports within a System. b) It is the responsibility of the Aggregator to distribute frame transmissions from the MAC Client to the various ports, and to collect received frames from the ports and pass them to the MAC Client transparently. c) A System may contain multiple Aggregators, serving multiple MAC Clients. A given port will bind to (at most) a single Aggregator at any time. A MAC Client is served by a single Aggregator at a time. d) The binding of ports to Aggregators within a System is managed by the Link Aggregation Control function for that System, which is responsible for determining which links may be aggregated, aggregating them, binding the ports within the System to an appropriate Aggregator, and monitoring conditions to determine when a change in aggregation is needed. e) Such determination and binding may be under manual control through direct manipulation of the state variables of Link Aggregation (e.g., Keys) by a network manager. In addition, automatic determination, configuration, binding, and monitoring may occur through the use of a Link Aggregation Control Protocol (LACP). The LACP uses peer exchanges across the links to determine, on an ongoing basis, the aggregation capability of the various links, and continuously provides the maximum level of aggregation capability achievable between a given pair of Systems. f) Frame ordering must be maintained for certain sequences of frame exchanges between MAC Clients (known as conversations, see 1.4). The Distributor ensures that all frames of a given conversation are passed to a single port. For any given port, the Collector is required to pass frames to the MAC Client in the order that they are received from that port. The Collector is otherwise free to select frames received from the aggregated ports in any order. Since there are no means for frames to be misordered on a single link, this guarantees that frame ordering is maintained for any conversation. g) Conversations may be moved among ports within an aggregation, both for load balancing and to maintain availability in the event of link failures. h) This standard does not impose any particular distribution algorithm on the Distributor. Whatever algorithm is used should be appropriate for the MAC Client being supported. i) Each port is assigned a unique, globally administered MAC address. This MAC address is used as the source address for frame exchanges that are initiated by entities within the Link Aggregation sublayer itself (i.e., LACP and Marker protocol exchanges). NOTE—The LACP and Marker protocols use a multicast destination address for all exchanges, and do not impose any requirement for a port to recognize more than one unicast address on received frames. j) Each Aggregator is assigned a unique, globally administered MAC address; this address is used as the MAC address of the aggregation from the perspective of the MAC Client, both as a source address for transmitted frames and as the destination address for received frames. The MAC address of the Aggregator may be one of the MAC addresses of a port in the associated Link Aggregation Group (see 43.2.10).

No.	’740 Patent Claim 3	IEEE 802.3
3	The method according to claim 1, wherein the second physical links comprise backplane traces formed on a backplane to which the one or more interface modules are coupled.	IEEE 802.3 discloses the method according to claim 1, wherein the second physical links comprise backplane traces formed on a backplane to which the one or more interface modules are coupled. For example, IEEE 802.3 discloses physical connections between the aggregator and control parser/multiplexer. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the IEEE 802.3 is found to not meet this limitation, wherein the second physical links comprise backplane traces formed on a back plane to which the one or more interface modules are coupled would have been obvious to a person having ordinary skill in the art, as explained below. IEEE 802.3 at Figure 43-2

No.	’740 Patent Claim 3	IEEE 802.3
		 Figure 43–2—Link Aggregation sublayer block diagram

MAC Client
802.3® MAC Service Interface
Agg: MA_UNITDATA ( )
Frame Collection
Frame Distribution
Link Aggregation Control
Link Aggregation Control Protocol
shared state variables
Marker Responder
Frame Collector
Marker Generator/ Receiver (optional)
Frame Distributor
Marker Frames
MAC Client Frames
Marker Frames
MAC Client Frames
802.3® MAC Service Interface(s)
AggMuxN: MA_UNITDATA ( )
Aggregator Parser/Multiplexer
Aggregator Parser/Multiplexer
Aggregator Parser/Multiplexer
Aggregation Control
Aggregator
Control frames
MAC Client and Marker frames
802.3® MAC Service Interface(s)
CtrlMuxN: MA_UNITDATA ( )
Control Parser/Multiplexer
Control Parser/Multiplexer
Control Parser/Multiplexer
Link Aggregation sublayer
802.3® MAC Service Interface
MacN: MA_UNITDATA ( )
MAC Control (optional)
MAC
Physical Layer
MAC Control (optional)
MAC
Physical Layer
MAC Control (optional)
MAC
Physical Layer

No.	’740 Patent Claim 3	IEEE 802.3
		IEEE 802.3 at 1468  **43.2 Link Aggregation operation** As depicted in Figure 43–2, the Link Aggregation sublayer comprises the following functions: a) *Frame Distribution*. This block is responsible for taking frames submitted by the MAC Client and submitting them for transmission on the appropriate port, based on a frame distribution algorithm employed by the Frame Distributor. Frame Distribution also includes an optional *Marker Generator/ Receiver* used for the Marker protocol. (See 43.2.4, 43.2.5, and 43.5.) b) *Frame Collection*. This block is responsible for passing frames received from the various ports to the MAC Client. Frame Collection also includes a *Marker Responder,* used for the Marker protocol. (See 43.2.3 and 43.5.) c) *Aggregator Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Distributor, Marker Generator, and/or Marker Responder to the appropriate port. On receive, these blocks distinguish among Marker Request, Marker Response, and MAC Client PDUs, and pass each to the appropriate entity (Marker Responder, Marker Receiver, and Collector, respectively). d) *Aggregator.* The combination of Frame Distribution and Collection, along with the Aggregator Parser/Multiplexers, is referred to as the Aggregator. e) *Aggregation Control*. This block is responsible for the configuration and control of Link Aggregation. It incorporates a *Link Aggregation Control Protocol* (LACP) that can be used for automatic communication of aggregation capabilities between Systems and automatic configuration of Link Aggregation. f) *Control Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Aggregator and Control entities to the appropriate port. On receive, these blocks distinguish Link Aggregation Control PDUs from other frames, passing the LACPDUs to the appropriate sublayer entity, and all other frames to the Aggregator. IEEE 802.3 at 1468-69

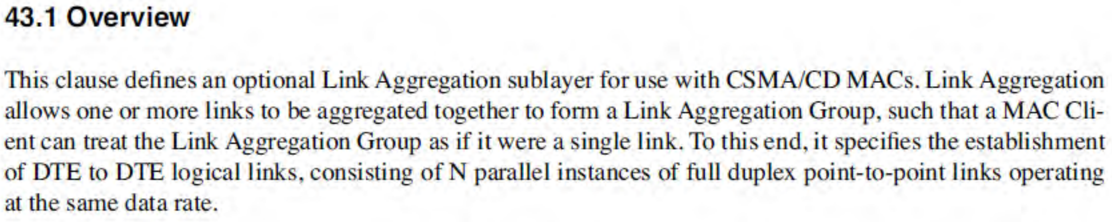
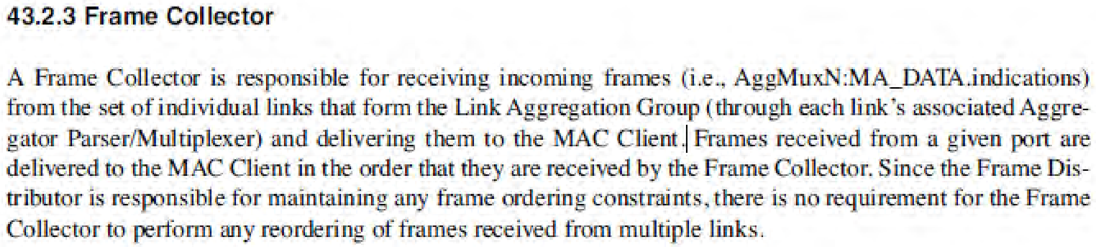
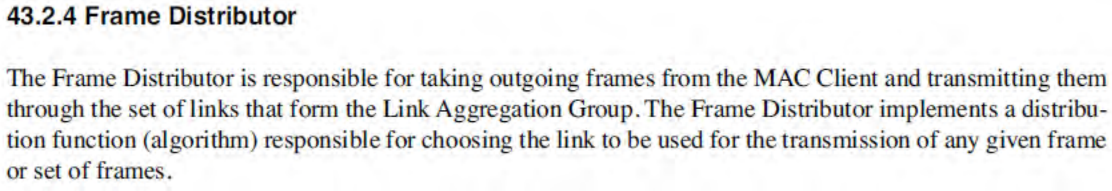
No.	’740 Patent Claim 3	IEEE 802.3
		 **43.2.1 Principles of Link Aggregation** Link Aggregation allows a MAC Client to treat a set of one or more ports as if it were a single port. In doing so, it employs the following principles and concepts: a) A MAC Client communicates with a set of ports through an Aggregator, which presents a standard IEEE 802.3® service interface to the MAC Client. The Aggregator binds to one or more ports within a System. b) It is the responsibility of the Aggregator to distribute frame transmissions from the MAC Client to the various ports, and to collect received frames from the ports and pass them to the MAC Client transparently. c) A System may contain multiple Aggregators, serving multiple MAC Clients. A given port will bind to (at most) a single Aggregator at any time. A MAC Client is served by a single Aggregator at a time. d) The binding of ports to Aggregators within a System is managed by the Link Aggregation Control function for that System, which is responsible for determining which links may be aggregated, aggregating them, binding the ports within the System to an appropriate Aggregator, and monitoring conditions to determine when a change in aggregation is needed. e) Such determination and binding may be under manual control through direct manipulation of the state variables of Link Aggregation (e.g., Keys) by a network manager. In addition, automatic determination, configuration, binding, and monitoring may occur through the use of a Link Aggregation Control Protocol (LACP). The LACP uses peer exchanges across the links to determine, on an ongoing basis, the aggregation capability of the various links, and continuously provides the maximum level of aggregation capability achievable between a given pair of Systems. f) Frame ordering must be maintained for certain sequences of frame exchanges between MAC Clients (known as conversations, see 1.4). The Distributor ensures that all frames of a given conversation are passed to a single port. For any given port, the Collector is required to pass frames to the MAC Client in the order that they are received from that port. The Collector is otherwise free to select frames received from the aggregated ports in any order. Since there are no means for frames to be misordered on a single link, this guarantees that frame ordering is maintained for any conversation. g) Conversations may be moved among ports within an aggregation, both for load balancing and to maintain availability in the event of link failures. h) This standard does not impose any particular distribution algorithm on the Distributor. Whatever algorithm is used should be appropriate for the MAC Client being supported. i) Each port is assigned a unique, globally administered MAC address. This MAC address is used as the source address for frame exchanges that are initiated by entities within the Link Aggregation sublayer itself (i.e., LACP and Marker protocol exchanges). NOTE—The LACP and Marker protocols use a multicast destination address for all exchanges, and do not impose any requirement for a port to recognize more than one unicast address on received frames. j) Each Aggregator is assigned a unique, globally administered MAC address; this address is used as the MAC address of the aggregation from the perspective of the MAC Client, both as a source address for transmitted frames and as the destination address for received frames. The MAC address of the Aggregator may be one of the MAC addresses of a port in the associated Link Aggregation Group (see 43.2.10).

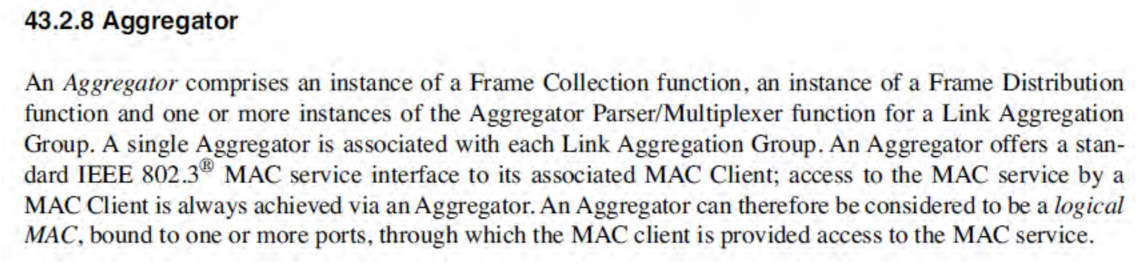
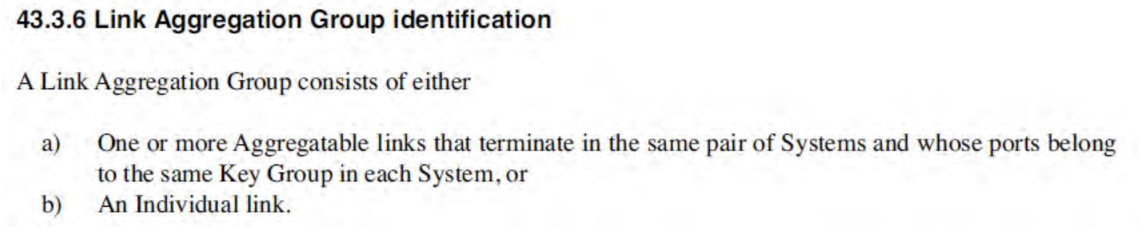
No.	’740 Patent Claim 3	IEEE 802.3
		Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, IEEE 802.3 in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 3 of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of three such references. For example, Ghosh discloses connecting the line cards to the active supervisor via the backplane using parallel interface circuitry. Ghosh at [0059] (“Line cards 803, 805, and 807 can communicate with an active supervisor 811 through interface circuitry 883, 885, and 887 and the backplane 815. According to various embodiments, each line card includes a plurality of ports that can act as either input ports or output ports for communication with external fibre channel network entities 851 and 853. The backplane 815 can provide a communi-cations channel for all traffic between line cards and super- visors. Individual line cards 803 and 807 can also be coupled to external fibre channel network entities 851 and 853 through fibre channel ports 843 and 847.”) For example, Bruckman discloses connecting the line cards to the network using traces comprising a backplane. Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link

No.	’740 Patent Claim 3	IEEE 802.3
		subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”) For example, Basso discloses coupling the blades to the communication network via link connections to the switch fabric. Basso at [0009] (“A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more ports. These ports may be connected to another one or more network devices. The switch fabric may be configured to direct incoming packets of data to particular blades where one or more of the network proces-sors in the recipient blade may be configured to process the received packets.”)

No.	’740 Patent Claim 4	IEEE 802.3
4[preamble]	A method for communication, comprising:	IEEE 802.3 discloses a method for communication. *See supra* at 1[preamble].
4[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	IEEE 802.3 discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].
4[b]	coupling each of the one or more interface modules to a communication network using a second group of	IEEE 802.3 discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra* at 1[c].

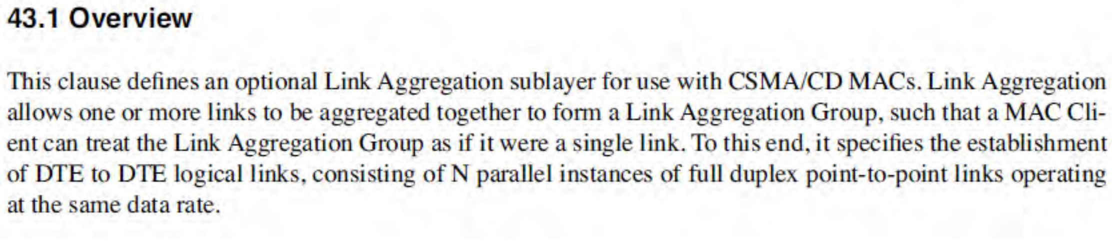
No.	’740 Patent Claim 4	IEEE 802.3
	second physical links arranged in parallel;	
4[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	IEEE 802.3 discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra* at 1[e].
4[d]	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	IEEE 802.3 discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. *See supra* at 1[f].
4[e]	sending the data frame over the selected first and second physical links,	IEEE 802.3 discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].
4[f]	at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group.	IEEE 802.3 discloses at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group. For example, IEEE 802.3 discloses the aggregation of one or more links together to form a Link Aggregation Group. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. IEEE 802.3 at 1465

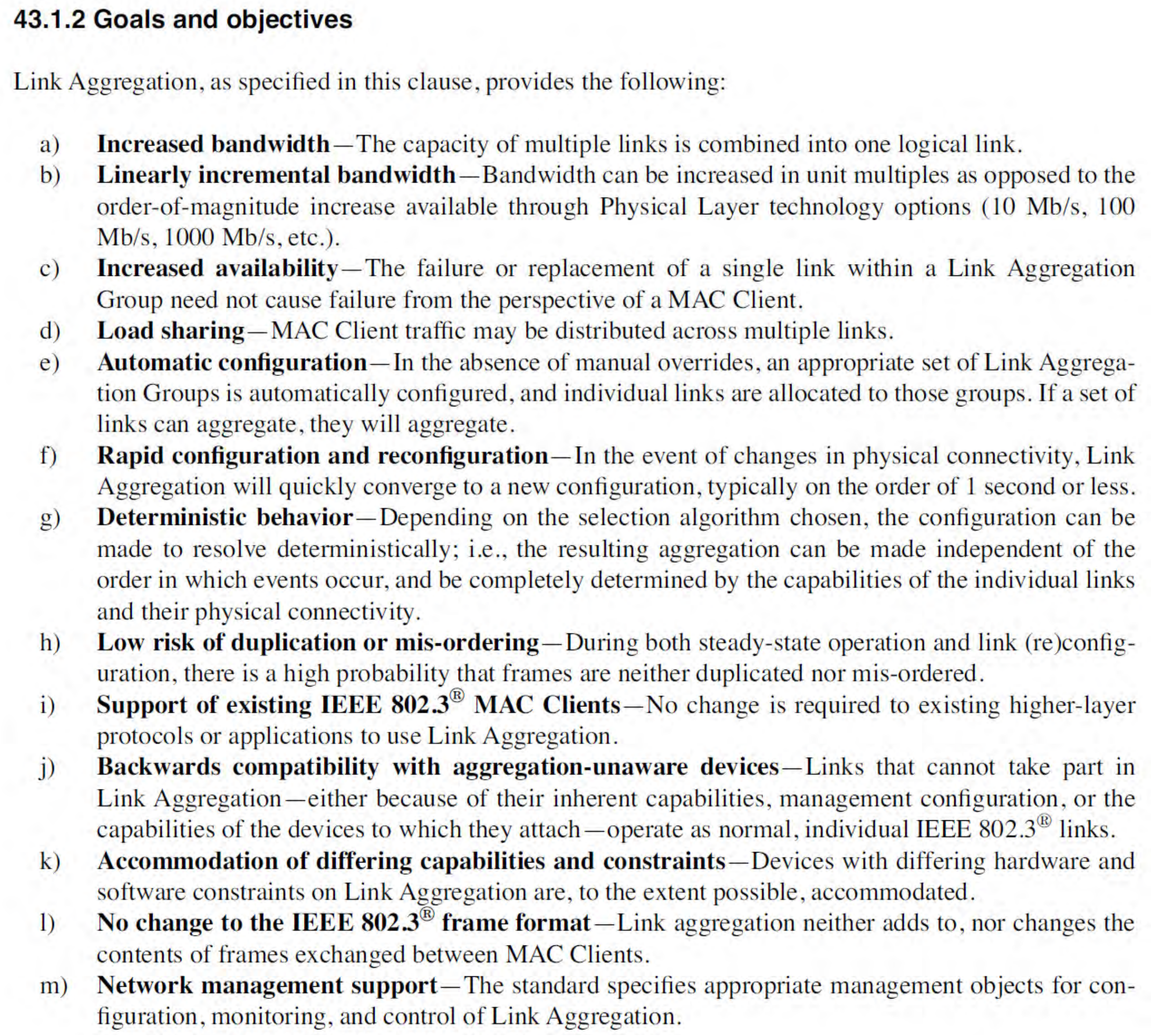
No.	’740 Patent Claim 4	IEEE 802.3
		 **43.1 Overview** This clause defines an optional Link Aggregation sublayer for use with CSMA/CD MACs. Link Aggregation allows one or more links to be aggregated together to form a Link Aggregation Group, such that a MAC Client can treat the Link Aggregation Group as if it were a single link. To this end, it specifies the establishment of DTE to DTE logical links, consisting of N parallel instances of full duplex point-to-point links operating at the same data rate. IEEE 802.3 at 1470  **43.2.3 Frame Collector** A Frame Collector is responsible for receiving incoming frames (i.e., AggMuxN:MA_DATA.indications) from the set of individual links that form the Link Aggregation Group (through each link’s associated Aggregator Parser/Multiplexer) and delivering them to the MAC Client. Frames received from a given port are delivered to the MAC Client in the order that they are received by the Frame Collector. Since the Frame Distributor is responsible for maintaining any frame ordering constraints, there is no requirement for the Frame Collector to perform any reordering of frames received from multiple links. IEEE 802.3 at 1471  **43.2.4 Frame Distributor** The Frame Distributor is responsible for taking outgoing frames from the MAC Client and transmitting them through the set of links that form the Link Aggregation Group. The Frame Distributor implements a distribution function (algorithm) responsible for choosing the link to be used for the transmission of any given frame or set of frames. IEEE 802.3 at 1474

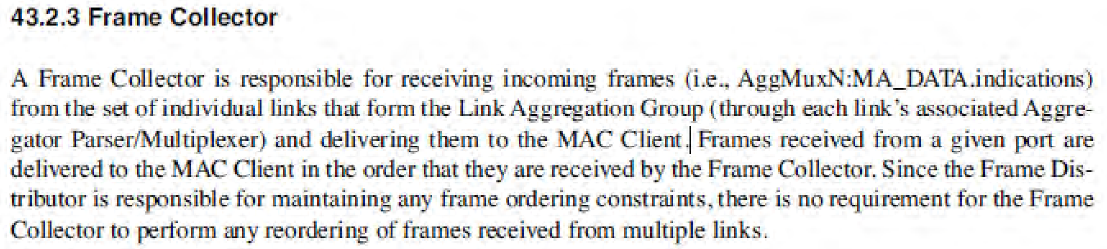
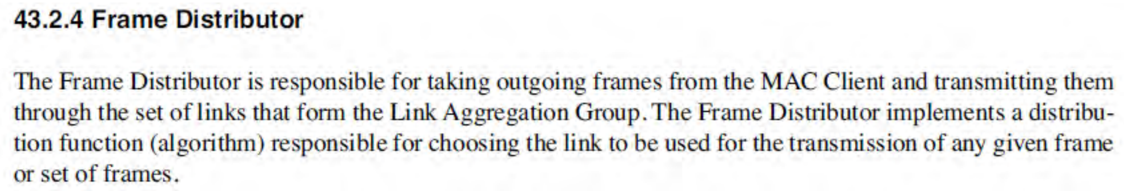
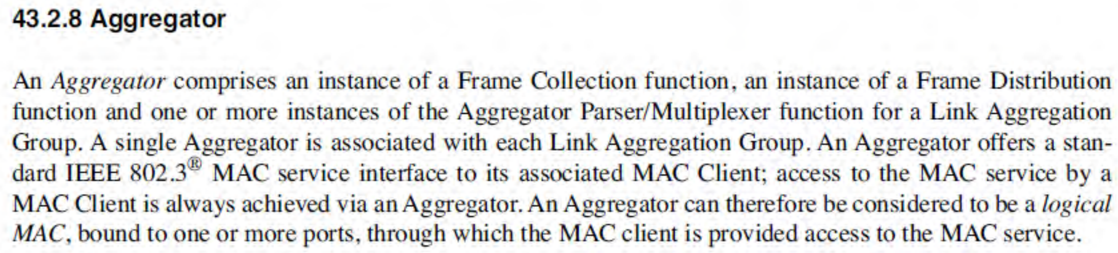
No.	’740 Patent Claim 4	IEEE 802.3
		 **43.2.8 Aggregator** An *Aggregator* comprises an instance of a Frame Collection function, an instance of a Frame Distribution function and one or more instances of the Aggregator Parser/Multiplexer function for a Link Aggregation Group. A single Aggregator is associated with each Link Aggregation Group. An Aggregator offers a standard IEEE 802.3® MAC service interface to its associated MAC Client; access to the MAC service by a MAC Client is always achieved via an Aggregator. An Aggregator can therefore be considered to be a *logical MAC*, bound to one or more ports, through which the MAC client is provided access to the MAC service. IEEE 802.3 at 1481  **43.3.6 Link Aggregation Group identification** A Link Aggregation Group consists of either a) One or more Aggregatable links that terminate in the same pair of Systems and whose ports belong to the same Key Group in each System, or b) An Individual link.

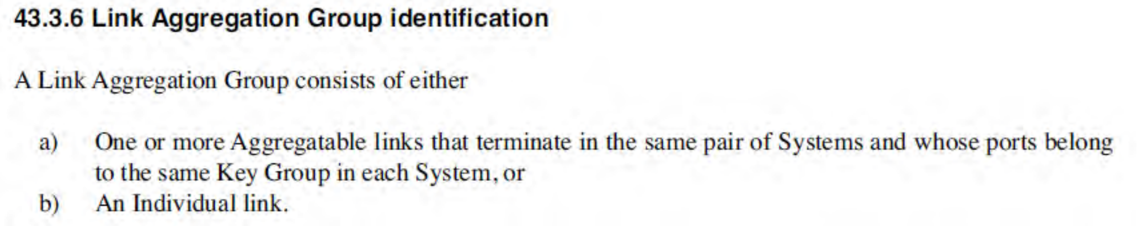
No.	’740 Patent Claim 5	IEEE 802.3
5[preamble]	A method for communication, comprising:	IEEE 802.3 discloses a method for communication. *See supra at* 1[preamble].
5[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	IEEE 802.3 discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].

No.	’740 Patent Claim 5	IEEE 802.3
5[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	IEEE 802.3 discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra at* 1[c].
5[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	IEEE 802.3 discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra at* 1[e].
5[d]	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	IEEE 802.3 discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. *See supra at* 1[f].
5[e]	sending the data frame over the selected first and	IEEE 802.3 discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].

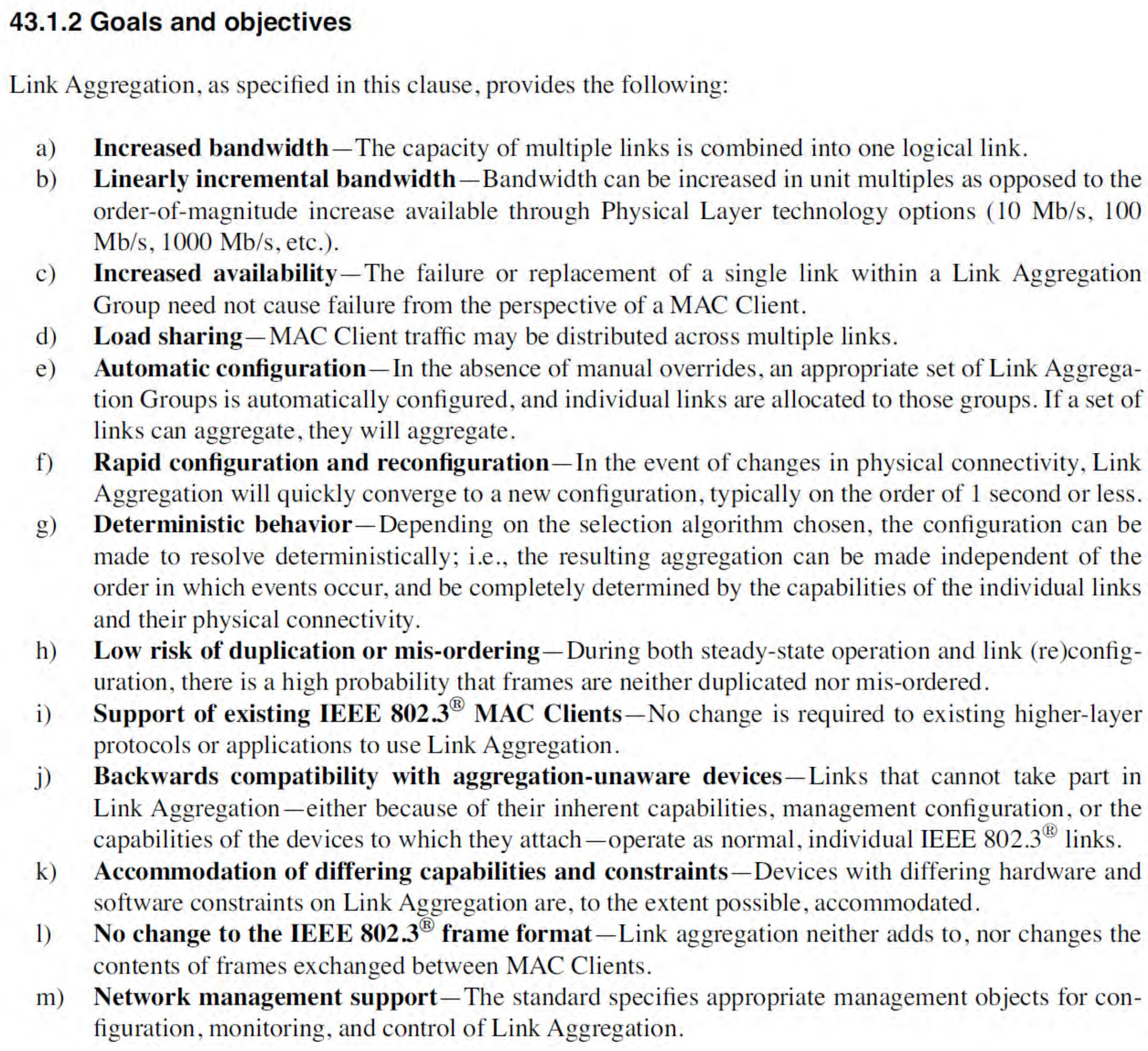
No.	’740 Patent Claim 5	IEEE 802.3
	second physical links,	
5[f]	coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node.	IEEE 802.3 discloses coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node. For example, IEEE 802.3 discloses connecting a MAC Client to the aggregator by aggregating one or more links together to form a Link Aggregation Group in order to increase bandwidth. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. IEEE 802.3 at 1465  **43.1 Overview** This clause defines an optional Link Aggregation sublayer for use with CSMA/CD MACs. Link Aggregation allows one or more links to be aggregated together to form a Link Aggregation Group, such that a MAC Client can treat the Link Aggregation Group as if it were a single link. To this end, it specifies the establishment of DTE to DTE logical links, consisting of N parallel instances of full duplex point-to-point links operating at the same data rate. IEEE 802.3 at 1465

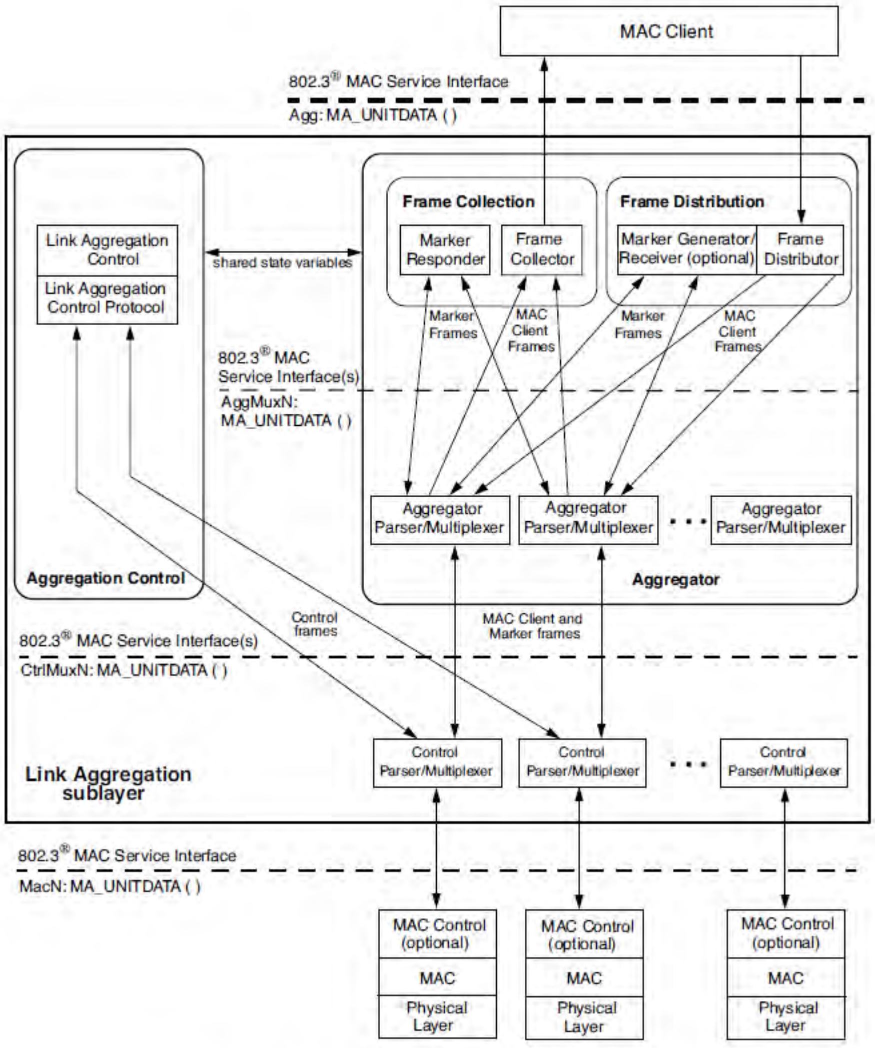
No.	’740 Patent Claim 5	IEEE 802.3
		 **43.1.2 Goals and objectives** Link Aggregation, as specified in this clause, provides the following: a) **Increased bandwidth**—The capacity of multiple links is combined into one logical link. b) **Linearly incremental bandwidth**—Bandwidth can be increased in unit multiples as opposed to the order-of-magnitude increase available through Physical Layer technology options (10 Mb/s, 100 Mb/s, 1000 Mb/s, etc.). c) **Increased availability**—The failure or replacement of a single link within a Link Aggregation Group need not cause failure from the perspective of a MAC Client. d) **Load sharing**—MAC Client traffic may be distributed across multiple links. e) **Automatic configuration**—In the absence of manual overrides, an appropriate set of Link Aggregation Groups is automatically configured, and individual links are allocated to those groups. If a set of links can aggregate, they will aggregate. f) **Rapid configuration and reconfiguration**—In the event of changes in physical connectivity, Link Aggregation will quickly converge to a new configuration, typically on the order of 1 second or less. g) **Deterministic behavior**—Depending on the selection algorithm chosen, the configuration can be made to resolve deterministically; i.e., the resulting aggregation can be made independent of the order in which events occur, and be completely determined by the capabilities of the individual links and their physical connectivity. h) **Low risk of duplication or mis-ordering**—During both steady-state operation and link (re)configuration, there is a high probability that frames are neither duplicated nor mis-ordered. i) **Support of existing IEEE 802.3® MAC Clients**—No change is required to existing higher-layer protocols or applications to use Link Aggregation. j) **Backwards compatibility with aggregation-unaware devices**—Links that cannot take part in Link Aggregation—either because of their inherent capabilities, management configuration, or the capabilities of the devices to which they attach—operate as normal, individual IEEE 802.3® links. k) **Accommodation of differing capabilities and constraints**—Devices with differing hardware and software constraints on Link Aggregation are, to the extent possible, accommodated. l) **No change to the IEEE 802.3® frame format**—Link aggregation neither adds to, nor changes the contents of frames exchanged between MAC Clients. m) **Network management support**—The standard specifies appropriate management objects for configuration, monitoring, and control of Link Aggregation.

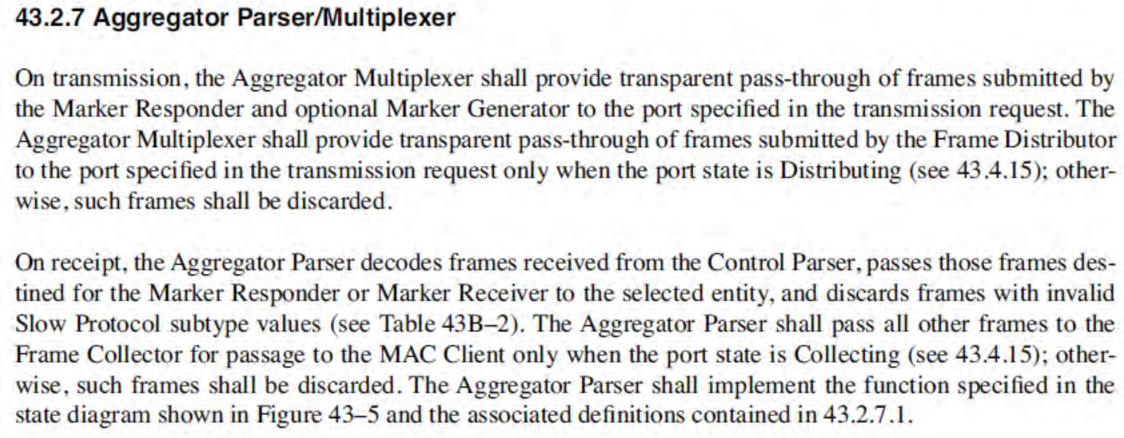
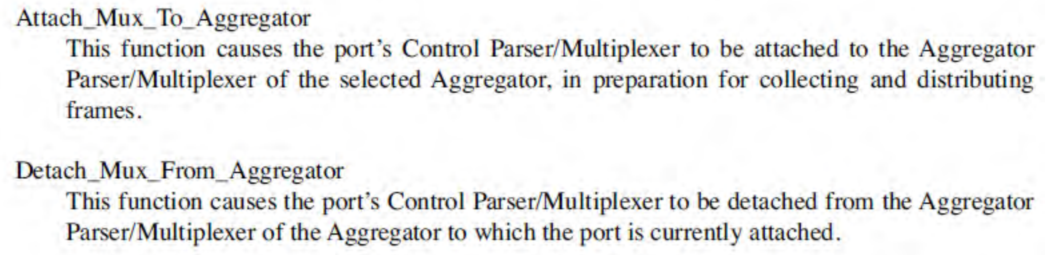
No.	’740 Patent Claim 5	IEEE 802.3
		IEEE 802.3 at 1470  **43.2.3 Frame Collector** A Frame Collector is responsible for receiving incoming frames (i.e., AggMuxN:MA_DATA.indications) from the set of individual links that form the Link Aggregation Group (through each link’s associated Aggregator Parser/Multiplexer) and delivering them to the MAC Client. Frames received from a given port are delivered to the MAC Client in the order that they are received by the Frame Collector. Since the Frame Distributor is responsible for maintaining any frame ordering constraints, there is no requirement for the Frame Collector to perform any reordering of frames received from multiple links. IEEE 802.3 at 1471  **43.2.4 Frame Distributor** The Frame Distributor is responsible for taking outgoing frames from the MAC Client and transmitting them through the set of links that form the Link Aggregation Group. The Frame Distributor implements a distribution function (algorithm) responsible for choosing the link to be used for the transmission of any given frame or set of frames. IEEE 802.3 at 1474  **43.2.8 Aggregator** An *Aggregator* comprises an instance of a Frame Collection function, an instance of a Frame Distribution function and one or more instances of the Aggregator Parser/Multiplexer function for a Link Aggregation Group. A single Aggregator is associated with each Link Aggregation Group. An Aggregator offers a standard IEEE 802.3® MAC service interface to its associated MAC Client; access to the MAC service by a MAC Client is always achieved via an Aggregator. An Aggregator can therefore be considered to be a *logical MAC*, bound to one or more ports, through which the MAC client is provided access to the MAC service. IEEE 802.3 at 1481

No.	’740 Patent Claim 5	IEEE 802.3
		 **43.3.6 Link Aggregation Group identification** A Link Aggregation Group consists of either a) One or more Aggregatable links that terminate in the same pair of Systems and whose ports belong to the same Key Group in each System, or b) An Individual link.

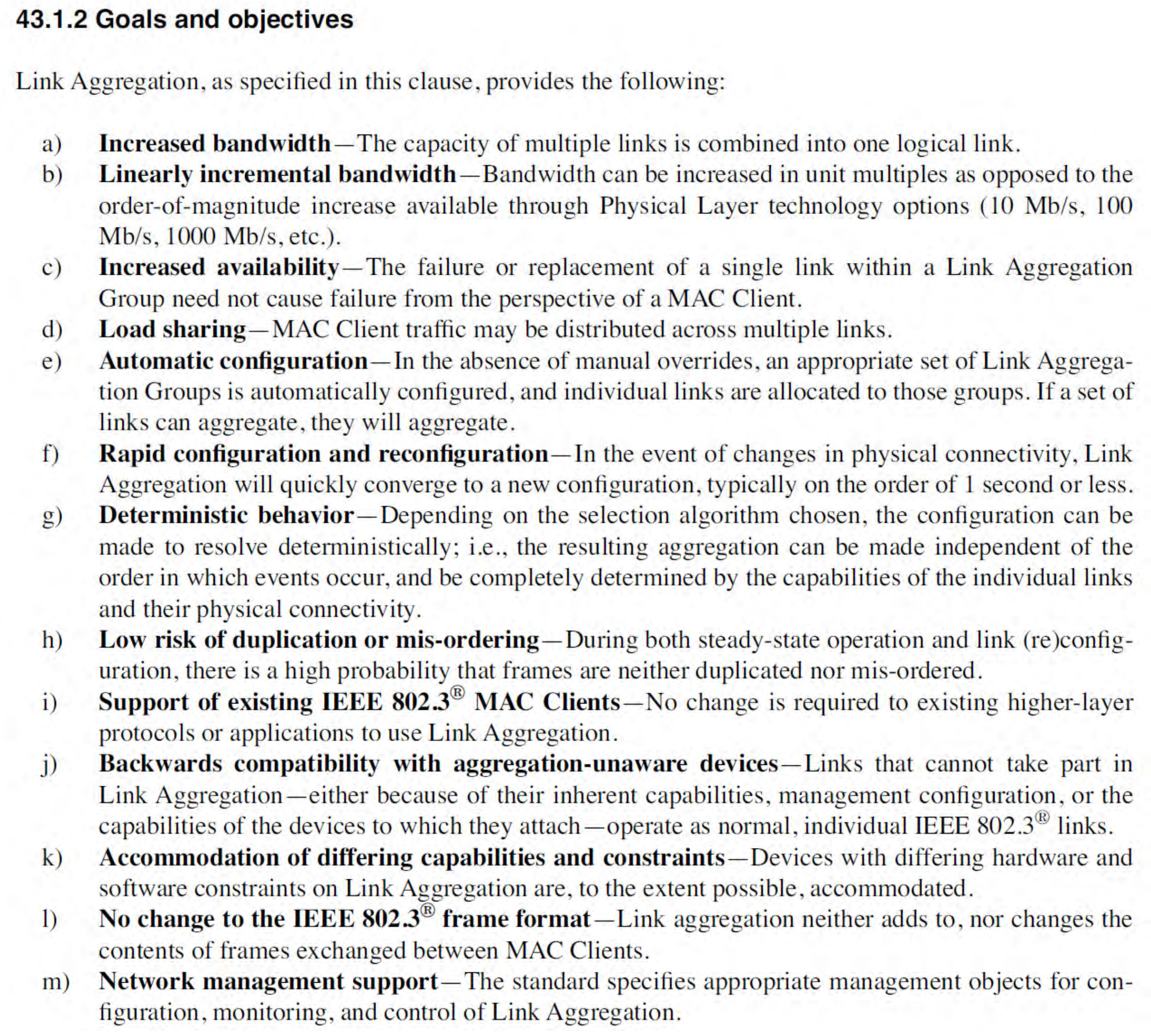
No.	’740 Patent Claim 6	IEEE 802.3
6	The method according to claim 1, wherein coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication	IEEE 802.3 discloses the method according to claim 1, wherein coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node. For example, IEEE 802.3 discloses multiplexing data frames via the aggregator multiplexer and parsing data frames via the aggregator parser. *See supra* Claim 1. IEEE 802.3 at 1465

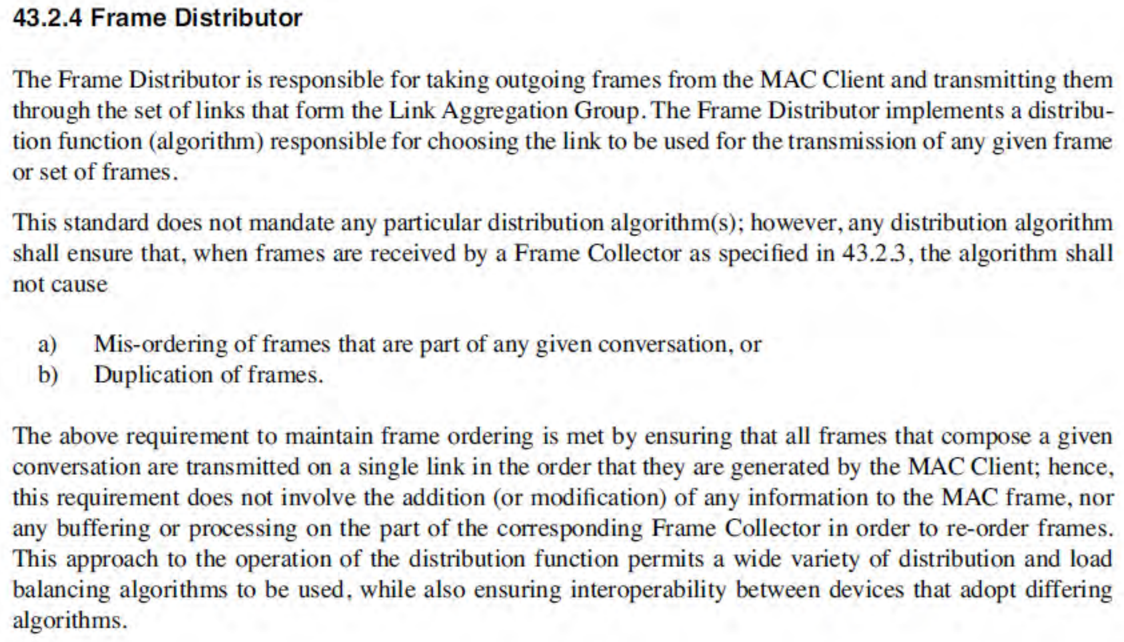
No.	’740 Patent Claim 6	IEEE 802.3
	network to the network node.	 **43.1.2 Goals and objectives** Link Aggregation, as specified in this clause, provides the following: a) **Increased bandwidth**—The capacity of multiple links is combined into one logical link. b) **Linearly incremental bandwidth**—Bandwidth can be increased in unit multiples as opposed to the order-of-magnitude increase available through Physical Layer technology options (10 Mb/s, 100 Mb/s, 1000 Mb/s, etc.). c) **Increased availability**—The failure or replacement of a single link within a Link Aggregation Group need not cause failure from the perspective of a MAC Client. d) **Load sharing**—MAC Client traffic may be distributed across multiple links. e) **Automatic configuration**—In the absence of manual overrides, an appropriate set of Link Aggregation Groups is automatically configured, and individual links are allocated to those groups. If a set of links can aggregate, they will aggregate. f) **Rapid configuration and reconfiguration**—In the event of changes in physical connectivity, Link Aggregation will quickly converge to a new configuration, typically on the order of 1 second or less. g) **Deterministic behavior**—Depending on the selection algorithm chosen, the configuration can be made to resolve deterministically; i.e., the resulting aggregation can be made independent of the order in which events occur, and be completely determined by the capabilities of the individual links and their physical connectivity. h) **Low risk of duplication or mis-ordering**—During both steady-state operation and link (re)configuration, there is a high probability that frames are neither duplicated nor mis-ordered. i) **Support of existing IEEE 802.3® MAC Clients**—No change is required to existing higher-layer protocols or applications to use Link Aggregation. j) **Backwards compatibility with aggregation-unaware devices**—Links that cannot take part in Link Aggregation—either because of their inherent capabilities, management configuration, or the capabilities of the devices to which they attach—operate as normal, individual IEEE 802.3® links. k) **Accommodation of differing capabilities and constraints**—Devices with differing hardware and software constraints on Link Aggregation are, to the extent possible, accommodated. l) **No change to the IEEE 802.3® frame format**—Link aggregation neither adds to, nor changes the contents of frames exchanged between MAC Clients. m) **Network management support**—The standard specifies appropriate management objects for configuration, monitoring, and control of Link Aggregation.

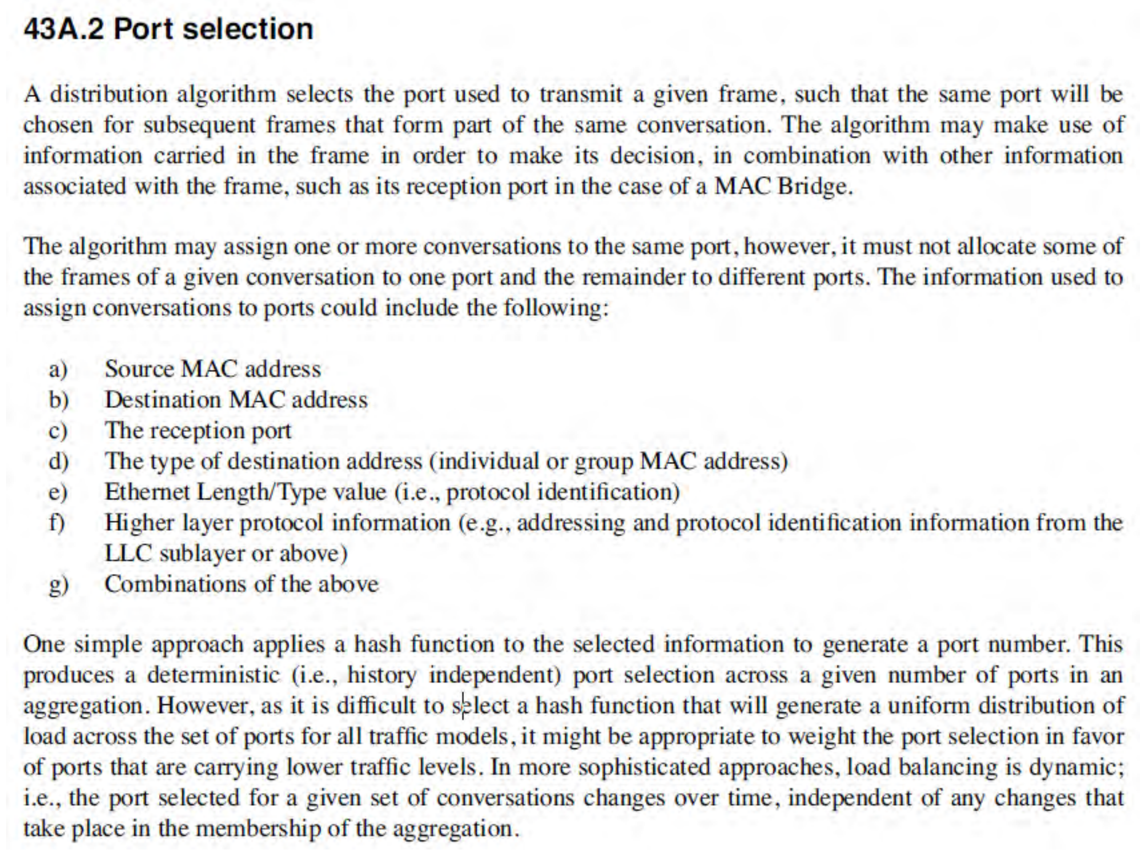
No.	’740 Patent Claim 6	IEEE 802.3
		IEEE 802.3 at Figure 43-2  MAC Client 802.3® MAC Service Interface Agg: MA_UNITDATA ( ) Link Aggregation Control Link Aggregation Control Protocol shared state variables Frame Collection Marker Responder Frame Collector Frame Distribution Marker Generator/ Receiver (optional) Frame Distributor Marker Frames MAC Client Frames Marker Frames MAC Client Frames 802.3® MAC Service Interface(s) AggMuxN: MA_UNITDATA ( ) Aggregator Parser/Multiplexer Aggregator Parser/Multiplexer Aggregator Parser/Multiplexer Aggregation Control Aggregator Control frames MAC Client and Marker frames 802.3® MAC Service Interface(s) CtrlMuxN: MA_UNITDATA ( ) Control Parser/Multiplexer Control Parser/Multiplexer Control Parser/Multiplexer Link Aggregation sublayer 802.3® MAC Service Interface MacN: MA_UNITDATA ( ) MAC Control (optional) MAC Physical Layer MAC Control (optional) MAC Physical Layer MAC Control (optional) MAC Physical Layer **Figure 43–2—Link Aggregation sublayer block diagram**

No.	’740 Patent Claim 6	IEEE 802.3
		IEEE 802.3 at 1473  **43.2.7 Aggregator Parser/Multiplexer** On transmission, the Aggregator Multiplexer shall provide transparent pass-through of frames submitted by the Marker Responder and optional Marker Generator to the port specified in the transmission request. The Aggregator Multiplexer shall provide transparent pass-through of frames submitted by the Frame Distributor to the port specified in the transmission request only when the port state is Distributing (see 43.4.15); otherwise, such frames shall be discarded. On receipt, the Aggregator Parser decodes frames received from the Control Parser, passes those frames destined for the Marker Responder or Marker Receiver to the selected entity, and discards frames with invalid Slow Protocol subtype values (see Table 43B–2). The Aggregator Parser shall pass all other frames to the Frame Collector for passage to the MAC Client only when the port state is Collecting (see 43.4.15); otherwise, such frames shall be discarded. The Aggregator Parser shall implement the function specified in the state diagram shown in Figure 43–5 and the associated definitions contained in 43.2.7.1. IEEE 802.3 at 1499  Attach_Mux_To_Aggregator This function causes the port’s Control Parser/Multiplexer to be attached to the Aggregator Parser/Multiplexer of the selected Aggregator, in preparation for collecting and distributing frames. Detach_Mux_From_Aggregator This function causes the port’s Control Parser/Multiplexer to be detached from the Aggregator Parser/Multiplexer of the Aggregator to which the port is currently attached.

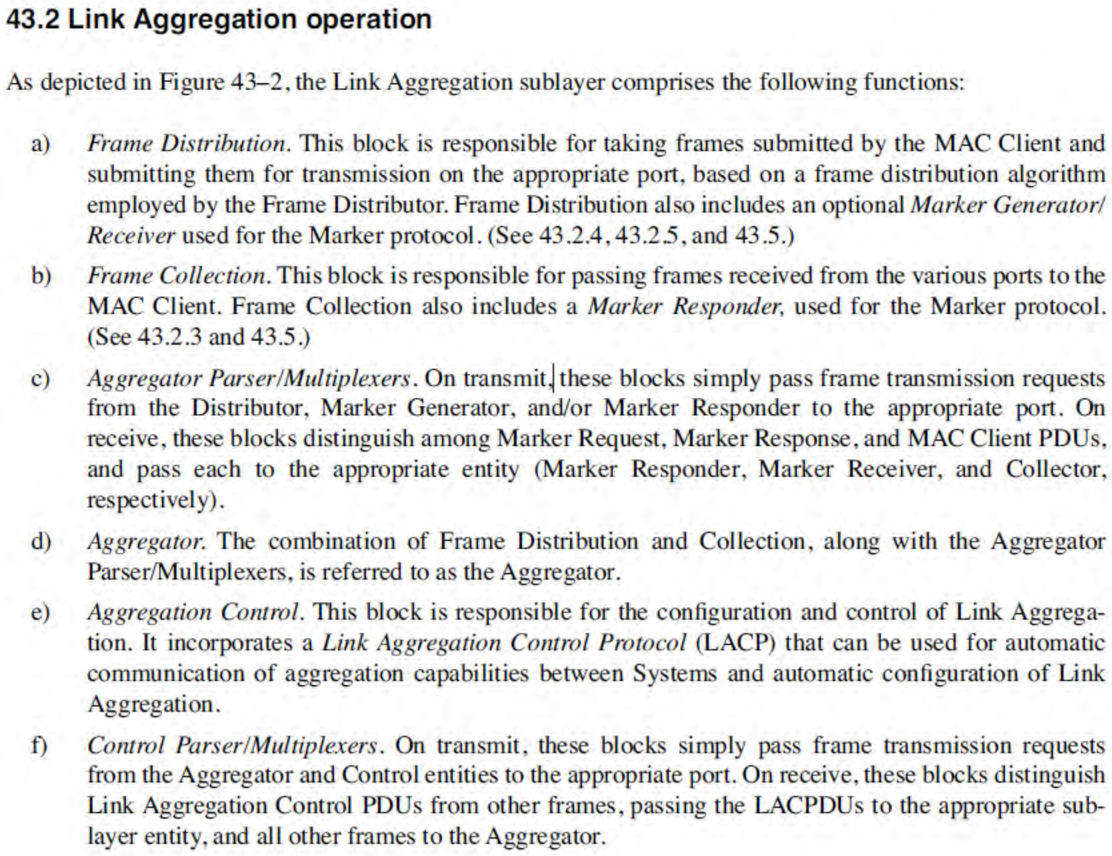
No.	’740 Patent Claim 7	IEEE 802.3
7	The method according to claim 1, wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the first and second physical links.	IEEE 802.3 discloses the method according to claim 1, wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the first and second physical links. For example, IEEE 802.3 discloses a distribution algorithm used to select links over which to send data frames that is based on load balancing and load sharing across the links. *See supra* Claim 1. IEEE 802.3 at 1465

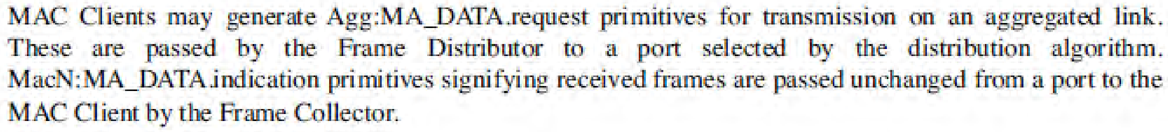
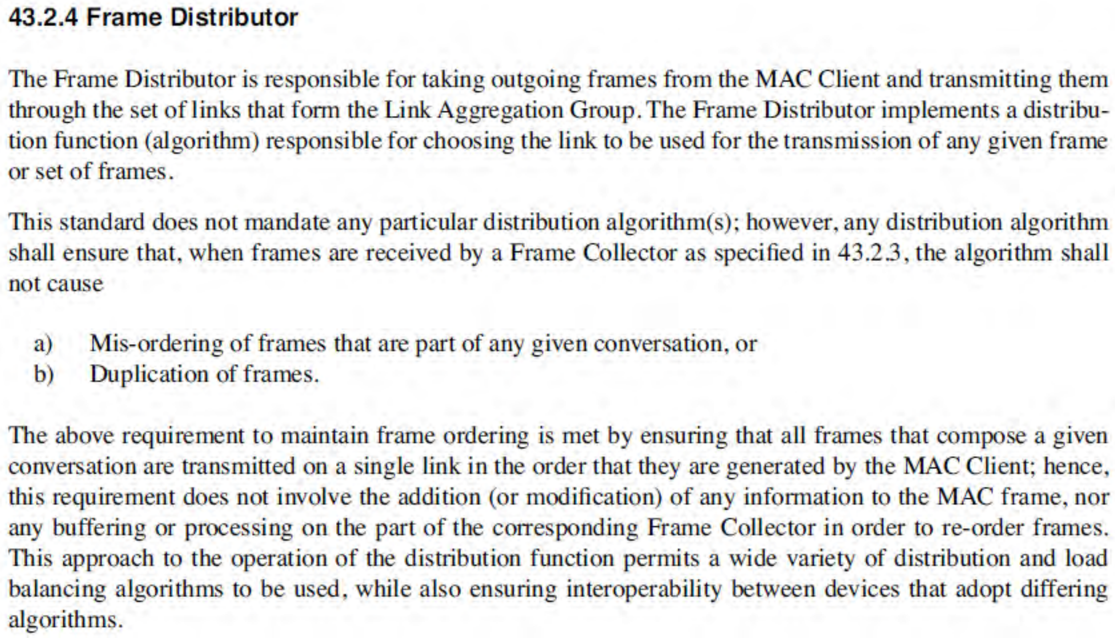
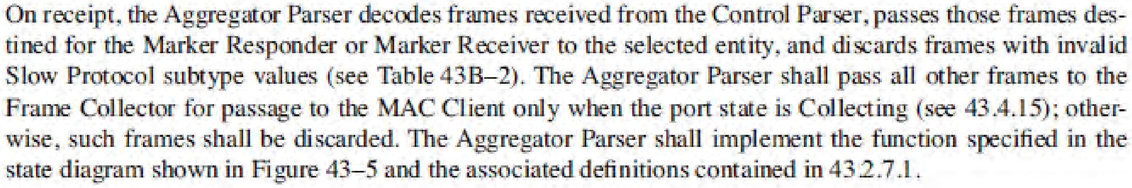
No.	’740 Patent Claim 7	IEEE 802.3
		 **43.1.2 Goals and objectives** Link Aggregation, as specified in this clause, provides the following: a) **Increased bandwidth**—The capacity of multiple links is combined into one logical link. b) **Linearly incremental bandwidth**—Bandwidth can be increased in unit multiples as opposed to the order-of-magnitude increase available through Physical Layer technology options (10 Mb/s, 100 Mb/s, 1000 Mb/s, etc.). c) **Increased availability**—The failure or replacement of a single link within a Link Aggregation Group need not cause failure from the perspective of a MAC Client. d) **Load sharing**—MAC Client traffic may be distributed across multiple links. e) **Automatic configuration**—In the absence of manual overrides, an appropriate set of Link Aggregation Groups is automatically configured, and individual links are allocated to those groups. If a set of links can aggregate, they will aggregate. f) **Rapid configuration and reconfiguration**—In the event of changes in physical connectivity, Link Aggregation will quickly converge to a new configuration, typically on the order of 1 second or less. g) **Deterministic behavior**—Depending on the selection algorithm chosen, the configuration can be made to resolve deterministically; i.e., the resulting aggregation can be made independent of the order in which events occur, and be completely determined by the capabilities of the individual links and their physical connectivity. h) **Low risk of duplication or mis-ordering**—During both steady-state operation and link (re)configuration, there is a high probability that frames are neither duplicated nor mis-ordered. i) **Support of existing IEEE 802.3$^{\circledR}$ MAC Clients**—No change is required to existing higher-layer protocols or applications to use Link Aggregation. j) **Backwards compatibility with aggregation-unaware devices**—Links that cannot take part in Link Aggregation—either because of their inherent capabilities, management configuration, or the capabilities of the devices to which they attach—operate as normal, individual IEEE 802.3$^{\circledR}$ links. k) **Accommodation of differing capabilities and constraints**—Devices with differing hardware and software constraints on Link Aggregation are, to the extent possible, accommodated. l) **No change to the IEEE 802.3$^{\circledR}$ frame format**—Link aggregation neither adds to, nor changes the contents of frames exchanged between MAC Clients. m) **Network management support**—The standard specifies appropriate management objects for configuration, monitoring, and control of Link Aggregation.

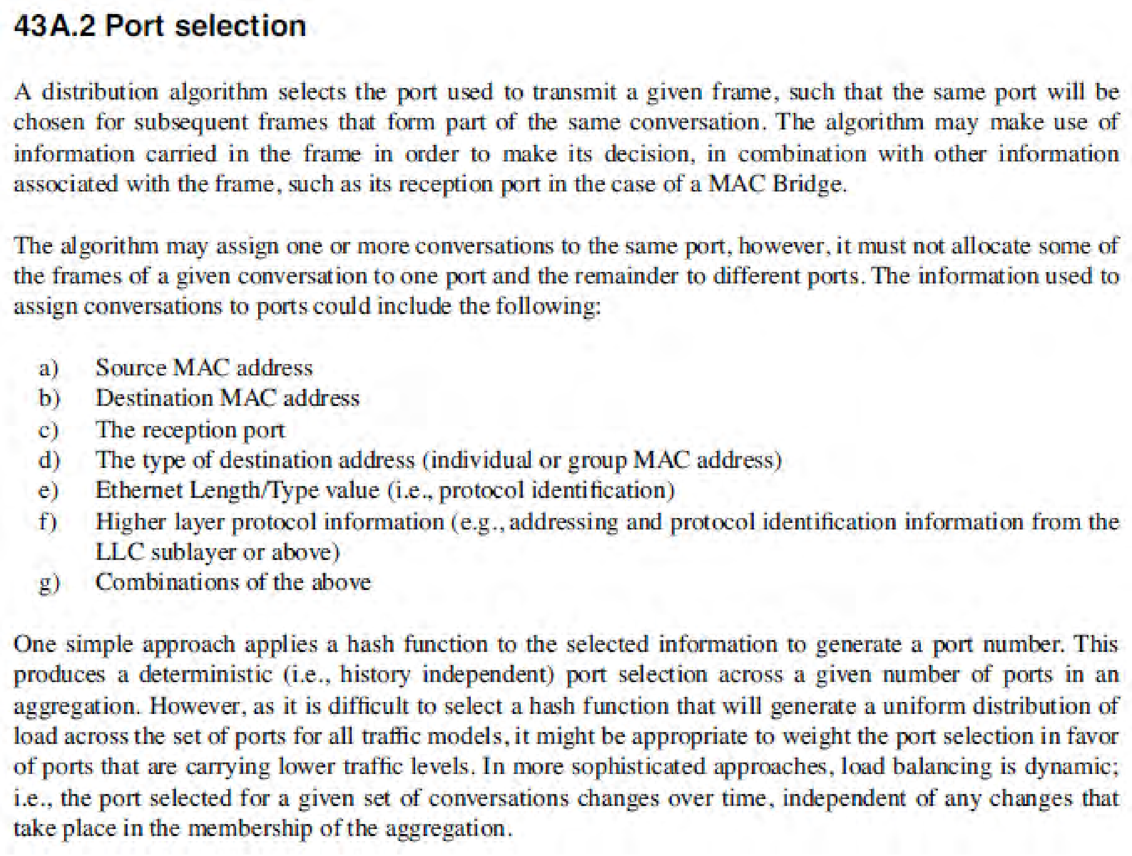
No.	’740 Patent Claim 7	IEEE 802.3
		IEEE 802.3 at 1471  43.2.4 Frame Distributor The Frame Distributor is responsible for taking outgoing frames from the MAC Client and transmitting them through the set of links that form the Link Aggregation Group. The Frame Distributor implements a distribution function (algorithm) responsible for choosing the link to be used for the transmission of any given frame or set of frames. This standard does not mandate any particular distribution algorithm(s); however, any distribution algorithm shall ensure that, when frames are received by a Frame Collector as specified in 43.2.3, the algorithm shall not cause a) Mis-ordering of frames that are part of any given conversation, or b) Duplication of frames. The above requirement to maintain frame ordering is met by ensuring that all frames that compose a given conversation are transmitted on a single link in the order that they are generated by the MAC Client; hence, this requirement does not involve the addition (or modification) of any information to the MAC frame, nor any buffering or processing on the part of the corresponding Frame Collector in order to re-order frames. This approach to the operation of the distribution function permits a wide variety of distribution and load balancing algorithms to be used, while also ensuring interoperability between devices that adopt differing algorithms. IEEE 802.3 at 1549

No.	’740 Patent Claim 7	IEEE 802.3
		 **43A.2 Port selection** A distribution algorithm selects the port used to transmit a given frame, such that the same port will be chosen for subsequent frames that form part of the same conversation. The algorithm may make use of information carried in the frame in order to make its decision, in combination with other information associated with the frame, such as its reception port in the case of a MAC Bridge. The algorithm may assign one or more conversations to the same port, however, it must not allocate some of the frames of a given conversation to one port and the remainder to different ports. The information used to assign conversations to ports could include the following: a) Source MAC address b) Destination MAC address c) The reception port d) The type of destination address (individual or group MAC address) e) Ethernet Length/Type value (i.e., protocol identification) f) Higher layer protocol information (e.g., addressing and protocol identification information from the LLC sublayer or above) g) Combinations of the above One simple approach applies a hash function to the selected information to generate a port number. This produces a deterministic (i.e., history independent) port selection across a given number of ports in an aggregation. However, as it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models, it might be appropriate to weight the port selection in favor of ports that are carrying lower traffic levels. In more sophisticated approaches, load balancing is dynamic; i.e., the port selected for a given set of conversations changes over time, independent of any changes that take place in the membership of the aggregation.

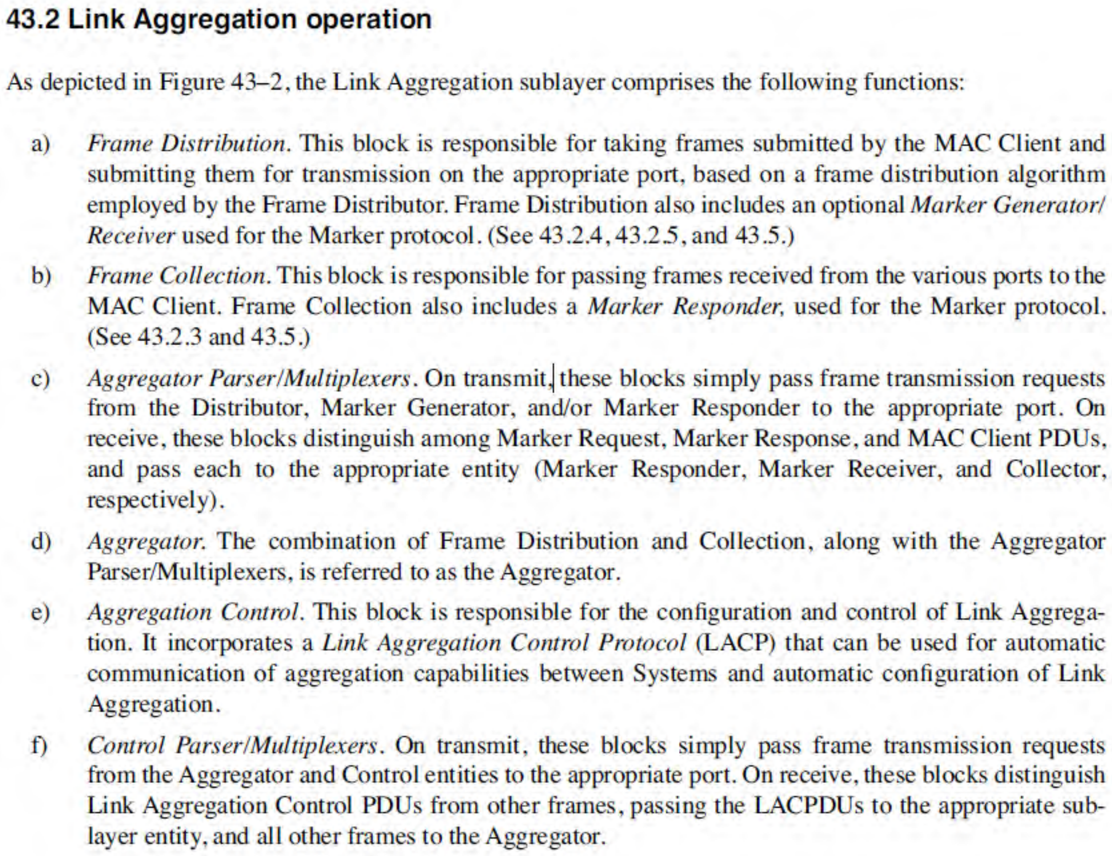
No.	’740 Patent Claim 8	IEEE 802.3
8	The method according to claim 1, wherein selecting the first and second physical links	IEEE 802.3 discloses the method according to claim 1, wherein selecting the first and second physical links comprises applying a mapping function to the at least one of the frame attributes.

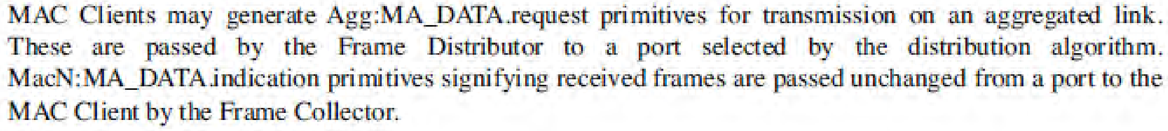
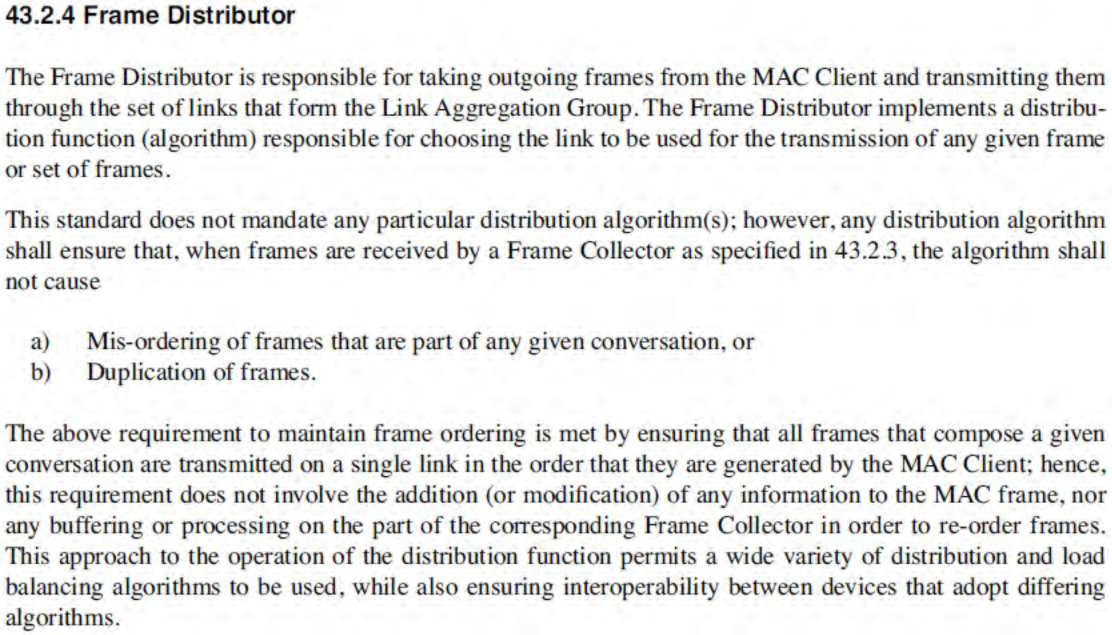
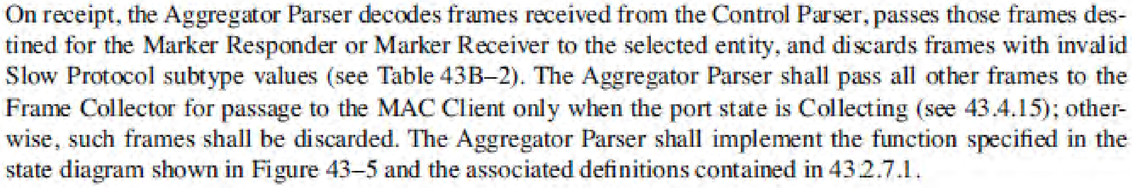
No.	’740 Patent Claim 8	IEEE 802.3
	comprises applying a mapping function to the at least one of the frame attributes.	For example, IEEE 802.3 discloses a distribution algorithm that maps the frames to the appropriate links based on frame information. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. *See supra* Claim 1 IEEE 802.3 at 1468  **43.2 Link Aggregation operation** As depicted in Figure 43–2, the Link Aggregation sublayer comprises the following functions: a) *Frame Distribution*. This block is responsible for taking frames submitted by the MAC Client and submitting them for transmission on the appropriate port, based on a frame distribution algorithm employed by the Frame Distributor. Frame Distribution also includes an optional *Marker Generator/Receiver* used for the Marker protocol. (See 43.2.4, 43.2.5, and 43.5.) b) *Frame Collection*. This block is responsible for passing frames received from the various ports to the MAC Client. Frame Collection also includes a *Marker Responder*, used for the Marker protocol. (See 43.2.3 and 43.5.) c) *Aggregator Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Distributor, Marker Generator, and/or Marker Responder to the appropriate port. On receive, these blocks distinguish among Marker Request, Marker Response, and MAC Client PDUs, and pass each to the appropriate entity (Marker Responder, Marker Receiver, and Collector, respectively). d) *Aggregator*. The combination of Frame Distribution and Collection, along with the Aggregator Parser/Multiplexers, is referred to as the Aggregator. e) *Aggregation Control*. This block is responsible for the configuration and control of Link Aggregation. It incorporates a *Link Aggregation Control Protocol* (LACP) that can be used for automatic communication of aggregation capabilities between Systems and automatic configuration of Link Aggregation. f) *Control Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Aggregator and Control entities to the appropriate port. On receive, these blocks distinguish Link Aggregation Control PDUs from other frames, passing the LACPDUs to the appropriate sublayer entity, and all other frames to the Aggregator. IEEE 802.3 at 1469

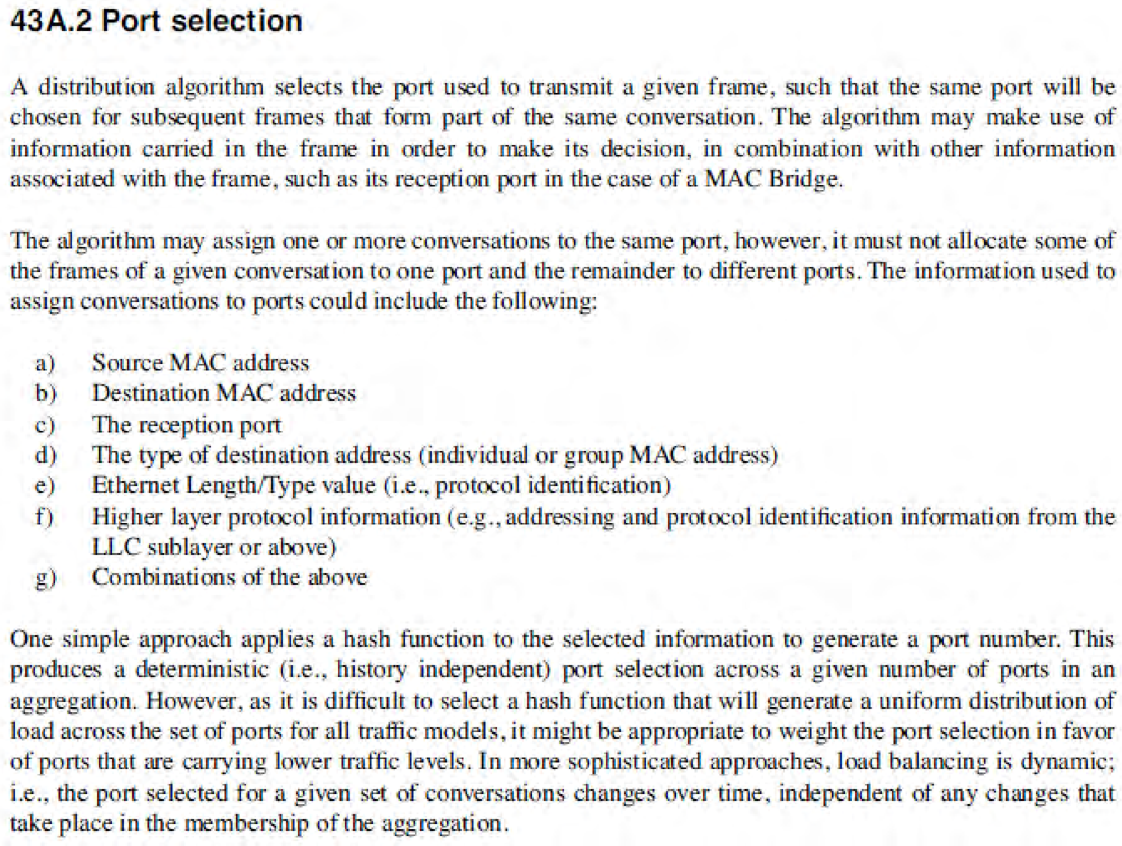
No.	’740 Patent Claim 8	IEEE 802.3
		 MAC Clients may generate Agg:MA_DATA.request primitives for transmission on an aggregated link. These are passed by the Frame Distributor to a port selected by the distribution algorithm. MacN:MA_DATA.indication primitives signifying received frames are passed unchanged from a port to the MAC Client by the Frame Collector. IEEE 802.3 at 1471  **43.2.4 Frame Distributor** The Frame Distributor is responsible for taking outgoing frames from the MAC Client and transmitting them through the set of links that form the Link Aggregation Group. The Frame Distributor implements a distribution function (algorithm) responsible for choosing the link to be used for the transmission of any given frame or set of frames. This standard does not mandate any particular distribution algorithm(s); however, any distribution algorithm shall ensure that, when frames are received by a Frame Collector as specified in 43.2.3, the algorithm shall not cause a) Mis-ordering of frames that are part of any given conversation, or b) Duplication of frames. The above requirement to maintain frame ordering is met by ensuring that all frames that compose a given conversation are transmitted on a single link in the order that they are generated by the MAC Client; hence, this requirement does not involve the addition (or modification) of any information to the MAC frame, nor any buffering or processing on the part of the corresponding Frame Collector in order to re-order frames. This approach to the operation of the distribution function permits a wide variety of distribution and load balancing algorithms to be used, while also ensuring interoperability between devices that adopt differing algorithms. IEEE 802.3 at 1473  On receipt, the Aggregator Parser decodes frames received from the Control Parser, passes those frames destined for the Marker Responder or Marker Receiver to the selected entity, and discards frames with invalid Slow Protocol subtype values (see Table 43B–2). The Aggregator Parser shall pass all other frames to the Frame Collector for passage to the MAC Client only when the port state is Collecting (see 43.4.15); otherwise, such frames shall be discarded. The Aggregator Parser shall implement the function specified in the state diagram shown in Figure 43–5 and the associated definitions contained in 43.2.7.1.

No.	’740 Patent Claim 8	IEEE 802.3
		IEEE 802.3 at 1549  **43A.2 Port selection** A distribution algorithm selects the port used to transmit a given frame, such that the same port will be chosen for subsequent frames that form part of the same conversation. The algorithm may make use of information carried in the frame in order to make its decision, in combination with other information associated with the frame, such as its reception port in the case of a MAC Bridge. The algorithm may assign one or more conversations to the same port, however, it must not allocate some of the frames of a given conversation to one port and the remainder to different ports. The information used to assign conversations to ports could include the following: a) Source MAC address b) Destination MAC address c) The reception port d) The type of destination address (individual or group MAC address) e) Ethernet Length/Type value (i.e., protocol identification) f) Higher layer protocol information (e.g., addressing and protocol identification information from the LLC sublayer or above) g) Combinations of the above One simple approach applies a hash function to the selected information to generate a port number. This produces a deterministic (i.e., history independent) port selection across a given number of ports in an aggregation. However, as it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models, it might be appropriate to weight the port selection in favor of ports that are carrying lower traffic levels. In more sophisticated approaches, load balancing is dynamic; i.e., the port selected for a given set of conversations changes over time, independent of any changes that take place in the membership of the aggregation.

No.	’740 Patent Claim 9	IEEE 802.3
9	The method according to claim 8, wherein applying the mapping function comprises	IEEE 802.3 discloses the method according to claim 8, wherein applying the mapping function comprises applying a hashing function.

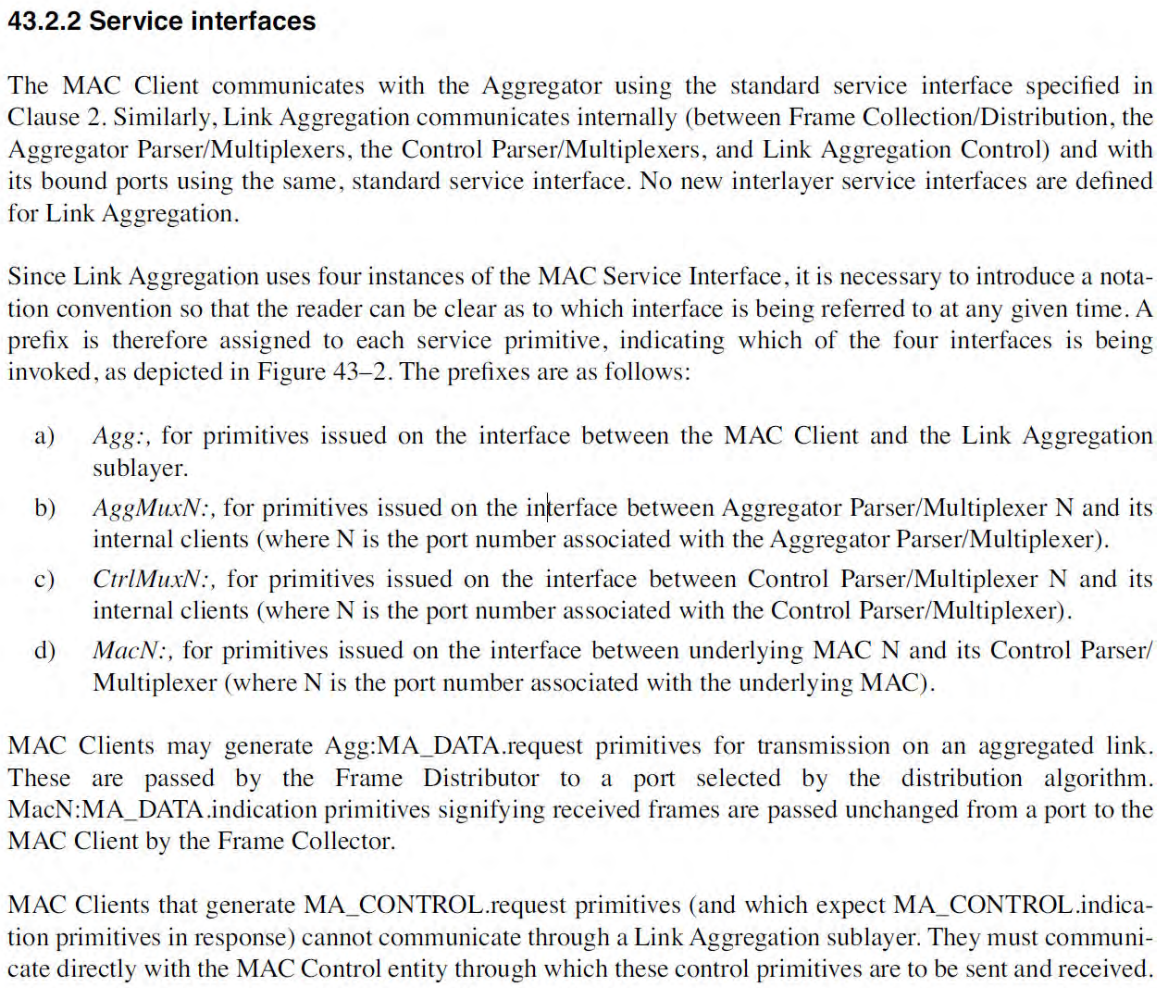
No.	’740 Patent Claim 9	IEEE 802.3
	applying a hashing function.	For example, IEEE 802.3 discloses a distribution (or mapping) algorithm that includes using a hash function on the frame information. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. *See supra* Claim 8 IEEE 802.3 at 1468  **43.2 Link Aggregation operation** As depicted in Figure 43–2, the Link Aggregation sublayer comprises the following functions: a) *Frame Distribution*. This block is responsible for taking frames submitted by the MAC Client and submitting them for transmission on the appropriate port, based on a frame distribution algorithm employed by the Frame Distributor. Frame Distribution also includes an optional *Marker Generator/Receiver* used for the Marker protocol. (See 43.2.4, 43.2.5, and 43.5.) b) *Frame Collection*. This block is responsible for passing frames received from the various ports to the MAC Client. Frame Collection also includes a *Marker Responder*, used for the Marker protocol. (See 43.2.3 and 43.5.) c) *Aggregator Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Distributor, Marker Generator, and/or Marker Responder to the appropriate port. On receive, these blocks distinguish among Marker Request, Marker Response, and MAC Client PDUs, and pass each to the appropriate entity (Marker Responder, Marker Receiver, and Collector, respectively). d) *Aggregator*. The combination of Frame Distribution and Collection, along with the Aggregator Parser/Multiplexers, is referred to as the Aggregator. e) *Aggregation Control*. This block is responsible for the configuration and control of Link Aggregation. It incorporates a *Link Aggregation Control Protocol* (LACP) that can be used for automatic communication of aggregation capabilities between Systems and automatic configuration of Link Aggregation. f) *Control Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Aggregator and Control entities to the appropriate port. On receive, these blocks distinguish Link Aggregation Control PDUs from other frames, passing the LACPDUs to the appropriate sublayer entity, and all other frames to the Aggregator. IEEE 802.3 at 1469

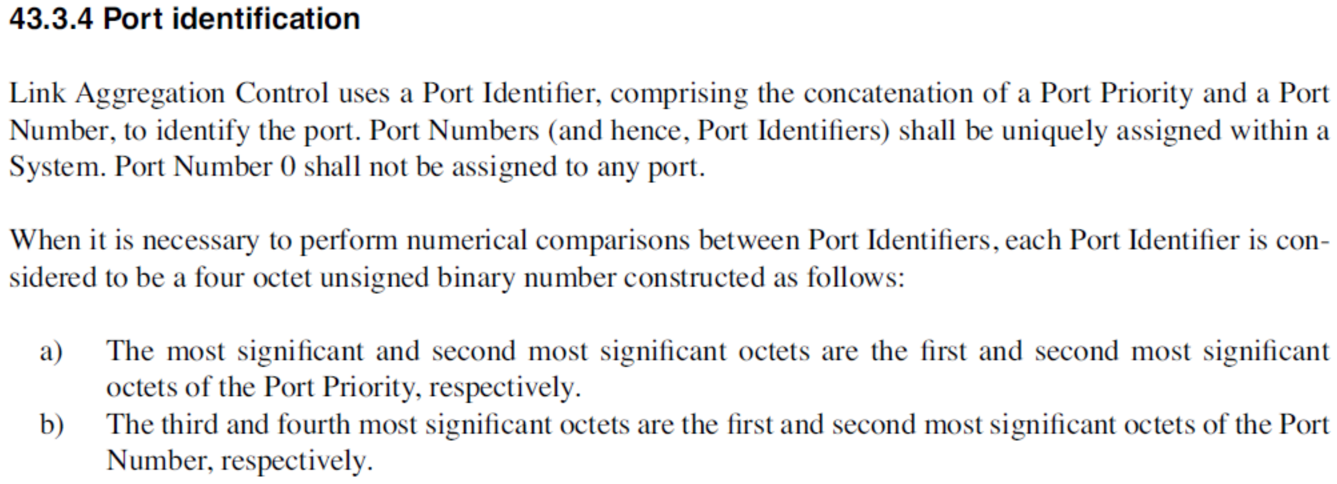
No.	’740 Patent Claim 9	IEEE 802.3
		 MAC Clients may generate Agg:MA_DATA.request primitives for transmission on an aggregated link. These are passed by the Frame Distributor to a port selected by the distribution algorithm. MacN:MA_DATA.indication primitives signifying received frames are passed unchanged from a port to the MAC Client by the Frame Collector. IEEE 802.3 at 1471  **43.2.4 Frame Distributor** The Frame Distributor is responsible for taking outgoing frames from the MAC Client and transmitting them through the set of links that form the Link Aggregation Group. The Frame Distributor implements a distribution function (algorithm) responsible for choosing the link to be used for the transmission of any given frame or set of frames. This standard does not mandate any particular distribution algorithm(s); however, any distribution algorithm shall ensure that, when frames are received by a Frame Collector as specified in 43.2.3, the algorithm shall not cause a) Mis-ordering of frames that are part of any given conversation, or b) Duplication of frames. The above requirement to maintain frame ordering is met by ensuring that all frames that compose a given conversation are transmitted on a single link in the order that they are generated by the MAC Client; hence, this requirement does not involve the addition (or modification) of any information to the MAC frame, nor any buffering or processing on the part of the corresponding Frame Collector in order to re-order frames. This approach to the operation of the distribution function permits a wide variety of distribution and load balancing algorithms to be used, while also ensuring interoperability between devices that adopt differing algorithms. IEEE 802.3 at 1473  On receipt, the Aggregator Parser decodes frames received from the Control Parser, passes those frames destined for the Marker Responder or Marker Receiver to the selected entity, and discards frames with invalid Slow Protocol subtype values (see Table 43B–2). The Aggregator Parser shall pass all other frames to the Frame Collector for passage to the MAC Client only when the port state is Collecting (see 43.4.15); otherwise, such frames shall be discarded. The Aggregator Parser shall implement the function specified in the state diagram shown in Figure 43–5 and the associated definitions contained in 43.2.7.1.

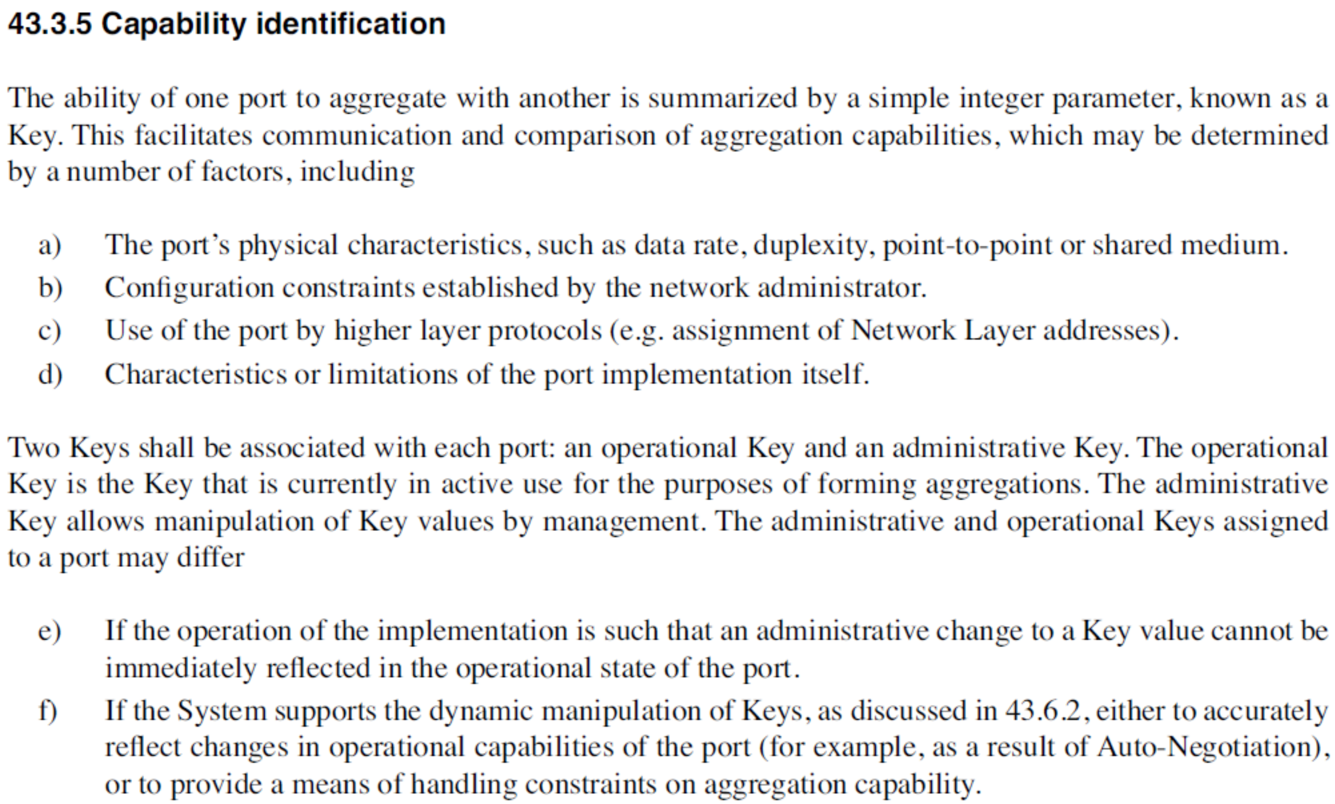
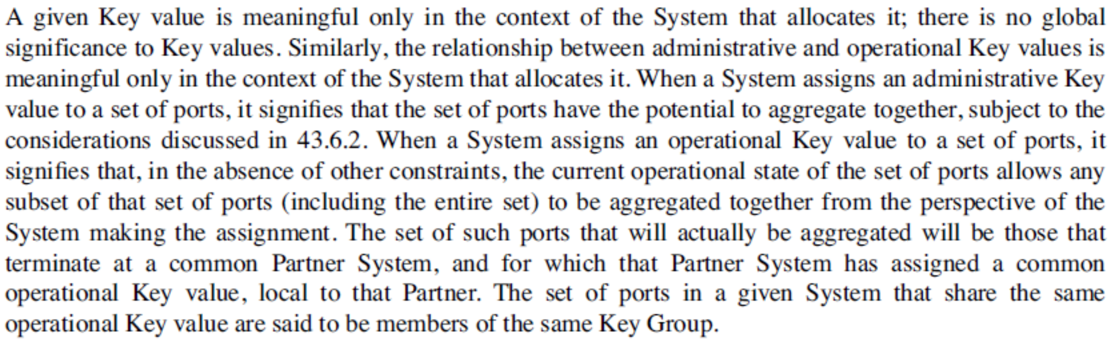
No.	’740 Patent Claim 9	IEEE 802.3
		IEEE 802.3 at 1549  **43A.2 Port selection** A distribution algorithm selects the port used to transmit a given frame, such that the same port will be chosen for subsequent frames that form part of the same conversation. The algorithm may make use of information carried in the frame in order to make its decision, in combination with other information associated with the frame, such as its reception port in the case of a MAC Bridge. The algorithm may assign one or more conversations to the same port, however, it must not allocate some of the frames of a given conversation to one port and the remainder to different ports. The information used to assign conversations to ports could include the following: a) Source MAC address b) Destination MAC address c) The reception port d) The type of destination address (individual or group MAC address) e) Ethernet Length/Type value (i.e., protocol identification) f) Higher layer protocol information (e.g., addressing and protocol identification information from the LLC sublayer or above) g) Combinations of the above One simple approach applies a hash function to the selected information to generate a port number. This produces a deterministic (i.e., history independent) port selection across a given number of ports in an aggregation. However, as it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models, it might be appropriate to weight the port selection in favor of ports that are carrying lower traffic levels. In more sophisticated approaches, load balancing is dynamic; i.e., the port selected for a given set of conversations changes over time, independent of any changes that take place in the membership of the aggregation.

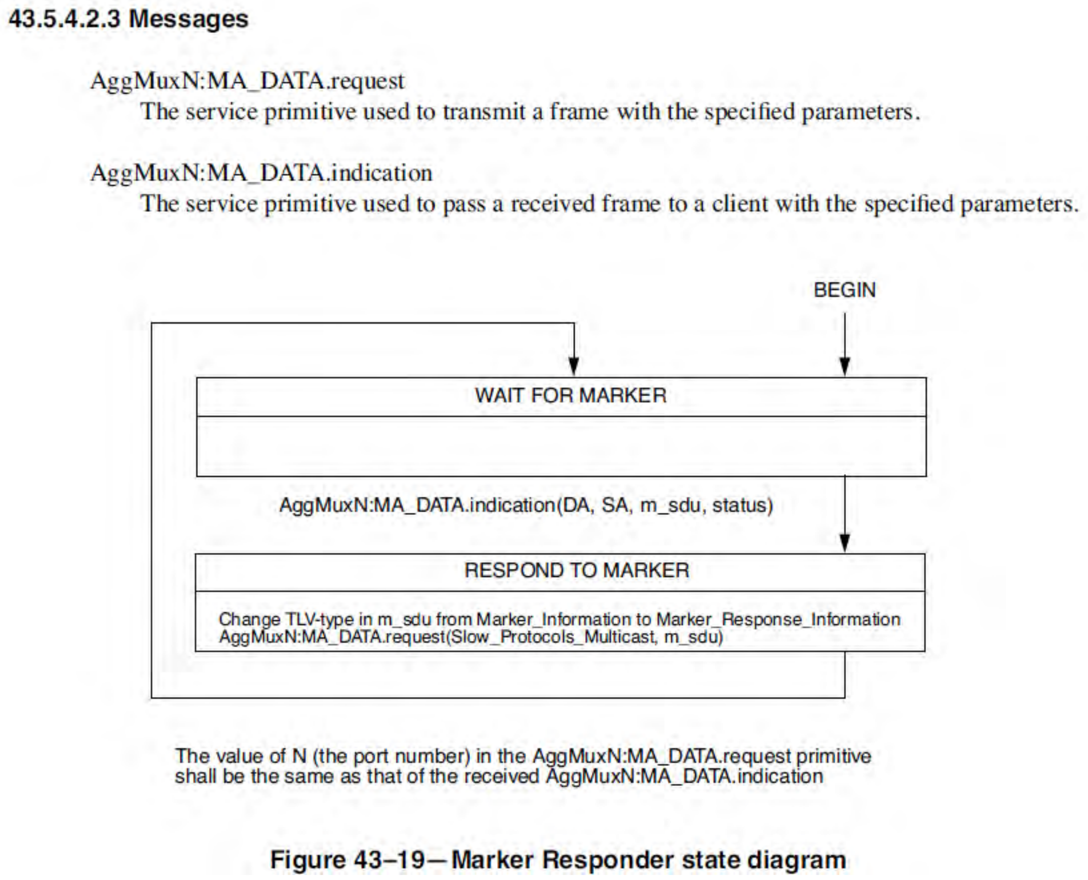
No.	’740 Patent Claim 10	IEEE 802.3
10[a]	The method according to	IEEE 802.3 discloses the method according to claim 9, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second physical links.

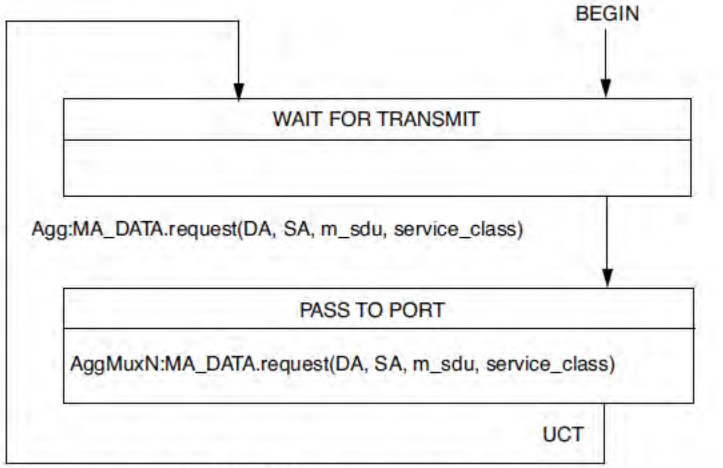
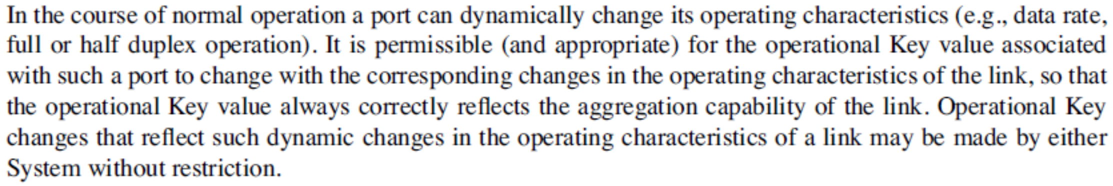
No.	’740 Patent Claim 10	IEEE 802.3
	claim 9, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second physical links,	For example, IEEE 802.3 discloses a hash function including determining a port number N. A person of ordinary skill in the art would understand that a hashing function is a standard algorithm used to route data traffic that involves information specific to the system architecture and includes calculations of the generated values. A person of ordinary skill in the art would further understand that the physical links over which the frame is sent are determined based on the calculations of the hashing function. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the IEEE 802.3 is found to not meet this limitation, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second physical links would have been obvious to a person having ordinary skill in the art, as explained below. *See supra* Claim 9. IEEE 802.3 at 1469

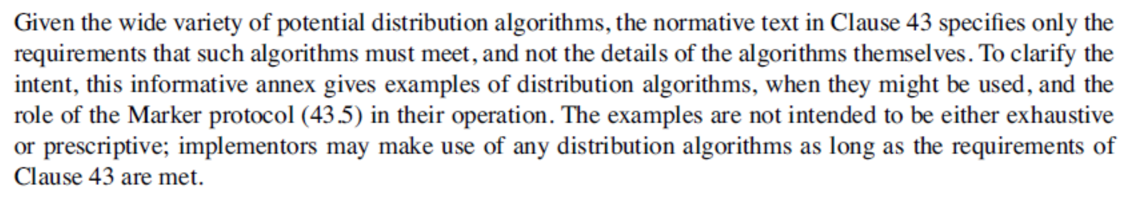
No.	’740 Patent Claim 10	IEEE 802.3
		 **43.2.2 Service interfaces** The MAC Client communicates with the Aggregator using the standard service interface specified in Clause 2. Similarly, Link Aggregation communicates internally (between Frame Collection/Distribution, the Aggregator Parser/Multiplexers, the Control Parser/Multiplexers, and Link Aggregation Control) and with its bound ports using the same, standard service interface. No new interlayer service interfaces are defined for Link Aggregation. Since Link Aggregation uses four instances of the MAC Service Interface, it is necessary to introduce a notation convention so that the reader can be clear as to which interface is being referred to at any given time. A prefix is therefore assigned to each service primitive, indicating which of the four interfaces is being invoked, as depicted in Figure 43–2. The prefixes are as follows: a) *Agg:*, for primitives issued on the interface between the MAC Client and the Link Aggregation sublayer. b) *AggMuxN:*, for primitives issued on the interface between Aggregator Parser/Multiplexer N and its internal clients (where N is the port number associated with the Aggregator Parser/Multiplexer). c) *CtrlMuxN:*, for primitives issued on the interface between Control Parser/Multiplexer N and its internal clients (where N is the port number associated with the Control Parser/Multiplexer). d) *MacN:*, for primitives issued on the interface between underlying MAC N and its Control Parser/Multiplexer (where N is the port number associated with the underlying MAC). MAC Clients may generate Agg:MA_DATA.request primitives for transmission on an aggregated link. These are passed by the Frame Distributor to a port selected by the distribution algorithm. MacN:MA_DATA.indication primitives signifying received frames are passed unchanged from a port to the MAC Client by the Frame Collector. MAC Clients that generate MA_CONTROL.request primitives (and which expect MA_CONTROL.indication primitives in response) cannot communicate through a Link Aggregation sublayer. They must communicate directly with the MAC Control entity through which these control primitives are to be sent and received. IEEE 802.3 at 1480

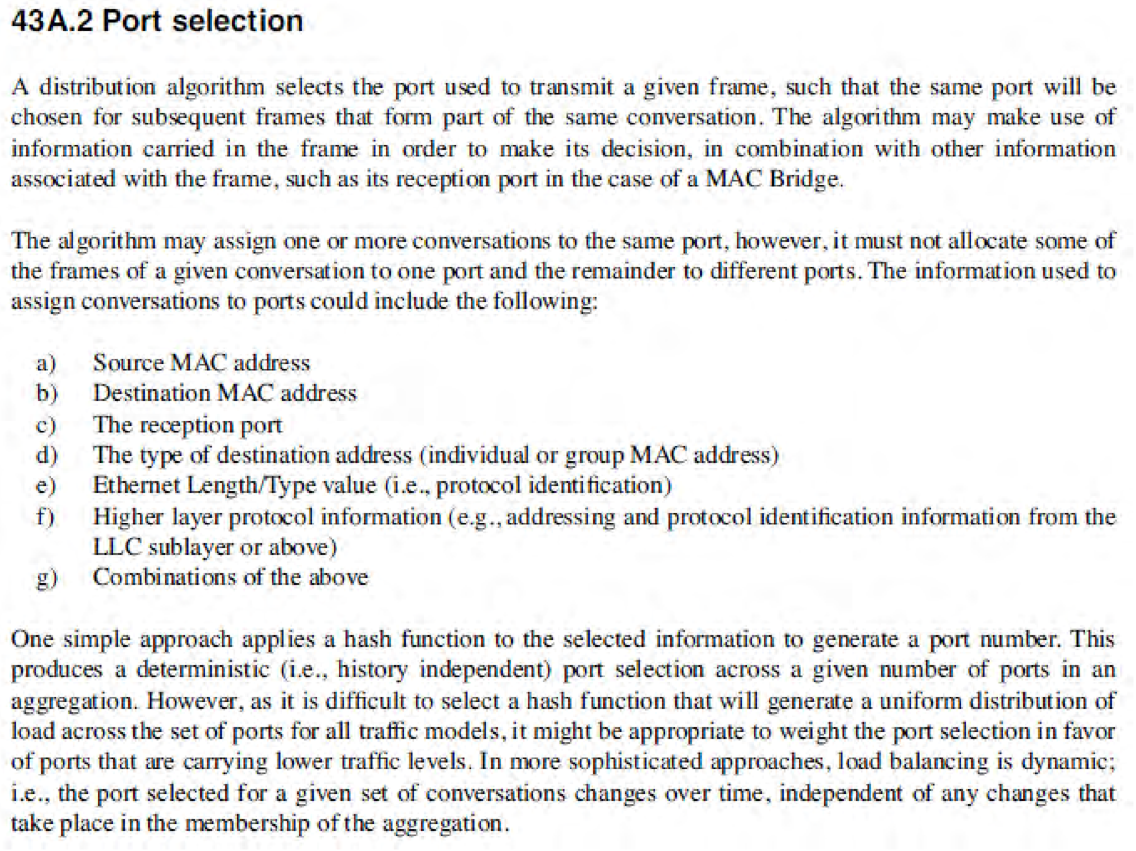
No.	’740 Patent Claim 10	IEEE 802.3
		 **43.3.4 Port identification** Link Aggregation Control uses a Port Identifier, comprising the concatenation of a Port Priority and a Port Number, to identify the port. Port Numbers (and hence, Port Identifiers) shall be uniquely assigned within a System. Port Number 0 shall not be assigned to any port. When it is necessary to perform numerical comparisons between Port Identifiers, each Port Identifier is considered to be a four octet unsigned binary number constructed as follows: a) The most significant and second most significant octets are the first and second most significant octets of the Port Priority, respectively. b) The third and fourth most significant octets are the first and second most significant octets of the Port Number, respectively. IEEE 802.3 at 1480

No.	’740 Patent Claim 10	IEEE 802.3
		 **43.3.5 Capability identification** The ability of one port to aggregate with another is summarized by a simple integer parameter, known as a Key. This facilitates communication and comparison of aggregation capabilities, which may be determined by a number of factors, including a) The port’s physical characteristics, such as data rate, duplexity, point-to-point or shared medium. b) Configuration constraints established by the network administrator. c) Use of the port by higher layer protocols (e.g. assignment of Network Layer addresses). d) Characteristics or limitations of the port implementation itself. Two Keys shall be associated with each port: an operational Key and an administrative Key. The operational Key is the Key that is currently in active use for the purposes of forming aggregations. The administrative Key allows manipulation of Key values by management. The administrative and operational Keys assigned to a port may differ e) If the operation of the implementation is such that an administrative change to a Key value cannot be immediately reflected in the operational state of the port. f) If the System supports the dynamic manipulation of Keys, as discussed in 43.6.2, either to accurately reflect changes in operational capabilities of the port (for example, as a result of Auto-Negotiation), or to provide a means of handling constraints on aggregation capability.  A given Key value is meaningful only in the context of the System that allocates it; there is no global significance to Key values. Similarly, the relationship between administrative and operational Key values is meaningful only in the context of the System that allocates it. When a System assigns an administrative Key value to a set of ports, it signifies that the set of ports have the potential to aggregate together, subject to the considerations discussed in 43.6.2. When a System assigns an operational Key value to a set of ports, it signifies that, in the absence of other constraints, the current operational state of the set of ports allows any subset of that set of ports (including the entire set) to be aggregated together from the perspective of the System making the assignment. The set of such ports that will actually be aggregated will be those that terminate at a common Partner System, and for which that Partner System has assigned a common operational Key value, local to that Partner. The set of ports in a given System that share the same operational Key value are said to be members of the same Key Group.

No.	’740 Patent Claim 10	IEEE 802.3
		IEEE 802.3 at 1516  **43.5.4.2.3 Messages** AggMuxN:MA_DATA.request The service primitive used to transmit a frame with the specified parameters. AggMuxN:MA_DATA.indication The service primitive used to pass a received frame to a client with the specified parameters. BEGIN WAIT FOR MARKER AggMuxN:MA_DATA.indication(DA, SA, m_sdu, status) RESPOND TO MARKER Change TLV-type in m_sdu from Marker_Information to Marker_Response_Information AggMuxN:MA_DATA.request(Slow_Protocols_Multicast, m_sdu) The value of N (the port number) in the AggMuxN:MA_DATA.request primitive shall be the same as that of the received AggMuxN:MA_DATA.indication **Figure 43–19—Marker Responder state diagram** IEEE 802.3 at 1472

No.	’740 Patent Claim 10	IEEE 802.3
		43.2.4.1.3 State diagram  BEGIN WAIT FOR TRANSMIT Agg:MA_DATA.request(DA, SA, m_sdu, service_class) PASS TO PORT AggMuxN:MA_DATA.request(DA, SA, m_sdu, service_class) UCT If a client issues an Agg:MA_DATA.request primitive that contains no SA parameter, the AggMuxN:MA_DATA.request primitive generated shall use the Aggregator’s MAC address for the SA. NOTE—The algorithm that the Frame Distributor uses to select the value of N in AggMuxN:MA_DATA.request for a given frame is unspecified. **Figure 43–4—Frame Distributor state diagram** IEEE at 802.3 at 1518  In the course of normal operation a port can dynamically change its operating characteristics (e.g., data rate, full or half duplex operation). It is permissible (and appropriate) for the operational Key value associated with such a port to change with the corresponding changes in the operating characteristics of the link, so that the operational Key value always correctly reflects the aggregation capability of the link. Operational Key changes that reflect such dynamic changes in the operating characteristics of a link may be made by either System without restriction.

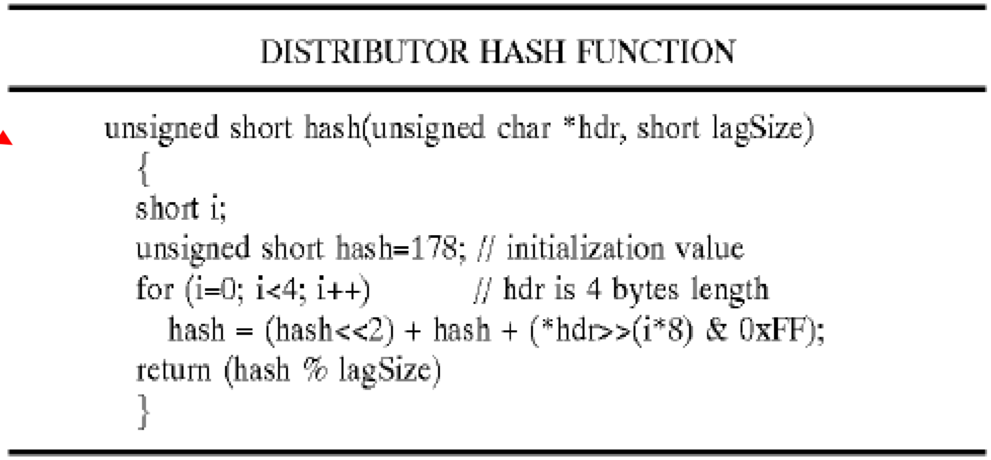
No.	’740 Patent Claim 10	IEEE 802.3
		IEEE 802.3 at 1548  Given the wide variety of potential distribution algorithms, the normative text in Clause 43 specifies only the requirements that such algorithms must meet, and not the details of the algorithms themselves. To clarify the intent, this informative annex gives examples of distribution algorithms, when they might be used, and the role of the Marker protocol (43.5) in their operation. The examples are not intended to be either exhaustive or prescriptive; implementors may make use of any distribution algorithms as long as the requirements of Clause 43 are met. IEEE 802.3 at 1549

No.	’740 Patent Claim 10	IEEE 802.3
		 **43A.2 Port selection** A distribution algorithm selects the port used to transmit a given frame, such that the same port will be chosen for subsequent frames that form part of the same conversation. The algorithm may make use of information carried in the frame in order to make its decision, in combination with other information associated with the frame, such as its reception port in the case of a MAC Bridge. The algorithm may assign one or more conversations to the same port, however, it must not allocate some of the frames of a given conversation to one port and the remainder to different ports. The information used to assign conversations to ports could include the following: a) Source MAC address b) Destination MAC address c) The reception port d) The type of destination address (individual or group MAC address) e) Ethernet Length/Type value (i.e., protocol identification) f) Higher layer protocol information (e.g., addressing and protocol identification information from the LLC sublayer or above) g) Combinations of the above One simple approach applies a hash function to the selected information to generate a port number. This produces a deterministic (i.e., history independent) port selection across a given number of ports in an aggregation. However, as it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models, it might be appropriate to weight the port selection in favor of ports that are carrying lower traffic levels. In more sophisticated approaches, load balancing is dynamic; i.e., the port selected for a given set of conversations changes over time, independent of any changes that take place in the membership of the aggregation. Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, IEEE 802.3 in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[a] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references.

No.	’740 Patent Claim 10	IEEE 802.3
		For example, Bruckman discloses applying a distributor hash function to the frame information which includes determining a number of the plurality of physical links. Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a

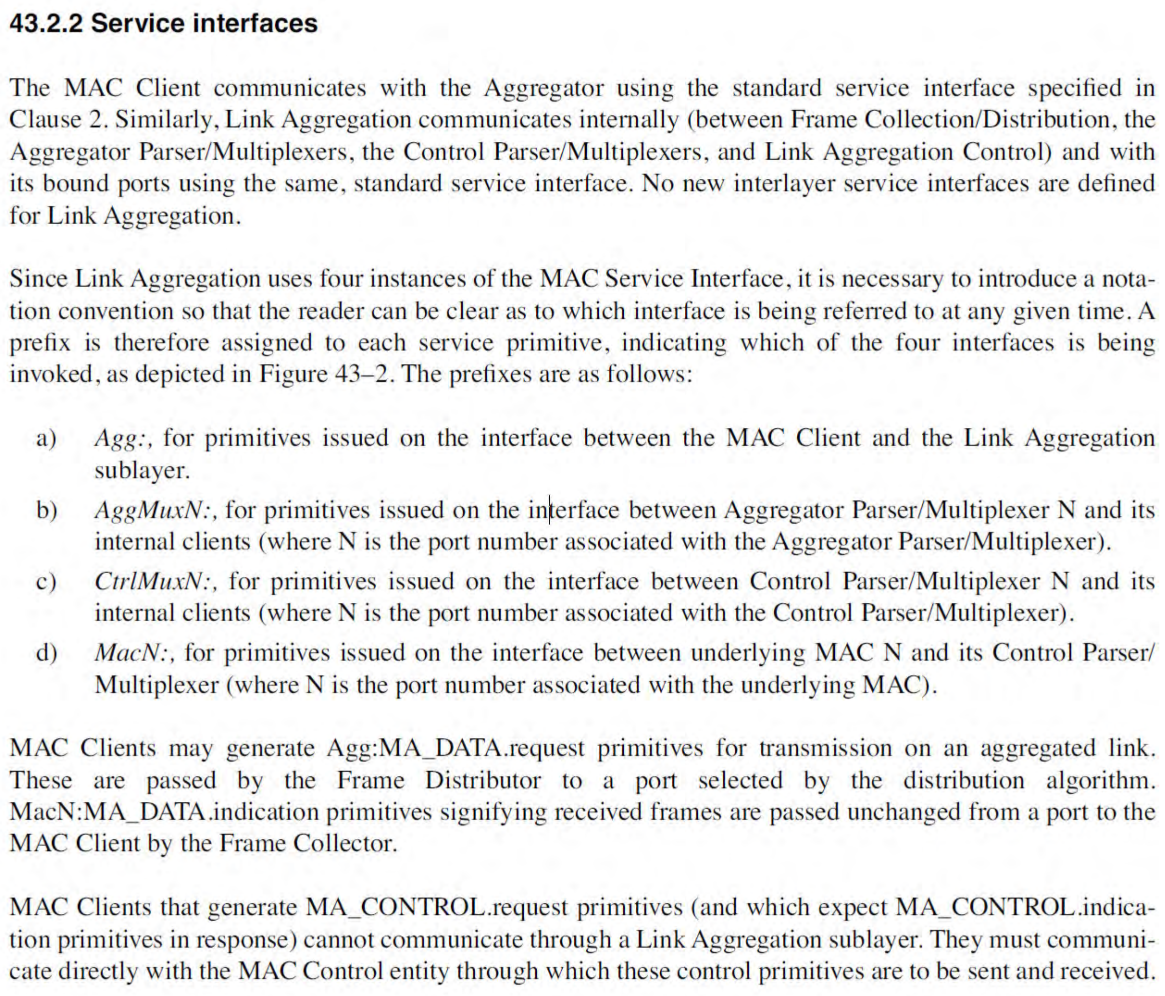
No.	’740 Patent Claim 10	IEEE 802.3
		program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”) Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I: Bruckman at Table 1 (annotated)

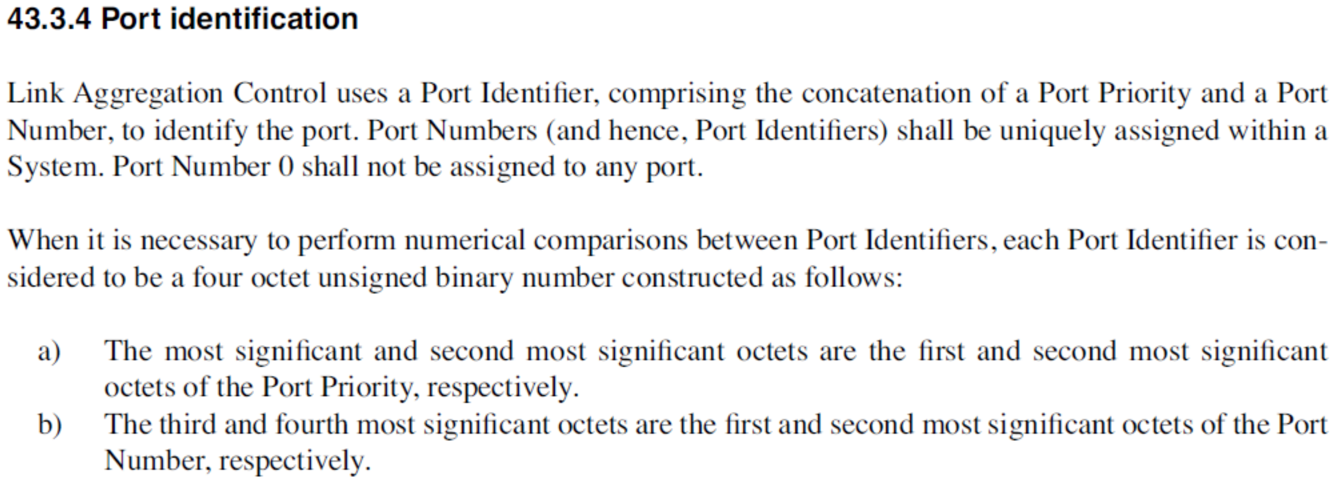
No.	’740 Patent Claim 10	IEEE 802.3

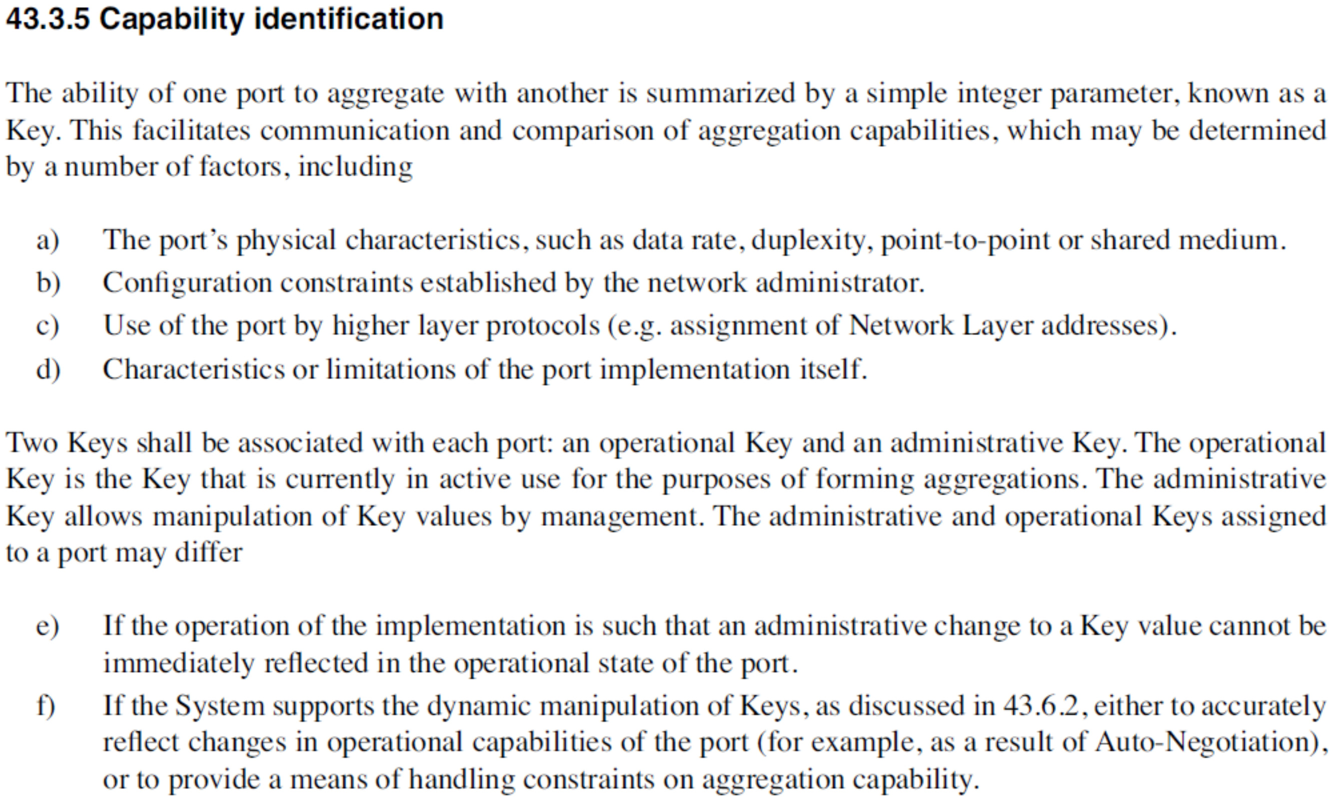
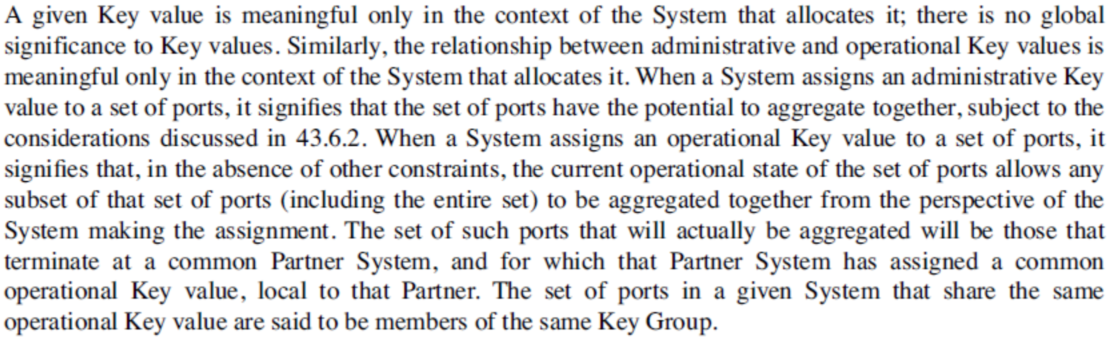
		hashing function *“mapping function”*
		
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		
		For example, Solomon discloses applying a distributor hash function to the frame information which includes determining a number of the plurality of physical links.
		
		Solomon at [0024] (“In another embodiment, switching the data packets includes mapping the data packets to the selected port responsively to the label. Additionally or alternatively, map-ping the data packets includes applying a hashing function to the label so as to determine a number of the selected port, and choosing the label includes applying an inverse of the hashing function to the number of the selected port.”)
		
		Solomon at [0048] (“The mapping function typically uses MPLS label 52 for mapping, since the MPLS label uniquely identifies MPLS tunnel 28, and it is required that all MPLS packets belonging to the same

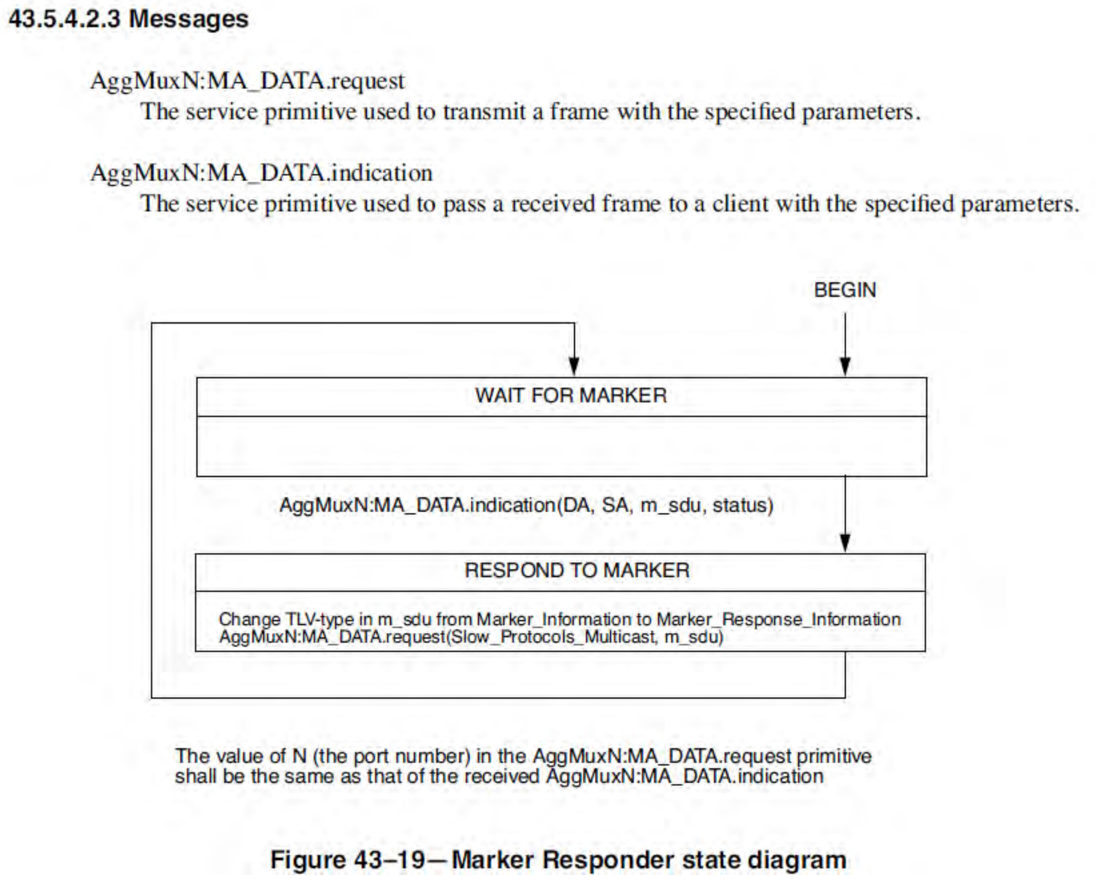
No.	’740 Patent Claim 10	IEEE 802.3
		tunnel be switched through the same physical port 24. Additionally or alternatively, the mapping function uses a "PW" label (pseudo wire label, formerly known as a virtual connection, or VC label), which is optionally added to MPLS header 50. The PW label com-prises information that the egress node requires for deliver-ing the packet to its destination, and is optionally added during the encapsulation of MPLS packets. Additional details regarding the VC label can be found in an IETF draft by Martini et al. entitled "Encapsulation Methods for Trans-port of Ethernet Frames Over IP/MPLS Networks" (IETF draft-ietf-pwe3-ethernet-encap-07.txt, May, 2004), which is incorporated herein by reference. In some embodiments, mapper 34 applies a hashing function to the MPLS and/or PW label, as will be described below.”) Solomon at [0059] (“In this method, the mapping function used by mapper 34 of switch A is a hashing function. Various hashing functions are known in the art, and any suitable hashing function may be used in mapper 34. Since the hashing operation is performed for each packet, it is desirable to have a hashing function that is computationally simple.”) Solomon at [0060] (“As mentioned above, the hashing function typically hashes the value of MPLS label 52 to determine the selected physical port, as the MPLS label uniquely identifies tunnel 28. For example, the following hashing function may be used by mapper 34: Selected port number=l+((MPLS label) mod N), wherein N denotes the number of physical Ethernet ports in LAG group 25, and "mod" denotes the modulus operator. Assuming the values of MPLS labels are distrib-uted uniformly over a certain range, this function achieves a uniform distribution of port allocations for the different MPLS labels. It can also be seen that all packets carrying the same MPLS label (in other words-belonging to the same MPLS tunnel) will be mapped to the same physical port.”) Solomon at [0065] (“Mapper 34 of switch A maps each received packet to the selected physical port of LAG group 25 using the hashing function, at a hashing step 90. Mapper 34 extracts the MPLS label from each received packet and uses the hashing function to calculate the serial number of the selected physical port, which was selected by the CAC processor at step 82. Following the numerical example given above, the mapper extracts MPLS label=65647 from the packet. Substituting this value and N=3 into the hashing function gives: Selected port number=1+(65647 mod 3)=2, which is indeed the port number selected in the example above.”)

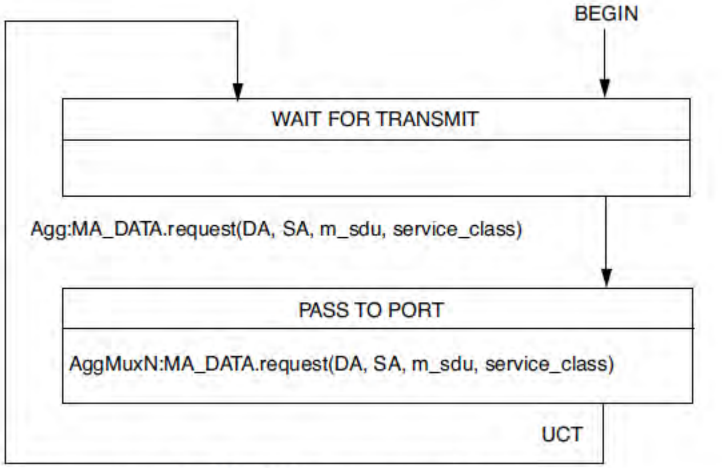
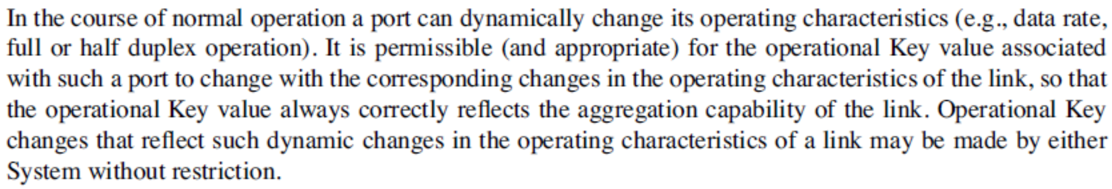
No.	’740 Patent Claim 10	IEEE 802.3
10[b]	applying the hashing function to the at least one of the frame attributes to produce a hashing key,	IEEE 802.3 discloses applying the hashing function to the at least one of the frame attributes to produce a hashing key. For example, IEEE 802.3 discloses producing a value from a hash function based on the selected frame information. A person of ordinary skill in the art would understand that a hashing function is a standard algorithm used to route data traffic that involves information specific to the system architecture and includes calculations of the generated values. A person of ordinary skill in the art would further understand that the physical links over which the frame is sent are determined based on the calculations of the hashing function. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the IEEE 802.3 is found to not meet this limitation, applying the hashing function to the at least one of the frame attributes to produce a hashing key would have been obvious to a person having ordinary skill in the art, as explained below. IEEE 802.3 at 1469

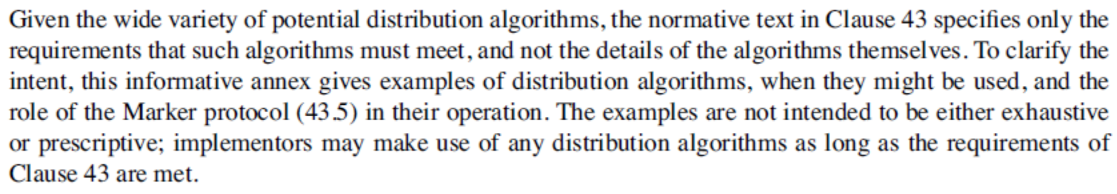
No.	’740 Patent Claim 10	IEEE 802.3
		 **43.2.2 Service interfaces** The MAC Client communicates with the Aggregator using the standard service interface specified in Clause 2. Similarly, Link Aggregation communicates internally (between Frame Collection/Distribution, the Aggregator Parser/Multiplexers, the Control Parser/Multiplexers, and Link Aggregation Control) and with its bound ports using the same, standard service interface. No new interlayer service interfaces are defined for Link Aggregation. Since Link Aggregation uses four instances of the MAC Service Interface, it is necessary to introduce a notation convention so that the reader can be clear as to which interface is being referred to at any given time. A prefix is therefore assigned to each service primitive, indicating which of the four interfaces is being invoked, as depicted in Figure 43–2. The prefixes are as follows: a) *Agg:*, for primitives issued on the interface between the MAC Client and the Link Aggregation sublayer. b) *AggMuxN:*, for primitives issued on the interface between Aggregator Parser/Multiplexer N and its internal clients (where N is the port number associated with the Aggregator Parser/Multiplexer). c) *CtrlMuxN:*, for primitives issued on the interface between Control Parser/Multiplexer N and its internal clients (where N is the port number associated with the Control Parser/Multiplexer). d) *MacN:*, for primitives issued on the interface between underlying MAC N and its Control Parser/Multiplexer (where N is the port number associated with the underlying MAC). MAC Clients may generate Agg:MA_DATA.request primitives for transmission on an aggregated link. These are passed by the Frame Distributor to a port selected by the distribution algorithm. MacN:MA_DATA.indication primitives signifying received frames are passed unchanged from a port to the MAC Client by the Frame Collector. MAC Clients that generate MA_CONTROL.request primitives (and which expect MA_CONTROL.indication primitives in response) cannot communicate through a Link Aggregation sublayer. They must communicate directly with the MAC Control entity through which these control primitives are to be sent and received. IEEE 802.3 at 1480

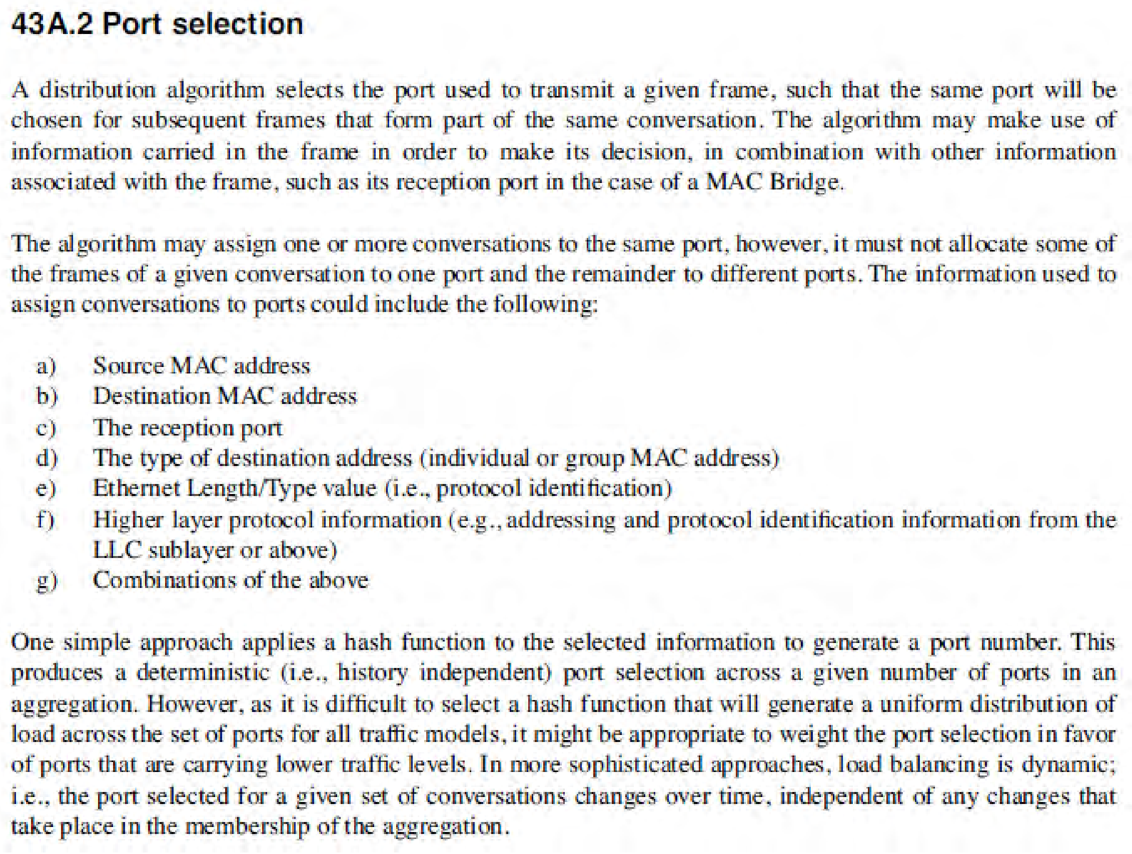
No.	’740 Patent Claim 10	IEEE 802.3
		 **43.3.4 Port identification** Link Aggregation Control uses a Port Identifier, comprising the concatenation of a Port Priority and a Port Number, to identify the port. Port Numbers (and hence, Port Identifiers) shall be uniquely assigned within a System. Port Number 0 shall not be assigned to any port. When it is necessary to perform numerical comparisons between Port Identifiers, each Port Identifier is considered to be a four octet unsigned binary number constructed as follows: a) The most significant and second most significant octets are the first and second most significant octets of the Port Priority, respectively. b) The third and fourth most significant octets are the first and second most significant octets of the Port Number, respectively. IEEE 802.3 at 1480

No.	’740 Patent Claim 10	IEEE 802.3
		 **43.3.5 Capability identification** The ability of one port to aggregate with another is summarized by a simple integer parameter, known as a Key. This facilitates communication and comparison of aggregation capabilities, which may be determined by a number of factors, including a) The port’s physical characteristics, such as data rate, duplexity, point-to-point or shared medium. b) Configuration constraints established by the network administrator. c) Use of the port by higher layer protocols (e.g. assignment of Network Layer addresses). d) Characteristics or limitations of the port implementation itself. Two Keys shall be associated with each port: an operational Key and an administrative Key. The operational Key is the Key that is currently in active use for the purposes of forming aggregations. The administrative Key allows manipulation of Key values by management. The administrative and operational Keys assigned to a port may differ e) If the operation of the implementation is such that an administrative change to a Key value cannot be immediately reflected in the operational state of the port. f) If the System supports the dynamic manipulation of Keys, as discussed in 43.6.2, either to accurately reflect changes in operational capabilities of the port (for example, as a result of Auto-Negotiation), or to provide a means of handling constraints on aggregation capability.  A given Key value is meaningful only in the context of the System that allocates it; there is no global significance to Key values. Similarly, the relationship between administrative and operational Key values is meaningful only in the context of the System that allocates it. When a System assigns an administrative Key value to a set of ports, it signifies that the set of ports have the potential to aggregate together, subject to the considerations discussed in 43.6.2. When a System assigns an operational Key value to a set of ports, it signifies that, in the absence of other constraints, the current operational state of the set of ports allows any subset of that set of ports (including the entire set) to be aggregated together from the perspective of the System making the assignment. The set of such ports that will actually be aggregated will be those that terminate at a common Partner System, and for which that Partner System has assigned a common operational Key value, local to that Partner. The set of ports in a given System that share the same operational Key value are said to be members of the same Key Group.

No.	’740 Patent Claim 10	IEEE 802.3
		IEEE 802.3 at 1516  **43.5.4.2.3 Messages** AggMuxN:MA_DATA.request The service primitive used to transmit a frame with the specified parameters. AggMuxN:MA_DATA.indication The service primitive used to pass a received frame to a client with the specified parameters. BEGIN WAIT FOR MARKER AggMuxN:MA_DATA.indication(DA, SA, m_sdu, status) RESPOND TO MARKER Change TLV-type in m_sdu from Marker_Information to Marker_Response_Information AggMuxN:MA_DATA.request(Slow_Protocols_Multicast, m_sdu) The value of N (the port number) in the AggMuxN:MA_DATA.request primitive shall be the same as that of the received AggMuxN:MA_DATA.indication **Figure 43–19—Marker Responder state diagram** IEEE 802.3 at 1472

No.	’740 Patent Claim 10	IEEE 802.3
		43.2.4.1.3 State diagram  BEGIN WAIT FOR TRANSMIT Agg:MA_DATA.request(DA, SA, m_sdu, service_class) PASS TO PORT AggMuxN:MA_DATA.request(DA, SA, m_sdu, service_class) UCT If a client issues an Agg:MA_DATA.request primitive that contains no SA parameter, the AggMuxN:MA_DATA.request primitive generated shall use the Aggregator’s MAC address for the SA. NOTE—The algorithm that the Frame Distributor uses to select the value of N in AggMuxN:MA_DATA.request for a given frame is unspecified. **Figure 43–4—Frame Distributor state diagram** IEEE at 802.3 at 1518  In the course of normal operation a port can dynamically change its operating characteristics (e.g., data rate, full or half duplex operation). It is permissible (and appropriate) for the operational Key value associated with such a port to change with the corresponding changes in the operating characteristics of the link, so that the operational Key value always correctly reflects the aggregation capability of the link. Operational Key changes that reflect such dynamic changes in the operating characteristics of a link may be made by either System without restriction.

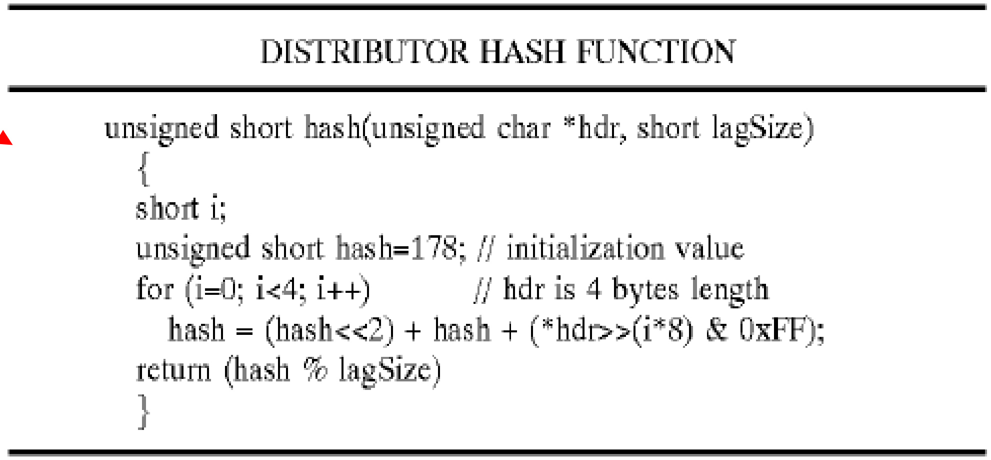
No.	’740 Patent Claim 10	IEEE 802.3
		IEEE 802.3 at 1548  Given the wide variety of potential distribution algorithms, the normative text in Clause 43 specifies only the requirements that such algorithms must meet, and not the details of the algorithms themselves. To clarify the intent, this informative annex gives examples of distribution algorithms, when they might be used, and the role of the Marker protocol (43.5) in their operation. The examples are not intended to be either exhaustive or prescriptive; implementors may make use of any distribution algorithms as long as the requirements of Clause 43 are met. IEEE 802.3 at 1549

No.	’740 Patent Claim 10	IEEE 802.3
		 43A.2 Port selection A distribution algorithm selects the port used to transmit a given frame, such that the same port will be chosen for subsequent frames that form part of the same conversation. The algorithm may make use of information carried in the frame in order to make its decision, in combination with other information associated with the frame, such as its reception port in the case of a MAC Bridge. The algorithm may assign one or more conversations to the same port, however, it must not allocate some of the frames of a given conversation to one port and the remainder to different ports. The information used to assign conversations to ports could include the following: a) Source MAC address b) Destination MAC address c) The reception port d) The type of destination address (individual or group MAC address) e) Ethernet Length/Type value (i.e., protocol identification) f) Higher layer protocol information (e.g., addressing and protocol identification information from the LLC sublayer or above) g) Combinations of the above One simple approach applies a hash function to the selected information to generate a port number. This produces a deterministic (i.e., history independent) port selection across a given number of ports in an aggregation. However, as it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models, it might be appropriate to weight the port selection in favor of ports that are carrying lower traffic levels. In more sophisticated approaches, load balancing is dynamic; i.e., the port selected for a given set of conversations changes over time, independent of any changes that take place in the membership of the aggregation. Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, IEEE 802.3 in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[b] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references.

No.	’740 Patent Claim 10	IEEE 802.3
		For example, Bruckman discloses applying a distributor hash function to the frame information which includes determining a number of the plurality of physical links. Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a

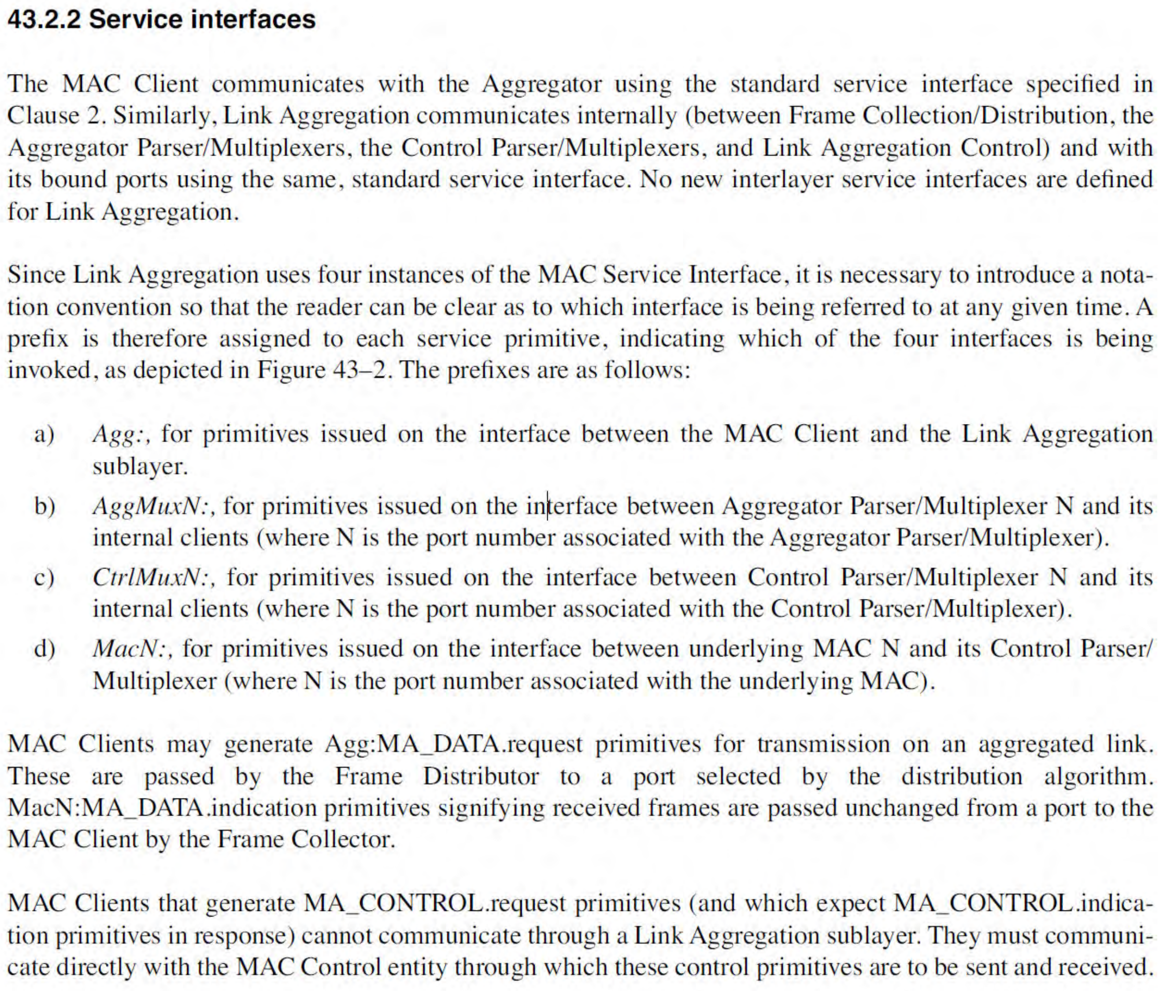
No.	’740 Patent Claim 10	IEEE 802.3
		program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”) Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I: Bruckman at Table 1 (annotated)

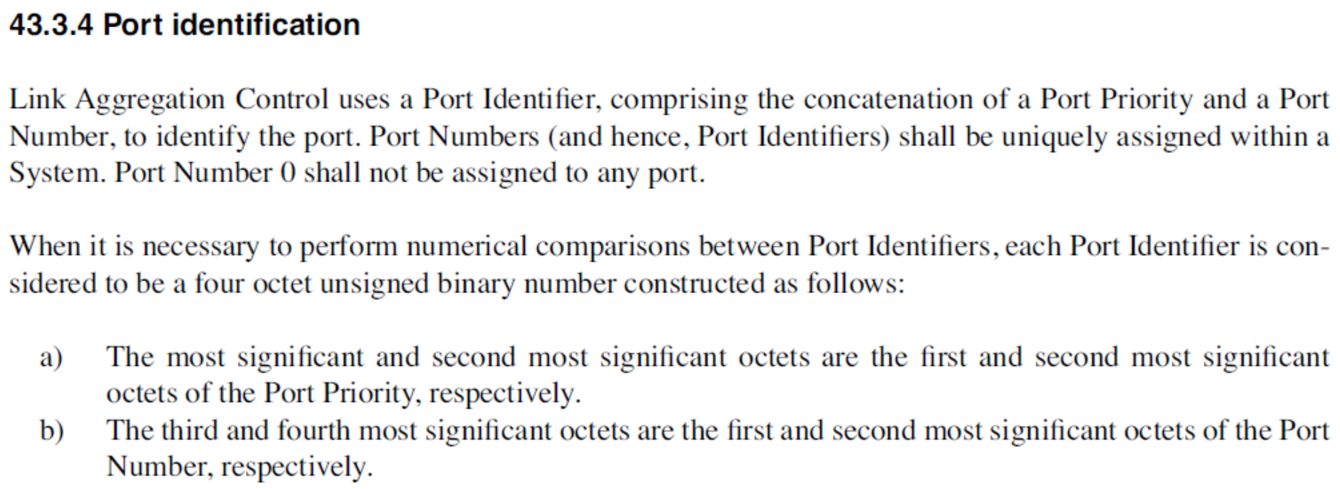
No.	’740 Patent Claim 10	IEEE 802.3

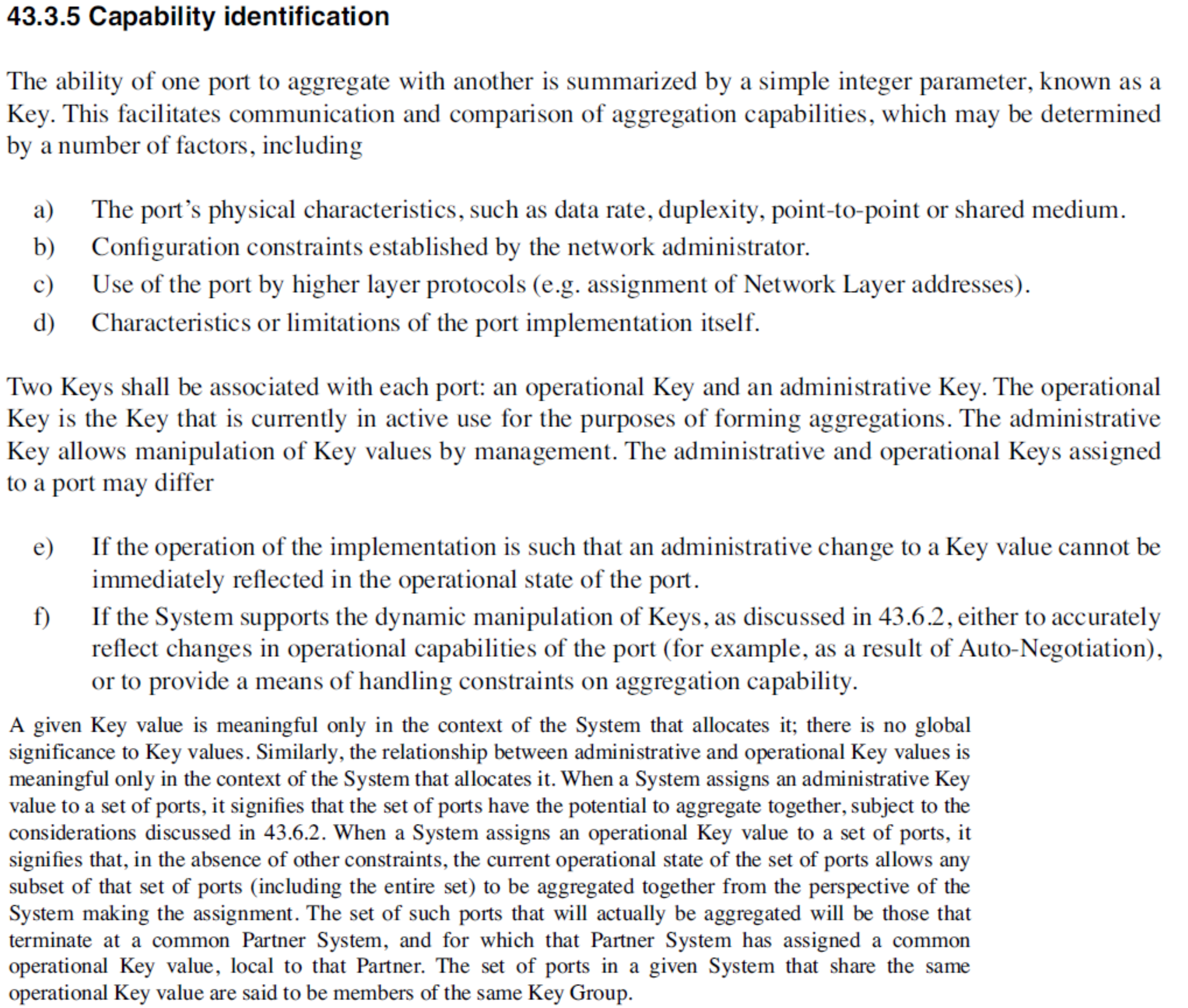
		hashing function *“mapping function”*
		
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		  hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		
		For example, Alexander discloses applying a distributor hash function to the frame information which includes determining a hash key based on packet information.
		
		Alexander at 3:1-40 (“The hash function is preferably selected such that suc-cessive application of the hash function to all source and destination addresses expected to be seen by the Ethernet switch will produce a lowest value hash key, a highest value hash key, and a group of hash keys having intermediate values distributed evenly between the lowest and highest
		values.
		
		The distribution table contains a separate port identifier look-up table for each aggregated grouping of outgoing ports. Advantageously, the hash key is an N bit hash key; and, each port identifier look-up table

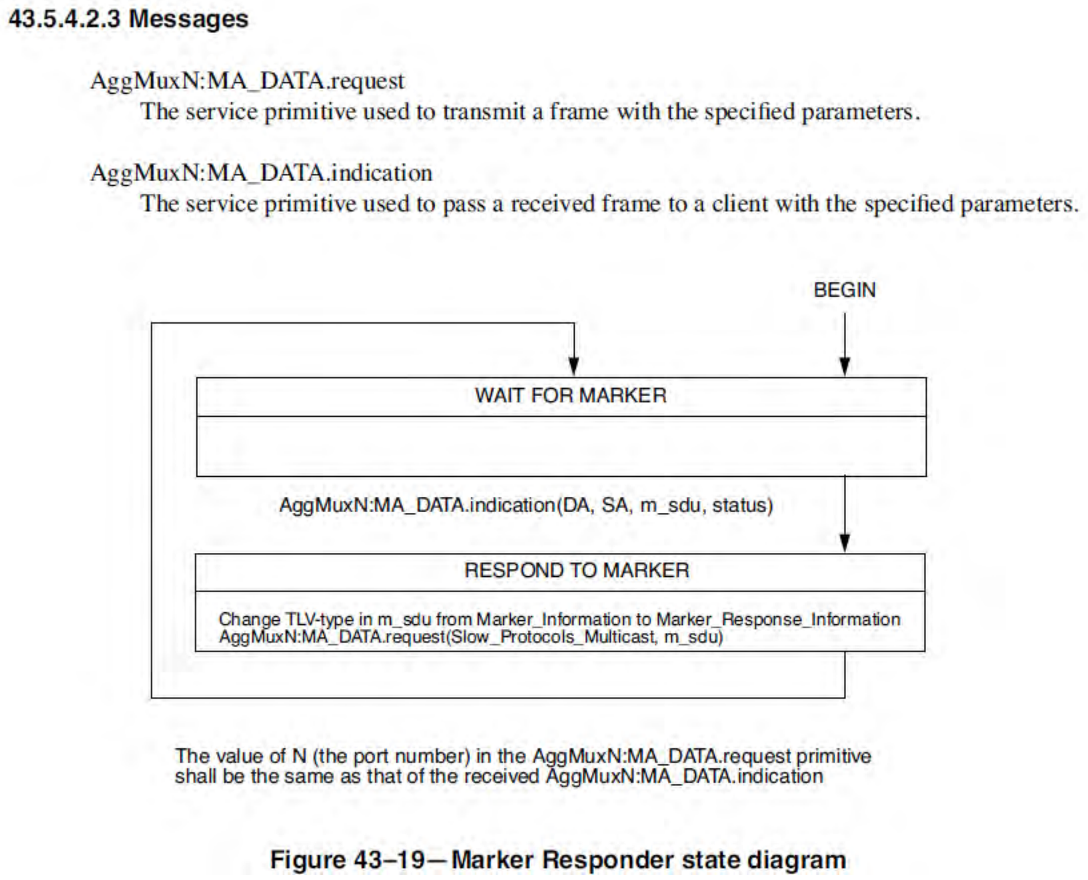
No.	’740 Patent Claim 10	IEEE 802.3
		contains 2^N entries occupying 2^N consecutive locations, with each entry being an identifier of a particular one of the physical outgoing ports. Identifiers for particular outgoing ports are retrieved from the distribution table by extracting first and second N bit hash keys which form part of the retrieved destination and source address contexts respectively. The hash keys are combined to form an N bit connection identifier. The port identifier look-up table corresponding to the aggregated grouping represented by the retrieved destination address is selected, and the entry at the table location corresponding to the value of the N bit connection identifier is retrieved. If the address look-up table does not contain a destination address corresponding to the extracted destination address then first and second hash keys are produced by applying a hash function to the extracted source and destination addresses respectively. The hash keys are combined to form an N bit connection identifier. The incoming port on which the packet containing the extracted source address was received is identified. All of the aggregated groupings are scanned to identify all outgoing ports to which packets may be directed from the incoming port on which the packet was received. For each one of those outgoing ports, the port identifier look-up table corresponding to the aggregated grouping containing that outgoing port is selected, the entry at the table location corresponding to the value of the N bit connection identifier is retrieved, and the received packet is queued for outgoing transmission on the outgoing port corresponding to the retrieved entry.”) Alexander at 5:10-35 (“If a packet arrives bearing a source Ethernet MAC address that was not found in look-up table 12 by address resolution unit 10, learning function 16 is invoked to update look-up table 12 with the new address (i.e. processing branches along the "No" exit from FIG. 2, block 36). Learning function 16 first computes a hash function on the source Ethernet MAC address, generating an N-bit hash key ("partial connection identifier") from the 48-bit MAC address, where N is some small integer in the range of 3 to 8 (FIG. 2, block 38). The physical port on which the packet arrived is then determined. If the physical port is found to be associated with an aggregate group (i.e., it is one of a set of ports that have been bound into a single logical port), then the logical identifier assigned to the aggregate group is also determined. The hash key is then stored into address look-up table 12 in conjunction with the actual Ethernet MAC address and the port identifier (FIG. 2, block 40). The physical port identifier is used if the port is not part of an aggregate group (i.e. if processing branched along the "No" exit from block 30 and through block 32), while the logical identifier is used for ports that have been aggregated (i.e. if processing

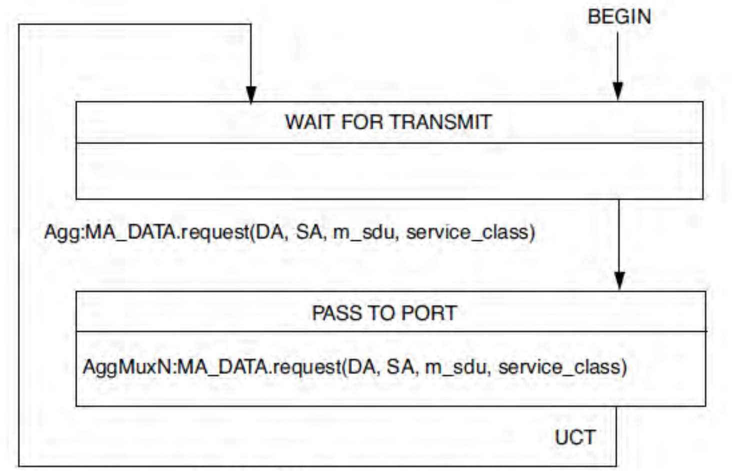
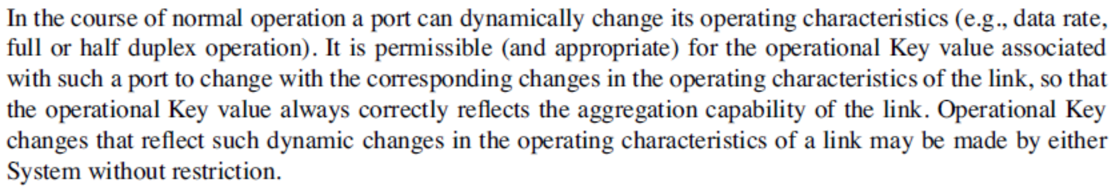
No.	’740 Patent Claim 10	IEEE 802.3
		branched along the "Yes" exit from block 30 and through block 34). The hash key and port identifier are considered to form the "context" for the given MAC address.”) Alexander at 5:36-46 (“The hash function should be selected to ensure an even distribution of hash key values over the range of MAC addresses that are expected to be seen by the Ethernet switch. As a specific example, the EXACT™ Ethernet switch system employs an exclusive-OR based hash function, wherein the 48-bit MAC address is divided into 16-bit blocks, which are then exclusive-ORed together to form a single 16-bit number; the 3 least significant bits (LSBs) of this number are taken to produce a 3-bit hash key. Other schemes such as CRC-based or checksum-based hashes may also be used.”) Alexander at 6:49-65 (“If the context information for the destination address indicates, however, that the target is an aggregate group (i.e. if processing branches along the "Yes" exit from FIG. 2, block 42) then the logical identifier assigned to the aggregate group is retrieved and is used to select the proper look-up table contained within the distribution table data structure. The hash keys (partial connection identifiers) stored into the contexts for the source and destination MAC addresses are obtained from address resolution unit 10 and combined to generate a "connection identifier" with the same number of bits (FIG. 2, block 44). (In the EXACT™ Ethernet switch, a Boolean exclusive-OR operation is used to combine the hash keys without increasing the number of bits.) This connection identifier is then used to index into the selected look-up table, and finally retrieve an actual physical port index on which the packet must be transmitted (FIG. 2, block 46).”)
10[c]	calculating a modulo of a division operation of the hashing key by the hashing size, and	IEEE 802.3 discloses calculating a modulo of a division operation of the hashing key by the hashing size. For example, IEEE 802.3 discloses a hash function that would include calculating a modulo based on the generated values. A person of ordinary skill in the art would understand that a hashing function is a standard algorithm used to route data traffic that involves information specific to the system architecture and includes calculations of the generated values. A person of ordinary skill in the art would further understand that the physical links over which the frame is sent are determined based on the calculations of the hashing function. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the IEEE 802.3 is found to not meet this limitation,

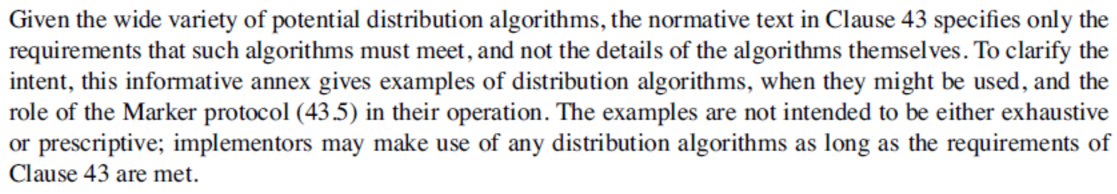
No.	’740 Patent Claim 10	IEEE 802.3
		calculating a modulo of a division operation of the hashing key by the hashing size would have been obvious to a person having ordinary skill in the art, as explained below. IEEE 802.3 at 1469  **43.2.2 Service interfaces** The MAC Client communicates with the Aggregator using the standard service interface specified in Clause 2. Similarly, Link Aggregation communicates internally (between Frame Collection/Distribution, the Aggregator Parser/Multiplexers, the Control Parser/Multiplexers, and Link Aggregation Control) and with its bound ports using the same, standard service interface. No new interlayer service interfaces are defined for Link Aggregation. Since Link Aggregation uses four instances of the MAC Service Interface, it is necessary to introduce a notation convention so that the reader can be clear as to which interface is being referred to at any given time. A prefix is therefore assigned to each service primitive, indicating which of the four interfaces is being invoked, as depicted in Figure 43–2. The prefixes are as follows: a) *Agg:*, for primitives issued on the interface between the MAC Client and the Link Aggregation sublayer. b) *AggMuxN:*, for primitives issued on the interface between Aggregator Parser/Multiplexer N and its internal clients (where N is the port number associated with the Aggregator Parser/Multiplexer). c) *CtrlMuxN:*, for primitives issued on the interface between Control Parser/Multiplexer N and its internal clients (where N is the port number associated with the Control Parser/Multiplexer). d) *MacN:*, for primitives issued on the interface between underlying MAC N and its Control Parser/Multiplexer (where N is the port number associated with the underlying MAC). MAC Clients may generate Agg:MA_DATA.request primitives for transmission on an aggregated link. These are passed by the Frame Distributor to a port selected by the distribution algorithm. MacN:MA_DATA.indication primitives signifying received frames are passed unchanged from a port to the MAC Client by the Frame Collector. MAC Clients that generate MA_CONTROL.request primitives (and which expect MA_CONTROL.indication primitives in response) cannot communicate through a Link Aggregation sublayer. They must communicate directly with the MAC Control entity through which these control primitives are to be sent and received.

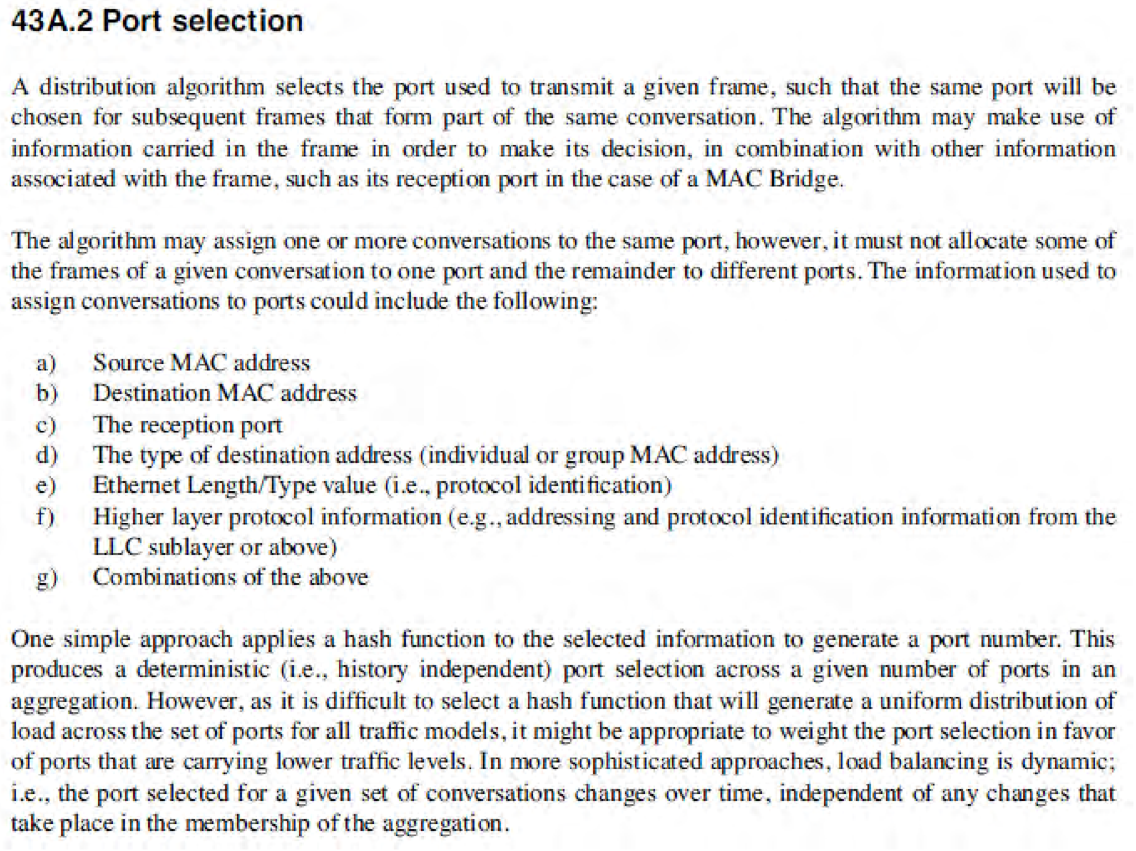
No.	’740 Patent Claim 10	IEEE 802.3
		IEEE 802.3 at 1480  **43.3.4 Port identification** Link Aggregation Control uses a Port Identifier, comprising the concatenation of a Port Priority and a Port Number, to identify the port. Port Numbers (and hence, Port Identifiers) shall be uniquely assigned within a System. Port Number 0 shall not be assigned to any port. When it is necessary to perform numerical comparisons between Port Identifiers, each Port Identifier is considered to be a four octet unsigned binary number constructed as follows: a) The most significant and second most significant octets are the first and second most significant octets of the Port Priority, respectively. b) The third and fourth most significant octets are the first and second most significant octets of the Port Number, respectively. IEEE 802.3 at 1480

No.	’740 Patent Claim 10	IEEE 802.3
		 **43.3.5 Capability identification** The ability of one port to aggregate with another is summarized by a simple integer parameter, known as a Key. This facilitates communication and comparison of aggregation capabilities, which may be determined by a number of factors, including a) The port’s physical characteristics, such as data rate, duplexity, point-to-point or shared medium. b) Configuration constraints established by the network administrator. c) Use of the port by higher layer protocols (e.g. assignment of Network Layer addresses). d) Characteristics or limitations of the port implementation itself. Two Keys shall be associated with each port: an operational Key and an administrative Key. The operational Key is the Key that is currently in active use for the purposes of forming aggregations. The administrative Key allows manipulation of Key values by management. The administrative and operational Keys assigned to a port may differ e) If the operation of the implementation is such that an administrative change to a Key value cannot be immediately reflected in the operational state of the port. f) If the System supports the dynamic manipulation of Keys, as discussed in 43.6.2, either to accurately reflect changes in operational capabilities of the port (for example, as a result of Auto-Negotiation), or to provide a means of handling constraints on aggregation capability. A given Key value is meaningful only in the context of the System that allocates it; there is no global significance to Key values. Similarly, the relationship between administrative and operational Key values is meaningful only in the context of the System that allocates it. When a System assigns an administrative Key value to a set of ports, it signifies that the set of ports have the potential to aggregate together, subject to the considerations discussed in 43.6.2. When a System assigns an operational Key value to a set of ports, it signifies that, in the absence of other constraints, the current operational state of the set of ports allows any subset of that set of ports (including the entire set) to be aggregated together from the perspective of the System making the assignment. The set of such ports that will actually be aggregated will be those that terminate at a common Partner System, and for which that Partner System has assigned a common operational Key value, local to that Partner. The set of ports in a given System that share the same operational Key value are said to be members of the same Key Group.

No.	’740 Patent Claim 10	IEEE 802.3
		IEEE 802.3 at 1516  **43.5.4.2.3 Messages** AggMuxN:MA_DATA.request The service primitive used to transmit a frame with the specified parameters. AggMuxN:MA_DATA.indication The service primitive used to pass a received frame to a client with the specified parameters. BEGIN WAIT FOR MARKER AggMuxN:MA_DATA.indication(DA, SA, m_sdu, status) RESPOND TO MARKER Change TLV-type in m_sdu from Marker_Information to Marker_Response_Information AggMuxN:MA_DATA.request(Slow_Protocols_Multicast, m_sdu) The value of N (the port number) in the AggMuxN:MA_DATA.request primitive shall be the same as that of the received AggMuxN:MA_DATA.indication **Figure 43–19—Marker Responder state diagram** IEEE 802.3 at 1472

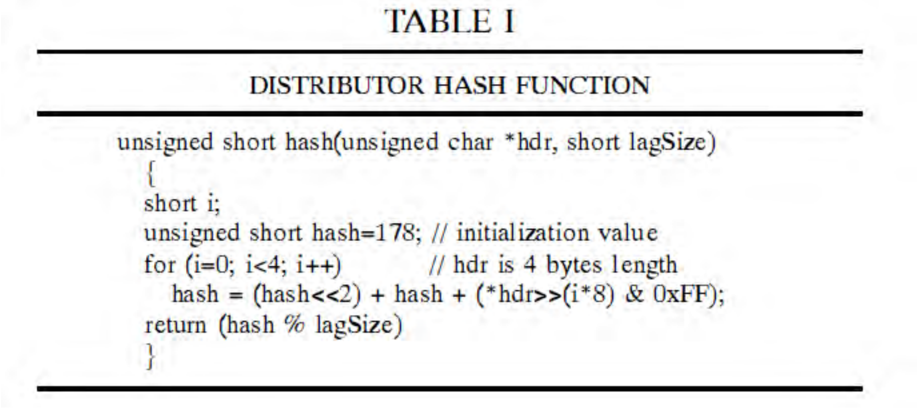
No.	’740 Patent Claim 10	IEEE 802.3
		43.2.4.1.3 State diagram  BEGIN WAIT FOR TRANSMIT Agg:MA_DATA.request(DA, SA, m_sdu, service_class) PASS TO PORT AggMuxN:MA_DATA.request(DA, SA, m_sdu, service_class) UCT If a client issues an Agg:MA_DATA.request primitive that contains no SA parameter, the AggMuxN:MA_DATA.request primitive generated shall use the Aggregator’s MAC address for the SA. NOTE—The algorithm that the Frame Distributor uses to select the value of N in AggMuxN:MA_DATA.request for a given frame is unspecified. Figure 43–4—Frame Distributor state diagram IEEE at 802.3 at 1518  In the course of normal operation a port can dynamically change its operating characteristics (e.g., data rate, full or half duplex operation). It is permissible (and appropriate) for the operational Key value associated with such a port to change with the corresponding changes in the operating characteristics of the link, so that the operational Key value always correctly reflects the aggregation capability of the link. Operational Key changes that reflect such dynamic changes in the operating characteristics of a link may be made by either System without restriction.

No.	’740 Patent Claim 10	IEEE 802.3
		IEEE 802.3 at 1548  Given the wide variety of potential distribution algorithms, the normative text in Clause 43 specifies only the requirements that such algorithms must meet, and not the details of the algorithms themselves. To clarify the intent, this informative annex gives examples of distribution algorithms, when they might be used, and the role of the Marker protocol (43.5) in their operation. The examples are not intended to be either exhaustive or prescriptive; implementors may make use of any distribution algorithms as long as the requirements of Clause 43 are met. IEEE 802.3 at 1549

No.	’740 Patent Claim 10	IEEE 802.3
		 43A.2 Port selection A distribution algorithm selects the port used to transmit a given frame, such that the same port will be chosen for subsequent frames that form part of the same conversation. The algorithm may make use of information carried in the frame in order to make its decision, in combination with other information associated with the frame, such as its reception port in the case of a MAC Bridge. The algorithm may assign one or more conversations to the same port, however, it must not allocate some of the frames of a given conversation to one port and the remainder to different ports. The information used to assign conversations to ports could include the following: a) Source MAC address b) Destination MAC address c) The reception port d) The type of destination address (individual or group MAC address) e) Ethernet Length/Type value (i.e., protocol identification) f) Higher layer protocol information (e.g., addressing and protocol identification information from the LLC sublayer or above) g) Combinations of the above One simple approach applies a hash function to the selected information to generate a port number. This produces a deterministic (i.e., history independent) port selection across a given number of ports in an aggregation. However, as it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models, it might be appropriate to weight the port selection in favor of ports that are carrying lower traffic levels. In more sophisticated approaches, load balancing is dynamic; i.e., the port selected for a given set of conversations changes over time, independent of any changes that take place in the membership of the aggregation. Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, IEEE 802.3 in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[c] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references.

No.	’740 Patent Claim 10	IEEE 802.3
		For example, Bruckman discloses distributing data frames over physical links and traces based on a hash function involving a division operation (%). Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0025] (“Typically, setting the protection policy includes determining a maximum number of the physical links that may fail while the logical link continues to provide at least the guaranteed bandwidth for the connection. In one embodiment, the guaranteed bandwidth is a bandwidth B, and the plurality of physical links consists of N links, and the maximum number is an integer P, and the link bandwidth allocated to each of the links is no less than B/(N-P). Conveying the data may further include managing the transmission of the data responsively to an actual number X of the physical links that have failed so that the guaranteed bandwidth on each of the links is limited to B/(N-X), X P, and an excess bandwidth on the physical links over the guaranteed bandwidth is available for other connections.”) Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”) Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:

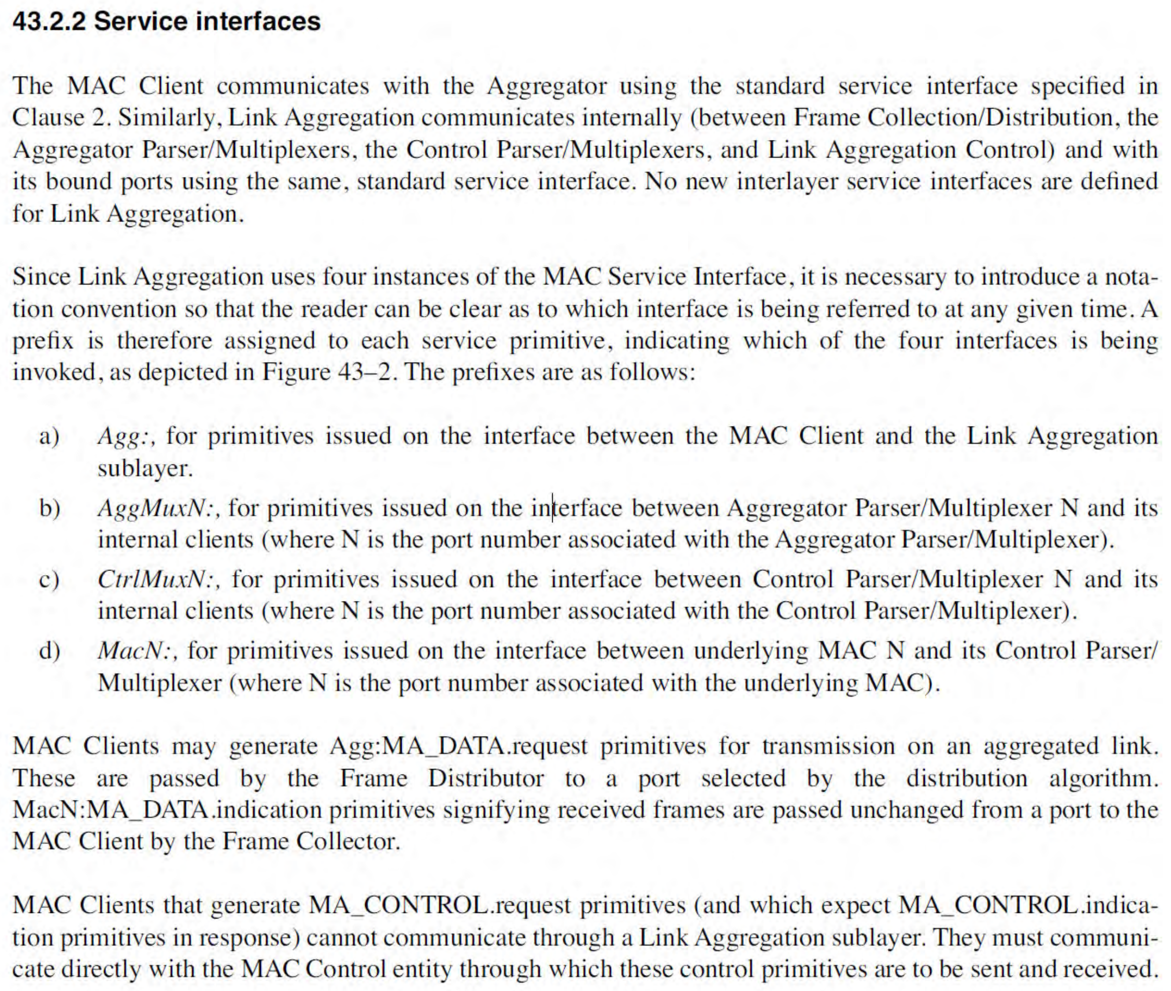
No.	’740 Patent Claim 10	IEEE 802.3

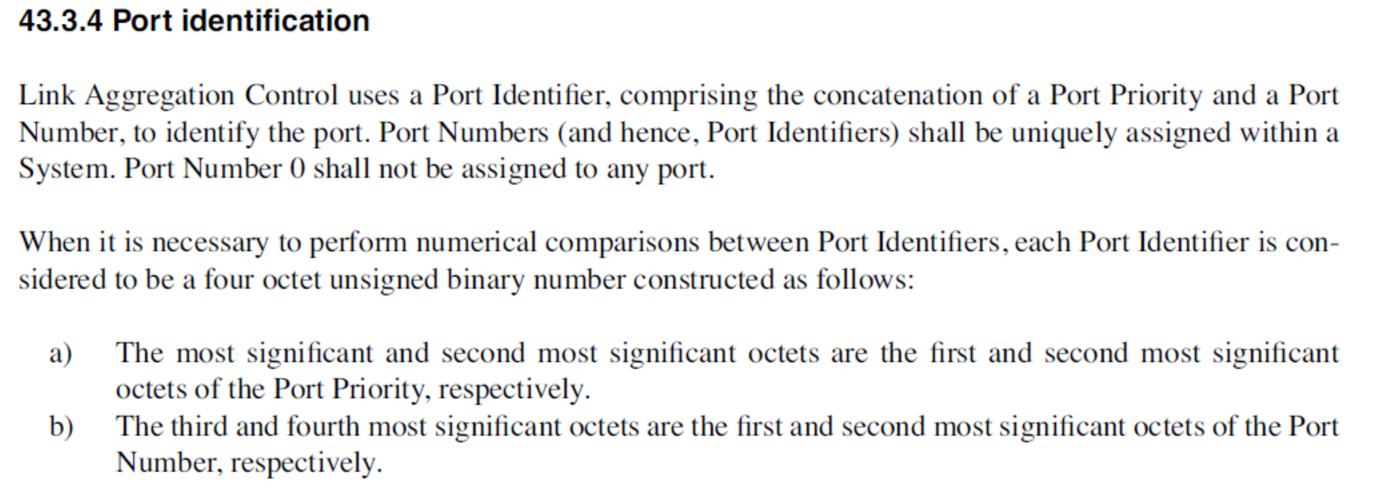
		
		TABLE I
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		”)
		
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		
		Bruckman at [0067] (“A similar problem may arise if there is a failure in a link in an aggregation group or in one of a number of line cards serving the aggregation group. In this case, to maintain the bandwidth allocation B made by CAC 44, each of the remaining links in the group must now carry, on average, B/(N-M) traffic, wherein M is the number of links in the group that are out of service. If only BIN has been allocated to each link, the remaining active links may not have sufficient bandwidth to continue to provide the bandwidth that has been guaranteed to the connections that they are required to carry. A similar problem arises with respect to loading of traces 52. For example, if there is a failure in LC2 or in one of links 30 in group 36 that connect to LC2, the trace connecting the multiplexer 50 in LCl will have to carry a substantially larger share of the bandwidth, or even all of the bandwidth, that is allocated to the connection in ques-tion.”)
		
		Bruckman at [0068] (“FIG. 3 is a flow chart that schematically illustrates a method for dealing with these problems of fluctuating bandwidth requirements, in accordance with an embodiment of the present invention. In order to provide sufficient bandwidth for failure protection, CAC 44 uses a safety margin

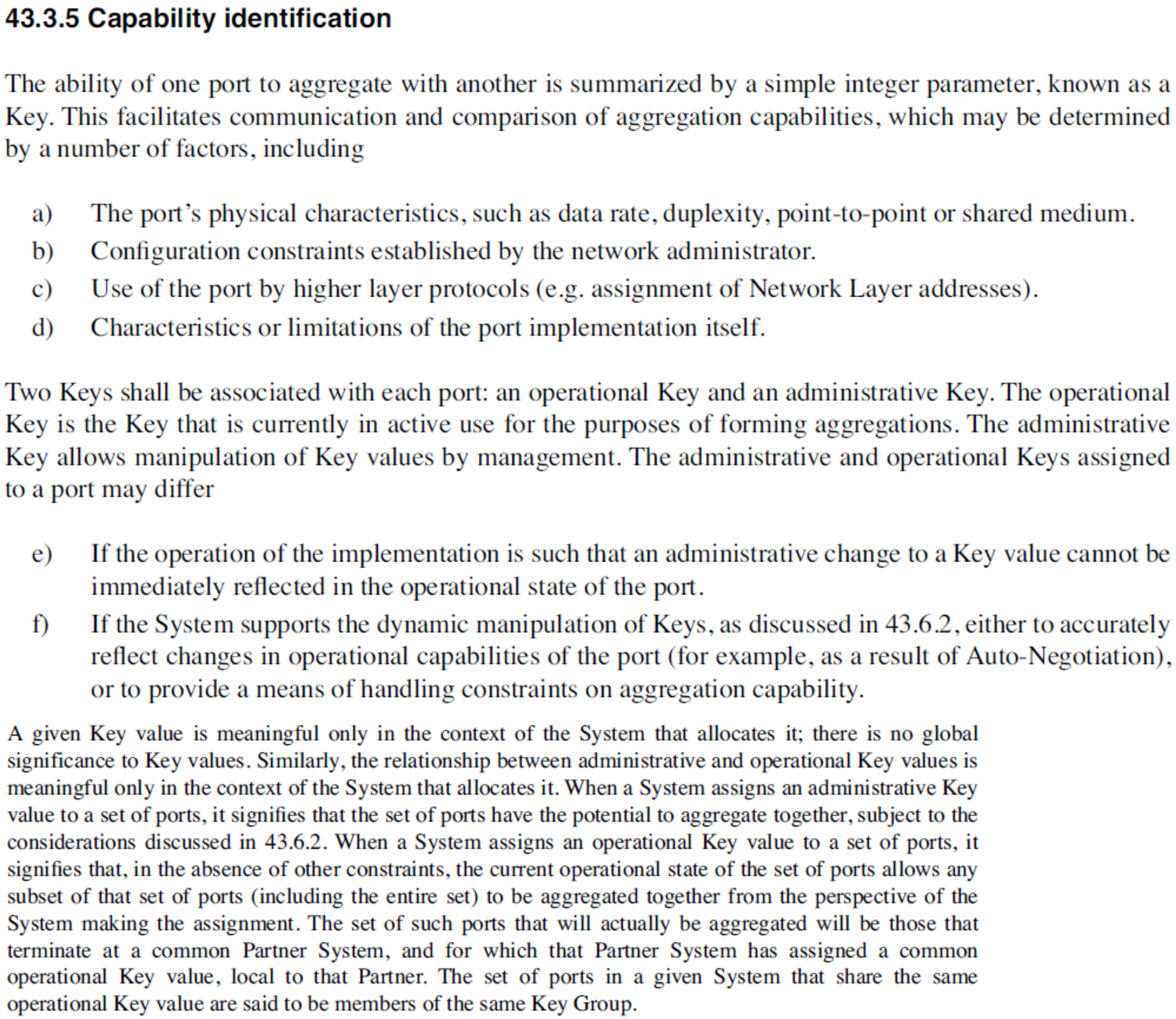
No.	’740 Patent Claim 10	IEEE 802.3
		based on a protection parameter P, which is assigned at a protection setting step 60. P represents the maximum number of links in the group that can be out of service while still permitting the aggregation group to provide a given connection with the bandwidth that has been guaranteed to the connection. CAC 44 will then allocate at least B/(N-P) bandwidth to each link in the group, so that if P links fail, the group still provides total bandwidth of (N-P)*B/(N-P)= B. Setting P=l is equivalent to 1:N protection, so that the group will be unaffected by failure of a single link. In the example of group 36, shown in FIG. 2, setting P=2 will give both facility and equipment protection, i.e., the group will be unaffected not only by failure of a link, but also by failure of one of line cards 34. In the extreme case, in which P=N-1, CAC 44 will allocate the full bandwidth B on each link in the group.”) As another example, Singh discloses determining a ratio between the number of ingress and egress links and the number of links carrying data to the backplane and using a modulo to correspond to the channel’s link number. Singh at 9:30-43 (“The ratio between the number of line ingress links and the number of links carrying data to the backplane gives the backplane speedup for the system. In this example, there are 10 ingress links into the MS and 20 links (2 backplane channels) carrying that data to the backplane. This gives a backplane speedup of 2x. As another example, with 8 ingress links and 12 backplane links, there is a speedup of 1.5x. It should be noted that in addition to the backplane speedup, there is also an ingress/egress speedup. With 10 ingress links capable of carrying 2 Gbps each of raw data, this presents a 20 Gbps interface to the MS. An OC-192 only has approximately 10 Gbps worth of data. Taking into account cell overhead and cell quantization inefficiencies, there still remains excess capacity in the links.”) Singh at 11:29-38 (“FIG. 9 is a diagram illustrating link to channel assignments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a channel, whether it is a backplane channel or an ingress/egress channel. There is no logical relationship between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.”)

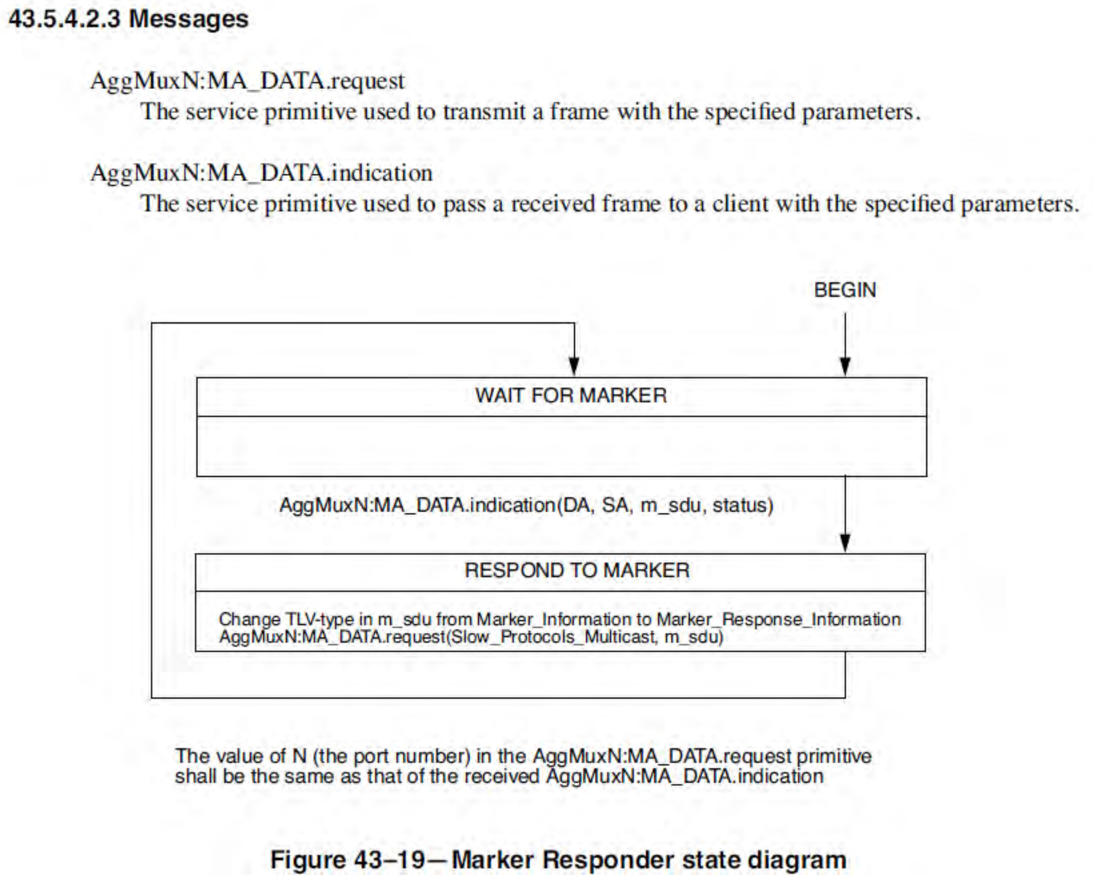
No.	’740 Patent Claim 10	IEEE 802.3
		Singh at 15:15-39 (“The number of crossbars that are required in a system is dependent on how many links are being used to create the backplane channels. There should be an even number of crossbars and they would be divided evenly across the switch cards. The following equation, for most cases, pro-vides the correct number of crossbars: # of Crossbars=(# links per ingress channelx# of ingress channels per portx# of port cardsx speedup)/32. For the 8x8 configuration, the # of crossbars should be multiplied by (4x# of iMS)/(# backplane channels per port card). The number of port cards should be rounded up to the nearest supported configuration, i.e. 8, 16, or 32. The speedup in the case of crossbars should be the fractional speedup that is desired. Example to determine the number of arbiters and cross-bars for the following system: 4 channel port cards (40 Gbps) 8 links per channel 16 port cards Speedup=l.5 # of arbiters=(4x2x2)/2=8 # of crossbars=(8x4x16xl.5)/32=24. This would give 3crossbars per arbiter.”) Singh at 16:28-44 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 16x16 and 32x32 is the organization of the switchplane. The port card remains the same. Backplane channels 1 and 2 are used for the backplane connectivity. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 16, correspond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”)

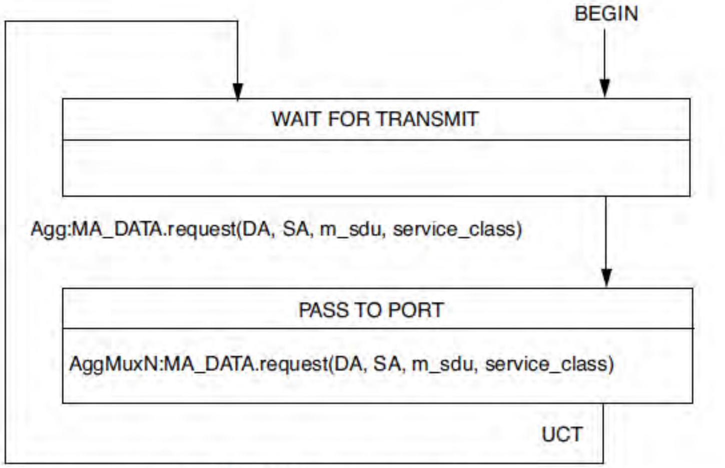
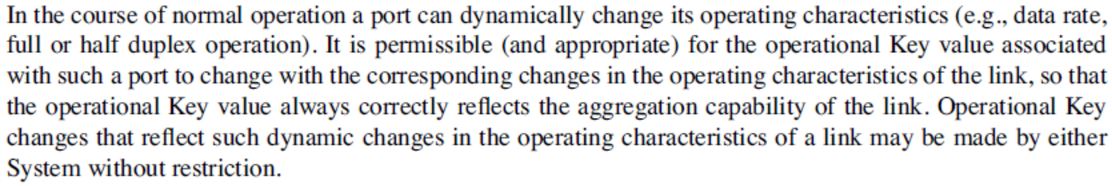
No.	’740 Patent Claim 10	IEEE 802.3
		Singh at 17:31-49 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 8x8 and 16x16 is the organi-zation of the switchplane. The port card remains the same. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Links 0-7 and 24-31 on the arbiters would not be used and would be powered off. Links 0-7 and 24-31 on the crossbars would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Backplane channels 1 and 2 are used for the backplane connectivity. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 8, corre-spond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”)
10[d]	selecting the first and second physical links responsively to the modulo.	IEEE 802.3 discloses selecting the first and second physical links responsively to the modulo. For example, IEEE 802.3 discloses selecting the physical links over which a frame is transmitted. A person of ordinary skill in the art would understand that a hashing function is a standard algorithm used to route data traffic that involves information specific to the system architecture and includes calculations of the generated values. A person of ordinary skill in the art would further understand that the physical links over which the frame is sent are determined based on the calculations of the hashing function. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the IEEE 802.3 is found to not meet this limitation, selecting the first and second physical links responsively to the modulo would have been obvious to a person having ordinary skill in the art, as explained below. IEEE 802.3 at 1469

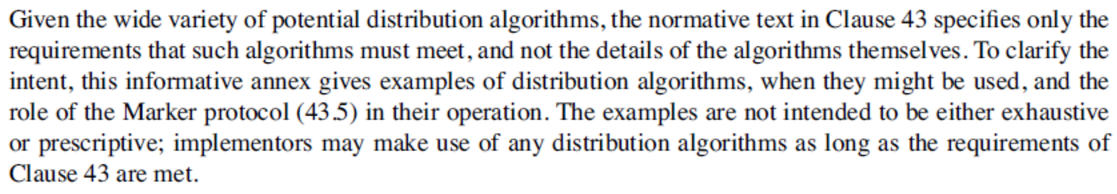
No.	’740 Patent Claim 10	IEEE 802.3
		 **43.2.2 Service interfaces** The MAC Client communicates with the Aggregator using the standard service interface specified in Clause 2. Similarly, Link Aggregation communicates internally (between Frame Collection/Distribution, the Aggregator Parser/Multiplexers, the Control Parser/Multiplexers, and Link Aggregation Control) and with its bound ports using the same, standard service interface. No new interlayer service interfaces are defined for Link Aggregation. Since Link Aggregation uses four instances of the MAC Service Interface, it is necessary to introduce a notation convention so that the reader can be clear as to which interface is being referred to at any given time. A prefix is therefore assigned to each service primitive, indicating which of the four interfaces is being invoked, as depicted in Figure 43–2. The prefixes are as follows: a) *Agg:*, for primitives issued on the interface between the MAC Client and the Link Aggregation sublayer. b) *AggMuxN:*, for primitives issued on the interface between Aggregator Parser/Multiplexer N and its internal clients (where N is the port number associated with the Aggregator Parser/Multiplexer). c) *CtrlMuxN:*, for primitives issued on the interface between Control Parser/Multiplexer N and its internal clients (where N is the port number associated with the Control Parser/Multiplexer). d) *MacN:*, for primitives issued on the interface between underlying MAC N and its Control Parser/Multiplexer (where N is the port number associated with the underlying MAC). MAC Clients may generate Agg:MA_DATA.request primitives for transmission on an aggregated link. These are passed by the Frame Distributor to a port selected by the distribution algorithm. MacN:MA_DATA.indication primitives signifying received frames are passed unchanged from a port to the MAC Client by the Frame Collector. MAC Clients that generate MA_CONTROL.request primitives (and which expect MA_CONTROL.indication primitives in response) cannot communicate through a Link Aggregation sublayer. They must communicate directly with the MAC Control entity through which these control primitives are to be sent and received. IEEE 802.3 at 1480

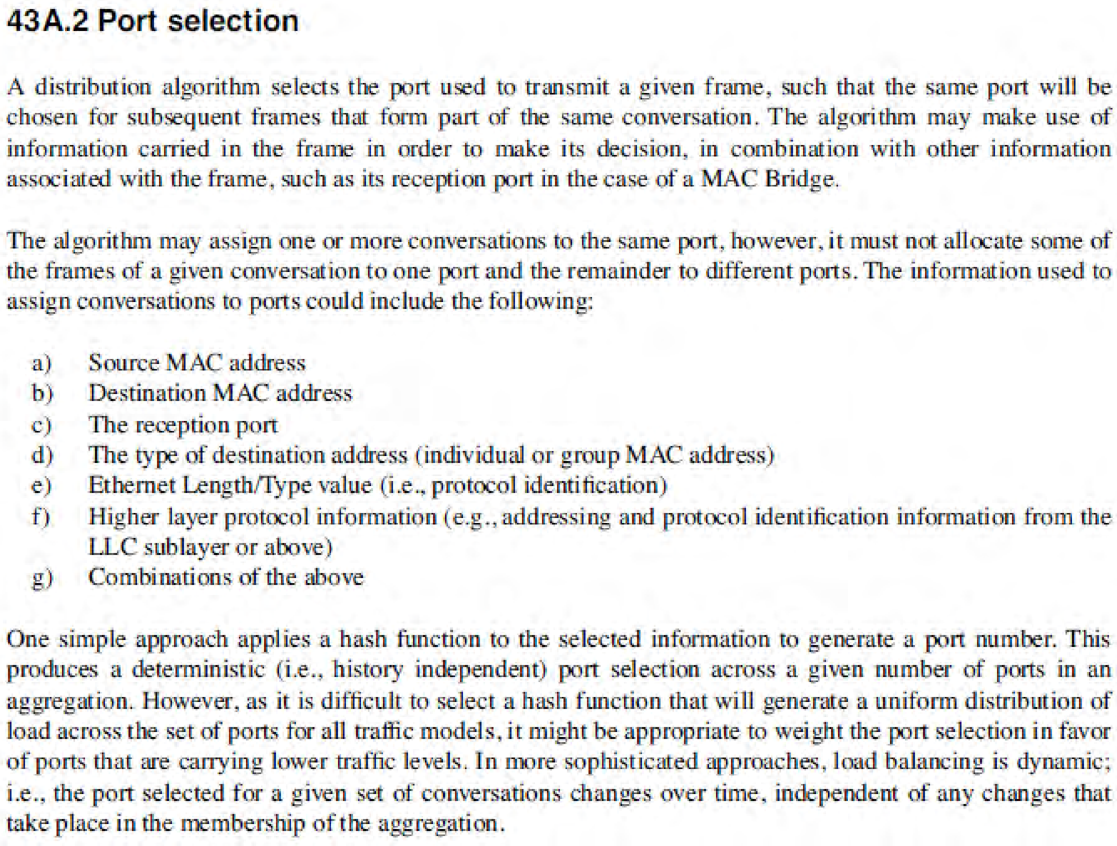
No.	’740 Patent Claim 10	IEEE 802.3
		 **43.3.4 Port identification** Link Aggregation Control uses a Port Identifier, comprising the concatenation of a Port Priority and a Port Number, to identify the port. Port Numbers (and hence, Port Identifiers) shall be uniquely assigned within a System. Port Number 0 shall not be assigned to any port. When it is necessary to perform numerical comparisons between Port Identifiers, each Port Identifier is considered to be a four octet unsigned binary number constructed as follows: a) The most significant and second most significant octets are the first and second most significant octets of the Port Priority, respectively. b) The third and fourth most significant octets are the first and second most significant octets of the Port Number, respectively. IEEE 802.3 at 1480

No.	’740 Patent Claim 10	IEEE 802.3
		 **43.3.5 Capability identification** The ability of one port to aggregate with another is summarized by a simple integer parameter, known as a Key. This facilitates communication and comparison of aggregation capabilities, which may be determined by a number of factors, including a) The port’s physical characteristics, such as data rate, duplexity, point-to-point or shared medium. b) Configuration constraints established by the network administrator. c) Use of the port by higher layer protocols (e.g. assignment of Network Layer addresses). d) Characteristics or limitations of the port implementation itself. Two Keys shall be associated with each port: an operational Key and an administrative Key. The operational Key is the Key that is currently in active use for the purposes of forming aggregations. The administrative Key allows manipulation of Key values by management. The administrative and operational Keys assigned to a port may differ e) If the operation of the implementation is such that an administrative change to a Key value cannot be immediately reflected in the operational state of the port. f) If the System supports the dynamic manipulation of Keys, as discussed in 43.6.2, either to accurately reflect changes in operational capabilities of the port (for example, as a result of Auto-Negotiation), or to provide a means of handling constraints on aggregation capability. A given Key value is meaningful only in the context of the System that allocates it; there is no global significance to Key values. Similarly, the relationship between administrative and operational Key values is meaningful only in the context of the System that allocates it. When a System assigns an administrative Key value to a set of ports, it signifies that the set of ports have the potential to aggregate together, subject to the considerations discussed in 43.6.2. When a System assigns an operational Key value to a set of ports, it signifies that, in the absence of other constraints, the current operational state of the set of ports allows any subset of that set of ports (including the entire set) to be aggregated together from the perspective of the System making the assignment. The set of such ports that will actually be aggregated will be those that terminate at a common Partner System, and for which that Partner System has assigned a common operational Key value, local to that Partner. The set of ports in a given System that share the same operational Key value are said to be members of the same Key Group.

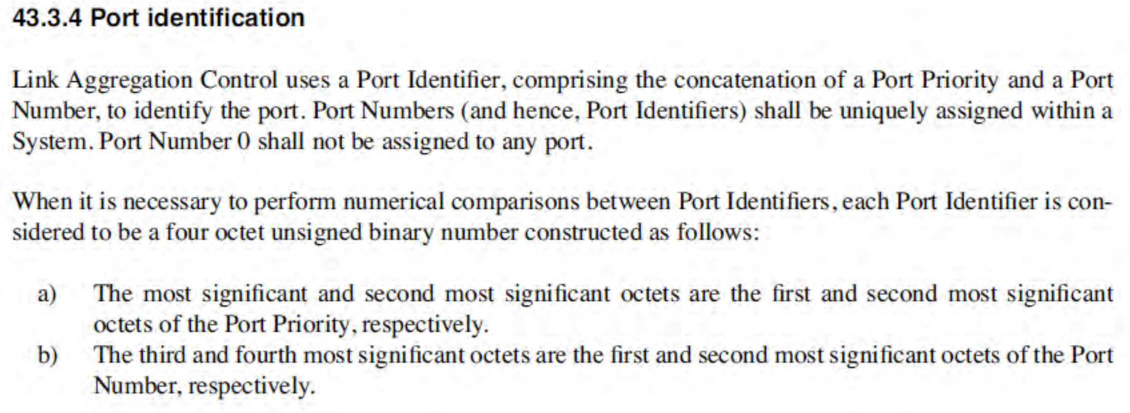
No.	’740 Patent Claim 10	IEEE 802.3
		IEEE 802.3 at 1516  **43.5.4.2.3 Messages** AggMuxN:MA_DATA.request The service primitive used to transmit a frame with the specified parameters. AggMuxN:MA_DATA.indication The service primitive used to pass a received frame to a client with the specified parameters. BEGIN WAIT FOR MARKER AggMuxN:MA_DATA.indication(DA, SA, m_sdu, status) RESPOND TO MARKER Change TLV-type in m_sdu from Marker_Information to Marker_Response_Information AggMuxN:MA_DATA.request(Slow_Protocols_Multicast, m_sdu) The value of N (the port number) in the AggMuxN:MA_DATA.request primitive shall be the same as that of the received AggMuxN:MA_DATA.indication **Figure 43–19—Marker Responder state diagram** IEEE 802.3 at 1472

No.	’740 Patent Claim 10	IEEE 802.3
		43.2.4.1.3 State diagram  BEGIN WAIT FOR TRANSMIT Agg:MA_DATA.request(DA, SA, m_sdu, service_class) PASS TO PORT AggMuxN:MA_DATA.request(DA, SA, m_sdu, service_class) UCT If a client issues an Agg:MA_DATA.request primitive that contains no SA parameter, the AggMuxN:MA_DATA.request primitive generated shall use the Aggregator’s MAC address for the SA. NOTE—The algorithm that the Frame Distributor uses to select the value of N in AggMuxN:MA_DATA.request for a given frame is unspecified. **Figure 43–4—Frame Distributor state diagram** IEEE at 802.3 at 1518  In the course of normal operation a port can dynamically change its operating characteristics (e.g., data rate, full or half duplex operation). It is permissible (and appropriate) for the operational Key value associated with such a port to change with the corresponding changes in the operating characteristics of the link, so that the operational Key value always correctly reflects the aggregation capability of the link. Operational Key changes that reflect such dynamic changes in the operating characteristics of a link may be made by either System without restriction.

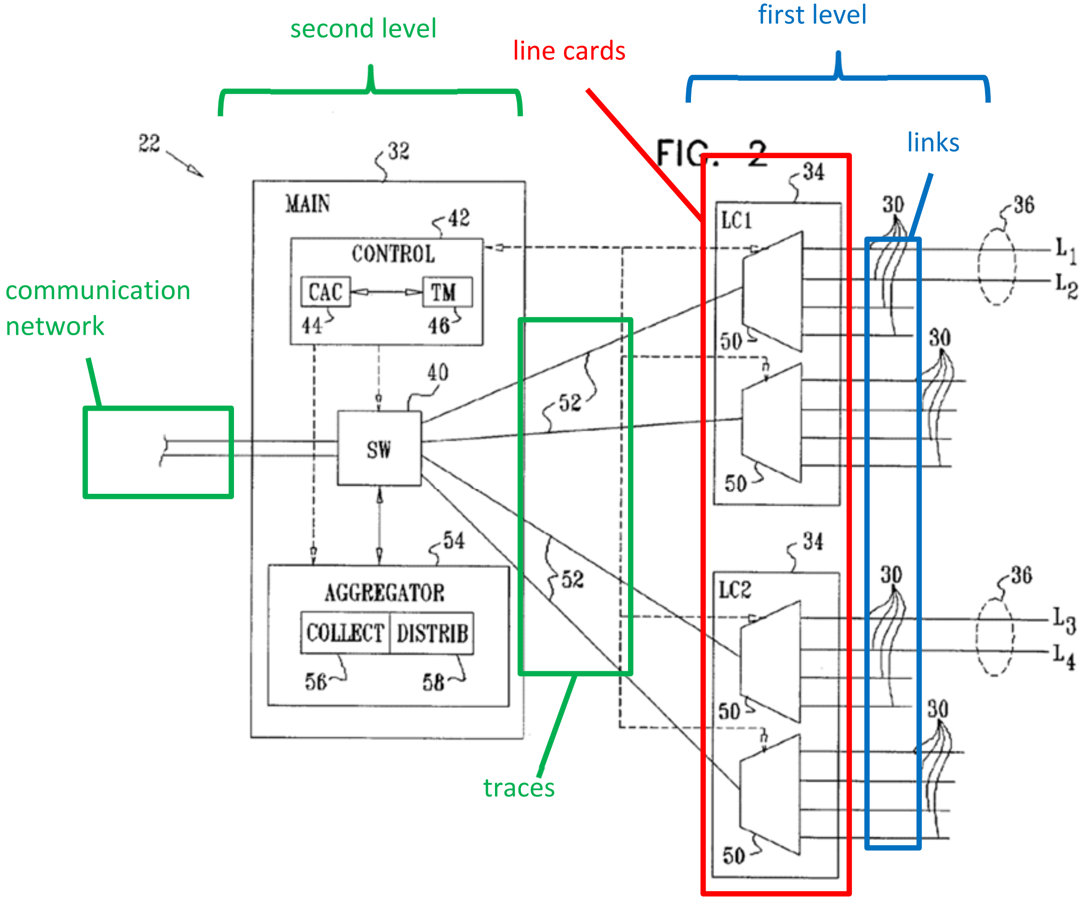
No.	’740 Patent Claim 10	IEEE 802.3
		IEEE 802.3 at 1548  Given the wide variety of potential distribution algorithms, the normative text in Clause 43 specifies only the requirements that such algorithms must meet, and not the details of the algorithms themselves. To clarify the intent, this informative annex gives examples of distribution algorithms, when they might be used, and the role of the Marker protocol (43.5) in their operation. The examples are not intended to be either exhaustive or prescriptive; implementors may make use of any distribution algorithms as long as the requirements of Clause 43 are met. IEEE 802.3 at 1549

No.	’740 Patent Claim 10	IEEE 802.3
		 **43A.2 Port selection** A distribution algorithm selects the port used to transmit a given frame, such that the same port will be chosen for subsequent frames that form part of the same conversation. The algorithm may make use of information carried in the frame in order to make its decision, in combination with other information associated with the frame, such as its reception port in the case of a MAC Bridge. The algorithm may assign one or more conversations to the same port, however, it must not allocate some of the frames of a given conversation to one port and the remainder to different ports. The information used to assign conversations to ports could include the following: a) Source MAC address b) Destination MAC address c) The reception port d) The type of destination address (individual or group MAC address) e) Ethernet Length/Type value (i.e., protocol identification) f) Higher layer protocol information (e.g., addressing and protocol identification information from the LLC sublayer or above) g) Combinations of the above One simple approach applies a hash function to the selected information to generate a port number. This produces a deterministic (i.e., history independent) port selection across a given number of ports in an aggregation. However, as it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models, it might be appropriate to weight the port selection in favor of ports that are carrying lower traffic levels. In more sophisticated approaches, load balancing is dynamic; i.e., the port selected for a given set of conversations changes over time, independent of any changes that take place in the membership of the aggregation.

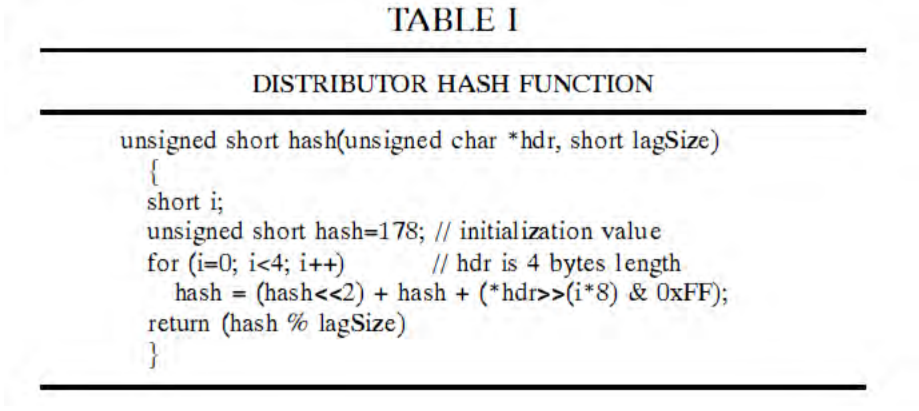
No.	’740 Patent Claim 11	IEEE 802.3
11	The method according to claim 10, wherein selecting the first	IEEE 802.3 discloses the method according to claim 10, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo.

No.	’740 Patent Claim 11	IEEE 802.3
	and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo.	For example, IEEE 802.3 discloses selecting the physical links over which a frame is transmitted. A person of ordinary skill in the art would understand that a hashing function is a standard algorithm used to route data traffic that involves information specific to the system architecture and includes calculations of the generated values. A person of ordinary skill in the art would further understand that the physical links over which the frame is sent are determined based on the calculations of the hashing function. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the IEEE 802.3 is found to not meet this limitation, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo would have been obvious to a person having ordinary skill in the art, as explained below. *See supra* Claim 10. IEEE 802.3 at 1480  **43.3.4 Port identification** Link Aggregation Control uses a Port Identifier, comprising the concatenation of a Port Priority and a Port Number, to identify the port. Port Numbers (and hence, Port Identifiers) shall be uniquely assigned within a System. Port Number 0 shall not be assigned to any port. When it is necessary to perform numerical comparisons between Port Identifiers, each Port Identifier is considered to be a four octet unsigned binary number constructed as follows: a) The most significant and second most significant octets are the first and second most significant octets of the Port Priority, respectively. b) The third and fourth most significant octets are the first and second most significant octets of the Port Number, respectively. Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, IEEE 802.3 in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the

No.	’740 Patent Claim 11	IEEE 802.3
		references identified in element 11 of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses distributing data frames over physical links and traces based on a division operation in the hash function, i.e., selecting the first and second physical links responsively to the modulo, involving specific byte lengths of the frame information, i.e., responsively to respective first and second subsets of bits in a binary representation of the modulo. Bruckman at Figure 2 (annotated)

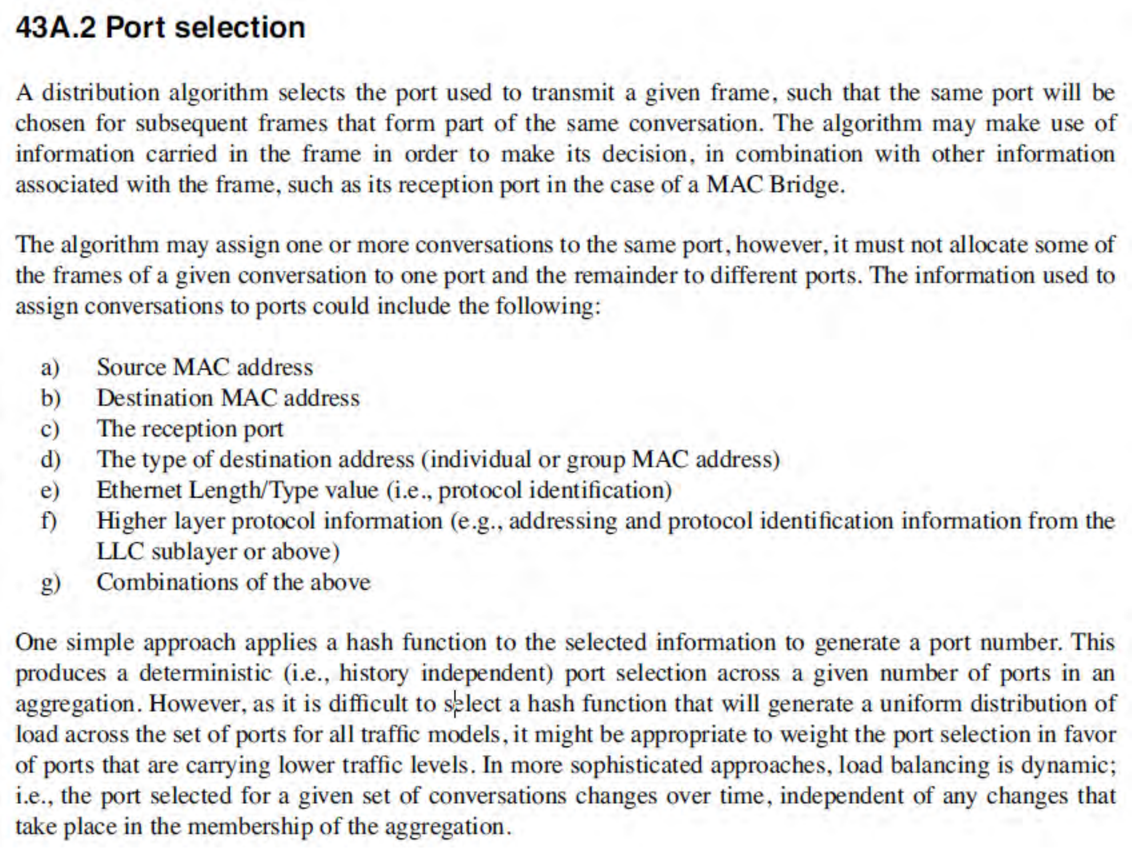
No.	’740 Patent Claim 11	IEEE 802.3
		 second level line cards first level links communication network traces 22 32 MAIN 42 CONTROL CAC TM 44 46 40 SW 52 52 54 AGGREGATOR COLLECT DISTRIB 56 58 34 LC1 50 50 34 LC2 50 50 30 30 30 30 36 36 L1 L2 L3 L4 FIG. 2 Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:

No.	’740 Patent Claim 11	IEEE 802.3

		
		TABLE I
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		”)
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		As another example, Solomon discloses using a subset of bits to encode for the selected physical port, i.e., selecting the first and second physical links responsively to the modulo, involving specific byte lengths of the frame information, i.e., responsively to respective first and second subsets of bits in a binary representation of the modulo.
		Solomon at [0054] (“Having selected a physical port, RSVP-TE proces-sor 30 of switch A now generates a suitable MPLS label, at a label generation step 64. The preceding node upstream of switch A will subsequently attach this MPLS label to all MPLS packets transmitted through tunnel 28 to switch A. The label is assigned, in conjunction with the mapping function of mapper 34, so as to ensure that all MPLS packets carrying this label are switched through the physical port that was selected for this tunnel at step 62. For this purpose, RSVP-TE processor 30 of switch A dedicates a sub-set of the bits of MPLS label 52 to encode the serial number of

No.	’740 Patent Claim 11	IEEE 802.3
		the selected physical port. For example, the four least-signifi-cant bits of MPLS label 52 may be used for encoding the selected port number. This configuration is suitable for representing LAG groups having up to 16 physical ports (N<16). The remaining bits of MPLS label 52 may be chosen at random or using any suitable method known in the art.”) Solomon at [0056] (“Mapper 34 of switch A maps the received packets belonging to tunnel 28 to the selected physical Ethernet port at a mapping step 70. For this purpose, mapper 34 extracts the MPLS label from each received packet and decodes the selected physical port number from the dedicated sub-set of bits, such as the four LSB, as described in step 64 above. The decoded value is used for mapping the packet to the selected physical port, which was allocated by the CAC processor at step 62 above. In the four-bit example described above, the mapping function may be written explicitly as: Selected port number=((MPLS label) and (0x0000F)), wherein "and" denotes the "bitwise and" operator.”)

No.	’740 Patent Claim 12	IEEE 802.3
12	The method according to claim 1, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source	IEEE 802.3 discloses the method according to claim 1, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port. For example, IEEE 802.3 discloses data frames with frame information including but not limited to source MAC address, destination MAC address, reception port, destination address, address and protocol identification information, combinations thereof, etc. *See supra* Claim 1. IEEE 802.3 at 1549

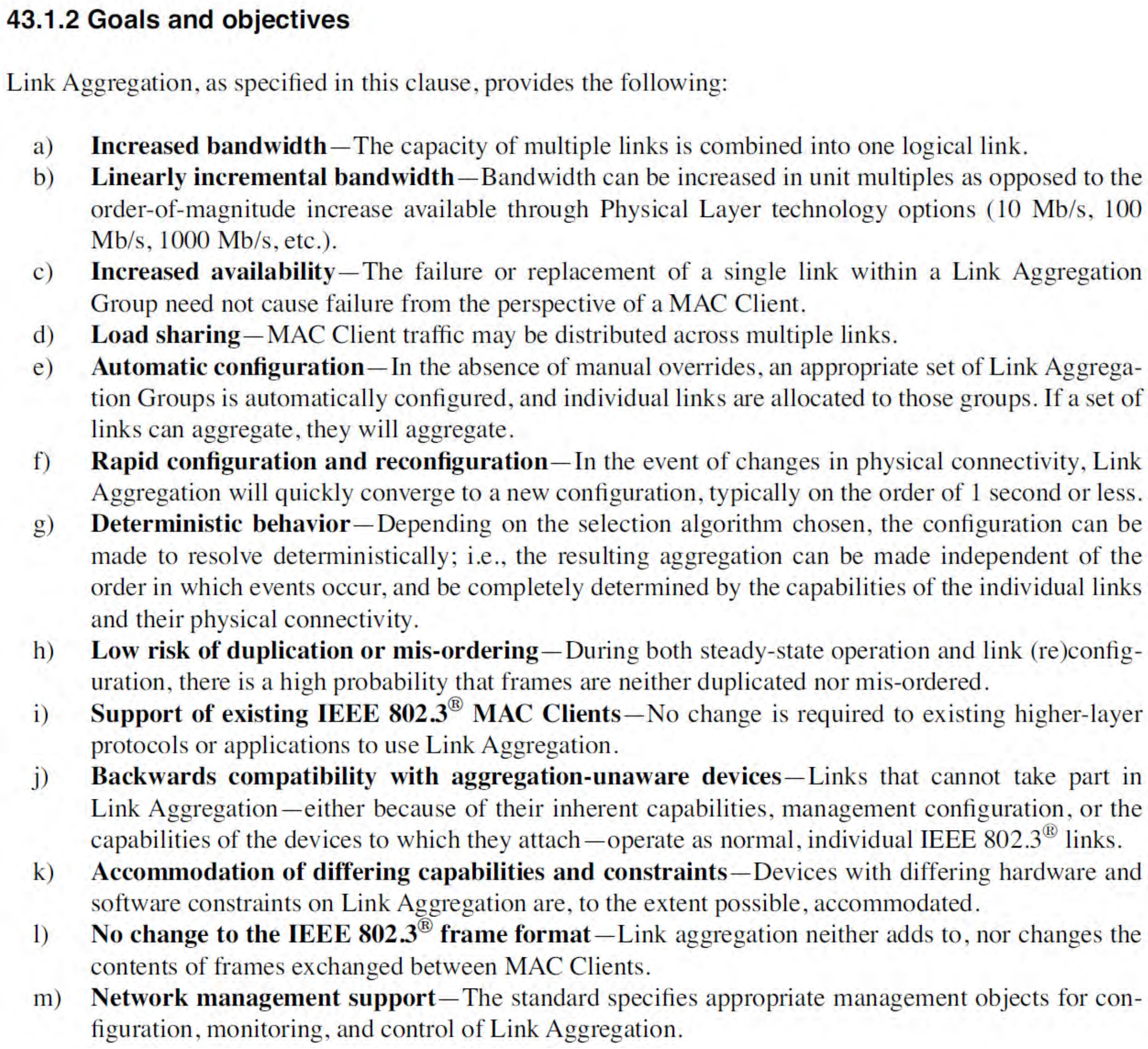
No.	’740 Patent Claim 12	IEEE 802.3
	medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port.	 **43A.2 Port selection** A distribution algorithm selects the port used to transmit a given frame, such that the same port will be chosen for subsequent frames that form part of the same conversation. The algorithm may make use of information carried in the frame in order to make its decision, in combination with other information associated with the frame, such as its reception port in the case of a MAC Bridge. The algorithm may assign one or more conversations to the same port, however, it must not allocate some of the frames of a given conversation to one port and the remainder to different ports. The information used to assign conversations to ports could include the following: a) Source MAC address b) Destination MAC address c) The reception port d) The type of destination address (individual or group MAC address) e) Ethernet Length/Type value (i.e., protocol identification) f) Higher layer protocol information (e.g., addressing and protocol identification information from the LLC sublayer or above) g) Combinations of the above One simple approach applies a hash function to the selected information to generate a port number. This produces a deterministic (i.e., history independent) port selection across a given number of ports in an aggregation. However, as it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models, it might be appropriate to weight the port selection in favor of ports that are carrying lower traffic levels. In more sophisticated approaches, load balancing is dynamic; i.e., the port selected for a given set of conversations changes over time, independent of any changes that take place in the membership of the aggregation.

No.	’740 Patent Claim 13	IEEE 802.3
13[preamble]	A method for communication, comprising:	IEEE 802.3 discloses a method for communication. *See supra at* 1[preamble].

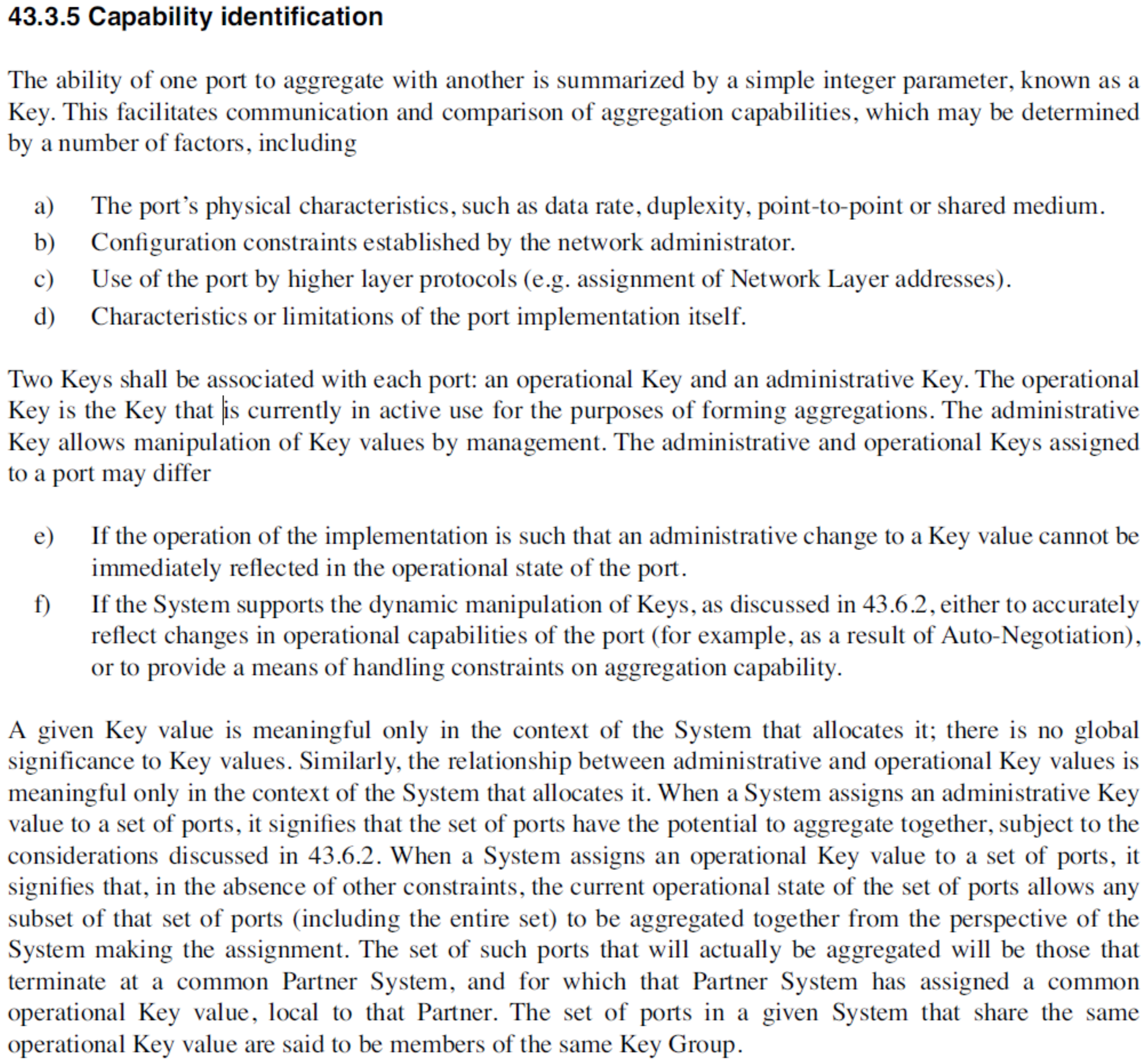
No.	’740 Patent Claim 13	IEEE 802.3
13[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	IEEE 802.3 discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].
13[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	IEEE 802.3 discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra at* 1[c].
13[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	IEEE 802.3 discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra at* 1[e].
13[d]	selecting, in a single computation based on at	IEEE 802.3 discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. *See supra at* 1[f].

No.	’740 Patent Claim 13	IEEE 802.3
	least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	
13[e]	sending the data frame over the selected first and second physical links,	IEEE 802.3 discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].
13[f]	coupling the network node to the one or more interface modules and	IEEE 802.3 discloses coupling the network node to the one or more interface modules. *See supra at* 1[a].
13[g]	coupling each of the one or more interface modules to the communication network comprising	IEEE 802.3 discloses coupling each of the one or more interface modules to the communication network. *See supra at* 1[c].
13[h]	specifying bandwidth requirements	IEEE 802.3 discloses specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node.

No.	’740 Patent Claim 13	IEEE 802.3
	comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node, and	For example, IEEE 802.3 discloses specific bandwidth and data rate capabilities of the links coupling the aggregator to the network. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the IEEE 802.3 is found to not meet this limitation, specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node would have been obvious to a person having ordinary skill in the art, as explained below. IEEE 802.3 at 1465

No.	’740 Patent Claim 13	IEEE 802.3
		 **43.1.2 Goals and objectives** Link Aggregation, as specified in this clause, provides the following: a) **Increased bandwidth**—The capacity of multiple links is combined into one logical link. b) **Linearly incremental bandwidth**—Bandwidth can be increased in unit multiples as opposed to the order-of-magnitude increase available through Physical Layer technology options (10 Mb/s, 100 Mb/s, 1000 Mb/s, etc.). c) **Increased availability**—The failure or replacement of a single link within a Link Aggregation Group need not cause failure from the perspective of a MAC Client. d) **Load sharing**—MAC Client traffic may be distributed across multiple links. e) **Automatic configuration**—In the absence of manual overrides, an appropriate set of Link Aggregation Groups is automatically configured, and individual links are allocated to those groups. If a set of links can aggregate, they will aggregate. f) **Rapid configuration and reconfiguration**—In the event of changes in physical connectivity, Link Aggregation will quickly converge to a new configuration, typically on the order of 1 second or less. g) **Deterministic behavior**—Depending on the selection algorithm chosen, the configuration can be made to resolve deterministically; i.e., the resulting aggregation can be made independent of the order in which events occur, and be completely determined by the capabilities of the individual links and their physical connectivity. h) **Low risk of duplication or mis-ordering**—During both steady-state operation and link (re)configuration, there is a high probability that frames are neither duplicated nor mis-ordered. i) **Support of existing IEEE 802.3® MAC Clients**—No change is required to existing higher-layer protocols or applications to use Link Aggregation. j) **Backwards compatibility with aggregation-unaware devices**—Links that cannot take part in Link Aggregation—either because of their inherent capabilities, management configuration, or the capabilities of the devices to which they attach—operate as normal, individual IEEE 802.3® links. k) **Accommodation of differing capabilities and constraints**—Devices with differing hardware and software constraints on Link Aggregation are, to the extent possible, accommodated. l) **No change to the IEEE 802.3® frame format**—Link aggregation neither adds to, nor changes the contents of frames exchanged between MAC Clients. m) **Network management support**—The standard specifies appropriate management objects for configuration, monitoring, and control of Link Aggregation.

No.	’740 Patent Claim 13	IEEE 802.3
		IEEE 802.3 at 1480



43.3.5 Capability identification

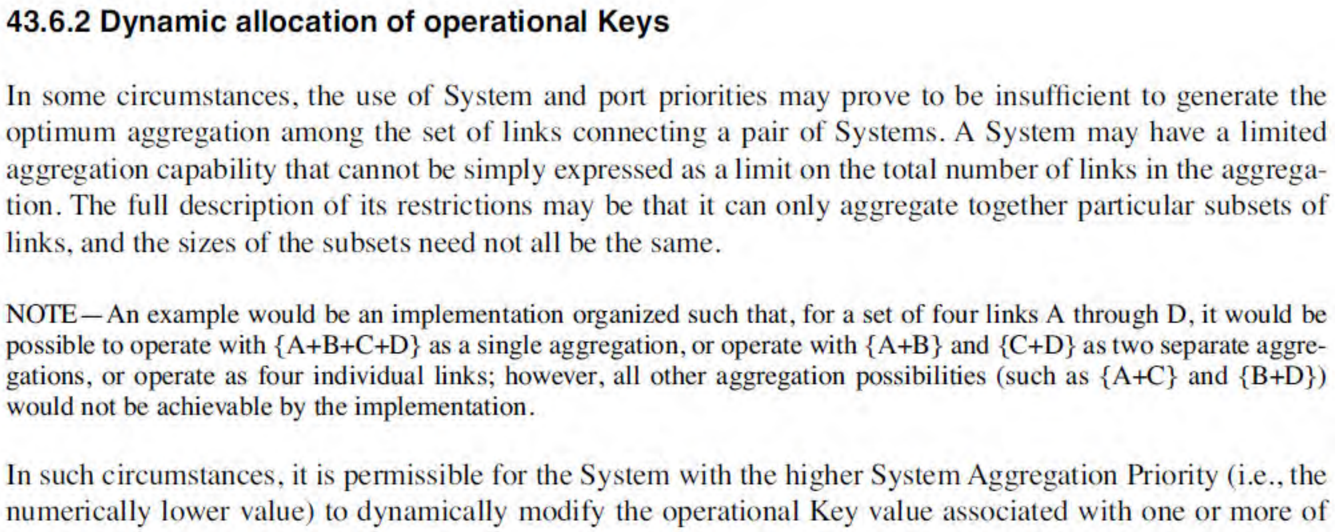
The ability of one port to aggregate with another is summarized by a simple integer parameter, known as a Key. This facilitates communication and comparison of aggregation capabilities, which may be determined by a number of factors, including

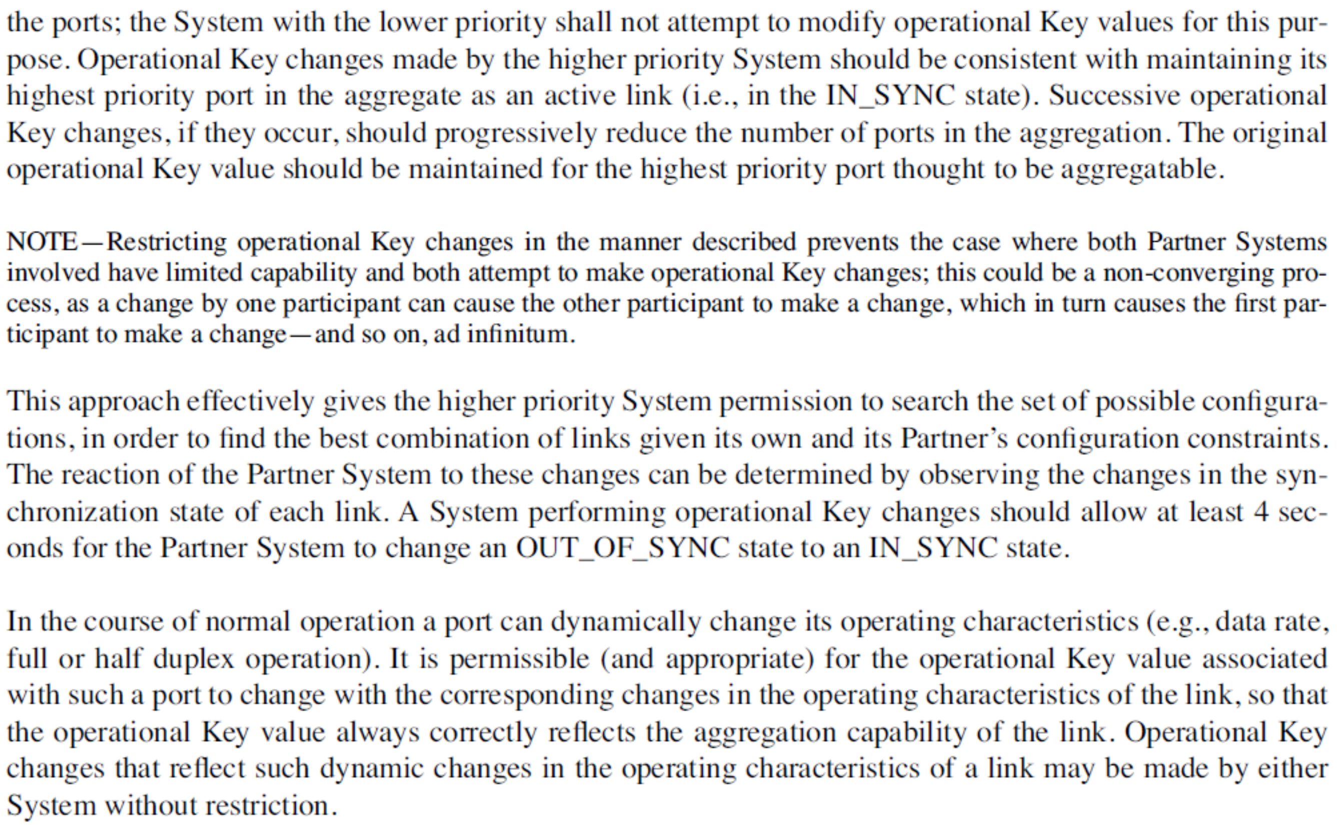
a) The port's physical characteristics, such as data rate, duplexity, point-to-point or shared medium.
b) Configuration constraints established by the network administrator.
c) Use of the port by higher layer protocols (e.g. assignment of Network Layer addresses).
d) Characteristics or limitations of the port implementation itself.

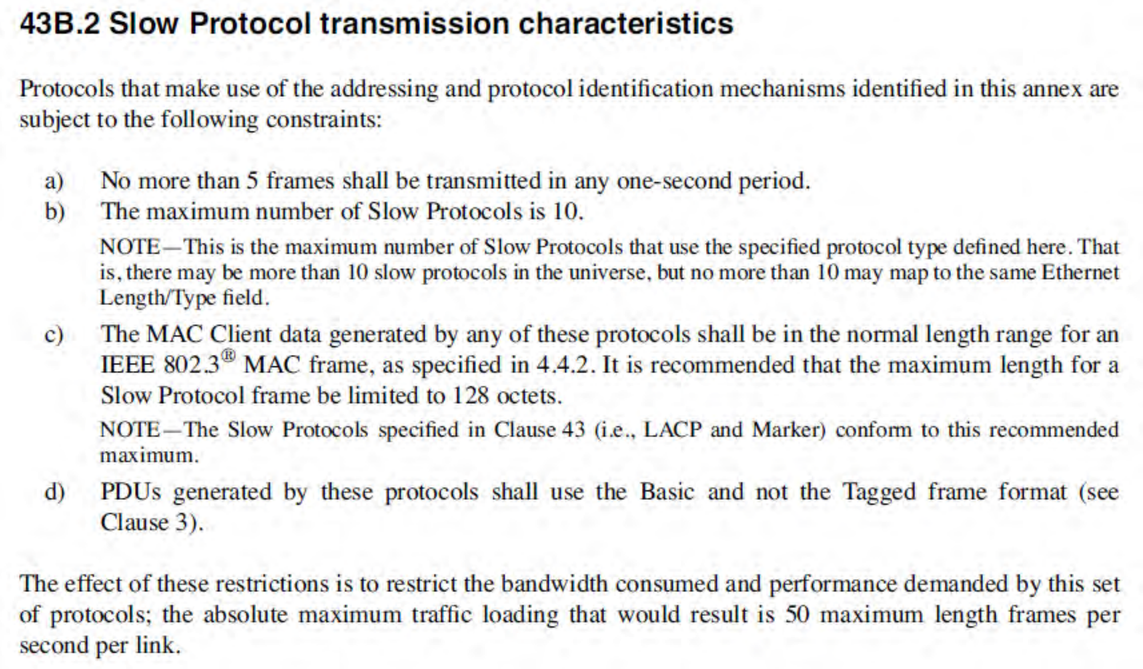
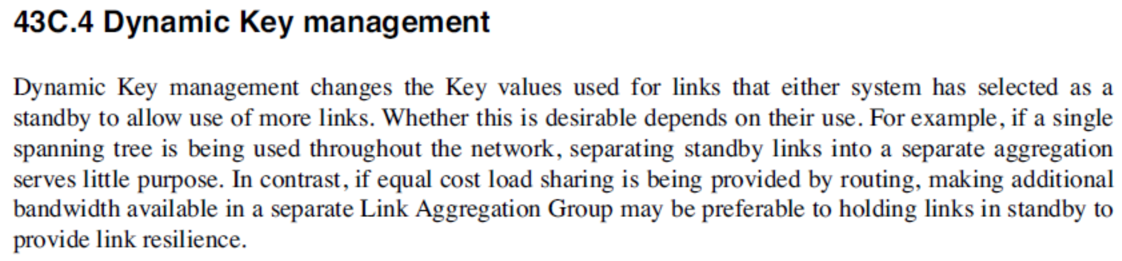
Two Keys shall be associated with each port: an operational Key and an administrative Key. The operational Key is the Key that is currently in active use for the purposes of forming aggregations. The administrative Key allows manipulation of Key values by management. The administrative and operational Keys assigned to a port may differ

e) If the operation of the implementation is such that an administrative change to a Key value cannot be immediately reflected in the operational state of the port.
f) If the System supports the dynamic manipulation of Keys, as discussed in 43.6.2, either to accurately reflect changes in operational capabilities of the port (for example, as a result of Auto-Negotiation), or to provide a means of handling constraints on aggregation capability.

A given Key value is meaningful only in the context of the System that allocates it; there is no global significance to Key values. Similarly, the relationship between administrative and operational Key values is meaningful only in the context of the System that allocates it. When a System assigns an administrative Key value to a set of ports, it signifies that the set of ports have the potential to aggregate together, subject to the considerations discussed in 43.6.2. When a System assigns an operational Key value to a set of ports, it signifies that, in the absence of other constraints, the current operational state of the set of ports allows any subset of that set of ports (including the entire set) to be aggregated together from the perspective of the System making the assignment. The set of such ports that will actually be aggregated will be those that terminate at a common Partner System, and for which that Partner System has assigned a common operational Key value, local to that Partner. The set of ports in a given System that share the same operational Key value are said to be members of the same Key Group.

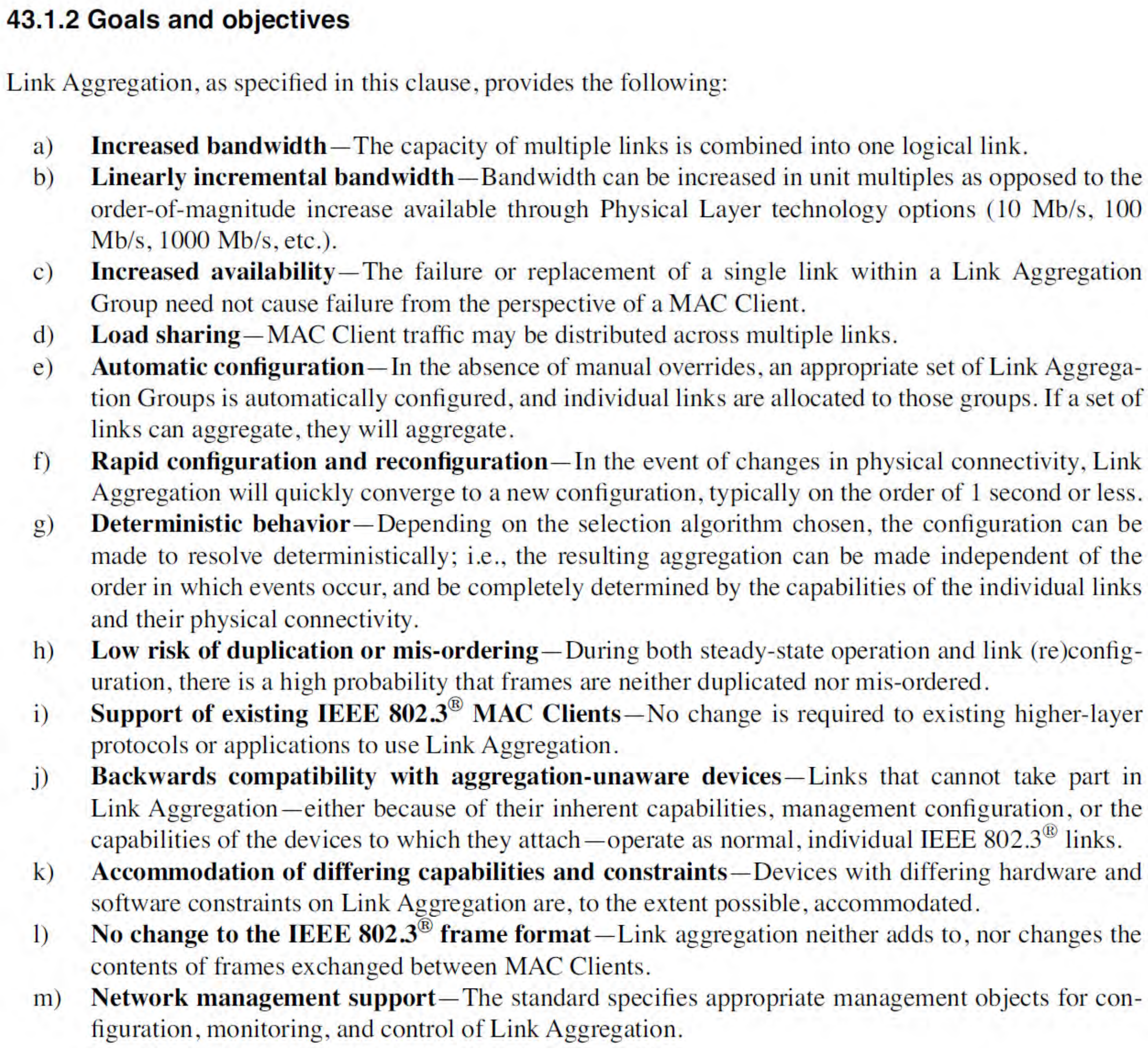
No.	’740 Patent Claim 13	IEEE 802.3
		IEEE 802.3 at 1517-18  43.6.2 Dynamic allocation of operational Keys In some circumstances, the use of System and port priorities may prove to be insufficient to generate the optimum aggregation among the set of links connecting a pair of Systems. A System may have a limited aggregation capability that cannot be simply expressed as a limit on the total number of links in the aggregation. The full description of its restrictions may be that it can only aggregate together particular subsets of links, and the sizes of the subsets need not all be the same. NOTE—An example would be an implementation organized such that, for a set of four links A through D, it would be possible to operate with {A+B+C+D} as a single aggregation, or operate with {A+B} and {C+D} as two separate aggregations, or operate as four individual links; however, all other aggregation possibilities (such as {A+C} and {B+D}) would not be achievable by the implementation. In such circumstances, it is permissible for the System with the higher System Aggregation Priority (i.e., the numerically lower value) to dynamically modify the operational Key value associated with one or more of

No.	’740 Patent Claim 13	IEEE 802.3
		 the ports; the System with the lower priority shall not attempt to modify operational Key values for this purpose. Operational Key changes made by the higher priority System should be consistent with maintaining its highest priority port in the aggregate as an active link (i.e., in the IN_SYNC state). Successive operational Key changes, if they occur, should progressively reduce the number of ports in the aggregation. The original operational Key value should be maintained for the highest priority port thought to be aggregatable. NOTE—Restricting operational Key changes in the manner described prevents the case where both Partner Systems involved have limited capability and both attempt to make operational Key changes; this could be a non-converging process, as a change by one participant can cause the other participant to make a change, which in turn causes the first participant to make a change—and so on, ad infinitum. This approach effectively gives the higher priority System permission to search the set of possible configurations, in order to find the best combination of links given its own and its Partner’s configuration constraints. The reaction of the Partner System to these changes can be determined by observing the changes in the synchronization state of each link. A System performing operational Key changes should allow at least 4 seconds for the Partner System to change an OUT_OF_SYNC state to an IN_SYNC state. In the course of normal operation a port can dynamically change its operating characteristics (e.g., data rate, full or half duplex operation). It is permissible (and appropriate) for the operational Key value associated with such a port to change with the corresponding changes in the operating characteristics of the link, so that the operational Key value always correctly reflects the aggregation capability of the link. Operational Key changes that reflect such dynamic changes in the operating characteristics of a link may be made by either System without restriction. IEEE 802.3 at 1552

No.	’740 Patent Claim 13	IEEE 802.3
		 **43B.2 Slow Protocol transmission characteristics** Protocols that make use of the addressing and protocol identification mechanisms identified in this annex are subject to the following constraints: a) No more than 5 frames shall be transmitted in any one-second period. b) The maximum number of Slow Protocols is 10. NOTE—This is the maximum number of Slow Protocols that use the specified protocol type defined here. That is, there may be more than 10 slow protocols in the universe, but no more than 10 may map to the same Ethernet Length/Type field. c) The MAC Client data generated by any of these protocols shall be in the normal length range for an IEEE 802.3® MAC frame, as specified in 4.4.2. It is recommended that the maximum length for a Slow Protocol frame be limited to 128 octets. NOTE—The Slow Protocols specified in Clause 43 (i.e., LACP and Marker) conform to this recommended maximum. d) PDUs generated by these protocols shall use the Basic and not the Tagged frame format (see Clause 3). The effect of these restrictions is to restrict the bandwidth consumed and performance demanded by this set of protocols; the absolute maximum traffic loading that would result is 50 maximum length frames per second per link. IEEE 802.3 at 1558  **43C.4 Dynamic Key management** Dynamic Key management changes the Key values used for links that either system has selected as a standby to allow use of more links. Whether this is desirable depends on their use. For example, if a single spanning tree is being used throughout the network, separating standby links into a separate aggregation serves little purpose. In contrast, if equal cost load sharing is being provided by routing, making additional bandwidth available in a separate Link Aggregation Group may be preferable to holding links in standby to provide link resilience. Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, IEEE 802.3 in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination

No.	’740 Patent Claim 13	IEEE 802.3
		with (2) each (individually, as well as one or more together) of the references identified in element 13[h] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses specifying certain committed, excess, and guaranteed bandwidths, including CIR, EIR, and PIR, respectively. Bruckman at [0013] (“Service level agreements between network service providers and customers commonly specify a certain com-mitted bandwidth, or committed information rate (CIR), which the service provider guarantees to provide to the customer at all times, regardless of bandwidth stress on the network. Additionally or alternatively, the agreement may specify an excess bandwidth, which is available to the customer when network traffic permits. The excess band-width is typically used by customers for lower-priority services, which do not require committed bandwidth. The network service provider may guarantee the customer a certain minimum excess bandwidth, or excess information rate (EIR), in order to avoid starvation of such services in case of bandwidth stress. In general, the bandwidth guaran-teed by a service provider, referred to as the peak informa-tion rate (PIR), may include either CIR, or EIR, or both CIR and EIR (in which case PIR=CIR+EIR). The term "guaran-teed bandwidth," as used in the context of the present patent application and in the claims, includes all these types of guaranteed bandwidth.”) As another example, Solomon discloses a service property of a guaranteed bandwidth, sometimes denoted as CIR-Committed Information Rate and PIR-Peak Information Rate. Solomon at [0023] (“In another embodiment, establishing the path includes receiving an indication of a requested service property of the flow, and selecting the port includes assign-ing the port to the flow so as to comply with the requested service property. In a disclosed embodiment, the requested service property includes at least one of a guaranteed bandwidth, a peak bandwidth and a class-of-service. Addi-tionally or alternatively, assigning the port includes selecting the port having a maximum available bandwidth out of the plurality of aggregated ports. Further additionally or alter-natively, assigning the port includes selecting the port hav-ing a minimum available bandwidth out of the plurality of aggregated ports, which is still greater than or equal to the guaranteed bandwidth.”)

No.	’740 Patent Claim 13	IEEE 802.3
		Solomon at [0050] (“The method of FIG. 3 begins when the preceding node asks to establish a part of tunnel 28 (comprising one or more hops) for sending MPLS packets to MPLS/LAG switch 26 A. The preceding node requests and then receives the MPLS label, which it will subsequently attach to all packets that are sent to MPLS/LAG switch 26 labeledA. The preceding node sends downstream an RSVP-TE PATH mes-sage augmented with a LABEL_REQUEST object, as defined by RSVP-TE, to MPLS/LAG switch A, at a label requesting step 60. The PATH message typically comprises information regarding service properties that are requested for tunnel 28. The service properties may comprise a guar-anteed bandwidth (sometimes denoted CIR-Committed Information Rate) and a peak bandwidth (sometimes denoted PIR-Peak Information Rate), as well as a requested CoS (Class of Service-a measure of packet priority).”)
13[i]	allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements.	IEEE 802.3 discloses allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements. For examples, IEEE 802.3 discloses dynamic reallocation of data traffic or conversations over different links based on bandwidth capabilities. IEEE 802.3 at 1465

No.	’740 Patent Claim 13	IEEE 802.3
		 **43.1.2 Goals and objectives** Link Aggregation, as specified in this clause, provides the following: a) **Increased bandwidth**—The capacity of multiple links is combined into one logical link. b) **Linearly incremental bandwidth**—Bandwidth can be increased in unit multiples as opposed to the order-of-magnitude increase available through Physical Layer technology options (10 Mb/s, 100 Mb/s, 1000 Mb/s, etc.). c) **Increased availability**—The failure or replacement of a single link within a Link Aggregation Group need not cause failure from the perspective of a MAC Client. d) **Load sharing**—MAC Client traffic may be distributed across multiple links. e) **Automatic configuration**—In the absence of manual overrides, an appropriate set of Link Aggregation Groups is automatically configured, and individual links are allocated to those groups. If a set of links can aggregate, they will aggregate. f) **Rapid configuration and reconfiguration**—In the event of changes in physical connectivity, Link Aggregation will quickly converge to a new configuration, typically on the order of 1 second or less. g) **Deterministic behavior**—Depending on the selection algorithm chosen, the configuration can be made to resolve deterministically; i.e., the resulting aggregation can be made independent of the order in which events occur, and be completely determined by the capabilities of the individual links and their physical connectivity. h) **Low risk of duplication or mis-ordering**—During both steady-state operation and link (re)configuration, there is a high probability that frames are neither duplicated nor mis-ordered. i) **Support of existing IEEE 802.3® MAC Clients**—No change is required to existing higher-layer protocols or applications to use Link Aggregation. j) **Backwards compatibility with aggregation-unaware devices**—Links that cannot take part in Link Aggregation—either because of their inherent capabilities, management configuration, or the capabilities of the devices to which they attach—operate as normal, individual IEEE 802.3® links. k) **Accommodation of differing capabilities and constraints**—Devices with differing hardware and software constraints on Link Aggregation are, to the extent possible, accommodated. l) **No change to the IEEE 802.3® frame format**—Link aggregation neither adds to, nor changes the contents of frames exchanged between MAC Clients. m) **Network management support**—The standard specifies appropriate management objects for configuration, monitoring, and control of Link Aggregation.

No.	’740 Patent Claim 13	IEEE 802.3
		IEEE 802.3 at 1480

43.3.5 Capability identification

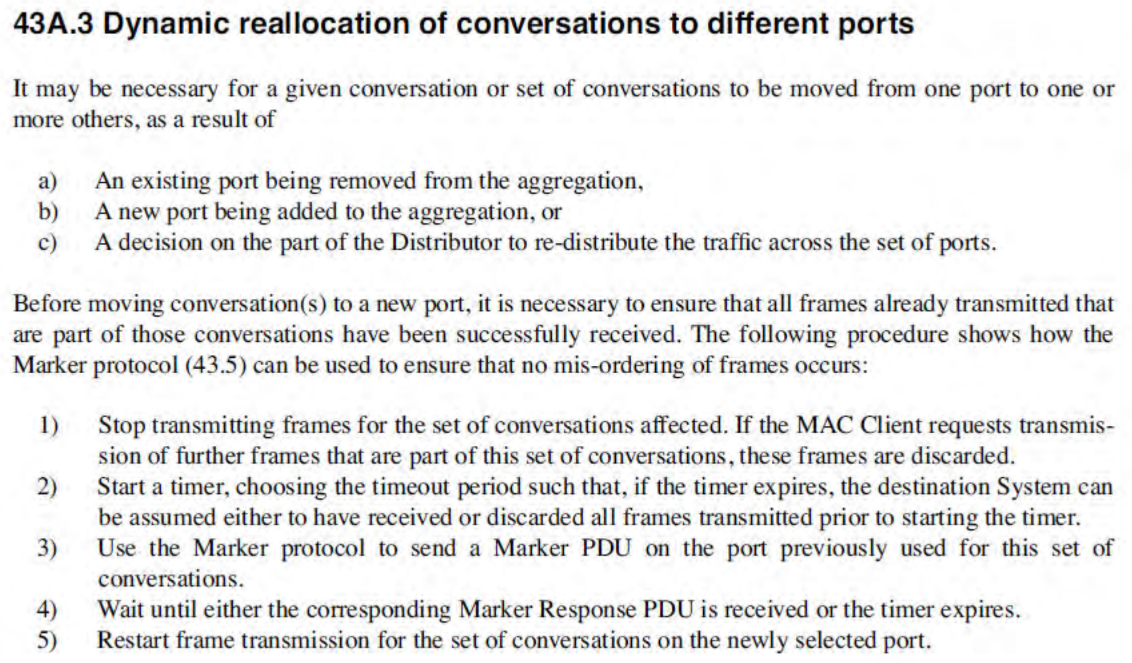
The ability of one port to aggregate with another is summarized by a simple integer parameter, known as a Key. This facilitates communication and comparison of aggregation capabilities, which may be determined by a number of factors, including

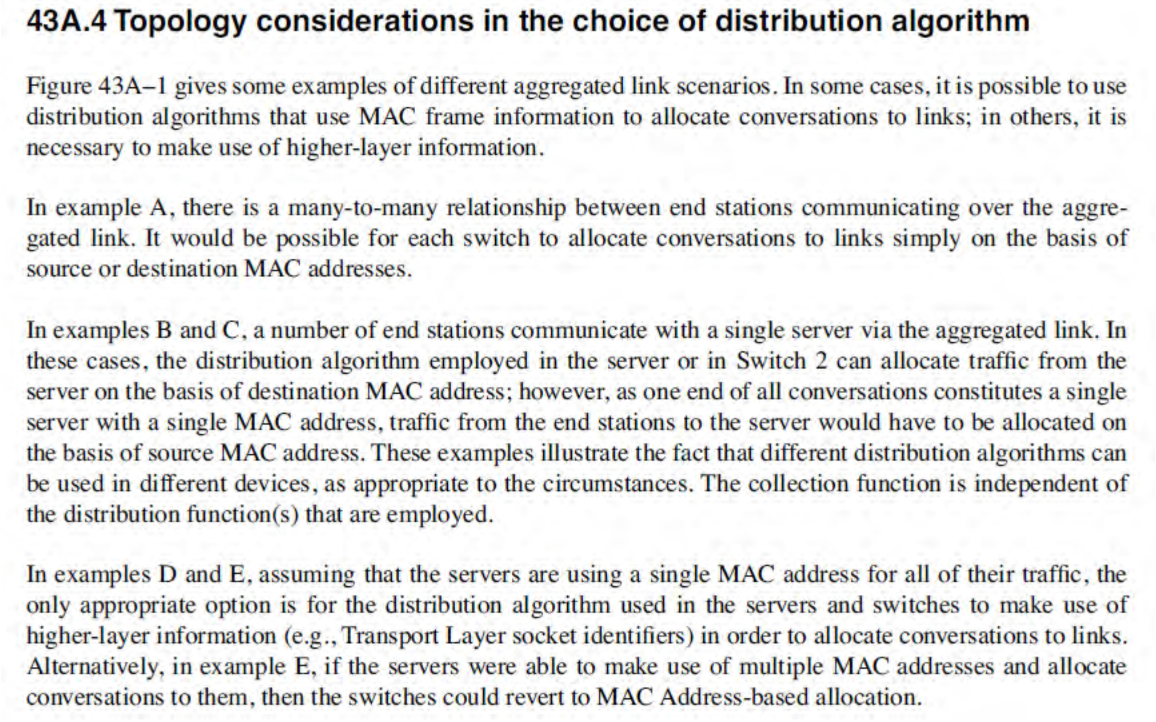
a) The port’s physical characteristics, such as data rate, duplexity, point-to-point or shared medium.
b) Configuration constraints established by the network administrator.
c) Use of the port by higher layer protocols (e.g. assignment of Network Layer addresses).
d) Characteristics or limitations of the port implementation itself.

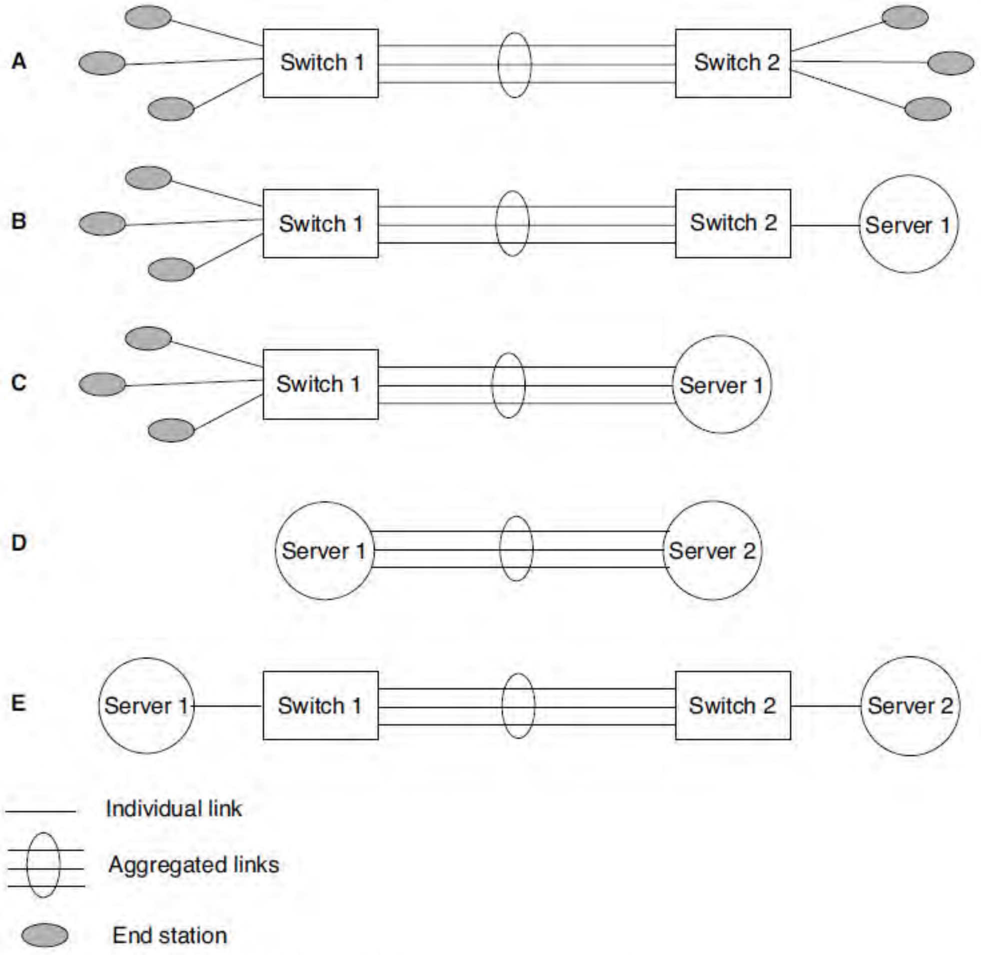
Two Keys shall be associated with each port: an operational Key and an administrative Key. The operational Key is the Key that is currently in active use for the purposes of forming aggregations. The administrative Key allows manipulation of Key values by management. The administrative and operational Keys assigned to a port may differ

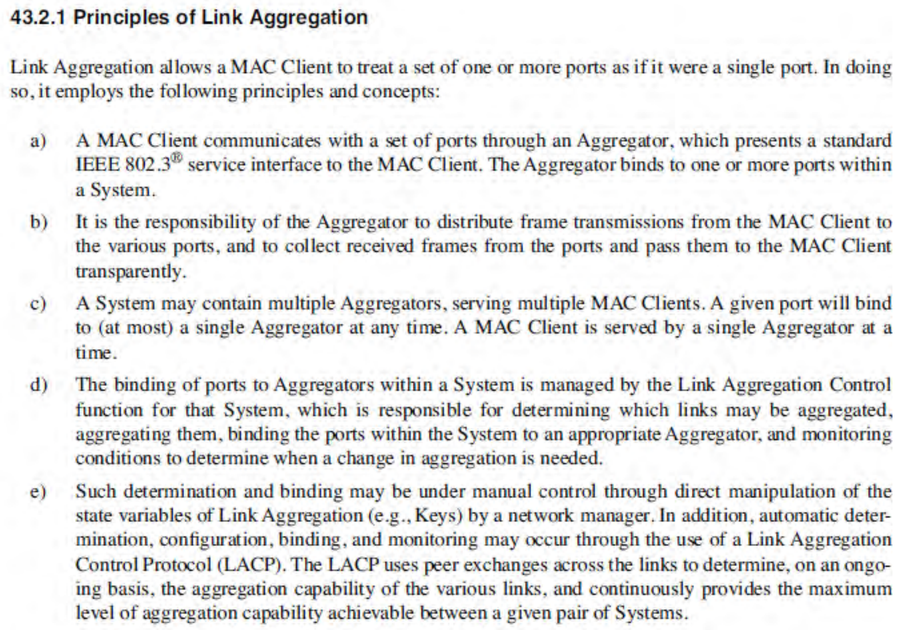
e) If the operation of the implementation is such that an administrative change to a Key value cannot be immediately reflected in the operational state of the port.
f) If the System supports the dynamic manipulation of Keys, as discussed in 43.6.2, either to accurately reflect changes in operational capabilities of the port (for example, as a result of Auto-Negotiation), or to provide a means of handling constraints on aggregation capability.

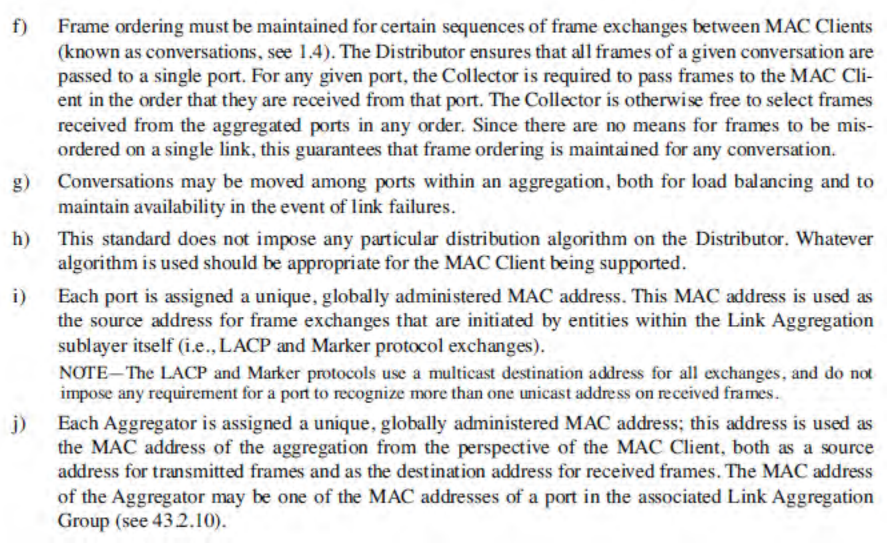
A given Key value is meaningful only in the context of the System that allocates it; there is no global significance to Key values. Similarly, the relationship between administrative and operational Key values is meaningful only in the context of the System that allocates it. When a System assigns an administrative Key value to a set of ports, it signifies that the set of ports have the potential to aggregate together, subject to the considerations discussed in 43.6.2. When a System assigns an operational Key value to a set of ports, it signifies that, in the absence of other constraints, the current operational state of the set of ports allows any subset of that set of ports (including the entire set) to be aggregated together from the perspective of the System making the assignment. The set of such ports that will actually be aggregated will be those that terminate at a common Partner System, and for which that Partner System has assigned a common operational Key value, local to that Partner. The set of ports in a given System that share the same operational Key value are said to be members of the same Key Group.

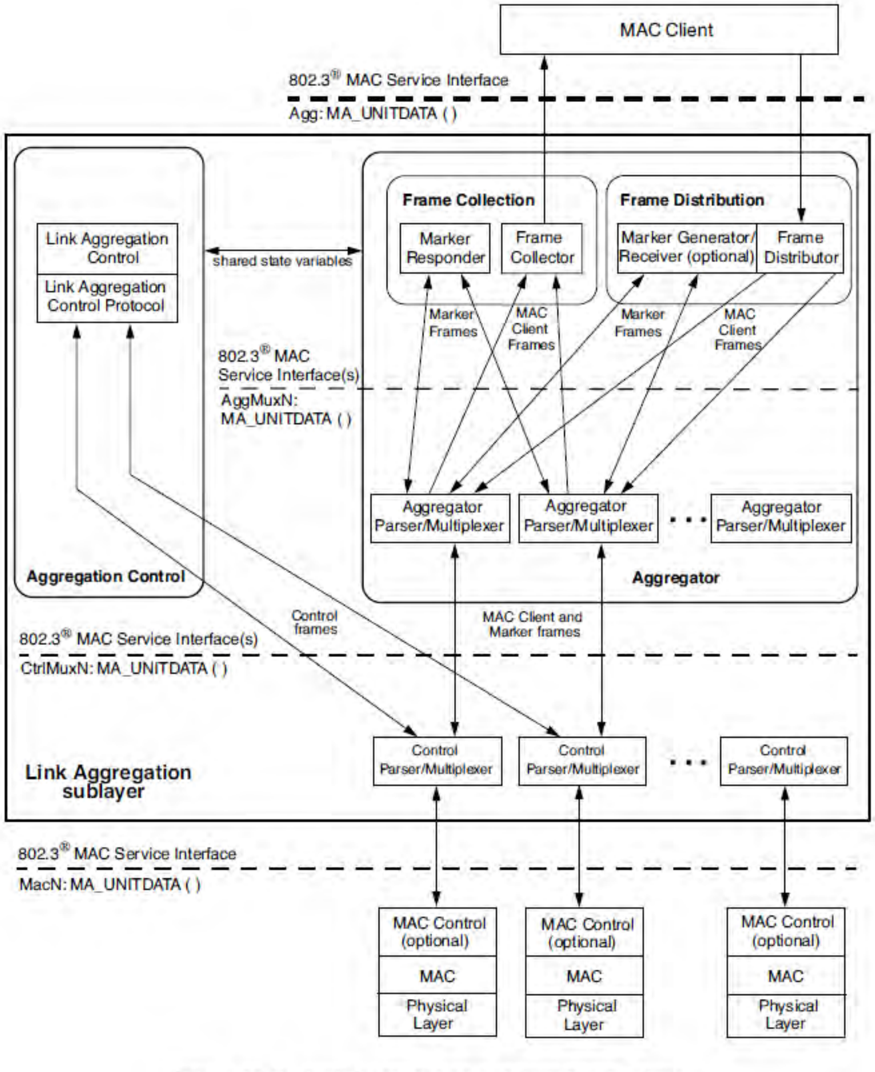
No.	’740 Patent Claim 13	IEEE 802.3
		IEEE 802.3 at 1549  43A.3 Dynamic reallocation of conversations to different ports It may be necessary for a given conversation or set of conversations to be moved from one port to one or more others, as a result of a) An existing port being removed from the aggregation, b) A new port being added to the aggregation, or c) A decision on the part of the Distributor to re-distribute the traffic across the set of ports. Before moving conversation(s) to a new port, it is necessary to ensure that all frames already transmitted that are part of those conversations have been successfully received. The following procedure shows how the Marker protocol (43.5) can be used to ensure that no mis-ordering of frames occurs: 1) Stop transmitting frames for the set of conversations affected. If the MAC Client requests transmission of further frames that are part of this set of conversations, these frames are discarded. 2) Start a timer, choosing the timeout period such that, if the timer expires, the destination System can be assumed either to have received or discarded all frames transmitted prior to starting the timer. 3) Use the Marker protocol to send a Marker PDU on the port previously used for this set of conversations. 4) Wait until either the corresponding Marker Response PDU is received or the timer expires. 5) Restart frame transmission for the set of conversations on the newly selected port. IEEE at 802.3 at 1550

No.	’740 Patent Claim 13	IEEE 802.3
		 43A.4 Topology considerations in the choice of distribution algorithm Figure 43A–1 gives some examples of different aggregated link scenarios. In some cases, it is possible to use distribution algorithms that use MAC frame information to allocate conversations to links; in others, it is necessary to make use of higher-layer information. In example A, there is a many-to-many relationship between end stations communicating over the aggregated link. It would be possible for each switch to allocate conversations to links simply on the basis of source or destination MAC addresses. In examples B and C, a number of end stations communicate with a single server via the aggregated link. In these cases, the distribution algorithm employed in the server or in Switch 2 can allocate traffic from the server on the basis of destination MAC address; however, as one end of all conversations constitutes a single server with a single MAC address, traffic from the end stations to the server would have to be allocated on the basis of source MAC address. These examples illustrate the fact that different distribution algorithms can be used in different devices, as appropriate to the circumstances. The collection function is independent of the distribution function(s) that are employed. In examples D and E, assuming that the servers are using a single MAC address for all of their traffic, the only appropriate option is for the distribution algorithm used in the servers and switches to make use of higher-layer information (e.g., Transport Layer socket identifiers) in order to allocate conversations to links. Alternatively, in example E, if the servers were able to make use of multiple MAC addresses and allocate conversations to them, then the switches could revert to MAC Address-based allocation. IEEE 802.3 at Figure 43A-1

No.	’740 Patent Claim 13	IEEE 802.3
		 A Switch 1 Switch 2 B Switch 1 Switch 2 Server 1 C Switch 1 Server 1 D Server 1 Server 2 E Server 1 Switch 1 Switch 2 Server 2 Individual link Aggregated links End station **Figure 43A–1—Link aggregation topology examples**

No.	’740 Patent Claim 14	IEEE 802.3
14[preamble]	A method for connecting user ports to a communication network, comprising:	IEEE 802.3 discloses a method for connecting user ports to a communication network. For example, IEEE 802.3 discloses a manner for connecting a set of ports to a network. IEEE 802.3 at 1468-69  **43.2.1 Principles of Link Aggregation** Link Aggregation allows a MAC Client to treat a set of one or more ports as if it were a single port. In doing so, it employs the following principles and concepts: a) A MAC Client communicates with a set of ports through an Aggregator, which presents a standard IEEE 802.3® service interface to the MAC Client. The Aggregator binds to one or more ports within a System. b) It is the responsibility of the Aggregator to distribute frame transmissions from the MAC Client to the various ports, and to collect received frames from the ports and pass them to the MAC Client transparently. c) A System may contain multiple Aggregators, serving multiple MAC Clients. A given port will bind to (at most) a single Aggregator at any time. A MAC Client is served by a single Aggregator at a time. d) The binding of ports to Aggregators within a System is managed by the Link Aggregation Control function for that System, which is responsible for determining which links may be aggregated, aggregating them, binding the ports within the System to an appropriate Aggregator, and monitoring conditions to determine when a change in aggregation is needed. e) Such determination and binding may be under manual control through direct manipulation of the state variables of Link Aggregation (e.g., Keys) by a network manager. In addition, automatic determination, configuration, binding, and monitoring may occur through the use of a Link Aggregation Control Protocol (LACP). The LACP uses peer exchanges across the links to determine, on an ongoing basis, the aggregation capability of the various links, and continuously provides the maximum level of aggregation capability achievable between a given pair of Systems.

No.	’740 Patent Claim 14	IEEE 802.3
		 f) Frame ordering must be maintained for certain sequences of frame exchanges between MAC Clients (known as conversations, see 1.4). The Distributor ensures that all frames of a given conversation are passed to a single port. For any given port, the Collector is required to pass frames to the MAC Client in the order that they are received from that port. The Collector is otherwise free to select frames received from the aggregated ports in any order. Since there are no means for frames to be misordered on a single link, this guarantees that frame ordering is maintained for any conversation. g) Conversations may be moved among ports within an aggregation, both for load balancing and to maintain availability in the event of link failures. h) This standard does not impose any particular distribution algorithm on the Distributor. Whatever algorithm is used should be appropriate for the MAC Client being supported. i) Each port is assigned a unique, globally administered MAC address. This MAC address is used as the source address for frame exchanges that are initiated by entities within the Link Aggregation sublayer itself (i.e., LACP and Marker protocol exchanges). NOTE—The LACP and Marker protocols use a multicast destination address for all exchanges, and do not impose any requirement for a port to recognize more than one unicast address on received frames. j) Each Aggregator is assigned a unique, globally administered MAC address; this address is used as the MAC address of the aggregation from the perspective of the MAC Client, both as a source address for transmitted frames and as the destination address for received frames. The MAC address of the Aggregator may be one of the MAC addresses of a port in the associated Link Aggregation Group (see 43.2.10). IEEE 802.3 at Figure 43-2

No.	’740 Patent Claim 14	IEEE 802.3
		 MAC Client 802.3® MAC Service Interface Agg: MA_UNITDATA ( ) Link Aggregation Control Link Aggregation Control Protocol shared state variables Frame Collection Marker Responder Frame Collector Frame Distribution Marker Generator/ Receiver (optional) Frame Distributor Marker Frames MAC Client Frames Marker Frames MAC Client Frames 802.3® MAC Service Interface(s) AggMuxN: MA_UNITDATA ( ) Aggregator Parser/Multiplexer Aggregator Parser/Multiplexer Aggregator Parser/Multiplexer Aggregation Control Aggregator Control frames MAC Client and Marker frames 802.3® MAC Service Interface(s) CtrlMuxN: MA_UNITDATA ( ) Control Parser/Multiplexer Control Parser/Multiplexer Control Parser/Multiplexer Link Aggregation sublayer 802.3® MAC Service Interface MacN: MA_UNITDATA ( ) MAC Control (optional) MAC Physical Layer MAC Control (optional) MAC Physical Layer MAC Control (optional) MAC Physical Layer **Figure 43–2—Link Aggregation sublayer block diagram**

No.	’740 Patent Claim 14	IEEE 802.3
14[a]	coupling the user ports to one or more user interface modules;	IEEE 802.3 discloses coupling the user ports to one or more user interface modules. *See supra at* 1[a].
14[b]	coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel,	IEEE 802.3 discloses coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel. *See supra at* 1[c], 3.
14[c]	at least one of said backplane traces being bi-directional and operative to communicate in both an upstream direction and a downstream direction;	IEEE 802.3 discloses at least one of said backplane traces being bi-directional and operative to communicate in both an upstream direction and a downstream direction. *See supra at* 14[b], 1[d].
14[d]	receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes;	IEEE 802.3 discloses receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes. *See supra at* 14[a], 1[e].
14[e]	for each data frame, selecting	IEEE 802.3 discloses for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces.

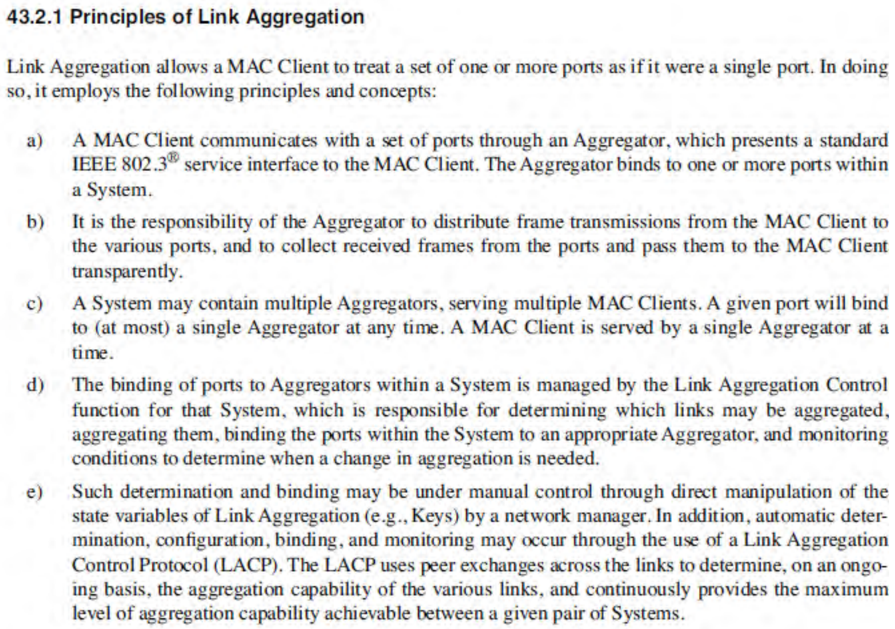
No.	’740 Patent Claim 14	IEEE 802.3
	responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces; and	*See supra at* 14[b], 1[f].
14[f]	sending the data frame over the selected backplane trace;	IEEE 802.3 discloses sending the data frame over the selected backplane trace. *See supra at* 14[e], 1[g].
14[g]	said sending comprising communicating along said at least one of said backplane traces.	IEEE 802.3 discloses said sending comprising communicating along said at least one of said backplane traces. *See supra at* 14[f], 1[h].

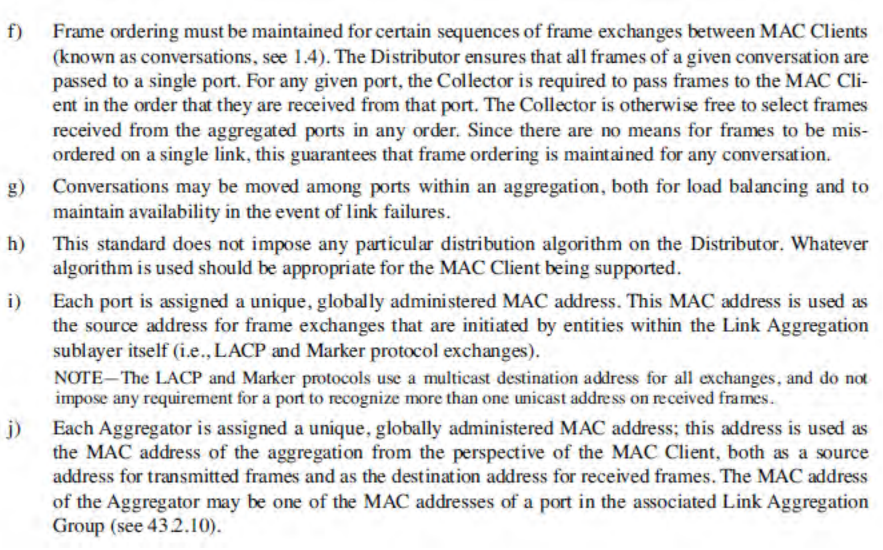
No.	’740 Patent Claim 15	IEEE 802.3
15[preamble]	A method for connecting user ports to a communication network, comprising:	IEEE 802.3 discloses a method for connecting user ports to a communication network. *See supra at* 14[preamble].
15[a]	coupling the user ports to one or more user interface modules;	IEEE 802.3 discloses coupling the user ports to one or more user interface modules. *See supra at* 14[a].
15[b]	coupling each user interface module to the communication network via a	IEEE 802.3 discloses coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel. *See supra at* 14[b].

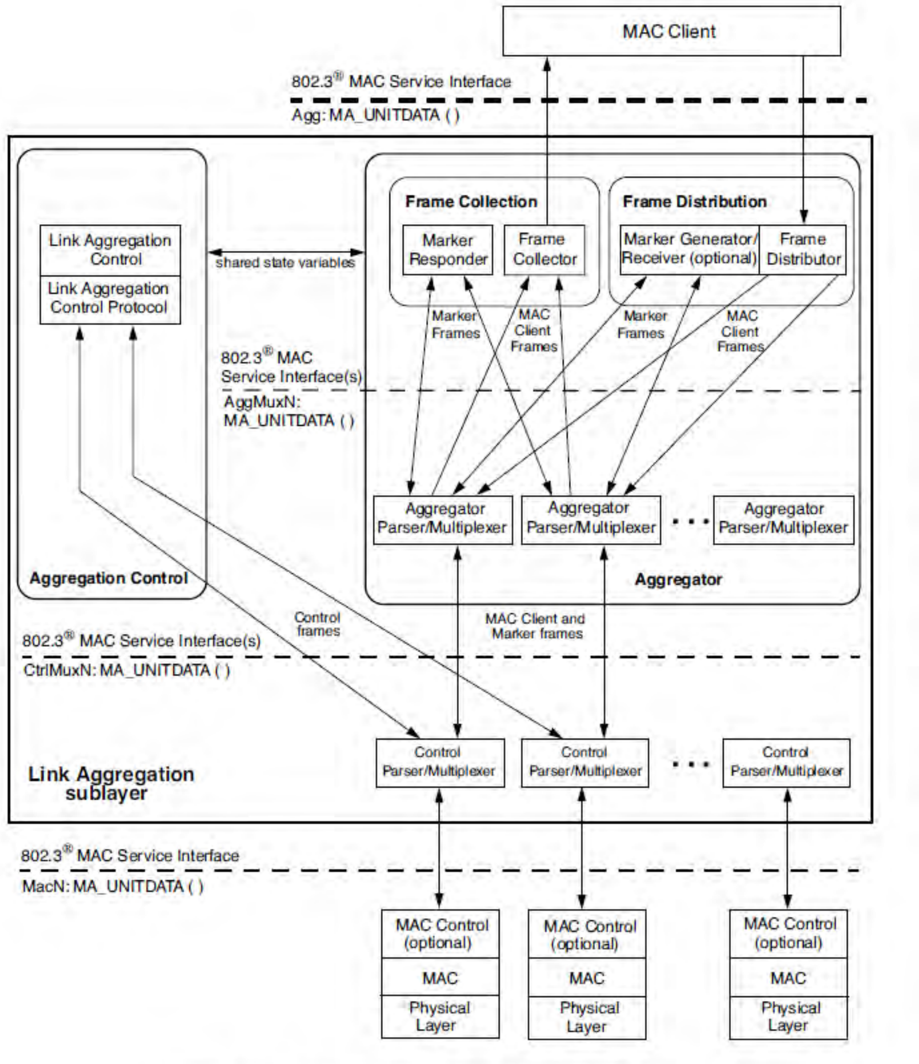
No.	’740 Patent Claim 15	IEEE 802.3
	backplane using two or more backplane traces arranged in parallel;	
15[c]	receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes;	IEEE 802.3 discloses receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes. *See supra at* 14[d].
15[d]	for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces; and	IEEE 802.3 discloses for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces. *See supra at* 14[e].
15[e]	sending the data frame over the selected backplane trace,	IEEE 802.3 discloses sending the data frame over the selected backplane trace. *See supra at* 14[f].
15[f]	at least some of the backplane traces being aggregated into an Ethernet link aggregation (LAG) group.	IEEE 802.3 discloses at least some of the backplane traces being aggregated into an Ethernet link aggregation (LAG) group. *See supra at* 15[e], 4[f], 3.

No.	**’740 Patent Claim 16**	**IEEE 802.3**
16	The method according to claim 14, wherein selecting the backplane trace comprises applying a hashing function to the at least one of the frame attributes.	IEEE 802.3 discloses the method according to claim 14, wherein selecting the backplane trace comprises applying a hashing function to the at least one of the frame attributes. *See supra* at 14, 9, 8.

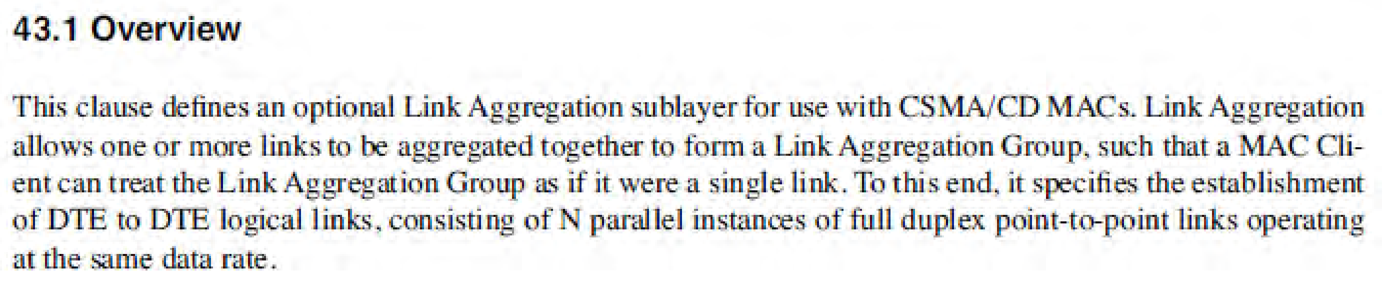
No.	**’740 Patent Claim 17**	**IEEE 802.3**
17[preamble]	Apparatus for connecting a network node with a communication network, comprising:	IEEE 802.3 discloses apparatus for connecting a network node with a communication network. For example, IEEE 802.3 discloses a System that connects a MAC Client to a network. IEEE 802.3 at 1468-69

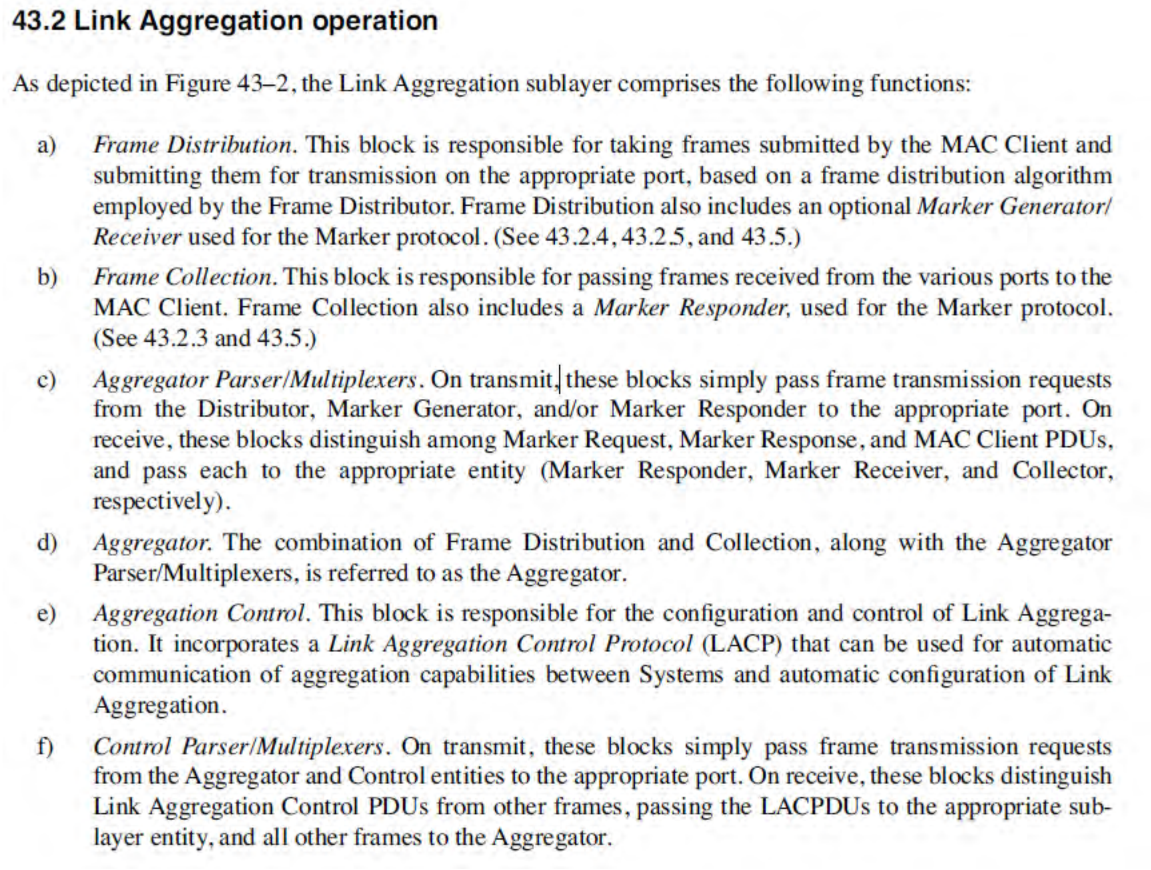
No.	’740 Patent Claim 17	IEEE 802.3
		 **43.2.1 Principles of Link Aggregation** Link Aggregation allows a MAC Client to treat a set of one or more ports as if it were a single port. In doing so, it employs the following principles and concepts: a) A MAC Client communicates with a set of ports through an Aggregator, which presents a standard IEEE 802.3® service interface to the MAC Client. The Aggregator binds to one or more ports within a System. b) It is the responsibility of the Aggregator to distribute frame transmissions from the MAC Client to the various ports, and to collect received frames from the ports and pass them to the MAC Client transparently. c) A System may contain multiple Aggregators, serving multiple MAC Clients. A given port will bind to (at most) a single Aggregator at any time. A MAC Client is served by a single Aggregator at a time. d) The binding of ports to Aggregators within a System is managed by the Link Aggregation Control function for that System, which is responsible for determining which links may be aggregated, aggregating them, binding the ports within the System to an appropriate Aggregator, and monitoring conditions to determine when a change in aggregation is needed. e) Such determination and binding may be under manual control through direct manipulation of the state variables of Link Aggregation (e.g., Keys) by a network manager. In addition, automatic determination, configuration, binding, and monitoring may occur through the use of a Link Aggregation Control Protocol (LACP). The LACP uses peer exchanges across the links to determine, on an ongoing basis, the aggregation capability of the various links, and continuously provides the maximum level of aggregation capability achievable between a given pair of Systems.

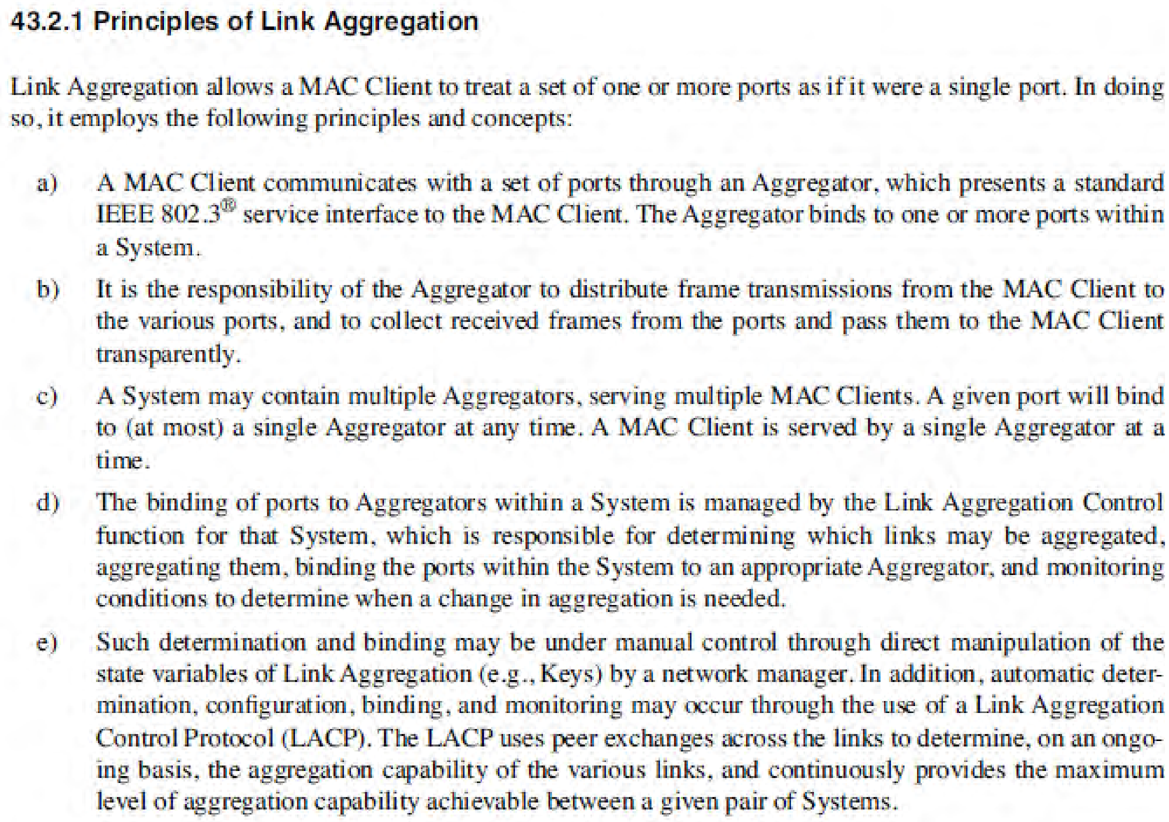
No.	’740 Patent Claim 17	IEEE 802.3
		 f) Frame ordering must be maintained for certain sequences of frame exchanges between MAC Clients (known as conversations, see 1.4). The Distributor ensures that all frames of a given conversation are passed to a single port. For any given port, the Collector is required to pass frames to the MAC Client in the order that they are received from that port. The Collector is otherwise free to select frames received from the aggregated ports in any order. Since there are no means for frames to be misordered on a single link, this guarantees that frame ordering is maintained for any conversation. g) Conversations may be moved among ports within an aggregation, both for load balancing and to maintain availability in the event of link failures. h) This standard does not impose any particular distribution algorithm on the Distributor. Whatever algorithm is used should be appropriate for the MAC Client being supported. i) Each port is assigned a unique, globally administered MAC address. This MAC address is used as the source address for frame exchanges that are initiated by entities within the Link Aggregation sublayer itself (i.e., LACP and Marker protocol exchanges). NOTE—The LACP and Marker protocols use a multicast destination address for all exchanges, and do not impose any requirement for a port to recognize more than one unicast address on received frames. j) Each Aggregator is assigned a unique, globally administered MAC address; this address is used as the MAC address of the aggregation from the perspective of the MAC Client, both as a source address for transmitted frames and as the destination address for received frames. The MAC address of the Aggregator may be one of the MAC addresses of a port in the associated Link Aggregation Group (see 43.2.10). IEEE 802.3 at Figure 43-2

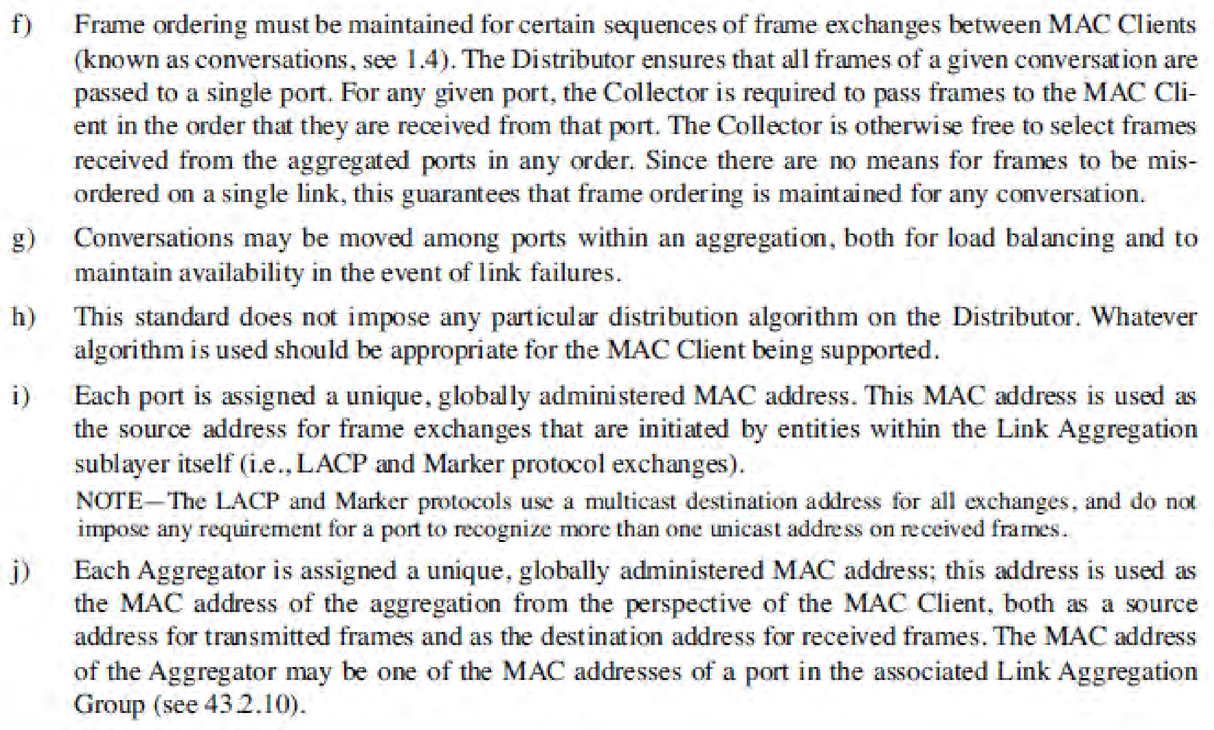
No.	’740 Patent Claim 17	IEEE 802.3
		 Figure 43–2—Link Aggregation sublayer block diagram

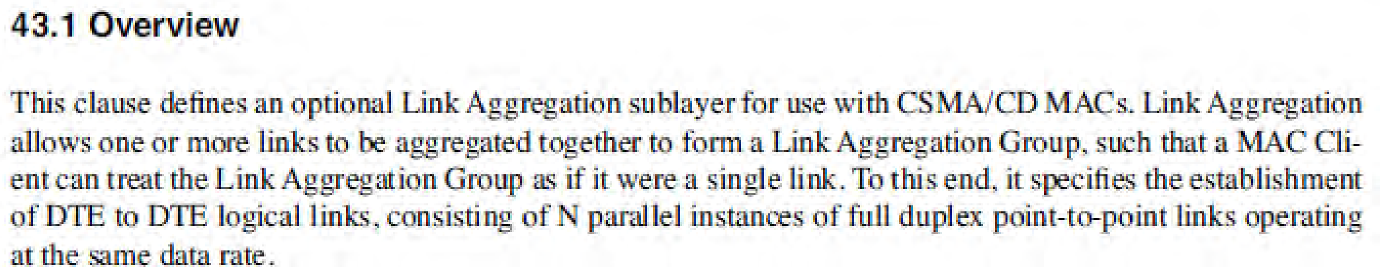
MAC Client
802.3® MAC Service Interface
Agg: MA_UNITDATA ( )
Frame Collection
Marker Responder
Frame Collector
Frame Distribution
Marker Generator/ Receiver (optional)
Frame Distributor
Link Aggregation Control
Link Aggregation Control Protocol
shared state variables
Marker Frames
MAC Client Frames
Marker Frames
MAC Client Frames
802.3® MAC Service Interface(s)
AggMuxN: MA_UNITDATA ( )
Aggregator Parser/Multiplexer
Aggregator Parser/Multiplexer
Aggregator Parser/Multiplexer
Aggregation Control
Aggregator
Control frames
MAC Client and Marker frames
802.3® MAC Service Interface(s)
CtrlMuxN: MA_UNITDATA ( )
Control Parser/Multiplexer
Control Parser/Multiplexer
Control Parser/Multiplexer
Link Aggregation sublayer
802.3® MAC Service Interface
MacN: MA_UNITDATA ( )
MAC Control (optional)
MAC
Physical Layer
MAC Control (optional)
MAC
Physical Layer
MAC Control (optional)
MAC
Physical Layer

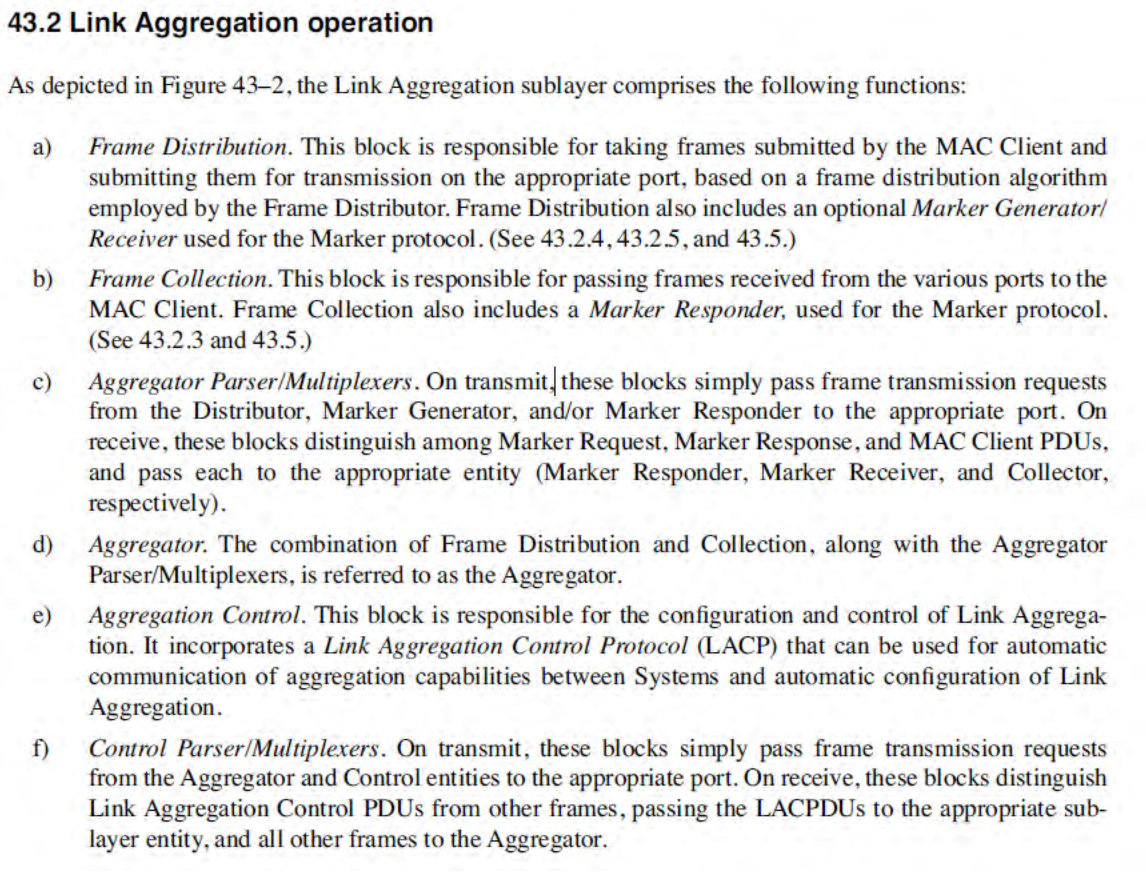
No.	’740 Patent Claim 17	IEEE 802.3
17[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network,	IEEE 802.3 discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. For example, IEEE 802.3 discloses an aggregator which processes data frames with frame information sent between the MAC client and network. IEEE 802.3 at 1465  **43.1 Overview** This clause defines an optional Link Aggregation sublayer for use with CSMA/CD MACs. Link Aggregation allows one or more links to be aggregated together to form a Link Aggregation Group, such that a MAC Client can treat the Link Aggregation Group as if it were a single link. To this end, it specifies the establishment of DTE to DTE logical links, consisting of N parallel instances of full duplex point-to-point links operating at the same data rate. IEEE 802.3 at 1468

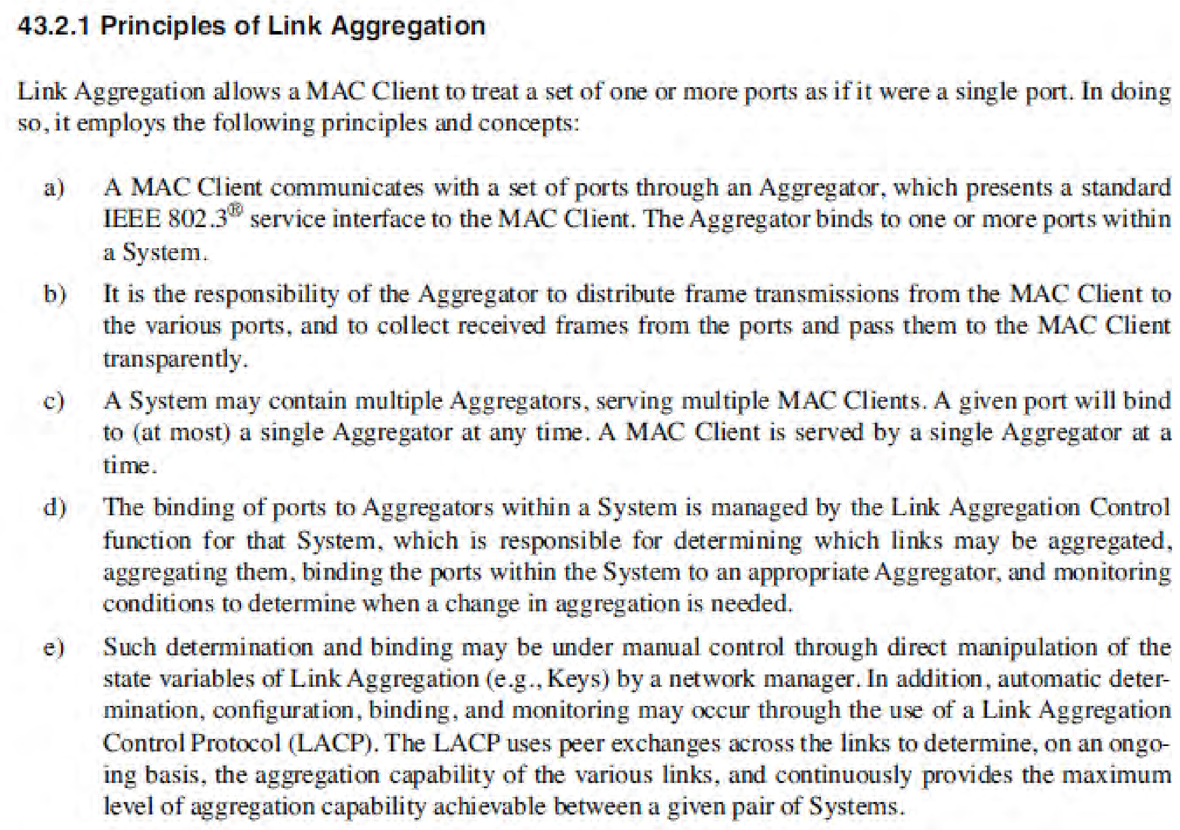
No.	’740 Patent Claim 17	IEEE 802.3
		 **43.2 Link Aggregation operation** As depicted in Figure 43–2, the Link Aggregation sublayer comprises the following functions: a) *Frame Distribution*. This block is responsible for taking frames submitted by the MAC Client and submitting them for transmission on the appropriate port, based on a frame distribution algorithm employed by the Frame Distributor. Frame Distribution also includes an optional *Marker Generator/Receiver* used for the Marker protocol. (See 43.2.4, 43.2.5, and 43.5.) b) *Frame Collection*. This block is responsible for passing frames received from the various ports to the MAC Client. Frame Collection also includes a *Marker Responder*, used for the Marker protocol. (See 43.2.3 and 43.5.) c) *Aggregator Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Distributor, Marker Generator, and/or Marker Responder to the appropriate port. On receive, these blocks distinguish among Marker Request, Marker Response, and MAC Client PDUs, and pass each to the appropriate entity (Marker Responder, Marker Receiver, and Collector, respectively). d) *Aggregator*. The combination of Frame Distribution and Collection, along with the Aggregator Parser/Multiplexers, is referred to as the Aggregator. e) *Aggregation Control*. This block is responsible for the configuration and control of Link Aggregation. It incorporates a *Link Aggregation Control Protocol* (LACP) that can be used for automatic communication of aggregation capabilities between Systems and automatic configuration of Link Aggregation. f) *Control Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Aggregator and Control entities to the appropriate port. On receive, these blocks distinguish Link Aggregation Control PDUs from other frames, passing the LACPDUs to the appropriate sublayer entity, and all other frames to the Aggregator. IEEE 802.3 at 1468-69

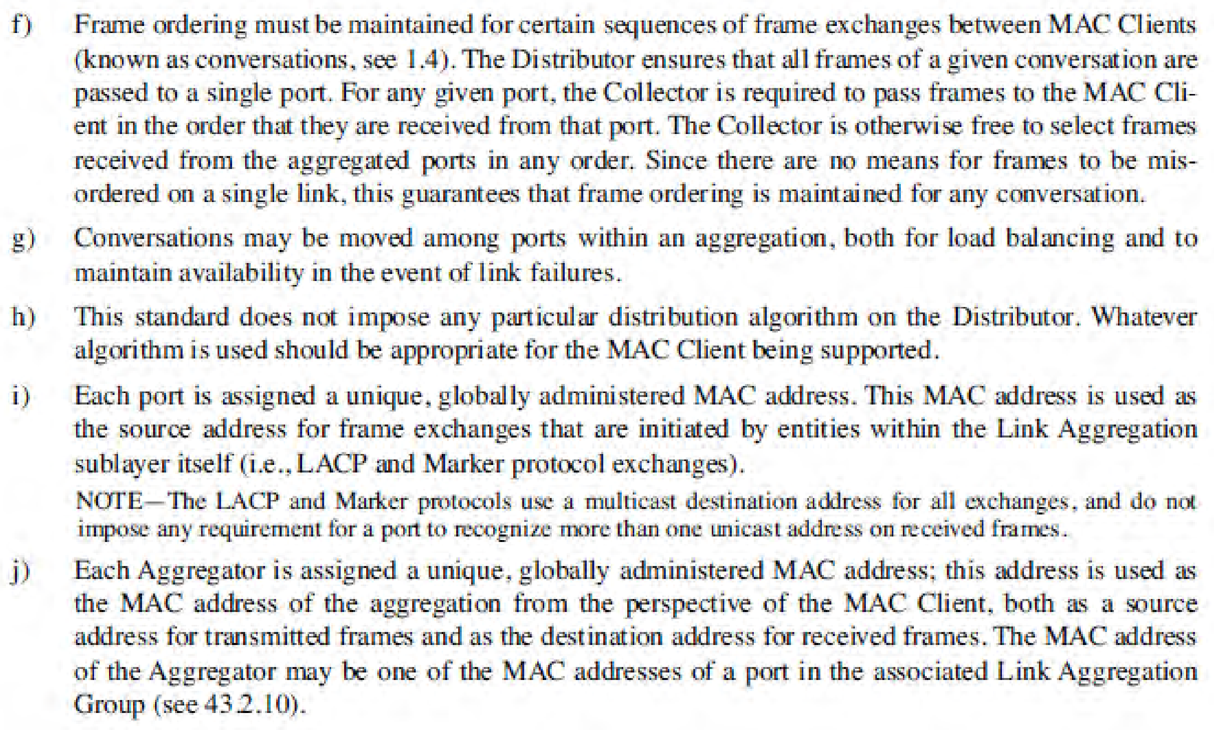
No.	’740 Patent Claim 17	IEEE 802.3
		 **43.2.1 Principles of Link Aggregation** Link Aggregation allows a MAC Client to treat a set of one or more ports as if it were a single port. In doing so, it employs the following principles and concepts: a) A MAC Client communicates with a set of ports through an Aggregator, which presents a standard IEEE 802.3® service interface to the MAC Client. The Aggregator binds to one or more ports within a System. b) It is the responsibility of the Aggregator to distribute frame transmissions from the MAC Client to the various ports, and to collect received frames from the ports and pass them to the MAC Client transparently. c) A System may contain multiple Aggregators, serving multiple MAC Clients. A given port will bind to (at most) a single Aggregator at any time. A MAC Client is served by a single Aggregator at a time. d) The binding of ports to Aggregators within a System is managed by the Link Aggregation Control function for that System, which is responsible for determining which links may be aggregated, aggregating them, binding the ports within the System to an appropriate Aggregator, and monitoring conditions to determine when a change in aggregation is needed. e) Such determination and binding may be under manual control through direct manipulation of the state variables of Link Aggregation (e.g., Keys) by a network manager. In addition, automatic determination, configuration, binding, and monitoring may occur through the use of a Link Aggregation Control Protocol (LACP). The LACP uses peer exchanges across the links to determine, on an ongoing basis, the aggregation capability of the various links, and continuously provides the maximum level of aggregation capability achievable between a given pair of Systems.

No.	’740 Patent Claim 17	IEEE 802.3
		 f) Frame ordering must be maintained for certain sequences of frame exchanges between MAC Clients (known as conversations, see 1.4). The Distributor ensures that all frames of a given conversation are passed to a single port. For any given port, the Collector is required to pass frames to the MAC Client in the order that they are received from that port. The Collector is otherwise free to select frames received from the aggregated ports in any order. Since there are no means for frames to be misordered on a single link, this guarantees that frame ordering is maintained for any conversation. g) Conversations may be moved among ports within an aggregation, both for load balancing and to maintain availability in the event of link failures. h) This standard does not impose any particular distribution algorithm on the Distributor. Whatever algorithm is used should be appropriate for the MAC Client being supported. i) Each port is assigned a unique, globally administered MAC address. This MAC address is used as the source address for frame exchanges that are initiated by entities within the Link Aggregation sublayer itself (i.e., LACP and Marker protocol exchanges). NOTE—The LACP and Marker protocols use a multicast destination address for all exchanges, and do not impose any requirement for a port to recognize more than one unicast address on received frames. j) Each Aggregator is assigned a unique, globally administered MAC address; this address is used as the MAC address of the aggregation from the perspective of the MAC Client, both as a source address for transmitted frames and as the destination address for received frames. The MAC address of the Aggregator may be one of the MAC addresses of a port in the associated Link Aggregation Group (see 43.2.10).
17[b]	at least one of said interface modules being operative to communicate in both an upstream direction and a downstream direction;	IEEE 802.3 discloses at least one of said interface modules being operative to communicate in both an upstream direction and a downstream direction. For example, IEEE 802.3 discloses an aggregator that both transmits and receives data frames between the MAC client and network. IEEE 802.3 at 1465

No.	’740 Patent Claim 17	IEEE 802.3
		 43.1 Overview This clause defines an optional Link Aggregation sublayer for use with CSMA/CD MACs. Link Aggregation allows one or more links to be aggregated together to form a Link Aggregation Group, such that a MAC Client can treat the Link Aggregation Group as if it were a single link. To this end, it specifies the establishment of DTE to DTE logical links, consisting of N parallel instances of full duplex point-to-point links operating at the same data rate. IEEE 802.3 at 1468

No.	’740 Patent Claim 17	IEEE 802.3
		 **43.2 Link Aggregation operation** As depicted in Figure 43–2, the Link Aggregation sublayer comprises the following functions: a) *Frame Distribution*. This block is responsible for taking frames submitted by the MAC Client and submitting them for transmission on the appropriate port, based on a frame distribution algorithm employed by the Frame Distributor. Frame Distribution also includes an optional *Marker Generator/Receiver* used for the Marker protocol. (See 43.2.4, 43.2.5, and 43.5.) b) *Frame Collection*. This block is responsible for passing frames received from the various ports to the MAC Client. Frame Collection also includes a *Marker Responder*, used for the Marker protocol. (See 43.2.3 and 43.5.) c) *Aggregator Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Distributor, Marker Generator, and/or Marker Responder to the appropriate port. On receive, these blocks distinguish among Marker Request, Marker Response, and MAC Client PDUs, and pass each to the appropriate entity (Marker Responder, Marker Receiver, and Collector, respectively). d) *Aggregator*. The combination of Frame Distribution and Collection, along with the Aggregator Parser/Multiplexers, is referred to as the Aggregator. e) *Aggregation Control*. This block is responsible for the configuration and control of Link Aggregation. It incorporates a *Link Aggregation Control Protocol* (LACP) that can be used for automatic communication of aggregation capabilities between Systems and automatic configuration of Link Aggregation. f) *Control Parser/Multiplexers*. On transmit, these blocks simply pass frame transmission requests from the Aggregator and Control entities to the appropriate port. On receive, these blocks distinguish Link Aggregation Control PDUs from other frames, passing the LACPDUs to the appropriate sublayer entity, and all other frames to the Aggregator. IEEE 802.3 at 1468-69

No.	’740 Patent Claim 17	IEEE 802.3
		 **43.2.1 Principles of Link Aggregation** Link Aggregation allows a MAC Client to treat a set of one or more ports as if it were a single port. In doing so, it employs the following principles and concepts: a) A MAC Client communicates with a set of ports through an Aggregator, which presents a standard IEEE 802.3® service interface to the MAC Client. The Aggregator binds to one or more ports within a System. b) It is the responsibility of the Aggregator to distribute frame transmissions from the MAC Client to the various ports, and to collect received frames from the ports and pass them to the MAC Client transparently. c) A System may contain multiple Aggregators, serving multiple MAC Clients. A given port will bind to (at most) a single Aggregator at any time. A MAC Client is served by a single Aggregator at a time. d) The binding of ports to Aggregators within a System is managed by the Link Aggregation Control function for that System, which is responsible for determining which links may be aggregated, aggregating them, binding the ports within the System to an appropriate Aggregator, and monitoring conditions to determine when a change in aggregation is needed. e) Such determination and binding may be under manual control through direct manipulation of the state variables of Link Aggregation (e.g., Keys) by a network manager. In addition, automatic determination, configuration, binding, and monitoring may occur through the use of a Link Aggregation Control Protocol (LACP). The LACP uses peer exchanges across the links to determine, on an ongoing basis, the aggregation capability of the various links, and continuously provides the maximum level of aggregation capability achievable between a given pair of Systems.

No.	’740 Patent Claim 17	IEEE 802.3
		 f) Frame ordering must be maintained for certain sequences of frame exchanges between MAC Clients (known as conversations, see 1.4). The Distributor ensures that all frames of a given conversation are passed to a single port. For any given port, the Collector is required to pass frames to the MAC Client in the order that they are received from that port. The Collector is otherwise free to select frames received from the aggregated ports in any order. Since there are no means for frames to be misordered on a single link, this guarantees that frame ordering is maintained for any conversation. g) Conversations may be moved among ports within an aggregation, both for load balancing and to maintain availability in the event of link failures. h) This standard does not impose any particular distribution algorithm on the Distributor. Whatever algorithm is used should be appropriate for the MAC Client being supported. i) Each port is assigned a unique, globally administered MAC address. This MAC address is used as the source address for frame exchanges that are initiated by entities within the Link Aggregation sublayer itself (i.e., LACP and Marker protocol exchanges). NOTE—The LACP and Marker protocols use a multicast destination address for all exchanges, and do not impose any requirement for a port to recognize more than one unicast address on received frames. j) Each Aggregator is assigned a unique, globally administered MAC address; this address is used as the MAC address of the aggregation from the perspective of the MAC Client, both as a source address for transmitted frames and as the destination address for received frames. The MAC address of the Aggregator may be one of the MAC addresses of a port in the associated Link Aggregation Group (see 43.2.10).
17[c]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	IEEE 802.3 discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 1[a].

No.	’740 Patent Claim 17	IEEE 802.3
17[d]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	IEEE 802.3 discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 1[c].
17[e]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second	IEEE 802.3 discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 1[f].

No.	’740 Patent Claim 17	IEEE 802.3
	physical link out of the second group over which to send the data frame;	
17[f]	at least one of said first physical links and at least one of said second links being bi-directional links operative to communicate in both said upstream direction and said downstream direction.	IEEE 802.3 discloses at least one of said first physical links and at least one of said second links being bi-directional links operative to communicate in both said upstream direction and said downstream direction. *See supra at* 1[b], 1[d].

No.	’740 Patent Claim 18	IEEE 802.3
18[a]	The apparatus according to claim 17, and comprising a backplane to which the one or more interface modules are coupled,	IEEE 802.3 discloses the apparatus according to claim 17, and comprising a backplane to which the one or more interface modules are coupled. *See supra at* 3, 17.
18[b]	wherein the second physical links comprise backplane traces formed on the backplane.	IEEE 802.3 discloses wherein the second physical links comprise backplane traces formed on the backplane. *See supra at* 3, 17.

No.	’740 Patent Claim 19	IEEE 802.3
19[preamble]	Apparatus for connecting a network node with a communication network, comprising:	IEEE 802.3 discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
19[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	IEEE 802.3 discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].

No.	’740 Patent Claim 19	IEEE 802.3
19[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	IEEE 802.3 discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].
19[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	IEEE 802.3 discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].
19[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second	IEEE 802.3 discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].

No.	’740 Patent Claim 19	IEEE 802.3
	group over which to send the data frame,	
19[e]	at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group.	IEEE 802.3 discloses at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group. *See supra at* 4[f].

No.	’740 Patent Claim 20	IEEE 802.3
20[preamble]	Apparatus for connecting a network node with a communication network, comprising:	IEEE 802.3 discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
20[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	IEEE 802.3 discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].
20[b]	a first group of first physical links arranged in parallel so as to couple the network node to the	IEEE 802.3 discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].

No.	’740 Patent Claim 20	IEEE 802.3
	one or more interface modules;	
20[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	IEEE 802.3 discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].
20[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,	IEEE 802.3 discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].
20[e]	two or more of the first physical links being aggregated into an external Ethernet	IEEE 802.3 discloses two or more of the first physical links being aggregated into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node.

No.	’740 Patent Claim 20	IEEE 802.3
	link aggregation (LAG) group so as to increase a data bandwidth provided to the network node.	*See supra at* 19[e], 5[f].

No.	’740 Patent Claim 21	IEEE 802.3
21	The apparatus according to claim 17, and comprising a multiplexer, which is arranged to perform at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node.	IEEE 802.3 discloses the apparatus according to claim 17, and comprising a multiplexer, which is arranged to perform at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node. *See supra at* 6, 17.

No.	’740 Patent Claim 22	IEEE 802.3
22	The apparatus according to claim 17, wherein the control module is arranged to balance a frame data rate among at least some of the first and second physical links.	IEEE 802.3 discloses the apparatus according to claim 17, wherein the control module is arranged to balance a frame data rate among at least some of the first and second physical links. *See supra at* 7, 17.

No.	’740 Patent Claim 23	IEEE 802.3
23	The apparatus according to claim 17, wherein the control module is arranged to apply a mapping function to the at least one of the frame attributes so as to select the first and second physical links.	IEEE 802.3 discloses the apparatus according to claim 17, wherein the control module is arranged to apply a mapping function to the at least one of the frame attributes so as to select the first and second physical links. *See supra at* 8, 17.

No.	’740 Patent Claim 24	IEEE 802.3
24	The apparatus according to claim 23, wherein the mapping function comprises a hashing function.	IEEE 802.3 discloses the apparatus according to claim 23, wherein the mapping function comprises a hashing function. *See supra at* 9, 23.

No.	’740 Patent Claim 25	IEEE 802.3
25[a]	The apparatus according to claim 24, wherein the control module is arranged to determine a hashing size responsively to a number of at least some of the first and second physical links,	IEEE 802.3 discloses the apparatus according to claim 24, wherein the control module is arranged to determine a hashing size responsively to a number of at least some of the first and second physical links. *See supra at* 10[a], 24.
25[b]	to apply the hashing function to the at least one of the frame attributes to produce a hashing key,	IEEE 802.3 discloses to apply the hashing function to the at least one of the frame attributes to produce a hashing key. *See supra at* 10[b].
25[c]	to calculate a modulo of a division operation of the hashing key by the hashing size, and	IEEE 802.3 discloses to calculate a modulo of a division operation of the hashing key by the hashing size. *See supra at* 10[c].
25[d]	to select the first and second physical links responsively to the modulo.	IEEE 802.3 discloses to select the first and second physical links responsively to the modulo. *See supra at* 10[d].

No.	’740 Patent Claim 26	IEEE 802.3
26	The apparatus according to claim 25, wherein the control module is arranged to select the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo.	IEEE 802.3 discloses the apparatus according to claim 25, wherein the control module is arranged to select the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo. *See supra at* 11, 25.

No.	’740 Patent Claim 27	IEEE 802.3
27	The apparatus according to claim 17, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port.	IEEE 802.3 discloses the apparatus according to claim 17, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port. *See supra at* 12, 17.

No.	’740 Patent Claim 28	IEEE 802.3
28[preamble]	Apparatus for connecting a network node with a communication network, comprising:	IEEE 802.3 discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
28[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	IEEE 802.3 discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].
28[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	IEEE 802.3 discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].
28[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	IEEE 802.3 discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].

28[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,	IEEE 802.3 discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].
28[e]	the communication network being arranged to provide a communication service to the network node,	IEEE 802.3 discloses the communication network being arranged to provide a communication service to the network node. *See supra at* 2[b].

28[f]	the service having specified bandwidth requirements comprising at least one of a committed information rate (CR), a peak information rate (PIR) and an excess information rate (EIR), and	IEEE 802.3 discloses the service having specified bandwidth requirements comprising at least one of a committed information rate (CR), a peak information rate (PIR) and an excess information rate (EIR). *See supra at* 13[i].
28[g]	the first and second groups of physical links being dimensioned to provide an allocated bandwidth for the communication service responsively to the bandwidth requirements.	IEEE 802.3 discloses the first and second groups of physical links being dimensioned to provide an allocated bandwidth for the communication service responsively to the bandwidth requirements. *See supra at* 13[j].

No.	**’740 Patent Claim 29**	**IEEE 802.3**
29[preamble]	Apparatus for connecting user ports to a communication network, comprising:	IEEE 802.3 discloses apparatus for connecting user ports to a communication network. *See supra at* 17[preamble], 14[preamble].
29[a]	one or more user interface modules coupled to the user ports, which are arranged to process	IEEE 802.3 discloses one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network. *See supra at* 17[a], 14[a].

No.	’740 Patent Claim 29	IEEE 802.3
	data frames having frame attributes sent between the user ports and the communication network,	
29[b]	at least one of said user interface modules being bi-directional and operative to communicate in both an upstream direction and a downstream direction;	IEEE 802.3 discloses at least one of said user interface modules being bi-directional and operative to communicate in both an upstream direction and a downstream direction. *See supra at* 17[b], 14[c].
29[c]	a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network,	IEEE 802.3 discloses a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network. *See supra at* 14[b]-[e].

No.	’740 Patent Claim 29	IEEE 802.3
29[d]	at least one of said backplane traces being bi-directional and operative to communicate in both said upstream direction and said downstream direction; and	IEEE 802.3 discloses at least one of said backplane traces being bi-directional and operative to communicate in both said upstream direction and said downstream direction. *See supra at* 14[c], 17[b].
29[e]	a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame.	IEEE 802.3 discloses a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame. *See supra at* 14[e], 17[e].

No.	’740 Patent Claim 30	IEEE 802.3
30[preamble]	Apparatus for connecting user ports to a communication network, comprising:	IEEE 802.3 discloses apparatus for connecting user ports to a communication network. *See supra at* 29[preamble].
30[a]	one or more user interface modules coupled to the user ports, which are	IEEE 802.3 discloses one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network.

No.	’740 Patent Claim 30	IEEE 802.3
	arranged to process data frames having frame attributes sent between the user ports and the communication network;	*See supra at* 29[a].
30[b]	a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network;	IEEE 802.3 discloses a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network. *See supra at* 29[c].
30[c]	a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over	IEEE 802.3 discloses a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame. *See supra at* 29[e].

No.	’740 Patent Claim 30	IEEE 802.3
	which to send the data frame;	
30[d]	at least some of the backplane traces are aggregated into an Ethernet link aggregation (LAG) group.	IEEE 802.3 discloses at least some of the backplane traces are aggregated into an Ethernet link aggregation (LAG) group. *See supra at* 4[f], 15[f].

No.	’740 Patent Claim 31	IEEE 802.3
31	The apparatus according to claim 29, wherein the control module is arranged to apply a hashing function to the at least one of the frame attributes so as to select the backplane trace.	IEEE 802.3 discloses the apparatus according to claim 29, wherein the control module is arranged to apply a hashing function to the at least one of the frame attributes so as to select the backplane trace. *See supra at* 16, 29, 30[c].

EXHIBIT C-5
Defendant's Preliminary Invalidity Contentions
Orckit Corporation v. Cisco Systems, Inc., 2:22-cv-00276-JRG-RSP

Chart for U.S. Patent 7,545,740 ("The '740 Patent")
Cisco EtherSwitch System ("Cisco EtherSwitch System")

Cisco EtherSwitch System product family including EtherSwitch EPS-500, EtherSwitch EPS-2015 RS, and EtherSwitch EPS-2115M, ("Cisco EtherSwitch System"), which were conceived, sold, publicly used, and reduced to practice no later than April 7, 2006, constitute prior art to the '740 Patent. As shown in the chart below, all Asserted Claims of the '740 Patent are invalid under (1) 35 U.S.C. §§ 102 (a) and (g) because the Cisco EtherSwitch System meet each element of those claims, 35 U.S.C. §§ (a) and (b) because the references describing the Cisco EtherSwitch System disclose every limitation of every Asserted Claim, and/or (3) 35 U.S.C. § 103 because the Cisco EtherSwitch System renders those claims obvious either alone, or in combination with the knowledge of a person having ordinary skill in the art, and in further combination with the references specifically identified below and in the following claim chart and/or one or more references identified in Defendant's Preliminary Invalidity Contentions. The following quotations and diagrams come from documentation describing the Cisco EtherSwitch System and its functionalities that were published prior to April 7, 2006.

The Cisco EtherSwitch System is described at least in the following documents and other materials published before April 7, 2006 cited in this chart:

- http://ciscoarchive.lunaimaging.com/MediaManager/srvr?mediafile=/MISC/CHMC~5~5/106/78-1080-07.pdf ("Cisco EtherSwitch System Catalog")
- http://ciscoarchive.lunaimaging.com/MediaManager/srvr?mediafile=/MISC/CHMC~5~5/106/78-5983-14.pdf (EtherSwitch Quick Reference Guide")
- https://www.cisco.com/c/dam/en/us/td/docs/ios/12_4/interface/configuration/guide/irh_bk.pdf ("Cisco IOS Interface and Hardware Component Configuration Guide")

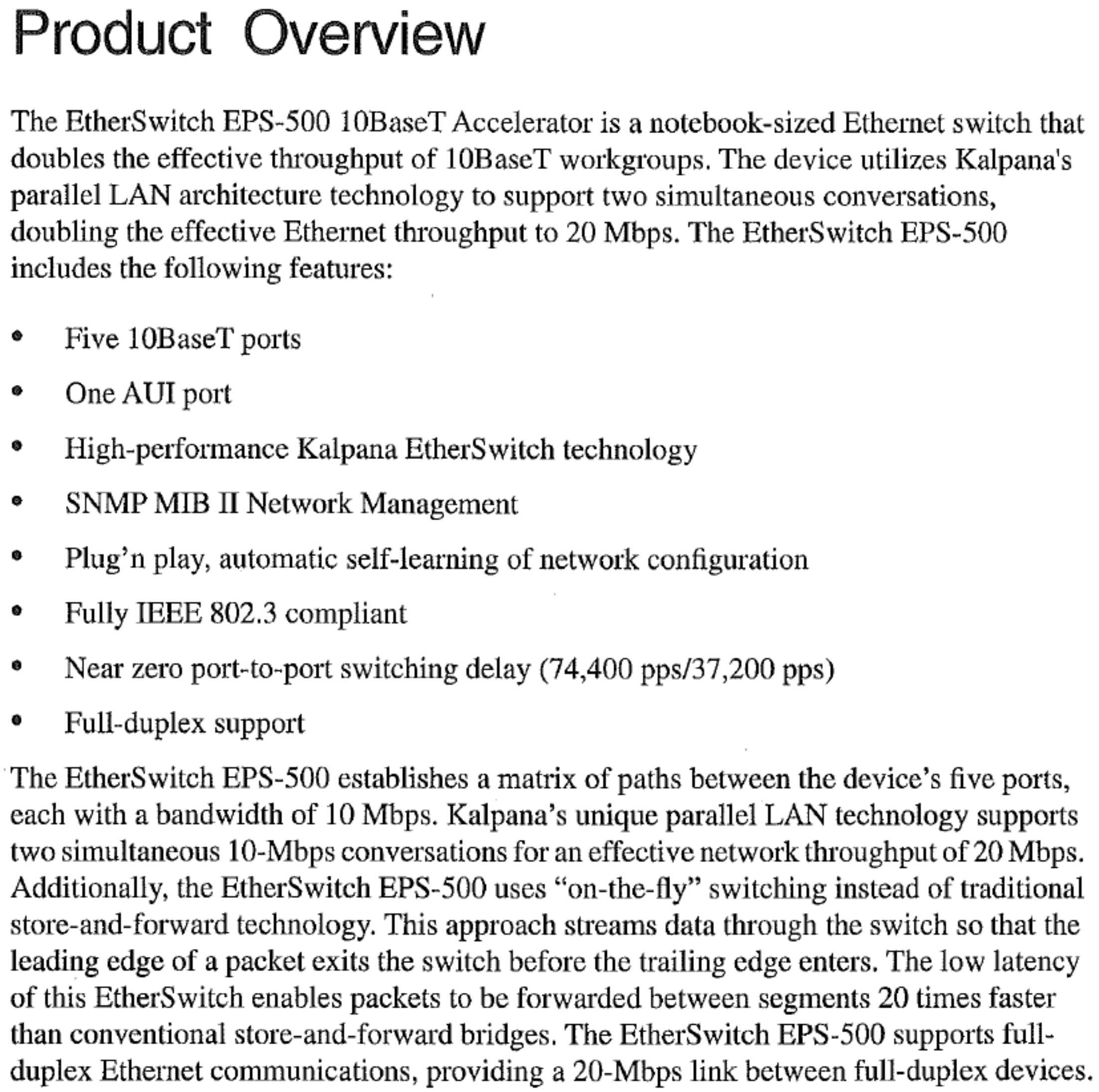
Motivations to combine the disclosures in Cisco EtherSwitch System with disclosures in other publications known in the art, as explained in this chart, include at least the similarity in subject matter between the references to the to the extent they concern methods of data communication systems, and specifically to methods and systems for link aggregation in a data communication network.

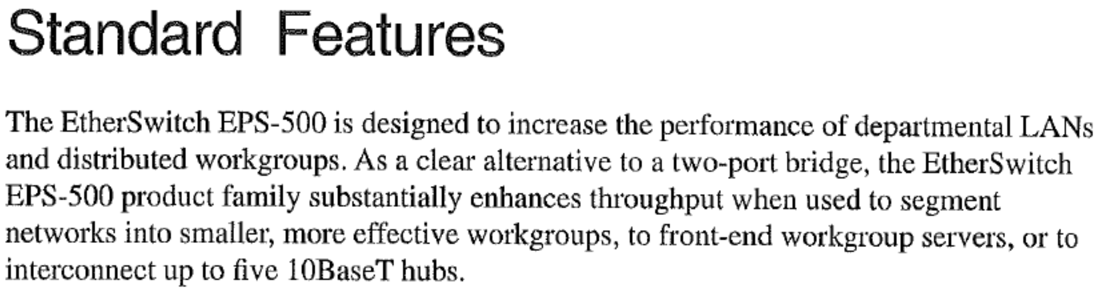
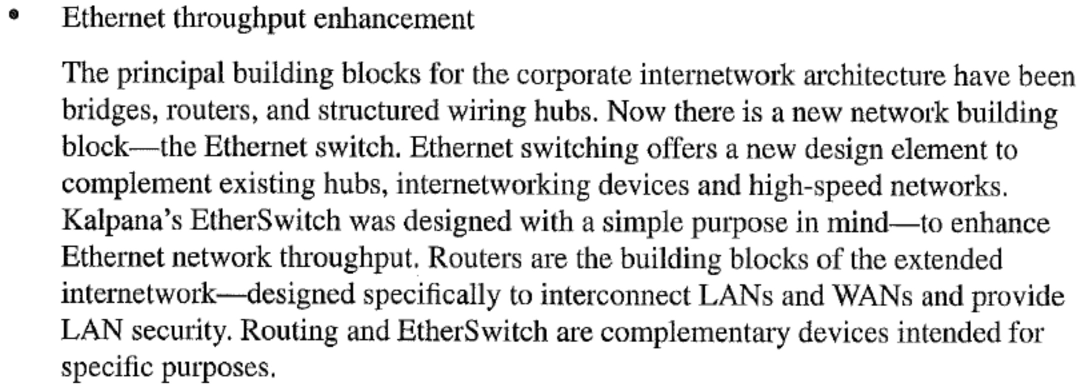
Insofar as the references cite other patents or publications, or suggest additional changes, one of ordinary skill in the art would look beyond a single reference to other references in the field.

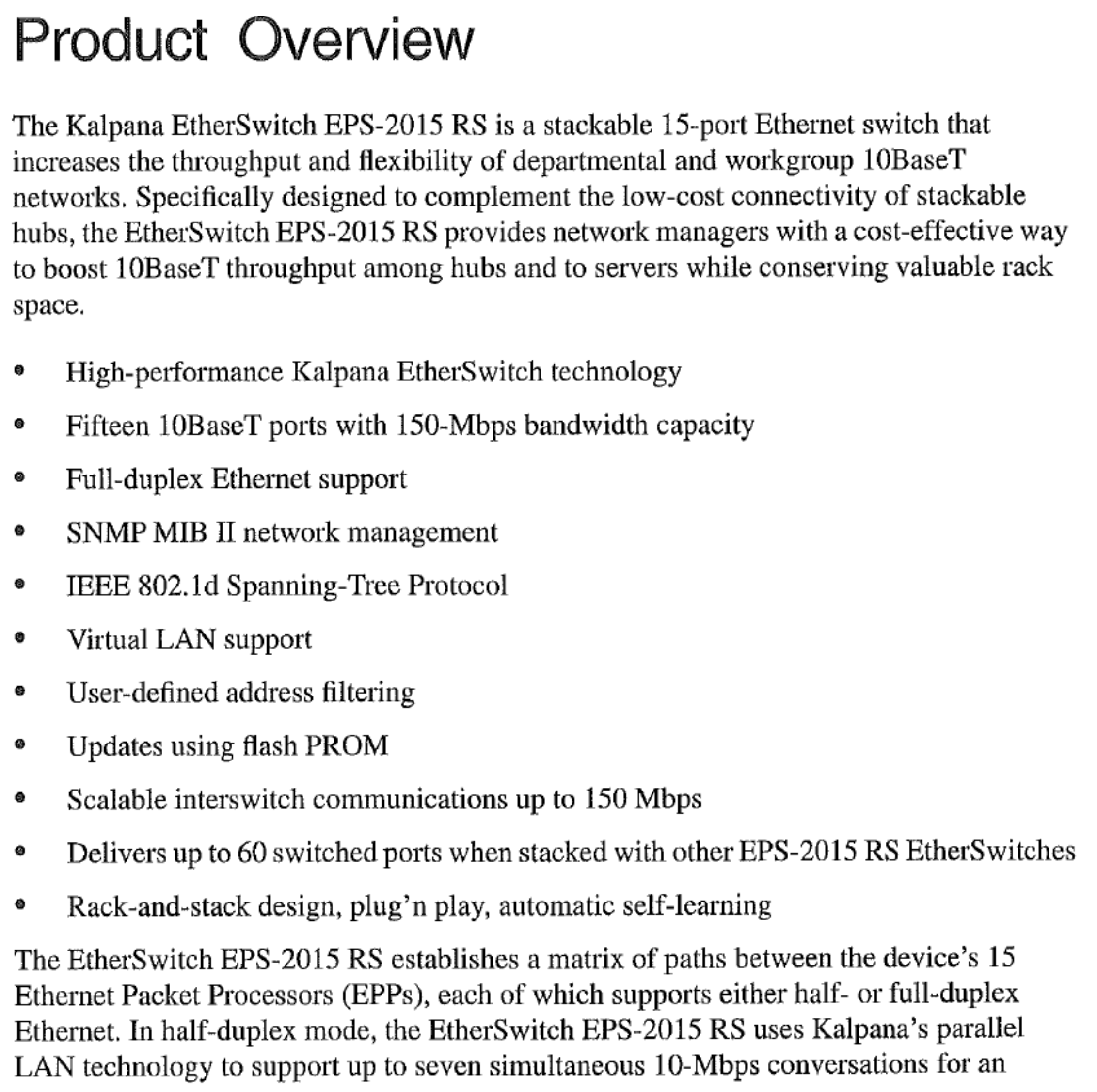
These invalidity contentions are based on Defendant’s present understanding of the asserted claims, and Orckit’s apparent construction of the claims in its November 3, 2022 Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1, and Orckit’s January 19, 2023 First Amended Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1 (Orckit’s “Infringement Disclosures”), which is deficient at least insofar as it fails to cite any documents or identify accused structures, acts, or materials in the Accused Products with particularity. Defendant does not agree with Orckit’s application of the claims, or that the claims satisfy the requirements of 35 U.S.C. § 112. Defendant’s contentions herein are not, and should in no way be seen as, admissions or adoptions as to any particular claim scope or construction, or as any admission that any particular element is met by any accused product in any particular way. Defendant objects to any attempt to imply claim construction from this chart. Defendant’s prior art invalidity contentions are made in a variety of alternatives and do not represent Defendant’s agreement or view as to the meaning, definiteness, written description support for, or enablement of any claim contained therein.

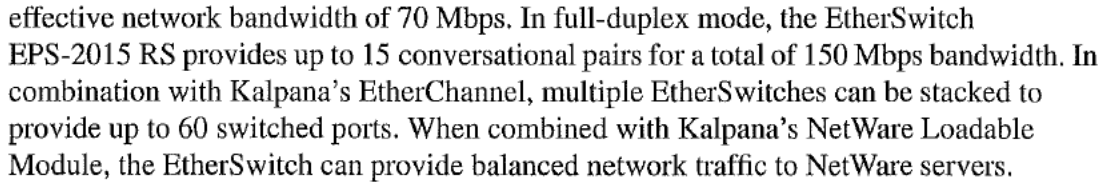
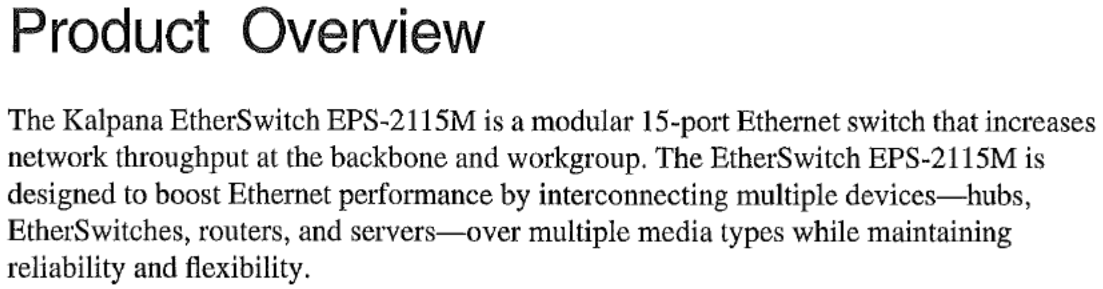
The following contentions are subject to revision and amendment pursuant to Federal Rule of Civil Procedure 26(e), the Local Rules, and the Orders of record in this matter subject to further investigation and discovery regarding the prior art and the Court’s construction of the claims at issue.

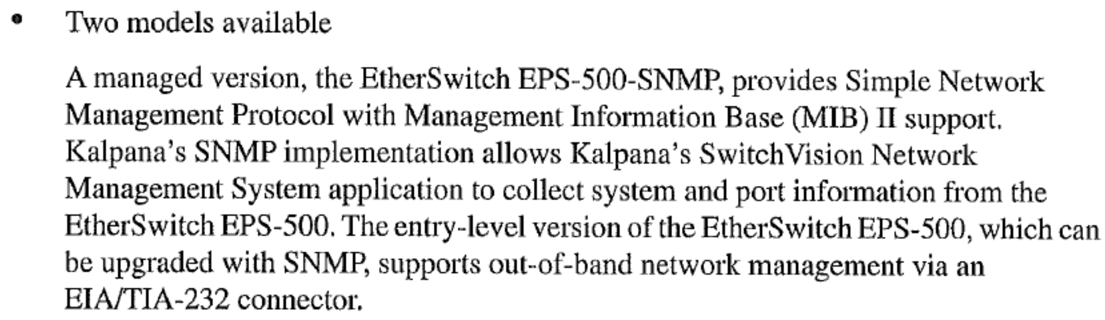
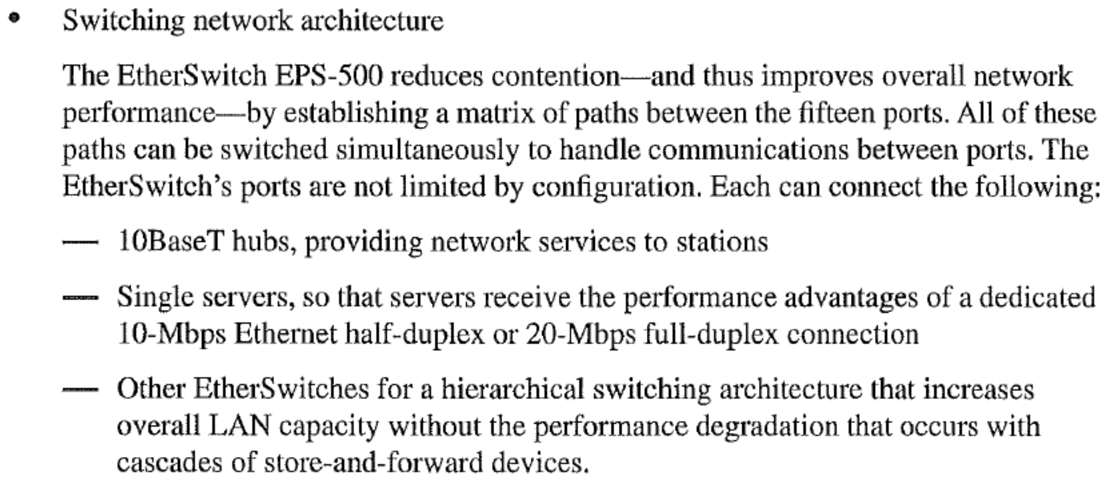
No.	**’740 Patent Claim 1**	**Cisco EtherSwitch System**
1[preamble]	A method for communication, comprising:	Cisco EtherSwitch System discloses a method for communication. For example, Cisco EtherSwitch System discloses an Ethernet switch that provides for increased throughput of data traffic in networks. Cisco EtherSwitch System Catalog at 111

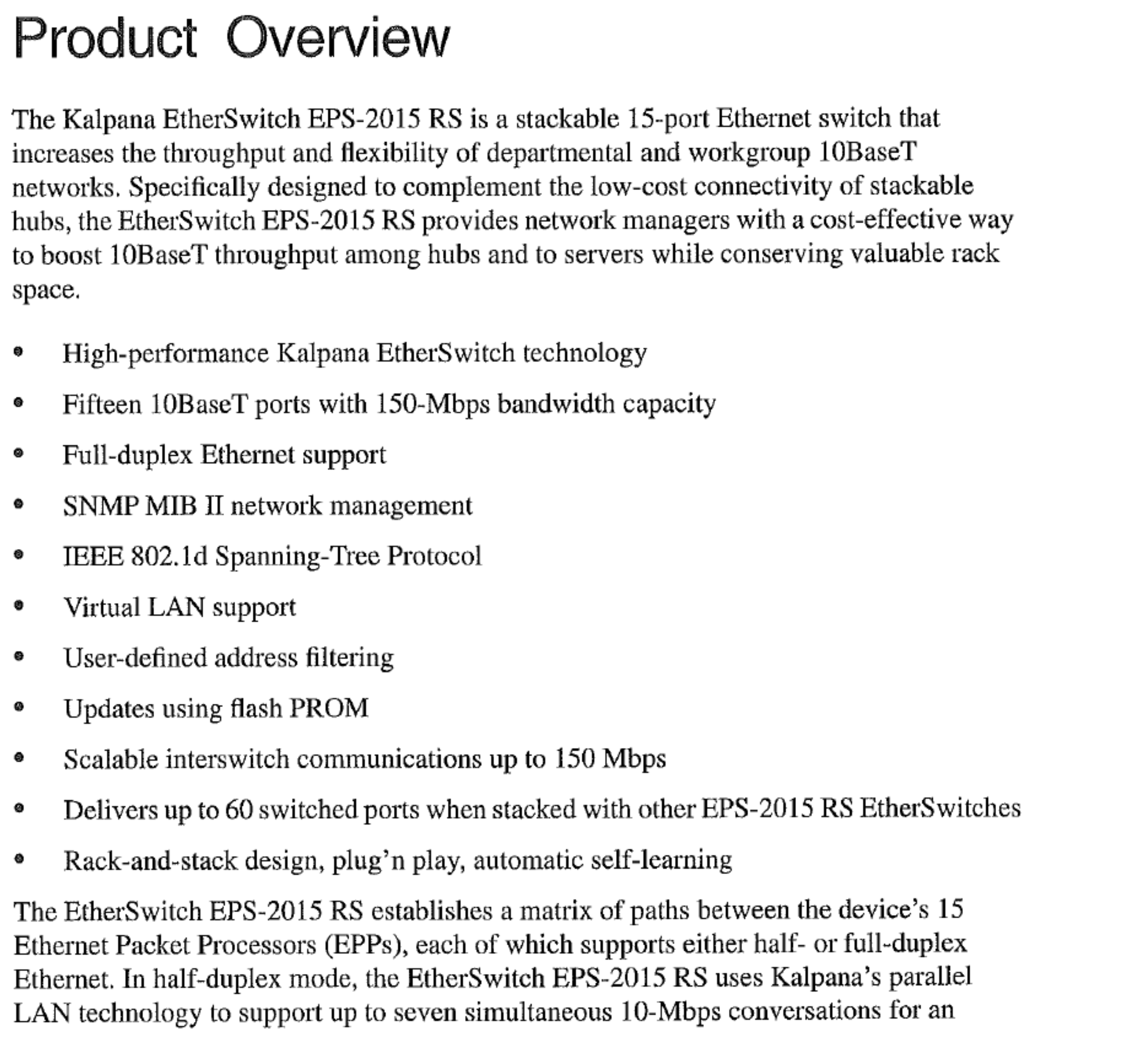
No.	’740 Patent Claim 1	Cisco EtherSwitch System
		 **Product Overview** The EtherSwitch EPS-500 10BaseT Accelerator is a notebook-sized Ethernet switch that doubles the effective throughput of 10BaseT workgroups. The device utilizes Kalpana's parallel LAN architecture technology to support two simultaneous conversations, doubling the effective Ethernet throughput to 20 Mbps. The EtherSwitch EPS-500 includes the following features: • Five 10BaseT ports • One AUI port • High-performance Kalpana EtherSwitch technology • SNMP MIB II Network Management • Plug’n play, automatic self-learning of network configuration • Fully IEEE 802.3 compliant • Near zero port-to-port switching delay (74,400 pps/37,200 pps) • Full-duplex support The EtherSwitch EPS-500 establishes a matrix of paths between the device’s five ports, each with a bandwidth of 10 Mbps. Kalpana’s unique parallel LAN technology supports two simultaneous 10-Mbps conversations for an effective network throughput of 20 Mbps. Additionally, the EtherSwitch EPS-500 uses “on-the-fly” switching instead of traditional store-and-forward technology. This approach streams data through the switch so that the leading edge of a packet exits the switch before the trailing edge enters. The low latency of this EtherSwitch enables packets to be forwarded between segments 20 times faster than conventional store-and-forward bridges. The EtherSwitch EPS-500 supports full-duplex Ethernet communications, providing a 20-Mbps link between full-duplex devices.

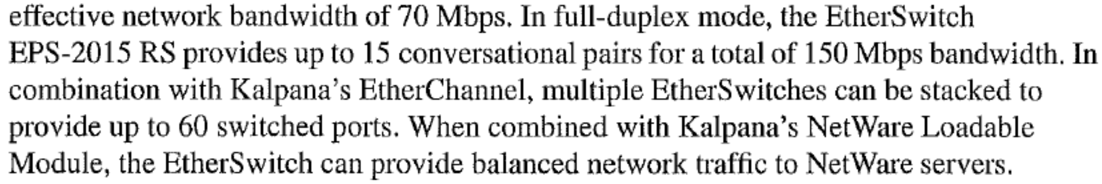
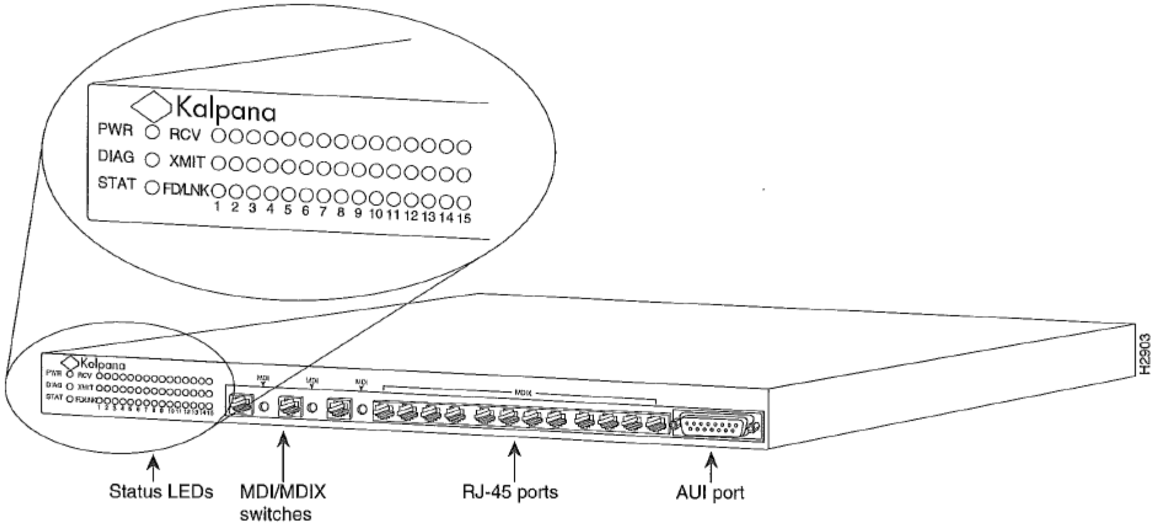
No.	’740 Patent Claim 1	Cisco EtherSwitch System
		Cisco EtherSwitch System Catalog at 113  **Standard Features** The EtherSwitch EPS-500 is designed to increase the performance of departmental LANs and distributed workgroups. As a clear alternative to a two-port bridge, the EtherSwitch EPS-500 product family substantially enhances throughput when used to segment networks into smaller, more effective workgroups, to front-end workgroup servers, or to interconnect up to five 10BaseT hubs. Cisco EtherSwitch System Catalog at 113  • Ethernet throughput enhancement The principal building blocks for the corporate internetwork architecture have been bridges, routers, and structured wiring hubs. Now there is a new network building block—the Ethernet switch. Ethernet switching offers a new design element to complement existing hubs, internetworking devices and high-speed networks. Kalpana’s EtherSwitch was designed with a simple purpose in mind—to enhance Ethernet network throughput. Routers are the building blocks of the extended internetwork—designed specifically to interconnect LANs and WANs and provide LAN security. Routing and EtherSwitch are complementary devices intended for specific purposes. Cisco EtherSwitch System Catalog at 115

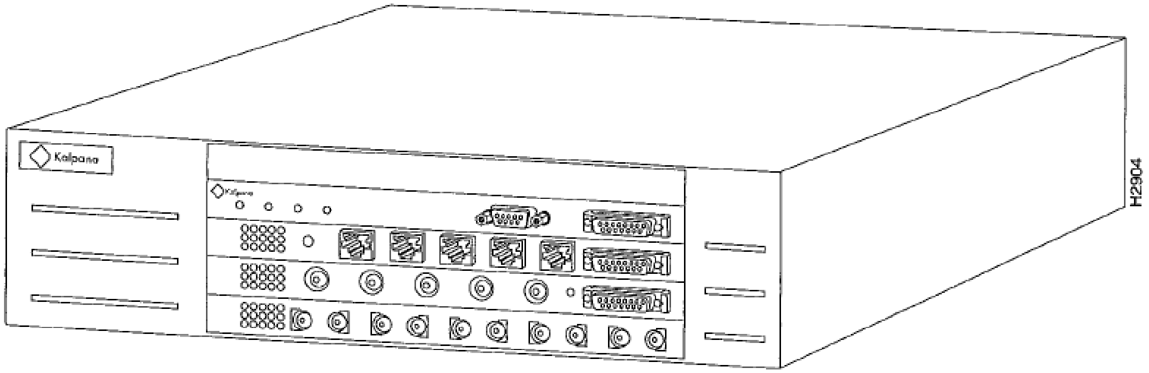
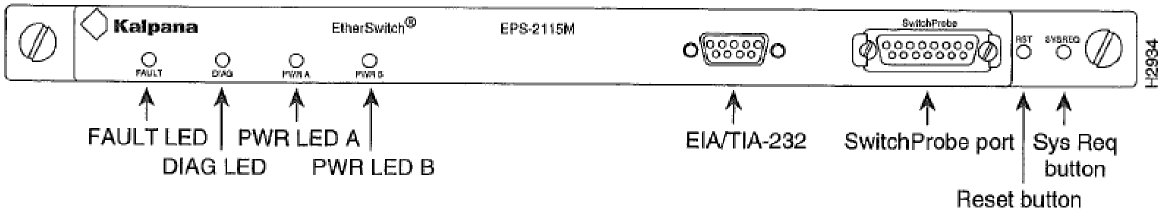
No.	’740 Patent Claim 1	Cisco EtherSwitch System
		 **Product Overview** The Kalpana EtherSwitch EPS-2015 RS is a stackable 15-port Ethernet switch that increases the throughput and flexibility of departmental and workgroup 10BaseT networks. Specifically designed to complement the low-cost connectivity of stackable hubs, the EtherSwitch EPS-2015 RS provides network managers with a cost-effective way to boost 10BaseT throughput among hubs and to servers while conserving valuable rack space. • High-performance Kalpana EtherSwitch technology • Fifteen 10BaseT ports with 150-Mbps bandwidth capacity • Full-duplex Ethernet support • SNMP MIB II network management • IEEE 802.1d Spanning-Tree Protocol • Virtual LAN support • User-defined address filtering • Updates using flash PROM • Scalable interswitch communications up to 150 Mbps • Delivers up to 60 switched ports when stacked with other EPS-2015 RS EtherSwitches • Rack-and-stack design, plug’n play, automatic self-learning The EtherSwitch EPS-2015 RS establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2015 RS uses Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an

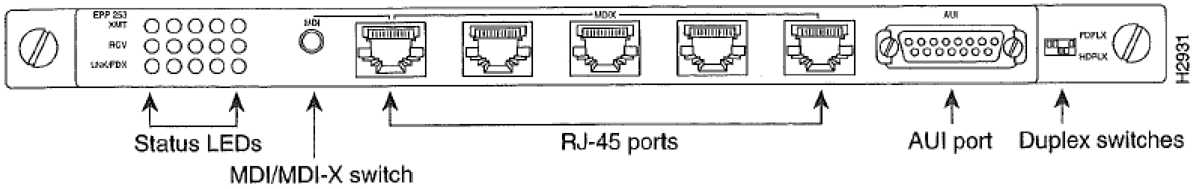
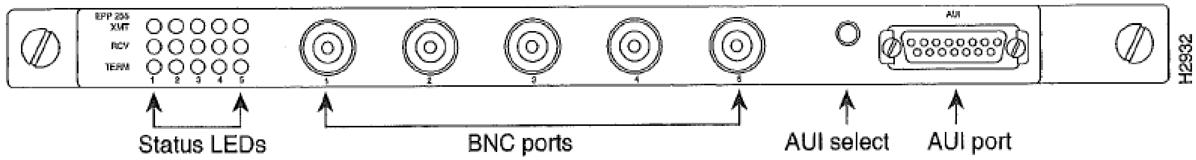
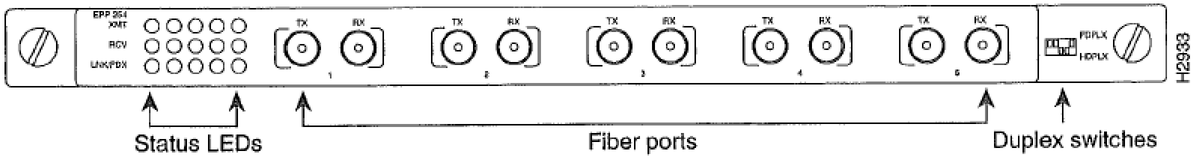
No.	’740 Patent Claim 1	Cisco EtherSwitch System
		 effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2015 RS provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module, the EtherSwitch can provide balanced network traffic to NetWare servers. Cisco EtherSwitch System Catalog at 119  Product Overview The Kalpana EtherSwitch EPS-2115M is a modular 15-port Ethernet switch that increases network throughput at the backbone and workgroup. The EtherSwitch EPS-2115M is designed to boost Ethernet performance by interconnecting multiple devices—hubs, EtherSwitches, routers, and servers—over multiple media types while maintaining reliability and flexibility.
1[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel,	Cisco EtherSwitch System discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. For example, Cisco EtherSwitch System discloses EtherSwitch network modules that are connected to network devices such as hubs, other EtherSwitches, routers, and servers via parallel ports. Cisco EtherSwitch System Catalog at 113

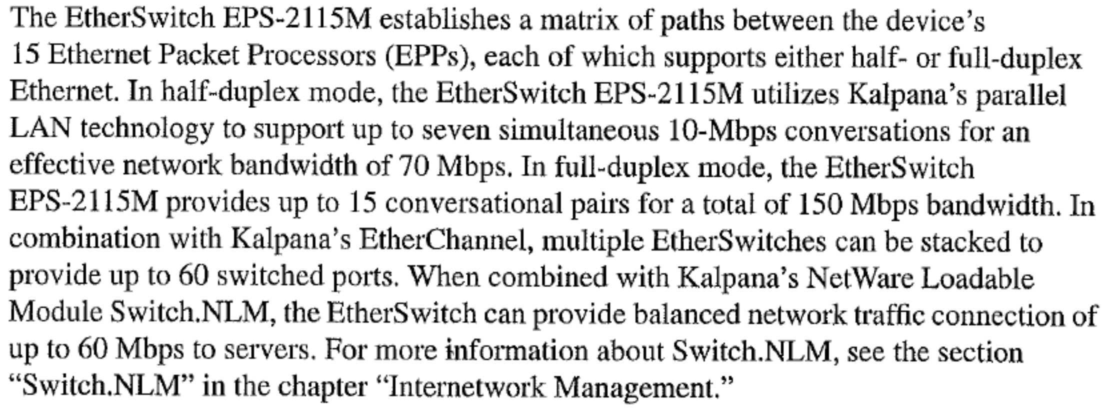
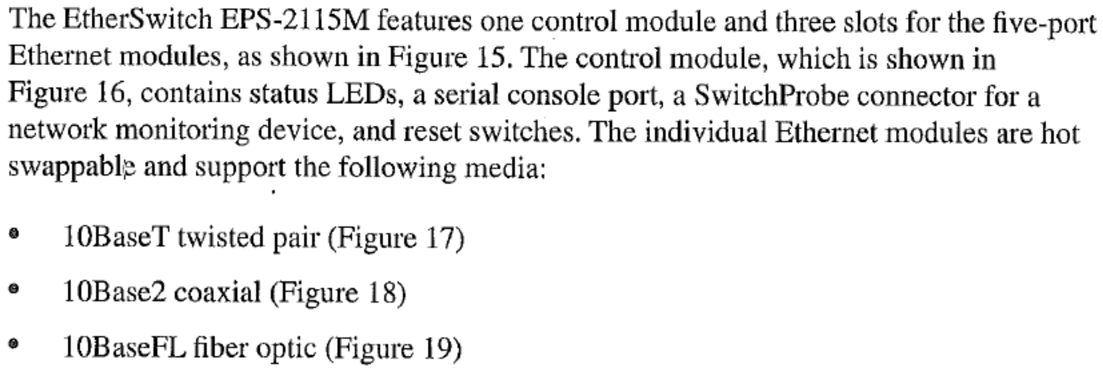
No.	’740 Patent Claim 1	Cisco EtherSwitch System
		 • Two models available A managed version, the EtherSwitch EPS-500-SNMP, provides Simple Network Management Protocol with Management Information Base (MIB) II support. Kalpana’s SNMP implementation allows Kalpana’s SwitchVision Network Management System application to collect system and port information from the EtherSwitch EPS-500. The entry-level version of the EtherSwitch EPS-500, which can be upgraded with SNMP, supports out-of-band network management via an EIA/TIA-232 connector. Cisco EtherSwitch System Catalog at 114  • Switching network architecture The EtherSwitch EPS-500 reduces contention—and thus improves overall network performance—by establishing a matrix of paths between the fifteen ports. All of these paths can be switched simultaneously to handle communications between ports. The EtherSwitch’s ports are not limited by configuration. Each can connect the following: — 10BaseT hubs, providing network services to stations — Single servers, so that servers receive the performance advantages of a dedicated 10-Mbps Ethernet half-duplex or 20-Mbps full-duplex connection — Other EtherSwitches for a hierarchical switching architecture that increases overall LAN capacity without the performance degradation that occurs with cascades of store-and-forward devices. Cisco EtherSwitch System Catalog at 115

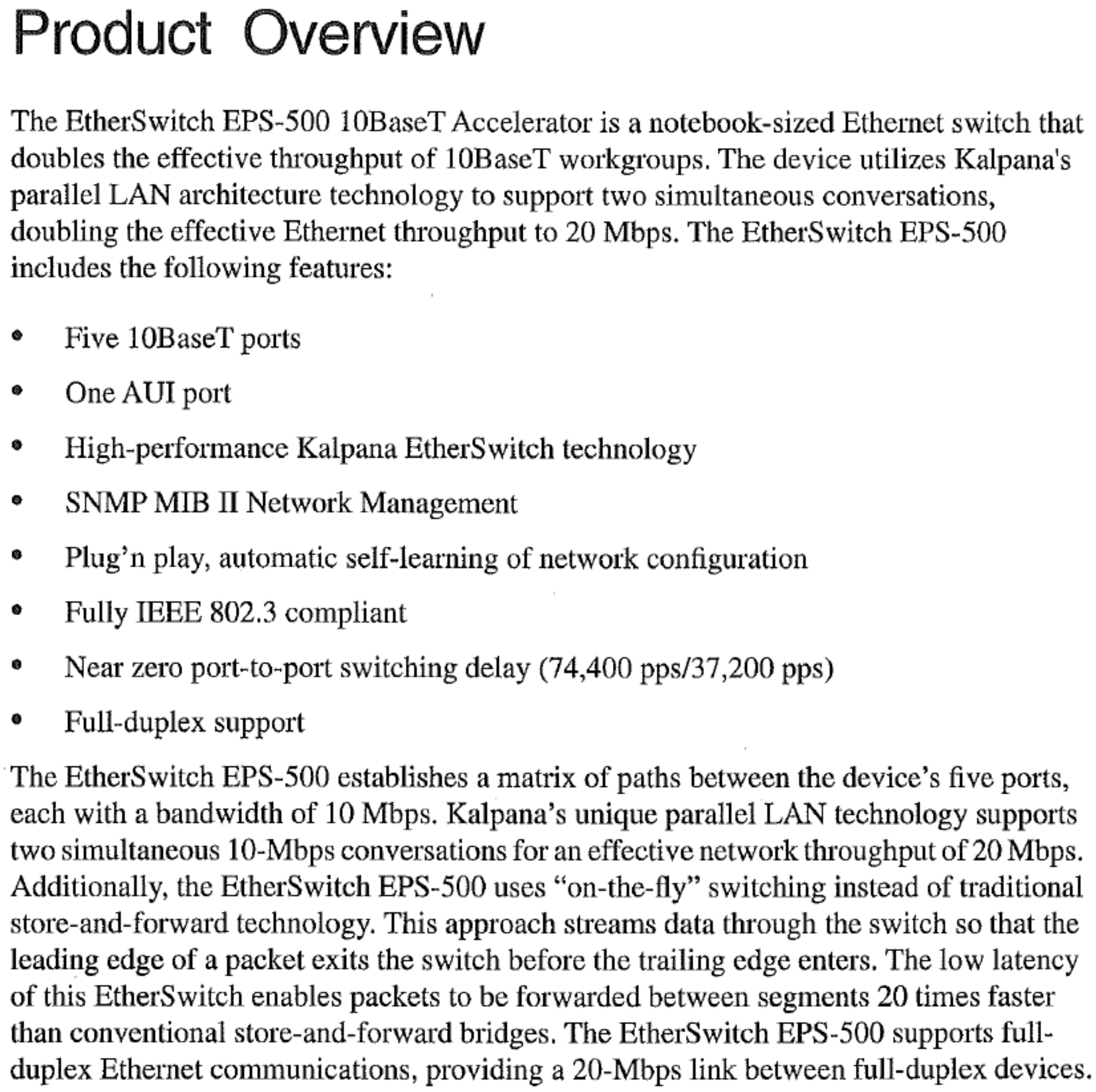
No.	’740 Patent Claim 1	Cisco EtherSwitch System
		 Product Overview The Kalpana EtherSwitch EPS-2015 RS is a stackable 15-port Ethernet switch that increases the throughput and flexibility of departmental and workgroup 10BaseT networks. Specifically designed to complement the low-cost connectivity of stackable hubs, the EtherSwitch EPS-2015 RS provides network managers with a cost-effective way to boost 10BaseT throughput among hubs and to servers while conserving valuable rack space. • High-performance Kalpana EtherSwitch technology • Fifteen 10BaseT ports with 150-Mbps bandwidth capacity • Full-duplex Ethernet support • SNMP MIB II network management • IEEE 802.1d Spanning-Tree Protocol • Virtual LAN support • User-defined address filtering • Updates using flash PROM • Scalable interswitch communications up to 150 Mbps • Delivers up to 60 switched ports when stacked with other EPS-2015 RS EtherSwitches • Rack-and-stack design, plug’n play, automatic self-learning The EtherSwitch EPS-2015 RS establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2015 RS uses Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an

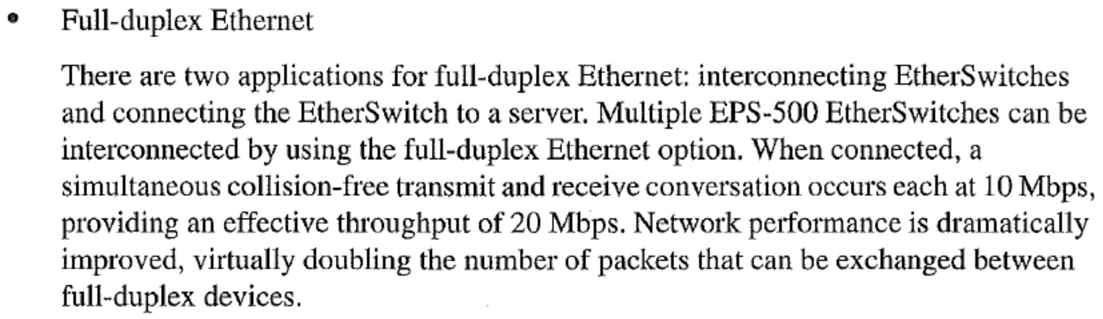
No.	’740 Patent Claim 1	Cisco EtherSwitch System
		 effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2015 RS provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module, the EtherSwitch can provide balanced network traffic to NetWare servers. Cisco EtherSwitch System Catalog at Figure 14 Figure 14 EtherSwitch EPS-2015 RS Front Panel  Kalpana PWR RCV DIAG XMIT STAT FD/LNK 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 MDI MDIX H2903 Status LEDs MDI/MDIX switches RJ-45 ports AUI port Cisco EtherSwitch System Catalog at Figures 15-19

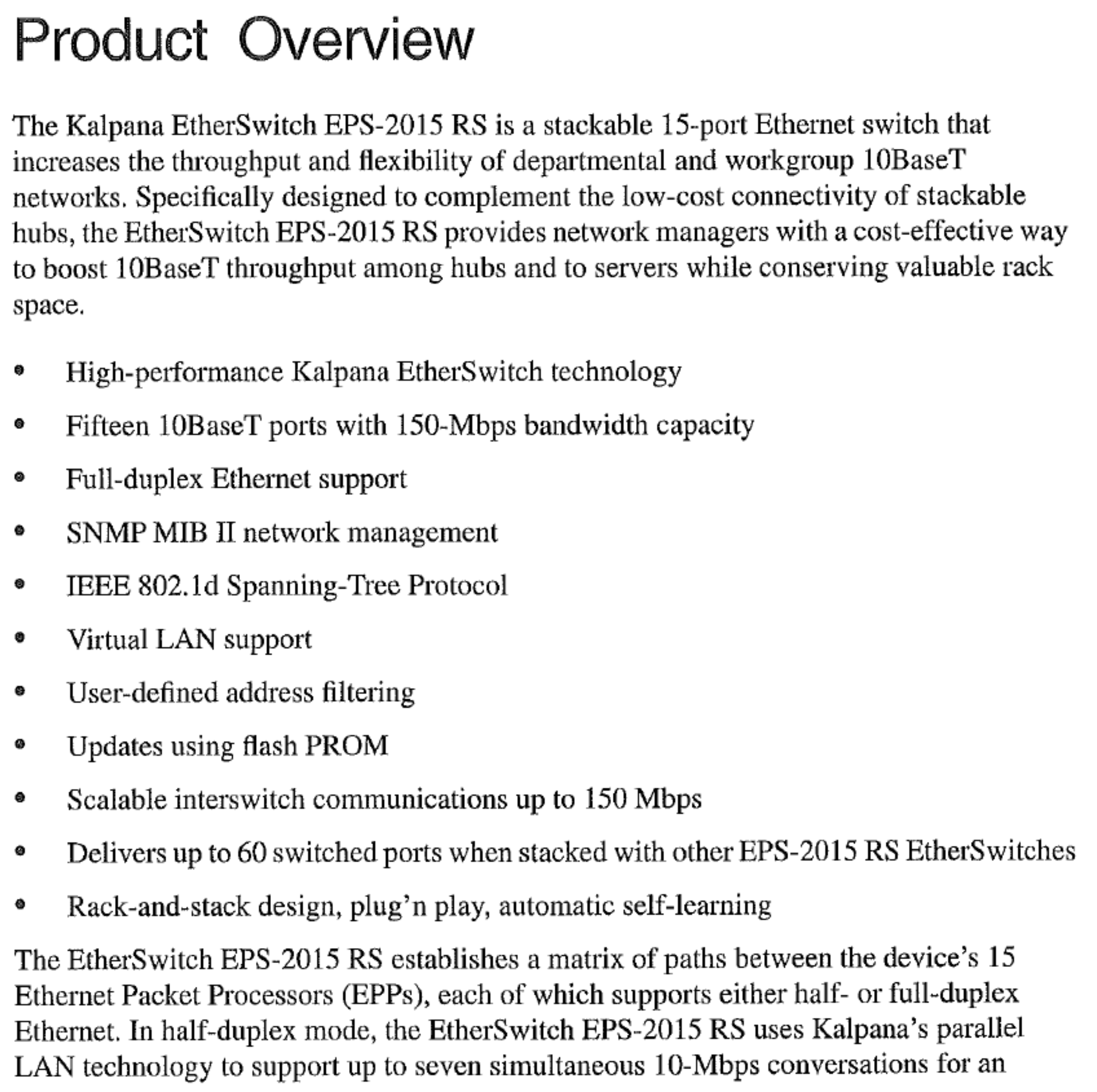
No.	’740 Patent Claim 1	Cisco EtherSwitch System
		Figure 15 EtherSwitch EPS-2115M Front Panel  Figure 16 EtherSwitch EPS-2115M Control Module  Kalpana EtherSwitch® EPS-2115M FAULT DIAG PWR A PWR B SwitchProbe RST SYSREQ H2934 FAULT LED DIAG LED PWR LED A PWR LED B EIA/TIA-232 SwitchProbe port Sys Req button Reset button

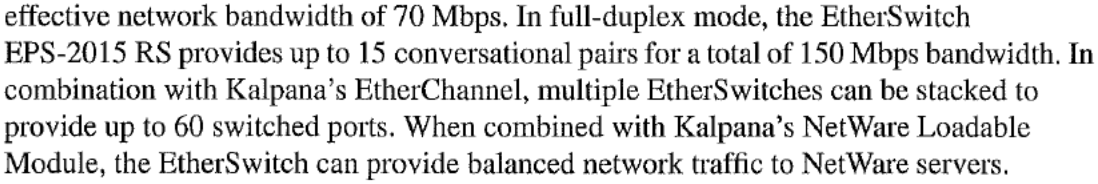
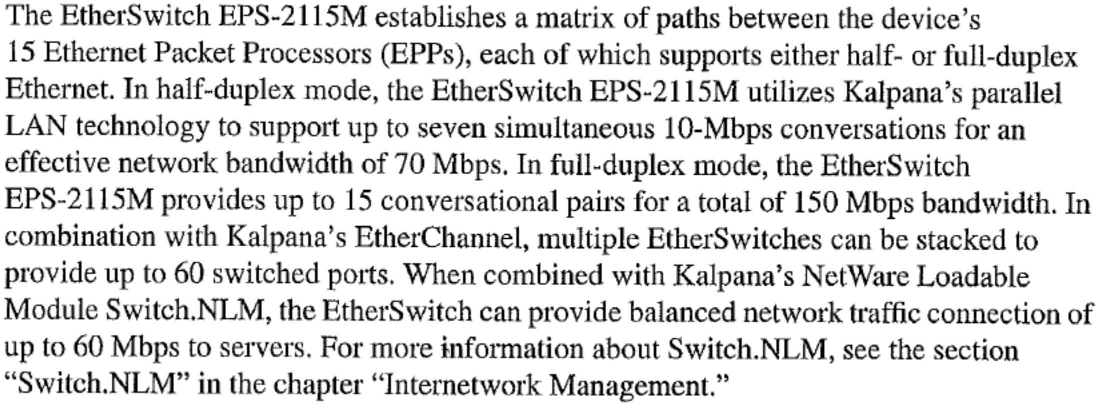
No.	’740 Patent Claim 1	Cisco EtherSwitch System
		Figure 17 EtherSwitch EPP-253 10BaseT Module  EPP 253 XMT RCV LNK/FDX MDI MDIX AUI FDPLX HDPLX H2931 Status LEDs MDI/MDI-X switch RJ-45 ports AUI port Duplex switches **Figure 18 EPP-255 10Base2 Module**  EPP 255 XMT RCV TERM AUI H2932 Status LEDs BNC ports AUI select AUI port **Figure 19 EPP-254 10BaseFL Module**  EPP 254 XMT RCV LNK/FDX TX RX FDPLX HDPLX H2933 Status LEDs Fiber ports Duplex switches Cisco EtherSwitch System Catalog at 120

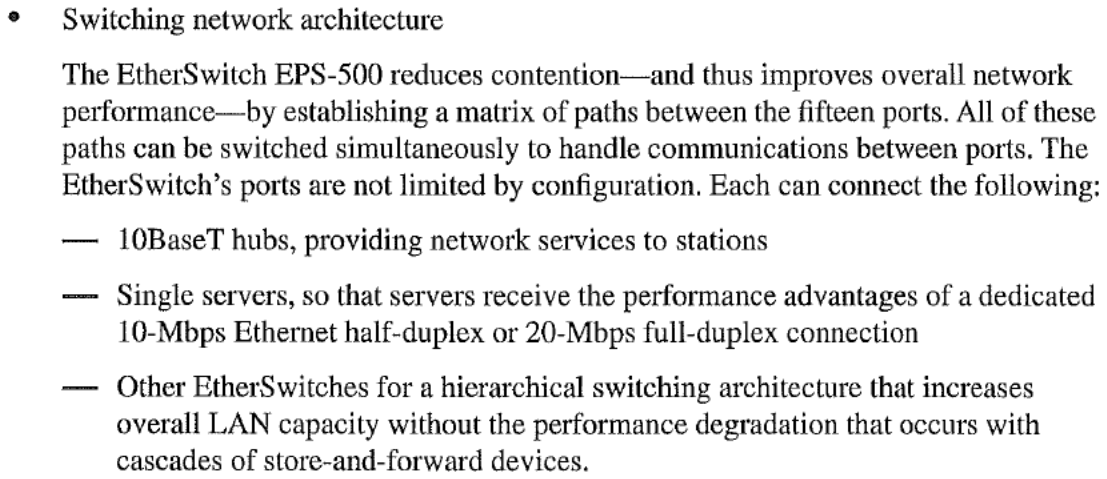
No.	’740 Patent Claim 1	Cisco EtherSwitch System
		 The EtherSwitch EPS-2115M establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2115M utilizes Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2115M provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module Switch.NLM, the EtherSwitch can provide balanced network traffic connection of up to 60 Mbps to servers. For more information about Switch.NLM, see the section “Switch.NLM” in the chapter “Internetwork Management.”  The EtherSwitch EPS-2115M features one control module and three slots for the five-port Ethernet modules, as shown in Figure 15. The control module, which is shown in Figure 16, contains status LEDs, a serial console port, a SwitchProbe connector for a network monitoring device, and reset switches. The individual Ethernet modules are hot swappable and support the following media: • 10BaseT twisted pair (Figure 17) • 10Base2 coaxial (Figure 18) • 10BaseFL fiber optic (Figure 19)
1[b]	at least one of said first physical links being a bi-directional link operative to communicate in both an upstream direction and a	Cisco EtherSwitch System discloses at least one of said first physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction. For example, Cisco EtherSwitch System discloses full duplex communication over the Ethernet ports and links connecting network devices. A full duplex communication means inbound and outbound traffic are capable of being exchanged. Cisco EtherSwitch System Catalog at 111

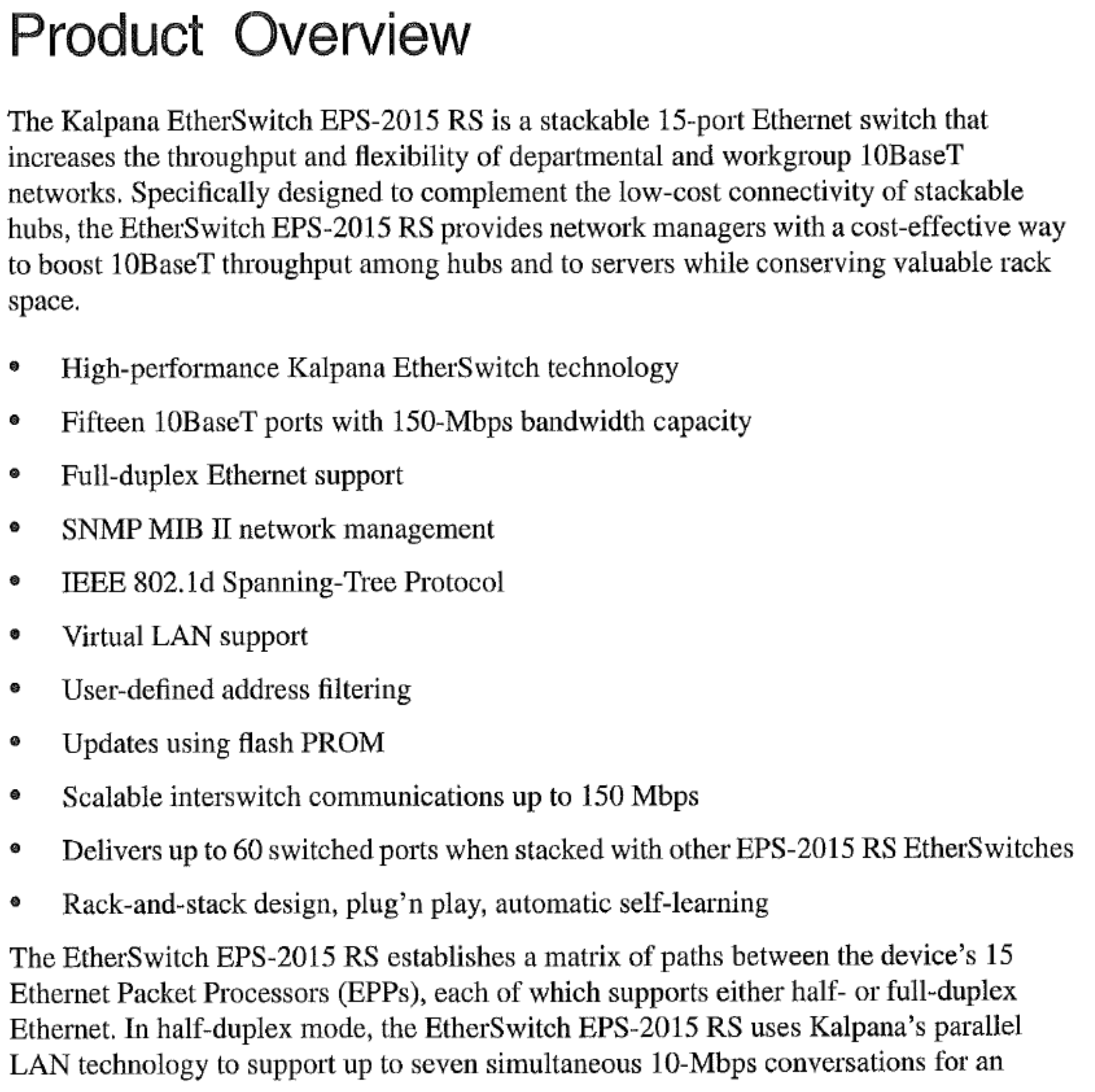
No.	’740 Patent Claim 1	Cisco EtherSwitch System
	downstream direction	 **Product Overview** The EtherSwitch EPS-500 10BaseT Accelerator is a notebook-sized Ethernet switch that doubles the effective throughput of 10BaseT workgroups. The device utilizes Kalpana's parallel LAN architecture technology to support two simultaneous conversations, doubling the effective Ethernet throughput to 20 Mbps. The EtherSwitch EPS-500 includes the following features: • Five 10BaseT ports • One AUI port • High-performance Kalpana EtherSwitch technology • SNMP MIB II Network Management • Plug’n play, automatic self-learning of network configuration • Fully IEEE 802.3 compliant • Near zero port-to-port switching delay (74,400 pps/37,200 pps) • Full-duplex support The EtherSwitch EPS-500 establishes a matrix of paths between the device’s five ports, each with a bandwidth of 10 Mbps. Kalpana’s unique parallel LAN technology supports two simultaneous 10-Mbps conversations for an effective network throughput of 20 Mbps. Additionally, the EtherSwitch EPS-500 uses “on-the-fly” switching instead of traditional store-and-forward technology. This approach streams data through the switch so that the leading edge of a packet exits the switch before the trailing edge enters. The low latency of this EtherSwitch enables packets to be forwarded between segments 20 times faster than conventional store-and-forward bridges. The EtherSwitch EPS-500 supports full-duplex Ethernet communications, providing a 20-Mbps link between full-duplex devices.

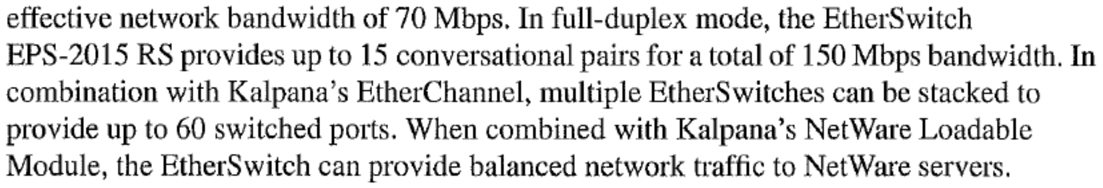
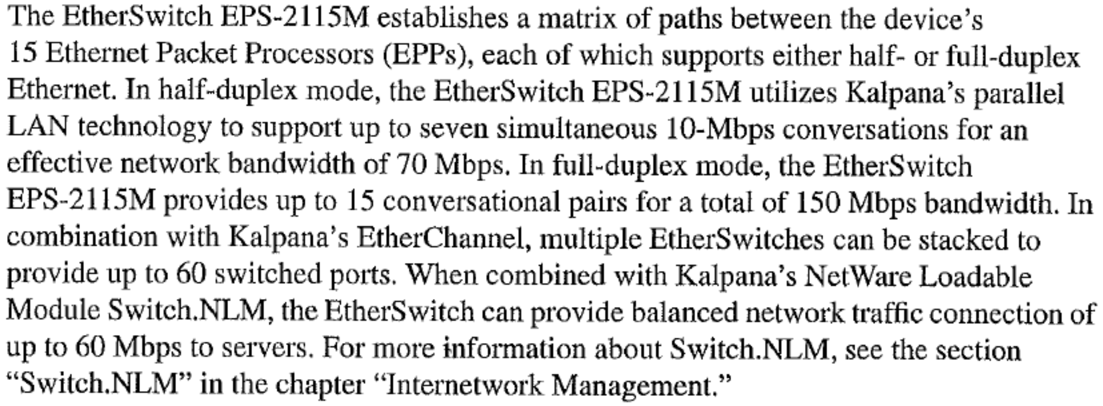
No.	’740 Patent Claim 1	Cisco EtherSwitch System
		Cisco EtherSwitch System Catalog at 114  • Full-duplex Ethernet There are two applications for full-duplex Ethernet: interconnecting EtherSwitches and connecting the EtherSwitch to a server. Multiple EPS-500 EtherSwitches can be interconnected by using the full-duplex Ethernet option. When connected, a simultaneous collision-free transmit and receive conversation occurs each at 10 Mbps, providing an effective throughput of 20 Mbps. Network performance is dramatically improved, virtually doubling the number of packets that can be exchanged between full-duplex devices. Cisco EtherSwitch System Catalog at 115

No.	’740 Patent Claim 1	Cisco EtherSwitch System
		 **Product Overview** The Kalpana EtherSwitch EPS-2015 RS is a stackable 15-port Ethernet switch that increases the throughput and flexibility of departmental and workgroup 10BaseT networks. Specifically designed to complement the low-cost connectivity of stackable hubs, the EtherSwitch EPS-2015 RS provides network managers with a cost-effective way to boost 10BaseT throughput among hubs and to servers while conserving valuable rack space. • High-performance Kalpana EtherSwitch technology • Fifteen 10BaseT ports with 150-Mbps bandwidth capacity • Full-duplex Ethernet support • SNMP MIB II network management • IEEE 802.1d Spanning-Tree Protocol • Virtual LAN support • User-defined address filtering • Updates using flash PROM • Scalable interswitch communications up to 150 Mbps • Delivers up to 60 switched ports when stacked with other EPS-2015 RS EtherSwitches • Rack-and-stack design, plug’n play, automatic self-learning The EtherSwitch EPS-2015 RS establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2015 RS uses Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an

No.	’740 Patent Claim 1	Cisco EtherSwitch System
		 effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2015 RS provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module, the EtherSwitch can provide balanced network traffic to NetWare servers. Cisco EtherSwitch System Catalog at 120  The EtherSwitch EPS-2115M establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2115M utilizes Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2115M provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module Switch.NLM, the EtherSwitch can provide balanced network traffic connection of up to 60 Mbps to servers. For more information about Switch.NLM, see the section “Switch.NLM” in the chapter “Internetwork Management.”
1[c]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel,	Cisco EtherSwitch System discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. For example, Cisco EtherSwitch System discloses a matrix of paths between network devices and the network to facilitate communication including EtherSwitch network modules that connect to the communication network via paths and connections that handle communications between ports. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Cisco EtherSwitch System is found to not meet this limitation, coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel would have been obvious to a person having ordinary skill in the art, as explained below.

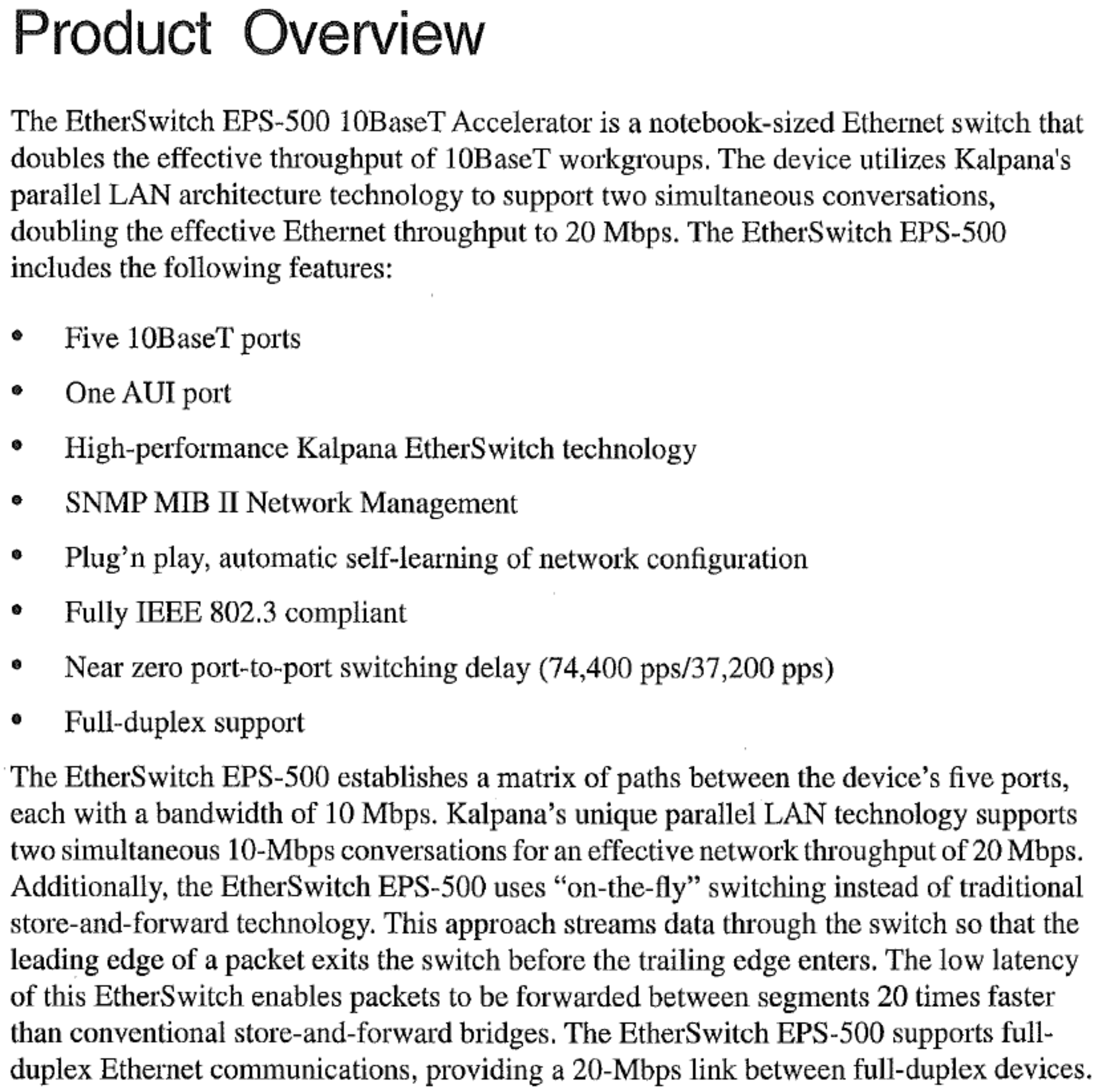
No.	’740 Patent Claim 1	Cisco EtherSwitch System
		Cisco EtherSwitch System Catalog at 114  • Switching network architecture The EtherSwitch EPS-500 reduces contention—and thus improves overall network performance—by establishing a matrix of paths between the fifteen ports. All of these paths can be switched simultaneously to handle communications between ports. The EtherSwitch’s ports are not limited by configuration. Each can connect the following: — 10BaseT hubs, providing network services to stations — Single servers, so that servers receive the performance advantages of a dedicated 10-Mbps Ethernet half-duplex or 20-Mbps full-duplex connection — Other EtherSwitches for a hierarchical switching architecture that increases overall LAN capacity without the performance degradation that occurs with cascades of store-and-forward devices. Cisco EtherSwitch System Catalog at 115

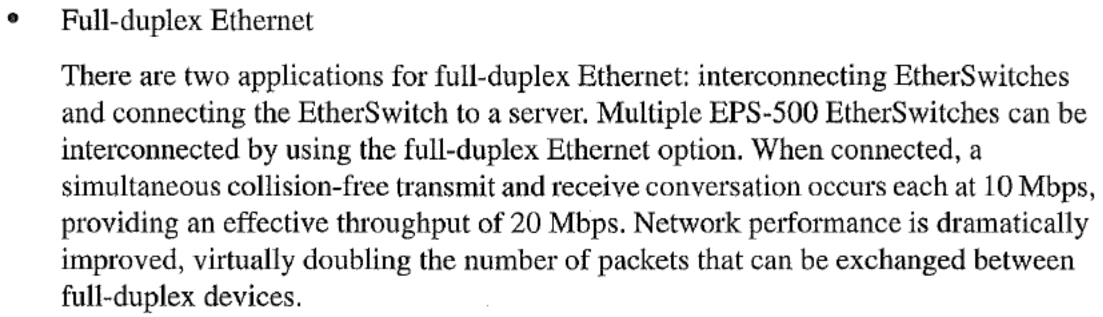
No.	’740 Patent Claim 1	Cisco EtherSwitch System
		 **Product Overview** The Kalpana EtherSwitch EPS-2015 RS is a stackable 15-port Ethernet switch that increases the throughput and flexibility of departmental and workgroup 10BaseT networks. Specifically designed to complement the low-cost connectivity of stackable hubs, the EtherSwitch EPS-2015 RS provides network managers with a cost-effective way to boost 10BaseT throughput among hubs and to servers while conserving valuable rack space. • High-performance Kalpana EtherSwitch technology • Fifteen 10BaseT ports with 150-Mbps bandwidth capacity • Full-duplex Ethernet support • SNMP MIB II network management • IEEE 802.1d Spanning-Tree Protocol • Virtual LAN support • User-defined address filtering • Updates using flash PROM • Scalable interswitch communications up to 150 Mbps • Delivers up to 60 switched ports when stacked with other EPS-2015 RS EtherSwitches • Rack-and-stack design, plug’n play, automatic self-learning The EtherSwitch EPS-2015 RS establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2015 RS uses Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an

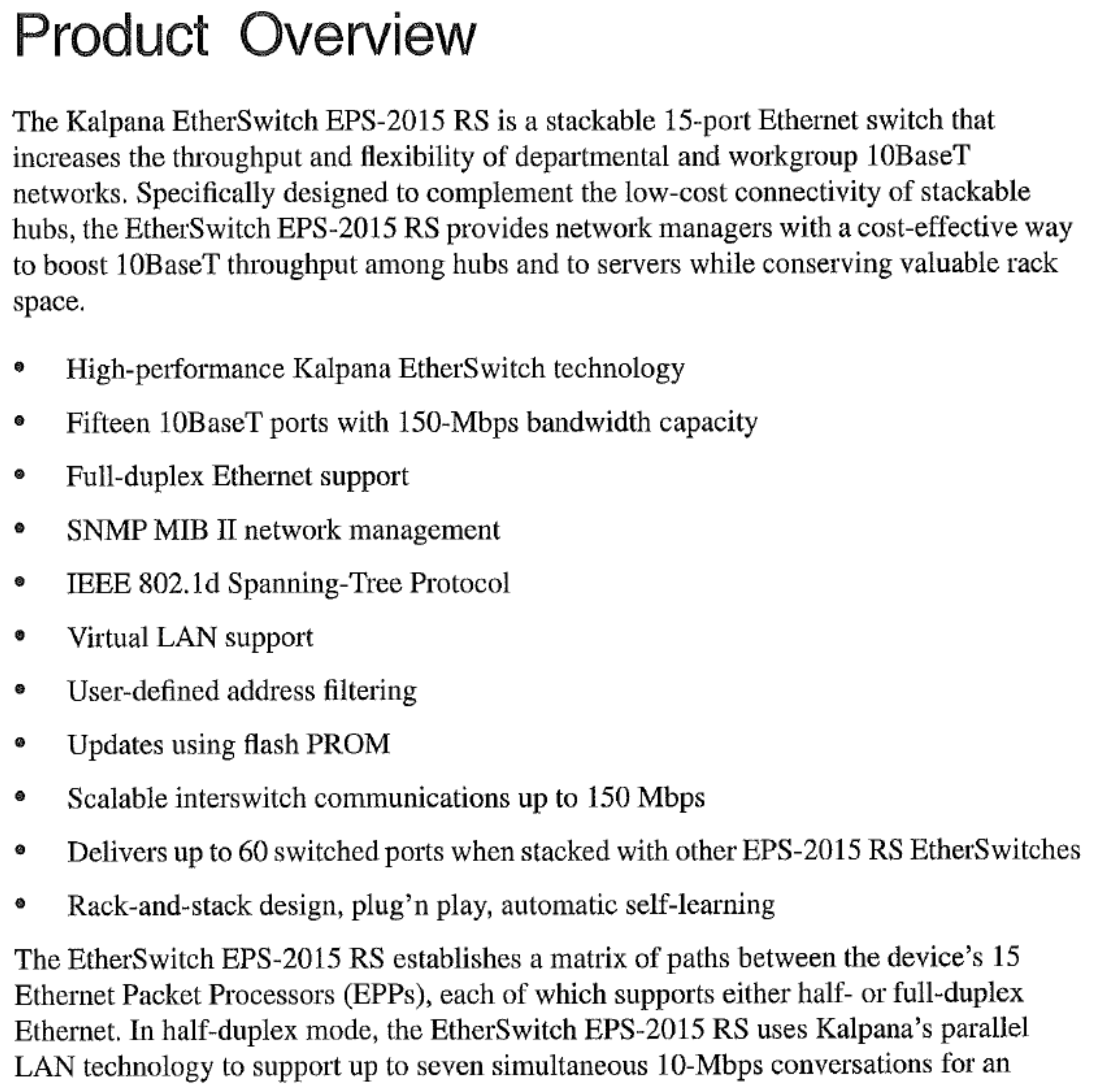
No.	’740 Patent Claim 1	Cisco EtherSwitch System
		 effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2015 RS provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module, the EtherSwitch can provide balanced network traffic to NetWare servers. Cisco EtherSwitch System Catalog at 120  The EtherSwitch EPS-2115M establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2115M utilizes Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2115M provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module Switch.NLM, the EtherSwitch can provide balanced network traffic connection of up to 60 Mbps to servers. For more information about Switch.NLM, see the section “Switch.NLM” in the chapter “Internetwork Management.” Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Cisco EtherSwitch System in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 1[c] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of three such references. For example, Ghosh discloses connecting the line cards to the active supervisor via the backplane using parallel interface circuitry. Ghosh at [0059] (“Line cards 803, 805, and 807 can communicate with an active supervisor 811 through interface circuitry 883, 885, and 887 and the backplane 815. According to various embodiments, each line card includes a plurality of ports that can act as either input ports or

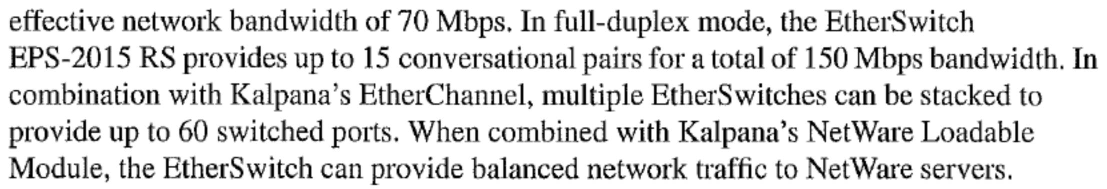
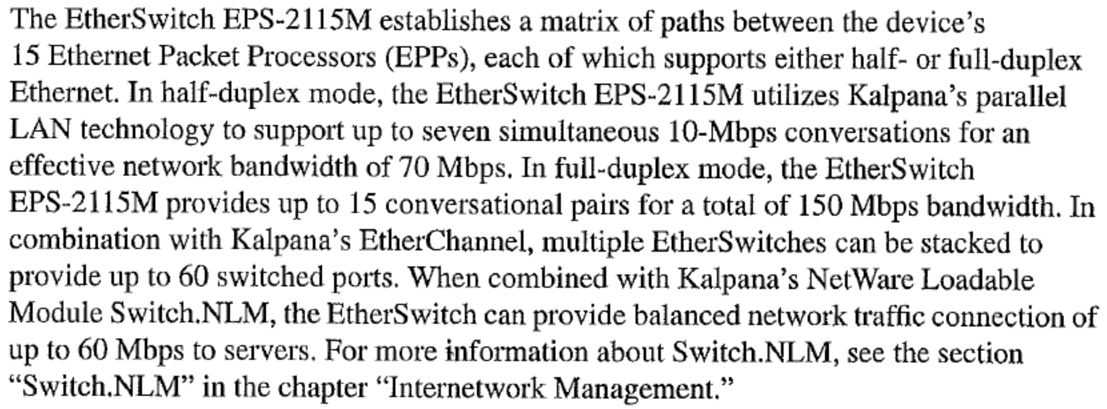
No.	’740 Patent Claim 1	Cisco EtherSwitch System
		output ports for communication with external fibre channel network entities 851 and 853. The backplane 815 can provide a communi-cations channel for all traffic between line cards and super- visors. Individual line cards 803 and 807 can also be coupled to external fibre channel network entities 851 and 853 through fibre channel ports 843 and 847.”) As another example, Bruckman discloses connecting the line cards to the network using traces comprising a backplane. Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”) For example, Basso discloses coupling the blades to the communication network via link connections to the switch fabric. Basso at [0009] (“A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more ports. These ports may be connected to another one or more network devices. The switch fabric may be configured to direct incoming packets of data to particular blades where one or more of the network proces-sors in the recipient blade may be configured to process the received packets.”)

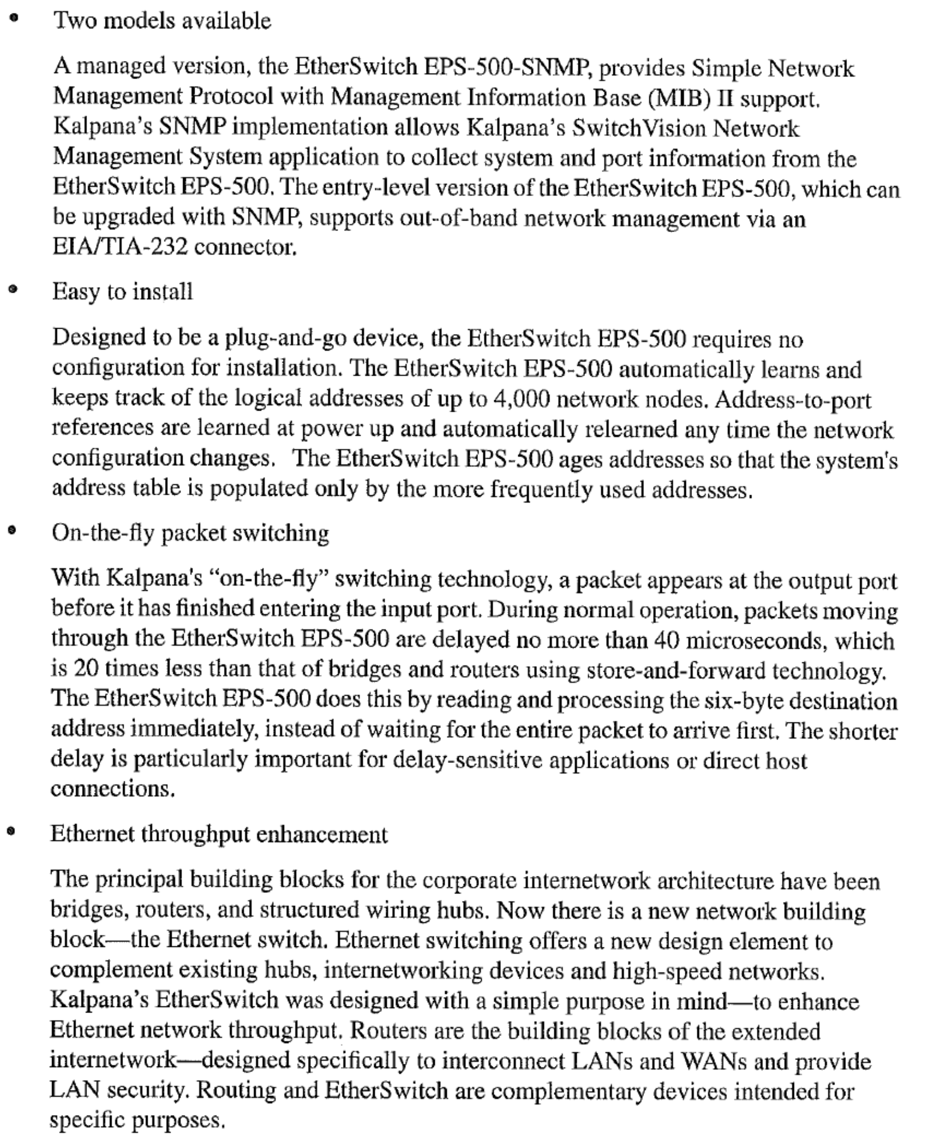
No.	’740 Patent Claim 1	Cisco EtherSwitch System
1[d]	at least one of said second physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction;	Cisco EtherSwitch System discloses at least one of said second physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction. For example, Cisco EtherSwitch System discloses full duplex communication over the paths and connections that handle communications between ports. A full duplex communication means inbound and outbound traffic are capable of being exchanged. Cisco EtherSwitch System Catalog at 111

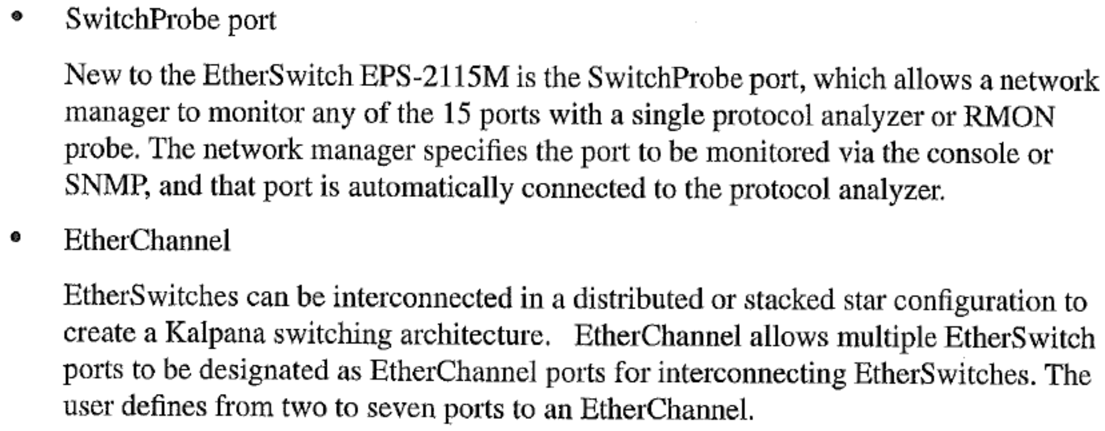
No.	’740 Patent Claim 1	Cisco EtherSwitch System
		 Product Overview The EtherSwitch EPS-500 10BaseT Accelerator is a notebook-sized Ethernet switch that doubles the effective throughput of 10BaseT workgroups. The device utilizes Kalpana's parallel LAN architecture technology to support two simultaneous conversations, doubling the effective Ethernet throughput to 20 Mbps. The EtherSwitch EPS-500 includes the following features: • Five 10BaseT ports • One AUI port • High-performance Kalpana EtherSwitch technology • SNMP MIB II Network Management • Plug’n play, automatic self-learning of network configuration • Fully IEEE 802.3 compliant • Near zero port-to-port switching delay (74,400 pps/37,200 pps) • Full-duplex support The EtherSwitch EPS-500 establishes a matrix of paths between the device’s five ports, each with a bandwidth of 10 Mbps. Kalpana’s unique parallel LAN technology supports two simultaneous 10-Mbps conversations for an effective network throughput of 20 Mbps. Additionally, the EtherSwitch EPS-500 uses “on-the-fly” switching instead of traditional store-and-forward technology. This approach streams data through the switch so that the leading edge of a packet exits the switch before the trailing edge enters. The low latency of this EtherSwitch enables packets to be forwarded between segments 20 times faster than conventional store-and-forward bridges. The EtherSwitch EPS-500 supports full-duplex Ethernet communications, providing a 20-Mbps link between full-duplex devices.

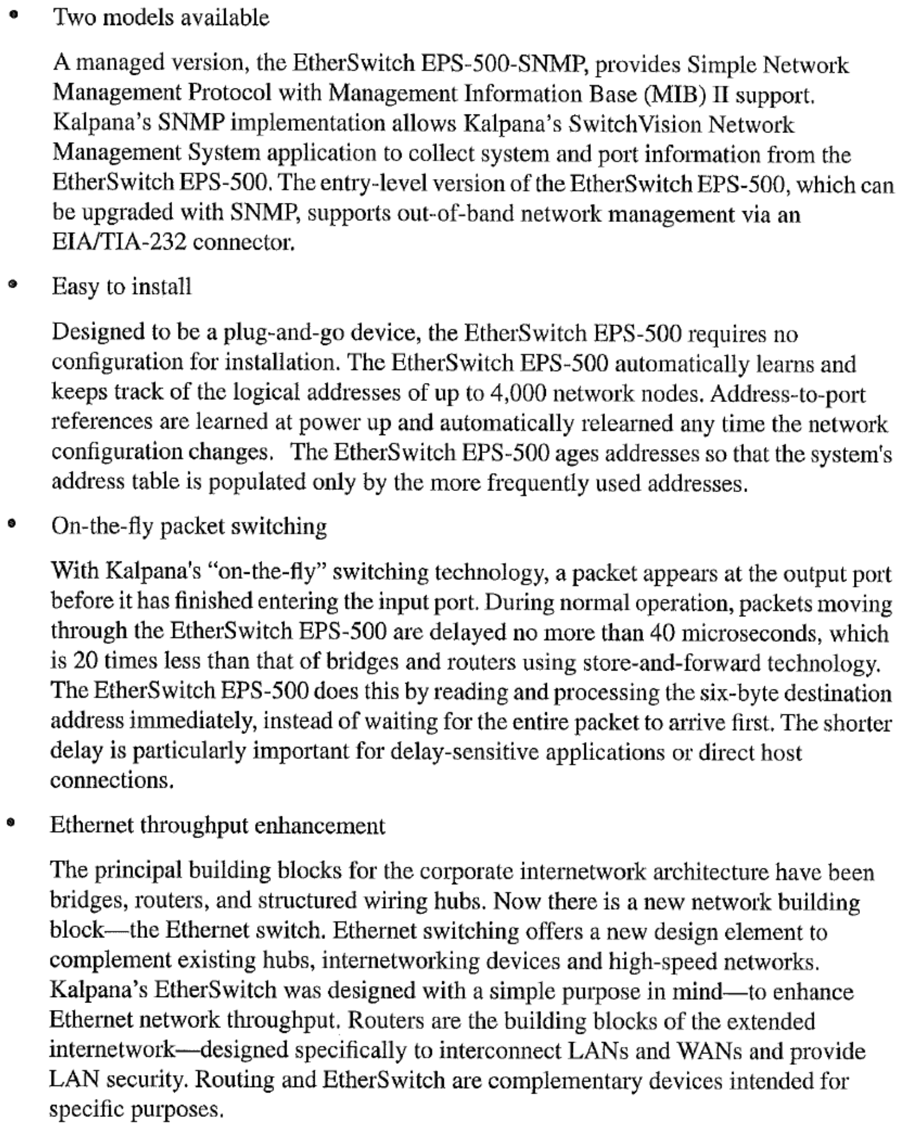
No.	’740 Patent Claim 1	Cisco EtherSwitch System
		Cisco EtherSwitch System Catalog at 114  - Full-duplex Ethernet There are two applications for full-duplex Ethernet: interconnecting EtherSwitches and connecting the EtherSwitch to a server. Multiple EPS-500 EtherSwitches can be interconnected by using the full-duplex Ethernet option. When connected, a simultaneous collision-free transmit and receive conversation occurs each at 10 Mbps, providing an effective throughput of 20 Mbps. Network performance is dramatically improved, virtually doubling the number of packets that can be exchanged between full-duplex devices. Cisco EtherSwitch System Catalog at 115

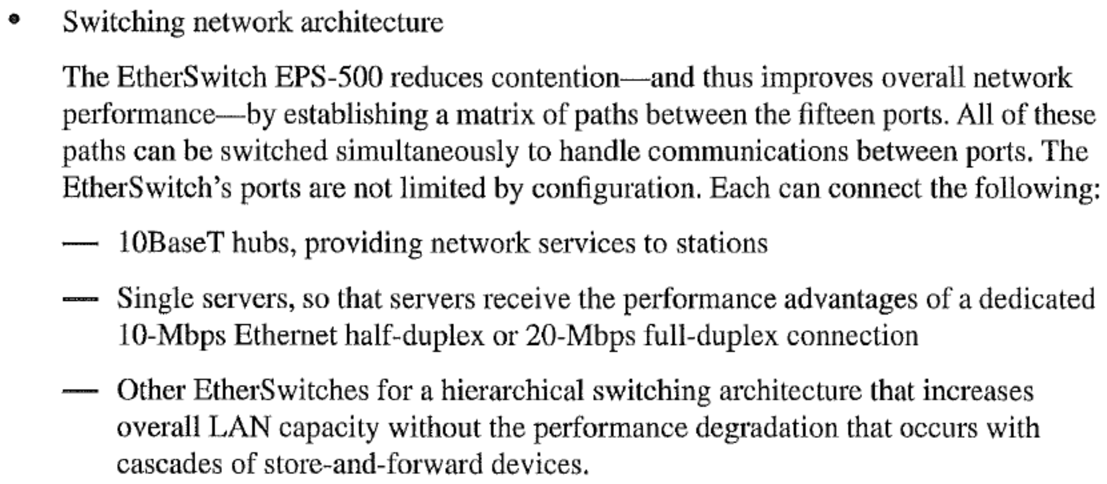
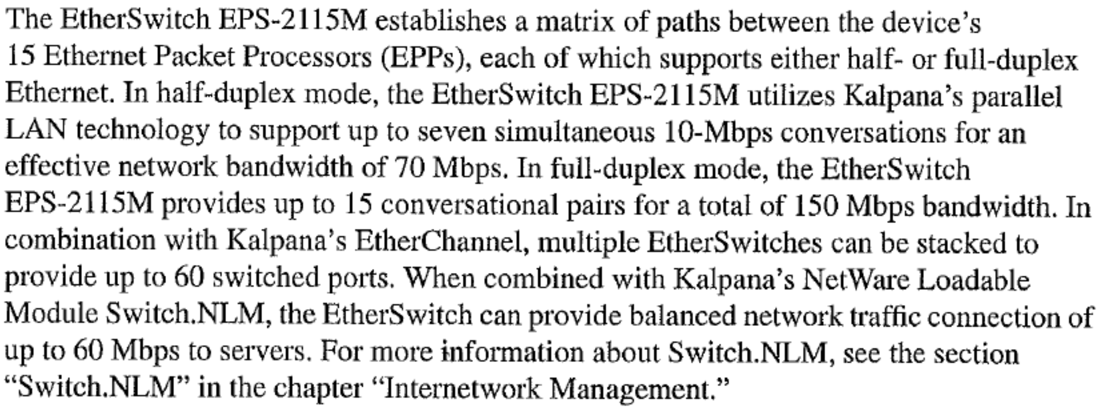
No.	’740 Patent Claim 1	Cisco EtherSwitch System
		 **Product Overview** The Kalpana EtherSwitch EPS-2015 RS is a stackable 15-port Ethernet switch that increases the throughput and flexibility of departmental and workgroup 10BaseT networks. Specifically designed to complement the low-cost connectivity of stackable hubs, the EtherSwitch EPS-2015 RS provides network managers with a cost-effective way to boost 10BaseT throughput among hubs and to servers while conserving valuable rack space. • High-performance Kalpana EtherSwitch technology • Fifteen 10BaseT ports with 150-Mbps bandwidth capacity • Full-duplex Ethernet support • SNMP MIB II network management • IEEE 802.1d Spanning-Tree Protocol • Virtual LAN support • User-defined address filtering • Updates using flash PROM • Scalable interswitch communications up to 150 Mbps • Delivers up to 60 switched ports when stacked with other EPS-2015 RS EtherSwitches • Rack-and-stack design, plug’n play, automatic self-learning The EtherSwitch EPS-2015 RS establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2015 RS uses Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an

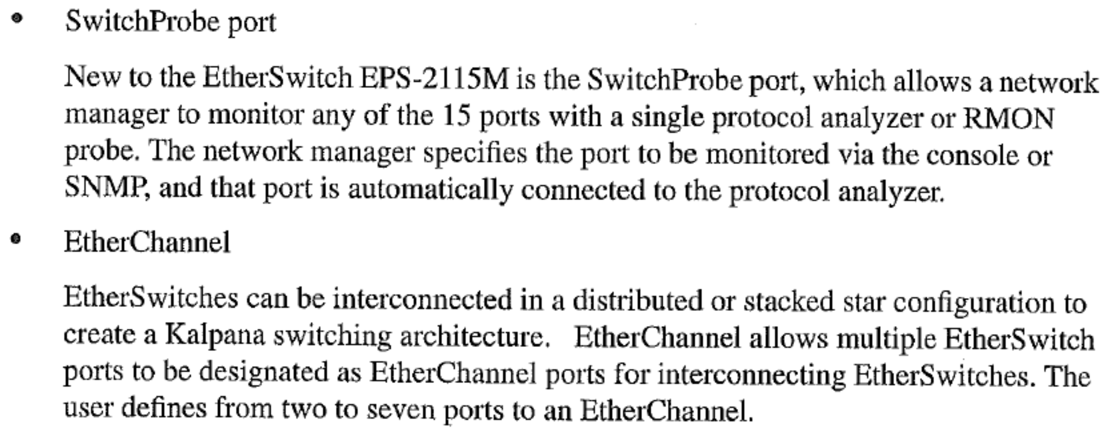
No.	’740 Patent Claim 1	Cisco EtherSwitch System
		 effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2015 RS provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module, the EtherSwitch can provide balanced network traffic to NetWare servers. Cisco EtherSwitch System Catalog at 120  The EtherSwitch EPS-2115M establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2115M utilizes Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2115M provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module Switch.NLM, the EtherSwitch can provide balanced network traffic connection of up to 60 Mbps to servers. For more information about Switch.NLM, see the section “Switch.NLM” in the chapter “Internetwork Management.”
1[e]	receiving a data frame having frame attributes sent between the communication network and the network node:	Cisco EtherSwitch System discloses receiving a data frame having frame attributes sent between the communication network and the network node. For example, Cisco EtherSwitch System discloses receiving data packets with specific data information, such as a destination address that is used for routing packets between the network and network devices. Cisco EtherSwitch System Catalog at 113

No.	’740 Patent Claim 1	Cisco EtherSwitch System
		 • Two models available A managed version, the EtherSwitch EPS-500-SNMP, provides Simple Network Management Protocol with Management Information Base (MIB) II support. Kalpana’s SNMP implementation allows Kalpana’s SwitchVision Network Management System application to collect system and port information from the EtherSwitch EPS-500. The entry-level version of the EtherSwitch EPS-500, which can be upgraded with SNMP, supports out-of-band network management via an EIA/TIA-232 connector. • Easy to install Designed to be a plug-and-go device, the EtherSwitch EPS-500 requires no configuration for installation. The EtherSwitch EPS-500 automatically learns and keeps track of the logical addresses of up to 4,000 network nodes. Address-to-port references are learned at power up and automatically relearned any time the network configuration changes. The EtherSwitch EPS-500 ages addresses so that the system's address table is populated only by the more frequently used addresses. • On-the-fly packet switching With Kalpana's “on-the-fly” switching technology, a packet appears at the output port before it has finished entering the input port. During normal operation, packets moving through the EtherSwitch EPS-500 are delayed no more than 40 microseconds, which is 20 times less than that of bridges and routers using store-and-forward technology. The EtherSwitch EPS-500 does this by reading and processing the six-byte destination address immediately, instead of waiting for the entire packet to arrive first. The shorter delay is particularly important for delay-sensitive applications or direct host connections. • Ethernet throughput enhancement The principal building blocks for the corporate internetwork architecture have been bridges, routers, and structured wiring hubs. Now there is a new network building block—the Ethernet switch. Ethernet switching offers a new design element to complement existing hubs, internetworking devices and high-speed networks. Kalpana’s EtherSwitch was designed with a simple purpose in mind—to enhance Ethernet network throughput. Routers are the building blocks of the extended internetwork—designed specifically to interconnect LANs and WANs and provide LAN security. Routing and EtherSwitch are complementary devices intended for specific purposes.

No.	’740 Patent Claim 1	Cisco EtherSwitch System
		Cisco EtherSwitch System Catalog at 122  • SwitchProbe port New to the EtherSwitch EPS-2115M is the SwitchProbe port, which allows a network manager to monitor any of the 15 ports with a single protocol analyzer or RMON probe. The network manager specifies the port to be monitored via the console or SNMP, and that port is automatically connected to the protocol analyzer. • EtherChannel EtherSwitches can be interconnected in a distributed or stacked star configuration to create a Kalpana switching architecture. EtherChannel allows multiple EtherSwitch ports to be designated as EtherChannel ports for interconnecting EtherSwitches. The user defines from two to seven ports to an EtherChannel.
1[f]:	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	Cisco EtherSwitch System discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. For example, Cisco EtherSwitch System discloses determining the path over which to send data packets based on packet information, such as the packet destination address by routing over ports and internal paths and connections. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Cisco EtherSwitch System is found to not meet this limitation, selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group would have been obvious to a person having ordinary skill in the art, as explained below. Cisco EtherSwitch System Catalog at 113

No.	’740 Patent Claim 1	Cisco EtherSwitch System
		 - Two models available A managed version, the EtherSwitch EPS-500-SNMP, provides Simple Network Management Protocol with Management Information Base (MIB) II support. Kalpana’s SNMP implementation allows Kalpana’s SwitchVision Network Management System application to collect system and port information from the EtherSwitch EPS-500. The entry-level version of the EtherSwitch EPS-500, which can be upgraded with SNMP, supports out-of-band network management via an EIA/TIA-232 connector. - Easy to install Designed to be a plug-and-go device, the EtherSwitch EPS-500 requires no configuration for installation. The EtherSwitch EPS-500 automatically learns and keeps track of the logical addresses of up to 4,000 network nodes. Address-to-port references are learned at power up and automatically relearned any time the network configuration changes. The EtherSwitch EPS-500 ages addresses so that the system's address table is populated only by the more frequently used addresses. - On-the-fly packet switching With Kalpana's “on-the-fly” switching technology, a packet appears at the output port before it has finished entering the input port. During normal operation, packets moving through the EtherSwitch EPS-500 are delayed no more than 40 microseconds, which is 20 times less than that of bridges and routers using store-and-forward technology. The EtherSwitch EPS-500 does this by reading and processing the six-byte destination address immediately, instead of waiting for the entire packet to arrive first. The shorter delay is particularly important for delay-sensitive applications or direct host connections. - Ethernet throughput enhancement The principal building blocks for the corporate internetwork architecture have been bridges, routers, and structured wiring hubs. Now there is a new network building block—the Ethernet switch. Ethernet switching offers a new design element to complement existing hubs, internetworking devices and high-speed networks. Kalpana’s EtherSwitch was designed with a simple purpose in mind—to enhance Ethernet network throughput. Routers are the building blocks of the extended internetwork—designed specifically to interconnect LANs and WANs and provide LAN security. Routing and EtherSwitch are complementary devices intended for specific purposes.

No.	’740 Patent Claim 1	Cisco EtherSwitch System
		Cisco EtherSwitch System Catalog at 114  • Switching network architecture The EtherSwitch EPS-500 reduces contention—and thus improves overall network performance—by establishing a matrix of paths between the fifteen ports. All of these paths can be switched simultaneously to handle communications between ports. The EtherSwitch’s ports are not limited by configuration. Each can connect the following: — 10BaseT hubs, providing network services to stations — Single servers, so that servers receive the performance advantages of a dedicated 10-Mbps Ethernet half-duplex or 20-Mbps full-duplex connection — Other EtherSwitches for a hierarchical switching architecture that increases overall LAN capacity without the performance degradation that occurs with cascades of store-and-forward devices. Cisco EtherSwitch System Catalog at 120  The EtherSwitch EPS-2115M establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2115M utilizes Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2115M provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module Switch.NLM, the EtherSwitch can provide balanced network traffic connection of up to 60 Mbps to servers. For more information about Switch.NLM, see the section “Switch.NLM” in the chapter “Internetwork Management.”

No.	’740 Patent Claim 1	Cisco EtherSwitch System
		Cisco EtherSwitch System Catalog at 122  • SwitchProbe port New to the EtherSwitch EPS-2115M is the SwitchProbe port, which allows a network manager to monitor any of the 15 ports with a single protocol analyzer or RMON probe. The network manager specifies the port to be monitored via the console or SNMP, and that port is automatically connected to the protocol analyzer. • EtherChannel EtherSwitches can be interconnected in a distributed or stacked star configuration to create a Kalpana switching architecture. EtherChannel allows multiple EtherSwitch ports to be designated as EtherChannel ports for interconnecting EtherSwitches. The user defines from two to seven ports to an EtherChannel. Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Cisco EtherSwitch System in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 1[f] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Basso discloses using a hash function and index table to select for blade/port combinations over which to send the packet over a user port a switch fabric link. Basso further discloses that this selection is performed based upon packet information. Basso at [0010] (“Upon a network processor receiving a packet of data, the network processor may index into a table, commonly referred to as a forwarding table, to determine the table associated with a particular logical interface as well as the next destination address. The forwarding table may comprise a plurality of entries where each entry may comprise information indicating a particular table associated with a particular logical interface as well as the next destination address. Each logical interface may be associated with a table storing a plurality of entries containing blade/ port

No.	’740 Patent Claim 1	Cisco EtherSwitch System
		combinations as discussed further below. In one embodiment, an entry may be indexed in the forwarding table using a destination address in the received packet header.”) Basso at [0011] (“A hash function may then be performed on the received packet to generate a hash value. In one embodiment, a hash function may be performed on the source and destination address in the packet header to generate a hash value.”) Basso at [0012] (“The hash value generated may be used to index into the table associated with a particular logical interface. Upon indexing into the table associated with the logical interface, an appropriate blade/port combination may be identified to transmit the received packet of data. In one embodiment, a blade/port combination may be selected in the indexed entry of the table associated with the logical interface by using a portion of the bits of the hashed value. The received packet may then be transmitted through the identified blade/port combination to the next destination (next destination previously identified by the next destination address in the forwarding table).”) Basso at [0035] (“By logically grouping a plurality of ports 404 coupled to a particular network device into a logical interface 405, network processor 403 may be configured to transmit processed packets to that particular network device via any blade 402/port 404 combination grouped in that logical interface 405. For example, referring to FIG. 4, ports 404A-404I are physically connected to router 104B. If ports 404A-404I were logically grouped into logical interface 405, then a particular network processor 403, e.g., network processor 403A, may be configured to transmit processed packets that are determined to be transmitted to router 104B through any of ports 404A-404I in blades 402A-C, respectively. Network processor 403, e.g., network processor 403A, may be configured to transmit the processed packets to router 104B through ports 404, e.g., ports 404D-I, not in its blade 402, e.g., blade 402A, by forwarding the processed packets to switch fabric 401 which may then direct the processed packets to another appropriate physical blade 402/port 404 combination. Network processor 403, e.g., network processor 403A, may further be configured to transmit the processed packets to router 104B through any ports 404, e.g., ports 404A-C, in its blade 402, e.g., blade402A, instead of just one physical port 404 in its blade 402, e.g., blade 402A. A more detailed description of routing packets implementing logical interface(s) 405 is provided below in FIG. 5.”)

No.	’740 Patent Claim 1	Cisco EtherSwitch System
		Basso at [0040] (“In step 502, network processor 403, e.g., network processor 403A, may receive a packet of data from switch fabric 401. Upon receiving the packet of data, network processor 403, in step 503, may index into a table, commonly referred to as a forwarding table, to determine the table associated with a particular logical interface 405 as well as the next destination address, i.e., the next hop address. The forwarding table may comprise a plurality of entries where each entry may comprise information indicating a particular table associated with a particular logical interface 405 as well as the next destination address. Each logical interface 405 may be associated with a table storing a plurality of entries containing blade 402/port 404 combinations as discussed further below. In one embodiment, an entry may be indexed in the forwarding table using a destination address in the received packet header. It is noted that an entry may be indexed in the forwarding table usingother means and that such means would be recognized by an artisan of ordinary skill in the art. It is further noted that embodiments implementing such means would fall within the scope of the present invention.”) Basso at [0041] (“In step 504, a hash function may be performed on the received packet to generate a hash value. In one embodiment, a hash function may be performed on the source and destination address in the packet header to generate a hash value. It is noted that in other embodiments a hash function may be performed on different fields, e.g., port, type of service, in the received packet to generate a hash value.”) Basso at [0042] (“In step 505, the hash value generated in step 504 may be used to index into the table associated with a particular logical interface 405 determined in step 503. Upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be identified in step 506 to transmit the received packet of data as explained below.”) Basso at [0043] (“As stated above, the table associated with a particular logical interface 405 may comprise a plurality of entries where each entry may comprise a threshold value associated with a particular blade 402/port 404 combination. The threshold value may represent a percentage of the total number of packets received by router 104A that may be transmitted through the blade 402/port

<table>
<tr><th>No.</th><th>’740 Patent Claim 1</th><th>Cisco EtherSwitch System</th></tr>
<tr><td></td><td></td><td>
404 combination associated with that threshold value. In one embodiment, the threshold value may be updated periodically by a user, e.g., system administrator, in control of router 104, e.g., router 104A. For example, the threshold value, e.g., twenty percent of the number of packets received by router 104A, associated with a particular blade 402/port 404 combination may be updated by lowering the threshold value by one percent during each update. An example of an entry of the table associated with a particular logical interface 405 is shown in Table 1 below:

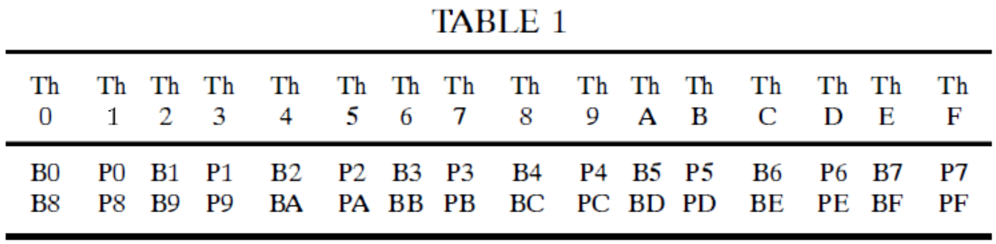


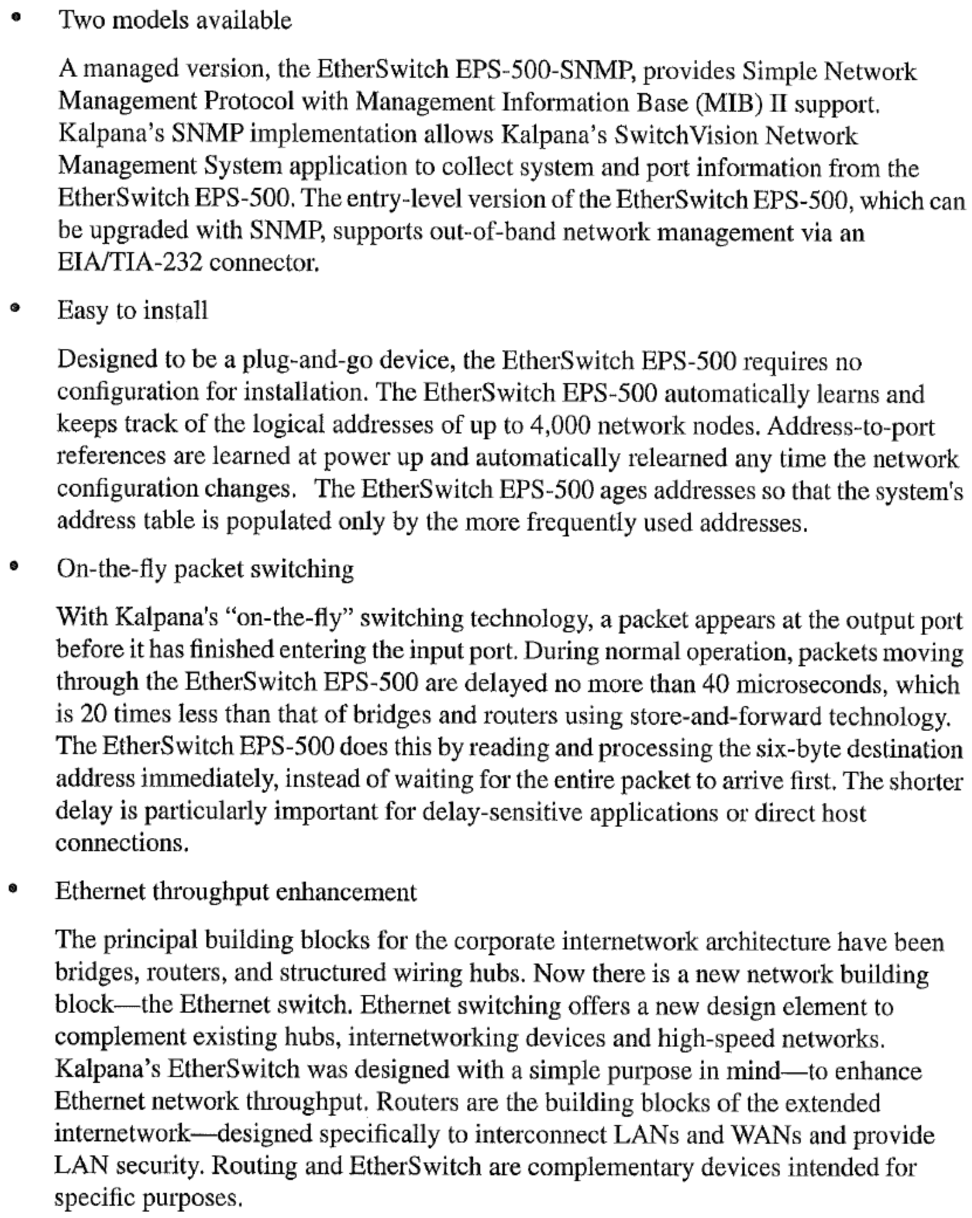
TABLE 1

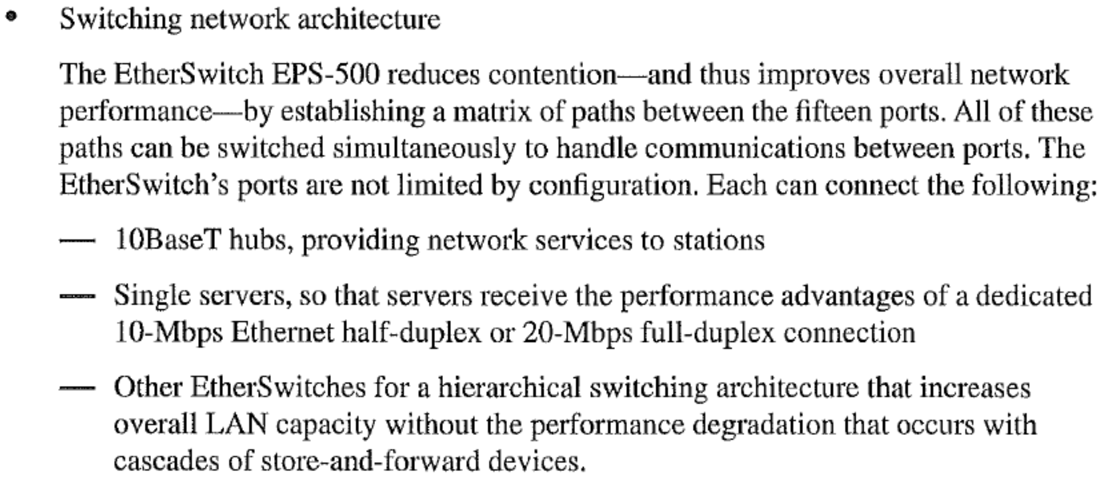
<table>
<tr><th>Th
0</th><th>Th
1</th><th>Th
2</th><th>Th
3</th><th>Th
4</th><th>Th
5</th><th>Th
6</th><th>Th
7</th><th>Th
8</th><th>Th
9</th><th>Th
A</th><th>Th
B</th><th>Th
C</th><th>Th
D</th><th>Th
E</th><th>Th
F</th></tr>
<tr><td>B0
B8</td><td>P0
P8</td><td>B1
B9</td><td>P1
P9</td><td>B2
BA</td><td>P2
PA</td><td>B3
BB</td><td>P3
PB</td><td>B4
BC</td><td>P4
PC</td><td>B5
BD</td><td>P5
PD</td><td>B6
BE</td><td>P6
PE</td><td>B7
BF</td><td>P7
PF</td></tr>
</table>

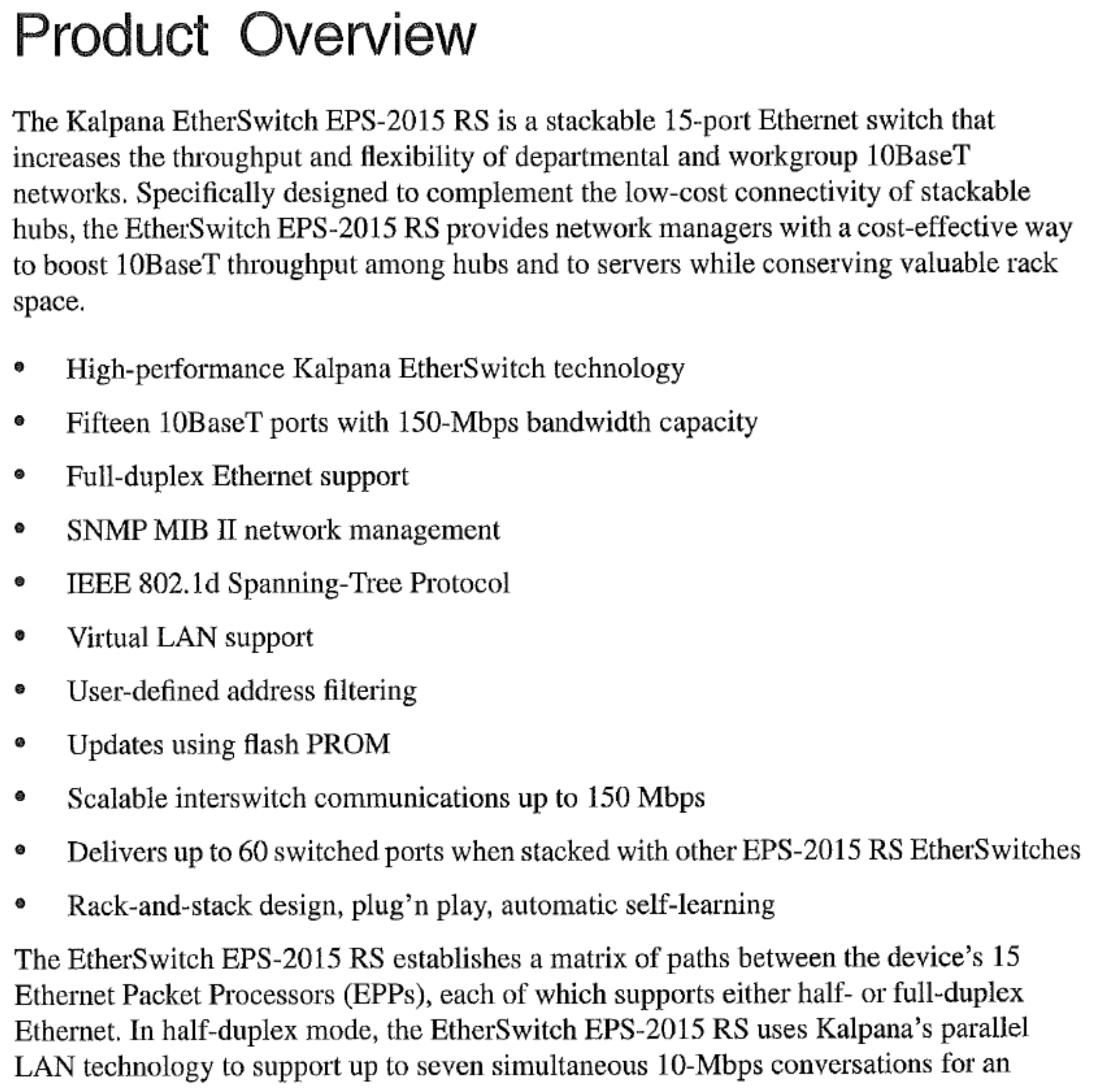
Basso at [0044] (“Table 1 above illustrates an exemplary entry in the table associated with a particular logical interface 405. Each entry may comprise a plurality of threshold values (16 threshold values in exemplary Table 1) where each threshold value is associated with a particular blade 402/port 404 combination. For example, threshold value (Th0) is associated with blade B0/port P0 combination where blade BO may refer to a particular blade 402, e.g., blade 402B, and port PO may refer to a particular port 404, e.g., port 404E. Threshold value (Thl) is associated with blade Bl/port Pl combination and so forth. As stated above, each threshold value may represent a percentage of the total number of packets received by router 104A that may be transmitted through the blade 402/port 404 combination associated with that threshold value. For example, threshold value (Th0) may represent a percentage of the total number of packets received by router 104A that may be transmitted through port PO in blade BO. If port PO refers to port 404D and blade BO refers to blade 402B, then if Th0 has a value of twenty percent, a maximum of twenty percent of the total packets received by router 104A may be transmitted through port 404D in blade 402B.”)
</td></tr>
</table>

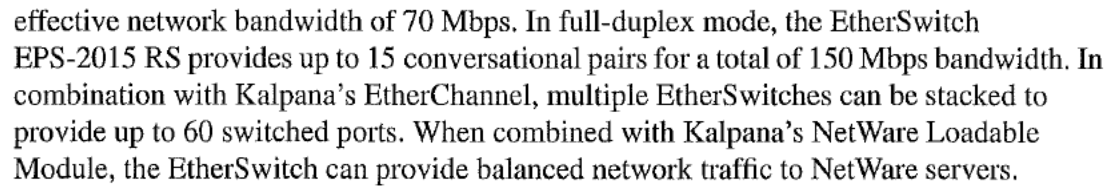
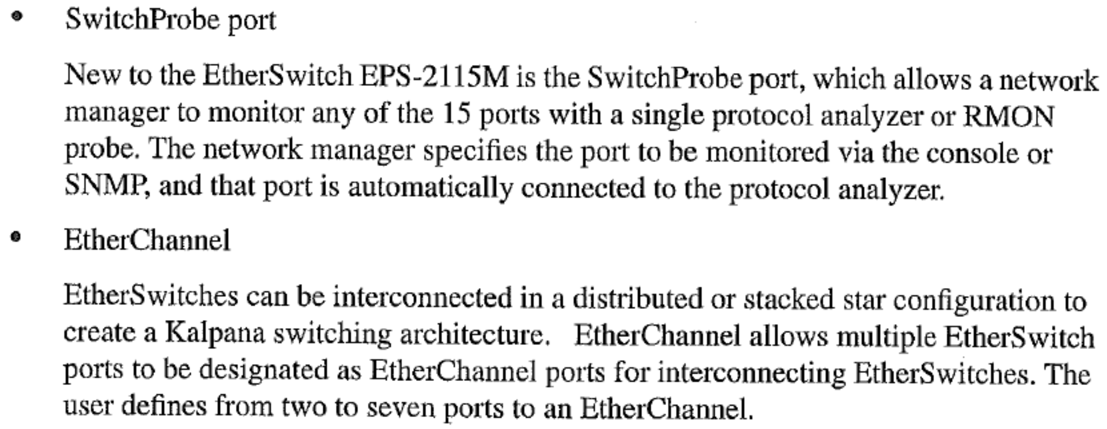
No.	’740 Patent Claim 1	Cisco EtherSwitch System
		Basso at [0045] (“As stated above, upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be identified in step 506 to transmit the received packet of data. In one embodiment, the hash value generated in step 504 may be used to select a particular threshold value and hence a blade 402/port 404 combination associated with the selected threshold value. In one embodiment, a portion of the bits of the hash value, e.g., most significant bits, may be used to select a particular threshold value in the entry indexed in step 505. For example, referring to Table 1, since there are 16 different threshold values in each entry of the table associated with logical interface 405, only four bits of the hash value generated in step 504 may be used to select a threshold value. Upon selecting a threshold value, the blade 402/port 404 combination associated with the selected threshold value may be used to transmit the received packet.”) As another example, Wiher discloses using cell header information to at each node to select and route the ATM data cell over a data link and selected backplane. Wiher at 3:43-65 (“In general, in another aspect, the invention features an apparatus for communicating data cells between a data link and a backplane. The apparatus includes transceiver circuitry to transmit and receive data cells over a data link and a plurality of backplane interfaces each including at least one cell signal terminal. Each of the backplane interface is coupled to a backplane interconnection circuit. Each backplane interconnection circuit transmits and receives cells over the cell signal terminals of its associated backplane interface. The apparatus also includes de-multiplexing circuitry coupling the transceiver circuitry to each of the backplane interconnection circuits. The de-multiplexing circuitry receives a data cell from the transceiver circuitry, select a backplane interconnection circuit associated with the data cell, and provide the data cell to the selected backplane interconnection circuit for transmission over the cell signal terminals of the associated backplane interface. The apparatus also includes multiplexing circuitry coupling the plurality of backplane interconnection circuits to the transceiver circuitry. The multiplexing circuitry receives data cells from each of the backplane interconnection circuits and provide the received data cells to the transceiver circuitry.”)

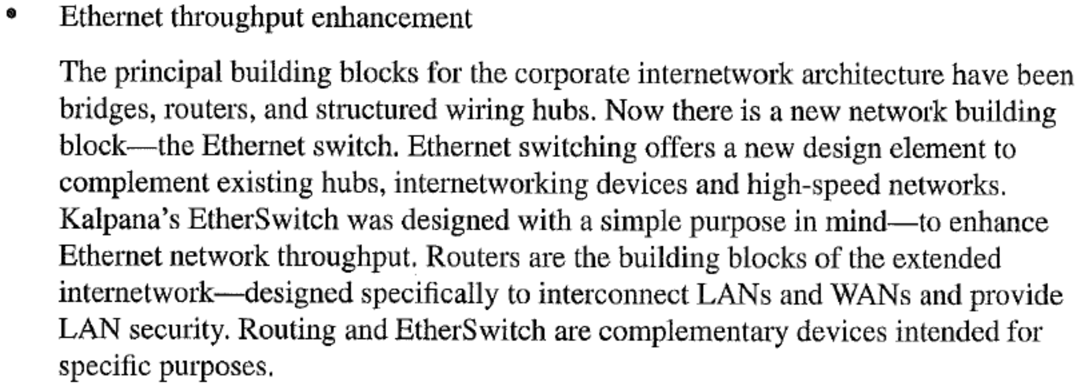
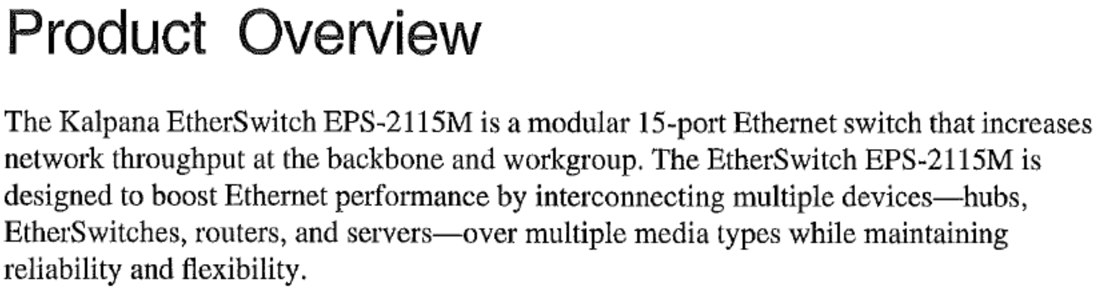
No.	’740 Patent Claim 1	Cisco EtherSwitch System
		Wiher at 3:66-4:22 (“Implementations of the invention may include one or more of the following features. The backplane interconnection circuits may independently receive and transmit data cells over the plurality of backplane interfaces. The de-multiplexing circuitry may select a backplane interface based on data in the header field of the data cell. The apparatus may include header translation circuitry to alter header data in cells sent between the plurality of backplane interfaces and the transceiver circuitry. Each of the plurality of backplane interfaces may include separate terminals to receive cells and separate terminals to transmit cells. The terminals to transmit cells may include a first and second control terminal and at least one outgoing cell data terminal. A backplane interface's backplane interconnection circuitry may accepts a signal on the first control terminal as indicating that a cell may be sent over the interface, asserts a 15 signal on the second control terminal to indicate that a cell is being transmitted, and transmits data bits of the cell on the outgoing cell data terminal. Each backplane interface may include a single outgoing cell data terminal and each bit of the cell may be serially transmitted over the single outgoing cell data terminal. Each backplane interface may include multiple outgoing cell data terminals and bits of the cell may be sent in parallel over the eight outgoing cell data terminals.”)
1[g]	sending the data frame over the selected first and second physical links,	Cisco EtherSwitch System discloses sending the data frame over the selected first and second physical links. For example, Cisco EtherSwitch System discloses sending packets over specific ports and paths. Cisco EtherSwitch System Catalog at 113

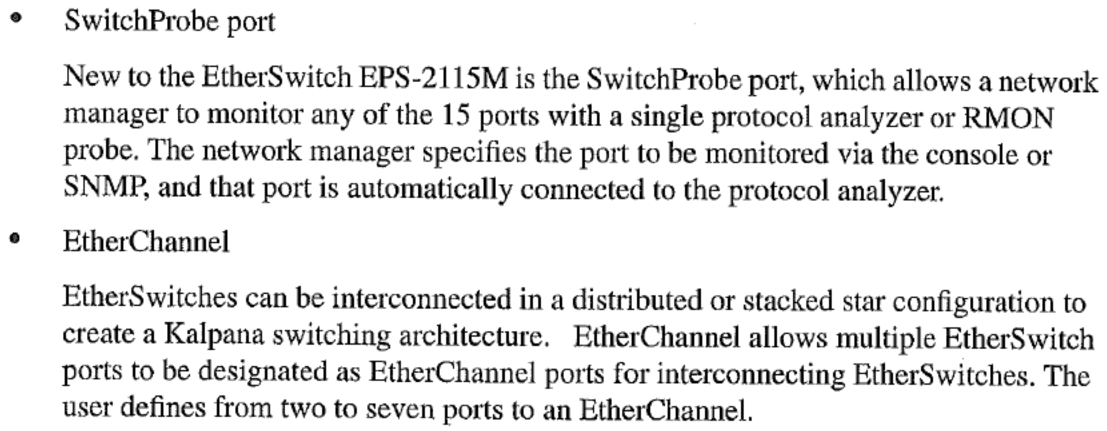
		 - Two models available A managed version, the EtherSwitch EPS-500-SNMP, provides Simple Network Management Protocol with Management Information Base (MIB) II support. Kalpana's SNMP implementation allows Kalpana's SwitchVision Network Management System application to collect system and port information from the EtherSwitch EPS-500. The entry-level version of the EtherSwitch EPS-500, which can be upgraded with SNMP, supports out-of-band network management via an EIA/TIA-232 connector. - Easy to install Designed to be a plug-and-go device, the EtherSwitch EPS-500 requires no configuration for installation. The EtherSwitch EPS-500 automatically learns and keeps track of the logical addresses of up to 4,000 network nodes. Address-to-port references are learned at power up and automatically relearned any time the network configuration changes. The EtherSwitch EPS-500 ages addresses so that the system's address table is populated only by the more frequently used addresses. - On-the-fly packet switching With Kalpana's "on-the-fly" switching technology, a packet appears at the output port before it has finished entering the input port. During normal operation, packets moving through the EtherSwitch EPS-500 are delayed no more than 40 microseconds, which is 20 times less than that of bridges and routers using store-and-forward technology. The EtherSwitch EPS-500 does this by reading and processing the six-byte destination address immediately, instead of waiting for the entire packet to arrive first. The shorter delay is particularly important for delay-sensitive applications or direct host connections. - Ethernet throughput enhancement The principal building blocks for the corporate internetwork architecture have been bridges, routers, and structured wiring hubs. Now there is a new network building block—the Ethernet switch. Ethernet switching offers a new design element to complement existing hubs, internetworking devices and high-speed networks. Kalpana's EtherSwitch was designed with a simple purpose in mind—to enhance Ethernet network throughput. Routers are the building blocks of the extended internetwork—designed specifically to interconnect LANs and WANs and provide LAN security. Routing and EtherSwitch are complementary devices intended for specific purposes.

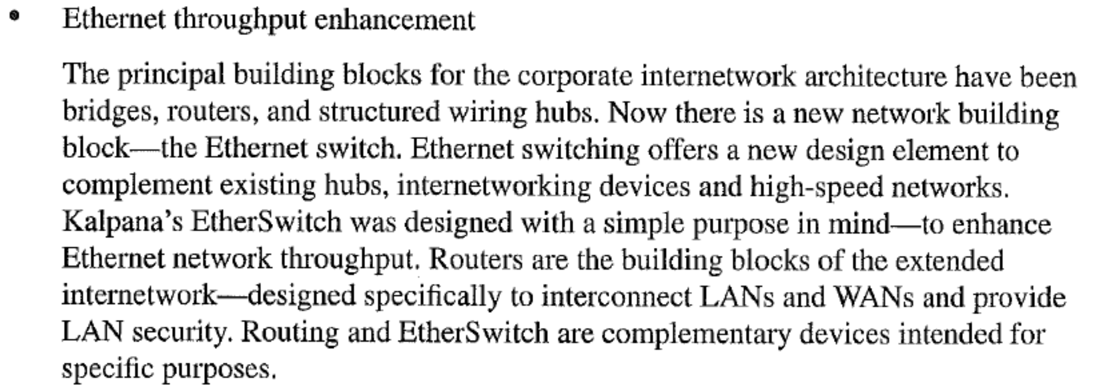
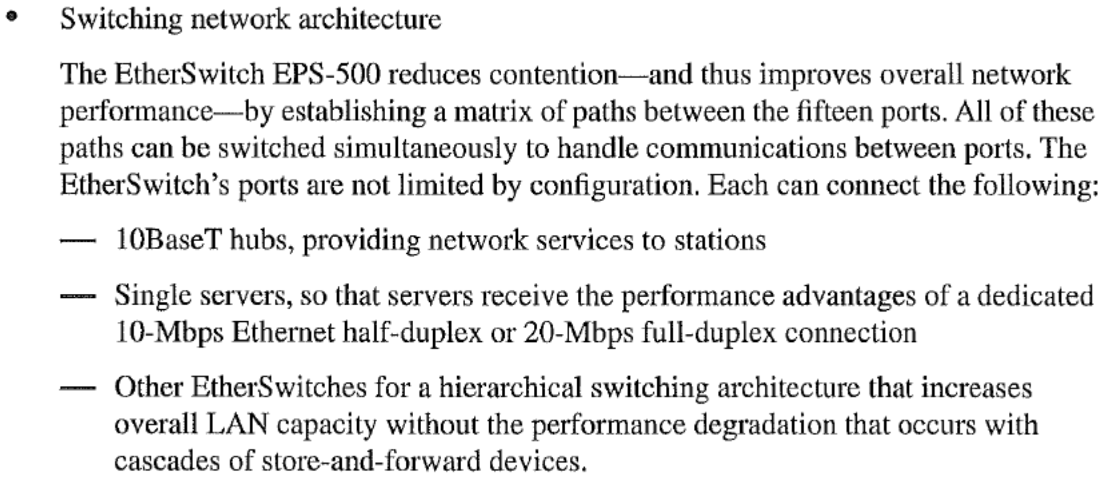
No.	’740 Patent Claim 1	Cisco EtherSwitch System
		Cisco EtherSwitch System Catalog at 114  • Switching network architecture The EtherSwitch EPS-500 reduces contention—and thus improves overall network performance—by establishing a matrix of paths between the fifteen ports. All of these paths can be switched simultaneously to handle communications between ports. The EtherSwitch’s ports are not limited by configuration. Each can connect the following: — 10BaseT hubs, providing network services to stations — Single servers, so that servers receive the performance advantages of a dedicated 10-Mbps Ethernet half-duplex or 20-Mbps full-duplex connection — Other EtherSwitches for a hierarchical switching architecture that increases overall LAN capacity without the performance degradation that occurs with cascades of store-and-forward devices. Cisco EtherSwitch System Catalog at 115

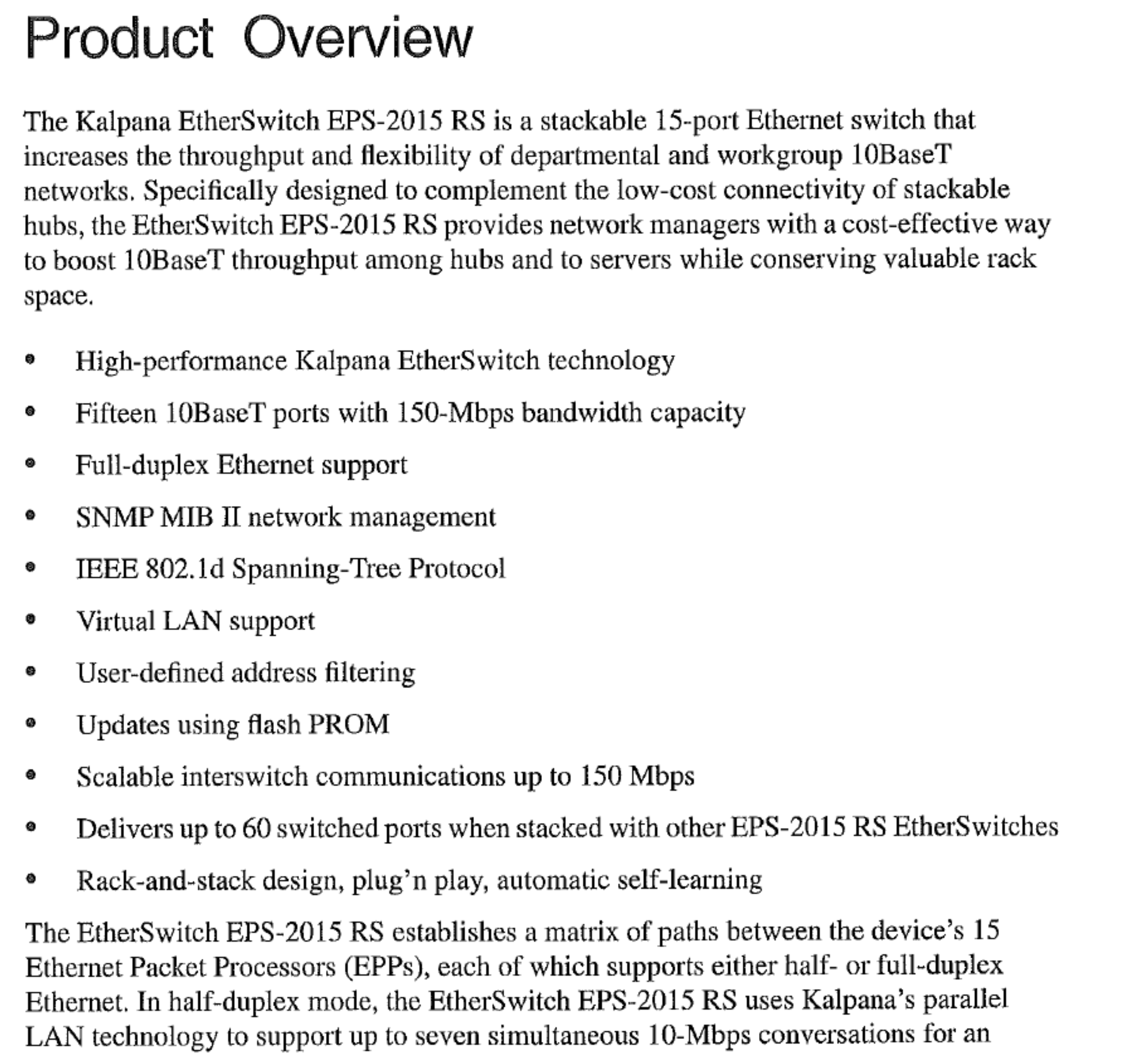
No.	’740 Patent Claim 1	Cisco EtherSwitch System
		 **Product Overview** The Kalpana EtherSwitch EPS-2015 RS is a stackable 15-port Ethernet switch that increases the throughput and flexibility of departmental and workgroup 10BaseT networks. Specifically designed to complement the low-cost connectivity of stackable hubs, the EtherSwitch EPS-2015 RS provides network managers with a cost-effective way to boost 10BaseT throughput among hubs and to servers while conserving valuable rack space. • High-performance Kalpana EtherSwitch technology • Fifteen 10BaseT ports with 150-Mbps bandwidth capacity • Full-duplex Ethernet support • SNMP MIB II network management • IEEE 802.1d Spanning-Tree Protocol • Virtual LAN support • User-defined address filtering • Updates using flash PROM • Scalable interswitch communications up to 150 Mbps • Delivers up to 60 switched ports when stacked with other EPS-2015 RS EtherSwitches • Rack-and-stack design, plug’n play, automatic self-learning The EtherSwitch EPS-2015 RS establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2015 RS uses Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an

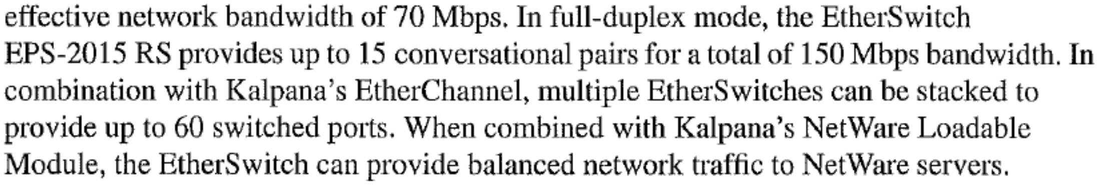
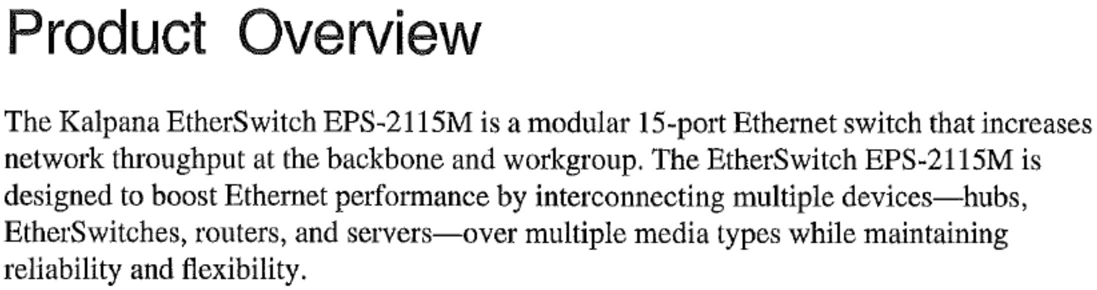
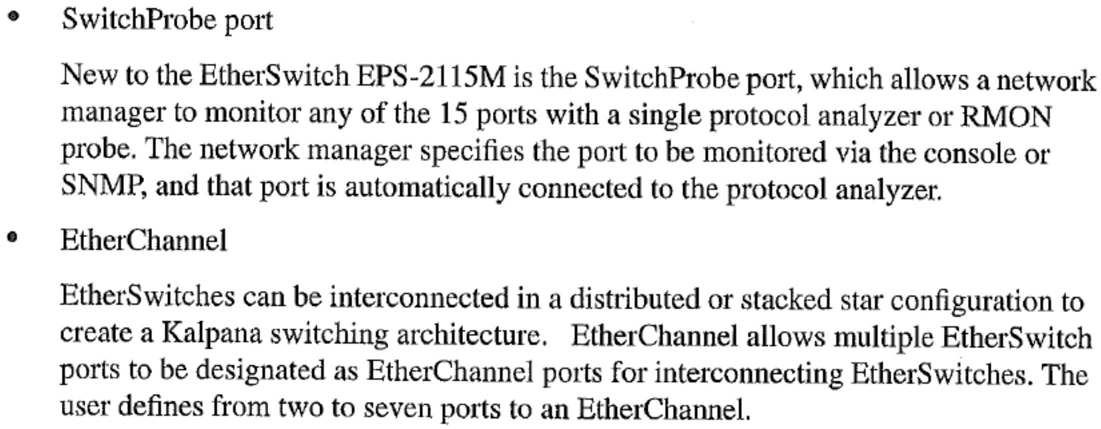
No.	’740 Patent Claim 1	Cisco EtherSwitch System
		 effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2015 RS provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module, the EtherSwitch can provide balanced network traffic to NetWare servers. Cisco EtherSwitch System Catalog at 122  - SwitchProbe port New to the EtherSwitch EPS-2115M is the SwitchProbe port, which allows a network manager to monitor any of the 15 ports with a single protocol analyzer or RMON probe. The network manager specifies the port to be monitored via the console or SNMP, and that port is automatically connected to the protocol analyzer. - EtherChannel EtherSwitches can be interconnected in a distributed or stacked star configuration to create a Kalpana switching architecture. EtherChannel allows multiple EtherSwitch ports to be designated as EtherChannel ports for interconnecting EtherSwitches. The user defines from two to seven ports to an EtherChannel.
1[h]	said sending comprising communicating along at least one of said bi-directional links.	Cisco EtherSwitch System discloses said sending comprising communicating along at least one of said bi-directional links. *See supra at* 1[b], 1[d], 1[g].

No.	’740 Patent Claim 2	Cisco EtherSwitch System
2[a]	The method according to claim 1, wherein the network node comprises a user node, and	Cisco EtherSwitch System discloses the method according to claim 1, wherein the network node comprises a user node. For example, Cisco EtherSwitch System discloses network devices, including hubs, EtherSwitches, routers, and servers, which are interconnected in an Ethernet network. A person of ordinary skill in the art would understand that these interconnect network devices could be user devices. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. Cisco EtherSwitch System Catalog at 113  • Ethernet throughput enhancement The principal building blocks for the corporate internetwork architecture have been bridges, routers, and structured wiring hubs. Now there is a new network building block—the Ethernet switch. Ethernet switching offers a new design element to complement existing hubs, internetworking devices and high-speed networks. Kalpana’s EtherSwitch was designed with a simple purpose in mind—to enhance Ethernet network throughput. Routers are the building blocks of the extended internetwork—designed specifically to interconnect LANs and WANs and provide LAN security. Routing and EtherSwitch are complementary devices intended for specific purposes. Cisco EtherSwitch System Catalog at 119  Product Overview The Kalpana EtherSwitch EPS-2115M is a modular 15-port Ethernet switch that increases network throughput at the backbone and workgroup. The EtherSwitch EPS-2115M is designed to boost Ethernet performance by interconnecting multiple devices—hubs, EtherSwitches, routers, and servers—over multiple media types while maintaining reliability and flexibility.

No.	’740 Patent Claim 2	Cisco EtherSwitch System
		Cisco EtherSwitch System Catalog at 122  • SwitchProbe port New to the EtherSwitch EPS-2115M is the SwitchProbe port, which allows a network manager to monitor any of the 15 ports with a single protocol analyzer or RMON probe. The network manager specifies the port to be monitored via the console or SNMP, and that port is automatically connected to the protocol analyzer. • EtherChannel EtherSwitches can be interconnected in a distributed or stacked star configuration to create a Kalpana switching architecture. EtherChannel allows multiple EtherSwitch ports to be designated as EtherChannel ports for interconnecting EtherSwitches. The user defines from two to seven ports to an EtherChannel.
2[b]	wherein sending the data frame comprises establishing a communication service between the user node and the communication network.	Cisco EtherSwitch System discloses wherein sending the data frame comprises establishing a communication service between the user node and the communication network. For example, Cisco EtherSwitch System discloses establishing a Ethernet network service with increased throughput for sending packets between a user network device and the network. Cisco EtherSwitch System Catalog at 113

No.	’740 Patent Claim 2	Cisco EtherSwitch System
		 - Ethernet throughput enhancement The principal building blocks for the corporate internetwork architecture have been bridges, routers, and structured wiring hubs. Now there is a new network building block—the Ethernet switch. Ethernet switching offers a new design element to complement existing hubs, internetworking devices and high-speed networks. Kalpana’s EtherSwitch was designed with a simple purpose in mind—to enhance Ethernet network throughput. Routers are the building blocks of the extended internetwork—designed specifically to interconnect LANs and WANs and provide LAN security. Routing and EtherSwitch are complementary devices intended for specific purposes. Cisco EtherSwitch System Catalog at 114  - Switching network architecture The EtherSwitch EPS-500 reduces contention—and thus improves overall network performance—by establishing a matrix of paths between the fifteen ports. All of these paths can be switched simultaneously to handle communications between ports. The EtherSwitch’s ports are not limited by configuration. Each can connect the following: — 10BaseT hubs, providing network services to stations — Single servers, so that servers receive the performance advantages of a dedicated 10-Mbps Ethernet half-duplex or 20-Mbps full-duplex connection — Other EtherSwitches for a hierarchical switching architecture that increases overall LAN capacity without the performance degradation that occurs with cascades of store-and-forward devices. Cisco EtherSwitch System Catalog at 115

No.	’740 Patent Claim 2	Cisco EtherSwitch System
		 **Product Overview** The Kalpana EtherSwitch EPS-2015 RS is a stackable 15-port Ethernet switch that increases the throughput and flexibility of departmental and workgroup 10BaseT networks. Specifically designed to complement the low-cost connectivity of stackable hubs, the EtherSwitch EPS-2015 RS provides network managers with a cost-effective way to boost 10BaseT throughput among hubs and to servers while conserving valuable rack space. • High-performance Kalpana EtherSwitch technology • Fifteen 10BaseT ports with 150-Mbps bandwidth capacity • Full-duplex Ethernet support • SNMP MIB II network management • IEEE 802.1d Spanning-Tree Protocol • Virtual LAN support • User-defined address filtering • Updates using flash PROM • Scalable interswitch communications up to 150 Mbps • Delivers up to 60 switched ports when stacked with other EPS-2015 RS EtherSwitches • Rack-and-stack design, plug’n play, automatic self-learning The EtherSwitch EPS-2015 RS establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2015 RS uses Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an

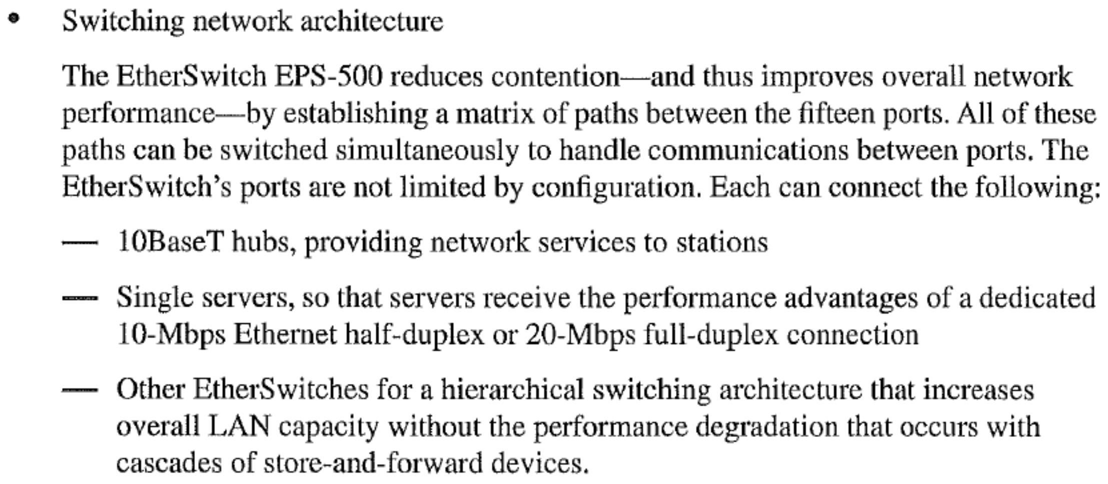
No.	’740 Patent Claim 2	Cisco EtherSwitch System
		 effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2015 RS provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module, the EtherSwitch can provide balanced network traffic to NetWare servers. Cisco EtherSwitch System Catalog at 119  Product Overview The Kalpana EtherSwitch EPS-2115M is a modular 15-port Ethernet switch that increases network throughput at the backbone and workgroup. The EtherSwitch EPS-2115M is designed to boost Ethernet performance by interconnecting multiple devices—hubs, EtherSwitches, routers, and servers—over multiple media types while maintaining reliability and flexibility. Cisco EtherSwitch System Catalog at 122  • SwitchProbe port New to the EtherSwitch EPS-2115M is the SwitchProbe port, which allows a network manager to monitor any of the 15 ports with a single protocol analyzer or RMON probe. The network manager specifies the port to be monitored via the console or SNMP, and that port is automatically connected to the protocol analyzer. • EtherChannel EtherSwitches can be interconnected in a distributed or stacked star configuration to create a Kalpana switching architecture. EtherChannel allows multiple EtherSwitch ports to be designated as EtherChannel ports for interconnecting EtherSwitches. The user defines from two to seven ports to an EtherChannel.

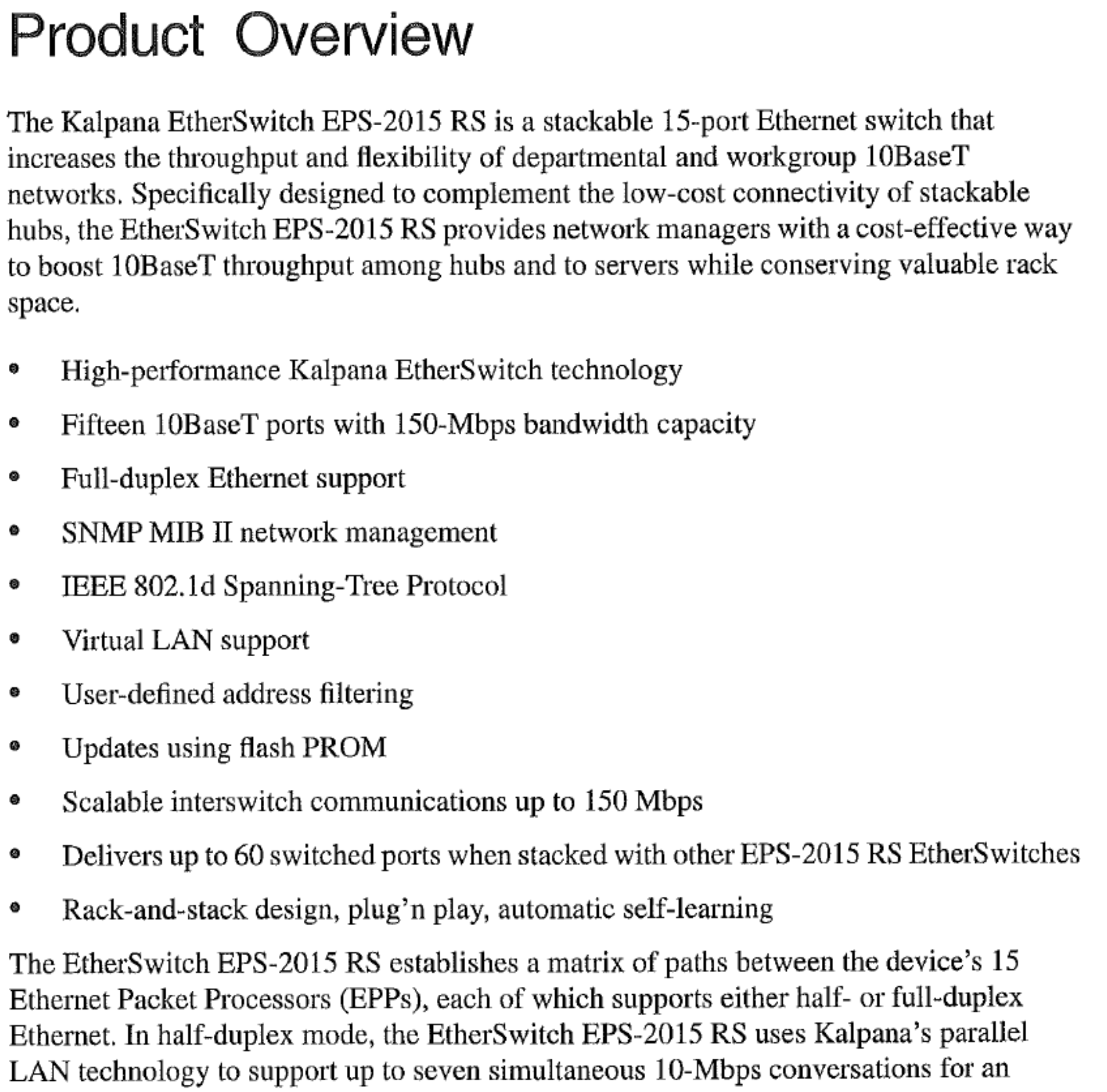
No.	’740 Patent Claim 3	Cisco EtherSwitch System
3	The method according to claim 1, wherein the second physical links comprise backplane traces formed on a backplane to which the one or more interface modules are coupled.	Cisco EtherSwitch System discloses the method according to claim 1, wherein the second physical links comprise backplane traces formed on a backplane to which the one or more interface modules are coupled. On information and belief, Cisco EtherSwitch System discloses a switch device with standard internal architecture, including a backplane. A person of ordinary skill in the art would understand that the multiple paths and connections include physical links formed on a backplane, a standard piece of switch architecture, which connect the EtherSwitch network modules to the backplane. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Cisco EtherSwitch System is found to not meet this limitation, wherein the second physical links comprise backplane traces formed on a back plane to which the one or more interface modules are coupled would have been obvious to a person having ordinary skill in the art, as explained below. Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Cisco EtherSwitch System in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 3 of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of three such references. For example, Ghosh discloses connecting the line cards, i.e., interface modules, to the active supervisor via the backplane using parallel interface circuitry. Ghosh at [0059] (“Line cards 803, 805, and 807 can communicate with an active supervisor 811 through interface circuitry 883, 885, and 887 and the backplane 815. According to various embodiments, each line card includes a plurality of ports that can act as either input ports or output ports for communication with external fibre channel network entities 851 and 853. The backplane 815 can provide a communi-cations channel for all traffic between line cards and super- visors. Individual line cards 803 and 807 can also be coupled to external fibre channel network entities 851 and 853 through fibre channel ports 843 and 847.”)

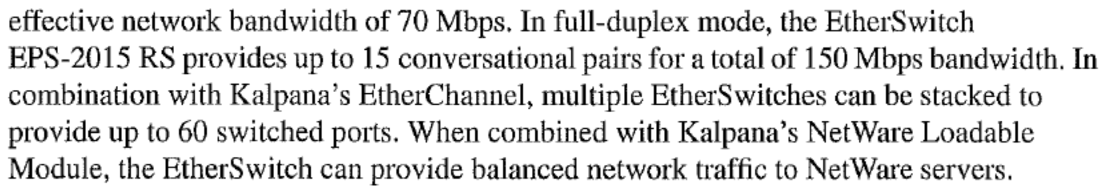
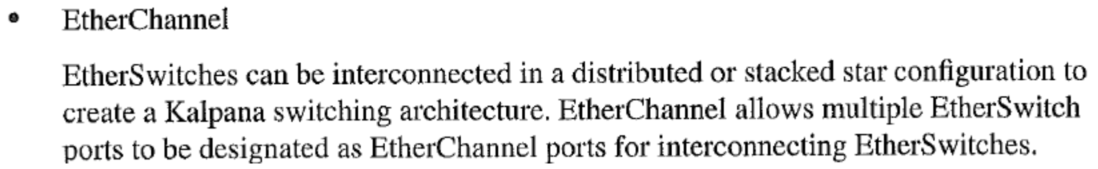
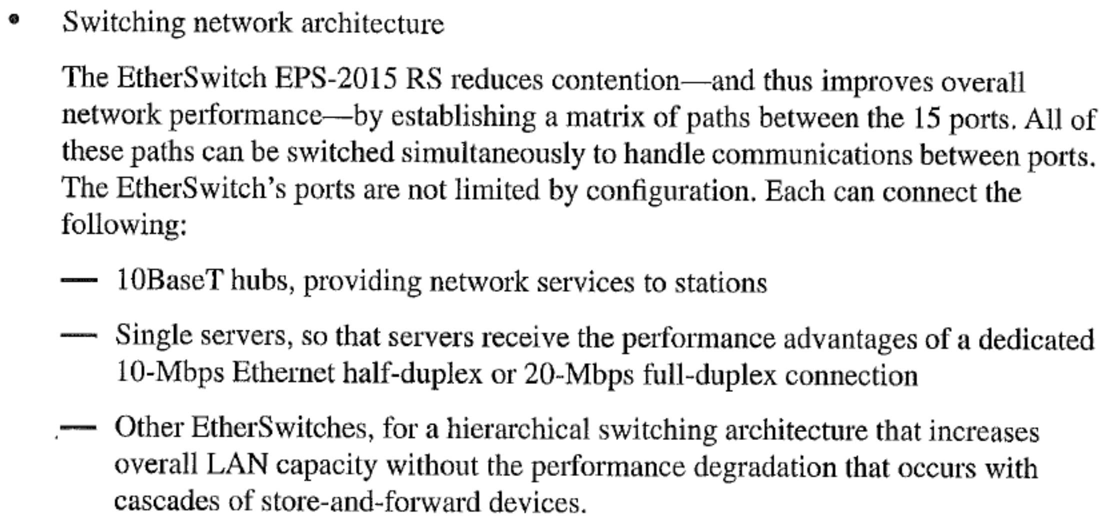
No.	’740 Patent Claim 3	Cisco EtherSwitch System
		For example, Bruckman discloses connecting the line cards to the network using traces comprising a backplane. Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”) For example, Basso discloses coupling the blades to the communication network via link connections to the switch fabric. Basso at [0009] (“A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more ports. These ports may be connected to another one or more network devices. The switch fabric may be configured to direct incoming packets of data to particular blades where one or more of the network proces-sors in the recipient blade may be configured to process the received packets.”)

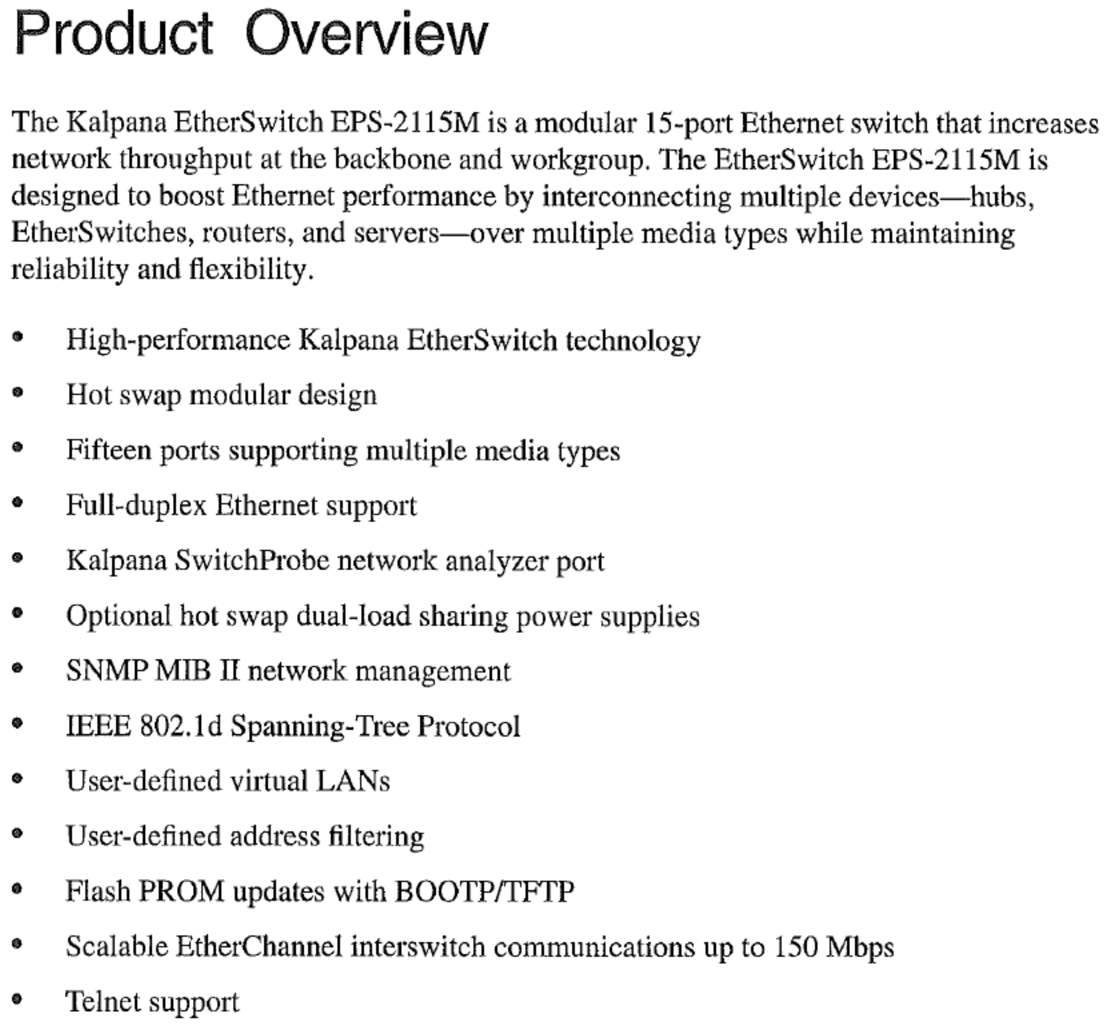
No.	’740 Patent Claim 4	Cisco EtherSwitch System
4[preamble]	A method for communication, comprising:	Cisco EtherSwitch System discloses a method for communication. *See supra* at 1[preamble].
4[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	Cisco EtherSwitch System discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].
4[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Cisco EtherSwitch System discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra* at 1[c].
4[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	Cisco EtherSwitch System discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra* at 1[e].

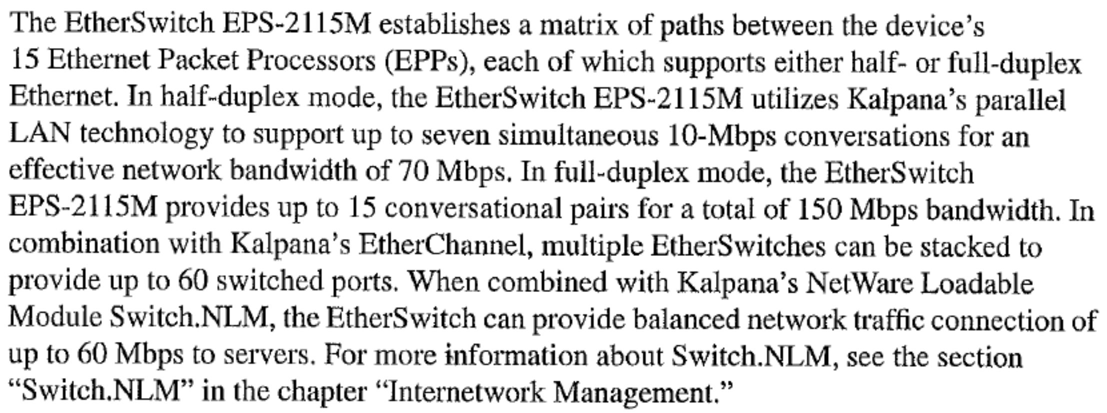
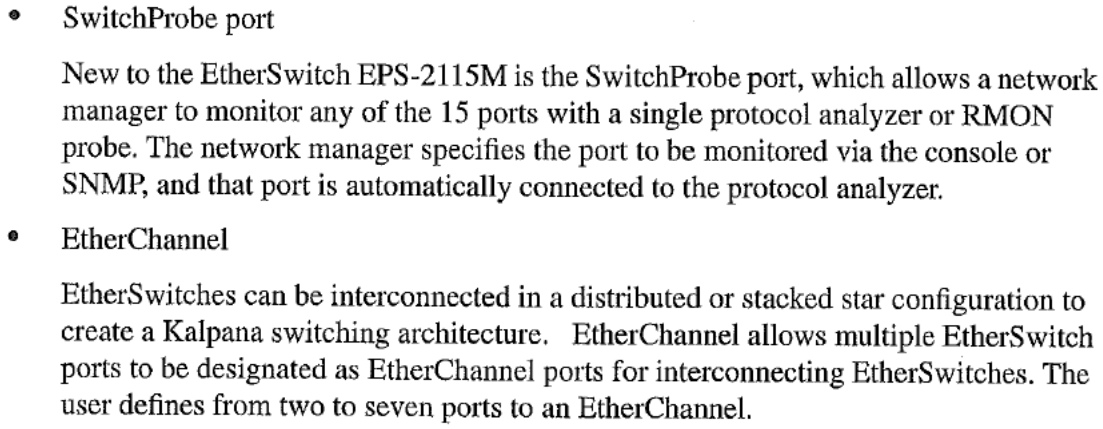
No.	’740 Patent Claim 4	Cisco EtherSwitch System
4[d]	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	Cisco EtherSwitch System discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. *See supra* at 1[f].
4[e]	sending the data frame over the selected first and second physical links,	Cisco EtherSwitch System discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].
4[f]	at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group.	Cisco EtherSwitch System discloses at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group. For example, Cisco EtherSwitch System discloses EtherChannel functionality, which allows multiple ports to be aggregated into an EtherChannel. A person of ordinary skill in the art would understands other physical links could be aggregated as well. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Cisco EtherSwitch System is found to not meet this limitation, at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group would have been obvious to a person having ordinary skill in the art, as explained below. Cisco EtherSwitch System Catalog at 114

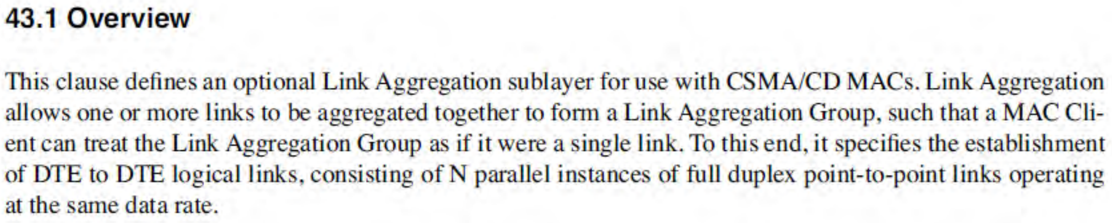
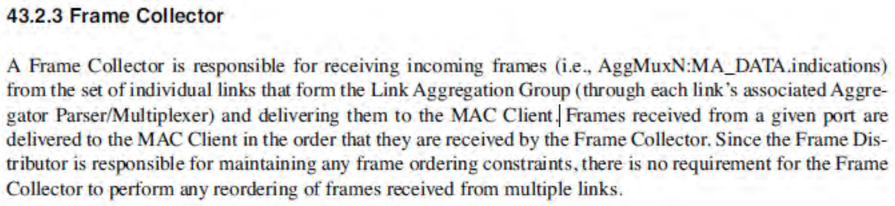
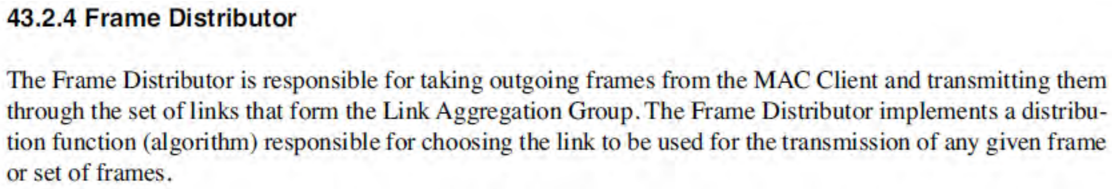
No.	’740 Patent Claim 4	Cisco EtherSwitch System
		 • Switching network architecture The EtherSwitch EPS-500 reduces contention—and thus improves overall network performance—by establishing a matrix of paths between the fifteen ports. All of these paths can be switched simultaneously to handle communications between ports. The EtherSwitch’s ports are not limited by configuration. Each can connect the following: — 10BaseT hubs, providing network services to stations — Single servers, so that servers receive the performance advantages of a dedicated 10-Mbps Ethernet half-duplex or 20-Mbps full-duplex connection — Other EtherSwitches for a hierarchical switching architecture that increases overall LAN capacity without the performance degradation that occurs with cascades of store-and-forward devices. Cisco EtherSwitch System Catalog at 115

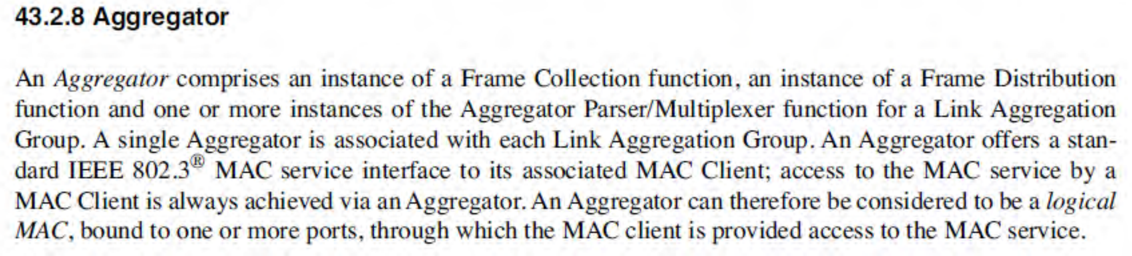
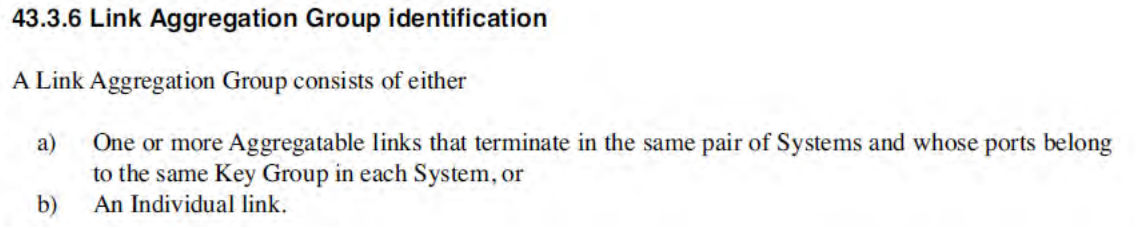
No.	’740 Patent Claim 4	Cisco EtherSwitch System
		 Product Overview The Kalpana EtherSwitch EPS-2015 RS is a stackable 15-port Ethernet switch that increases the throughput and flexibility of departmental and workgroup 10BaseT networks. Specifically designed to complement the low-cost connectivity of stackable hubs, the EtherSwitch EPS-2015 RS provides network managers with a cost-effective way to boost 10BaseT throughput among hubs and to servers while conserving valuable rack space. • High-performance Kalpana EtherSwitch technology • Fifteen 10BaseT ports with 150-Mbps bandwidth capacity • Full-duplex Ethernet support • SNMP MIB II network management • IEEE 802.1d Spanning-Tree Protocol • Virtual LAN support • User-defined address filtering • Updates using flash PROM • Scalable interswitch communications up to 150 Mbps • Delivers up to 60 switched ports when stacked with other EPS-2015 RS EtherSwitches • Rack-and-stack design, plug’n play, automatic self-learning The EtherSwitch EPS-2015 RS establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2015 RS uses Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an

No.	’740 Patent Claim 4	Cisco EtherSwitch System
		 effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2015 RS provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module, the EtherSwitch can provide balanced network traffic to NetWare servers. Cisco EtherSwitch System Catalog at 117  • EtherChannel EtherSwitches can be interconnected in a distributed or stacked star configuration to create a Kalpana switching architecture. EtherChannel allows multiple EtherSwitch ports to be designated as EtherChannel ports for interconnecting EtherSwitches. Cisco EtherSwitch System Catalog at 118  • Switching network architecture The EtherSwitch EPS-2015 RS reduces contention—and thus improves overall network performance—by establishing a matrix of paths between the 15 ports. All of these paths can be switched simultaneously to handle communications between ports. The EtherSwitch’s ports are not limited by configuration. Each can connect the following: — 10BaseT hubs, providing network services to stations — Single servers, so that servers receive the performance advantages of a dedicated 10-Mbps Ethernet half-duplex or 20-Mbps full-duplex connection — Other EtherSwitches, for a hierarchical switching architecture that increases overall LAN capacity without the performance degradation that occurs with cascades of store-and-forward devices.

No.	’740 Patent Claim 4	Cisco EtherSwitch System
		Cisco EtherSwitch System Catalog at 119  Product Overview The Kalpana EtherSwitch EPS-2115M is a modular 15-port Ethernet switch that increases network throughput at the backbone and workgroup. The EtherSwitch EPS-2115M is designed to boost Ethernet performance by interconnecting multiple devices—hubs, EtherSwitches, routers, and servers—over multiple media types while maintaining reliability and flexibility. • High-performance Kalpana EtherSwitch technology • Hot swap modular design • Fifteen ports supporting multiple media types • Full-duplex Ethernet support • Kalpana SwitchProbe network analyzer port • Optional hot swap dual-load sharing power supplies • SNMP MIB II network management • IEEE 802.1d Spanning-Tree Protocol • User-defined virtual LANs • User-defined address filtering • Flash PROM updates with BOOTP/TFTP • Scalable EtherChannel interswitch communications up to 150 Mbps • Telnet support Cisco EtherSwitch System Catalog at 120

No.	’740 Patent Claim 4	Cisco EtherSwitch System
		 The EtherSwitch EPS-2115M establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2115M utilizes Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2115M provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module Switch.NLM, the EtherSwitch can provide balanced network traffic connection of up to 60 Mbps to servers. For more information about Switch.NLM, see the section “Switch.NLM” in the chapter “Internetwork Management.” Cisco EtherSwitch System Catalog at 122  • SwitchProbe port New to the EtherSwitch EPS-2115M is the SwitchProbe port, which allows a network manager to monitor any of the 15 ports with a single protocol analyzer or RMON probe. The network manager specifies the port to be monitored via the console or SNMP, and that port is automatically connected to the protocol analyzer. • EtherChannel EtherSwitches can be interconnected in a distributed or stacked star configuration to create a Kalpana switching architecture. EtherChannel allows multiple EtherSwitch ports to be designated as EtherChannel ports for interconnecting EtherSwitches. The user defines from two to seven ports to an EtherChannel. Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Cisco EtherSwitch System in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the

No.	’740 Patent Claim 4	Cisco EtherSwitch System
		references identified in element 4[f] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, IEEE 802.3 discloses the aggregation of one or more links together to form a Link Aggregation Group. IEEE 802.3 at 1465  **43.1 Overview** This clause defines an optional Link Aggregation sublayer for use with CSMA/CD MACs. Link Aggregation allows one or more links to be aggregated together to form a Link Aggregation Group, such that a MAC Client can treat the Link Aggregation Group as if it were a single link. To this end, it specifies the establishment of DTE to DTE logical links, consisting of N parallel instances of full duplex point-to-point links operating at the same data rate. IEEE 802.3 at 1470  **43.2.3 Frame Collector** A Frame Collector is responsible for receiving incoming frames (i.e., AggMuxN:MA_DATA.indications) from the set of individual links that form the Link Aggregation Group (through each link’s associated Aggregator Parser/Multiplexer) and delivering them to the MAC Client. Frames received from a given port are delivered to the MAC Client in the order that they are received by the Frame Collector. Since the Frame Distributor is responsible for maintaining any frame ordering constraints, there is no requirement for the Frame Collector to perform any reordering of frames received from multiple links. IEEE 802.3 at 1471  **43.2.4 Frame Distributor** The Frame Distributor is responsible for taking outgoing frames from the MAC Client and transmitting them through the set of links that form the Link Aggregation Group. The Frame Distributor implements a distribution function (algorithm) responsible for choosing the link to be used for the transmission of any given frame or set of frames.

No.	’740 Patent Claim 4	Cisco EtherSwitch System
		IEEE 802.3 at 1474  **43.2.8 Aggregator** An *Aggregator* comprises an instance of a Frame Collection function, an instance of a Frame Distribution function and one or more instances of the Aggregator Parser/Multiplexer function for a Link Aggregation Group. A single Aggregator is associated with each Link Aggregation Group. An Aggregator offers a standard IEEE 802.3® MAC service interface to its associated MAC Client; access to the MAC service by a MAC Client is always achieved via an Aggregator. An Aggregator can therefore be considered to be a *logical MAC*, bound to one or more ports, through which the MAC client is provided access to the MAC service. IEEE 802.3 at 1481  **43.3.6 Link Aggregation Group identification** A Link Aggregation Group consists of either a) One or more Aggregatable links that terminate in the same pair of Systems and whose ports belong to the same Key Group in each System, or b) An Individual link. For example, Ghosh discloses aggregating physical links, including ports, into aggregate port channels that form a single logical link to increase bandwidth. Ghosh at Abstract (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggregation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent gen-eration of port channel map tables.”) Ghosh at [0004] (“Neighboring nodes in a fibre channel network are typically interconnected through multiple physical links. For example, a local fibre channel switch may be connected to a remote fibre channel switch through four physical links. In many instances, it may be beneficial

No.	’740 Patent Claim 4	Cisco EtherSwitch System
		to aggregate some of the physical links into logical links. That is, multiple physi-cal links can be combined to form a logical interface to provide higher aggregate bandwidth, load balancing, and link redundancy. When a frame is being transmitted over a logical link, it does not matter what particular physical link is being used as long as all the frames of a given flow are transmitted through the same link. If a constituent physical link goes down, the logical link can still remain operational.”) Ghosh at [0007] (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggre-gation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent generation of port channel map tables.”) Ghosh at [0008] (“In one embodiment, a method for aggregating ports in a fibre channel fabric is provided. It is determined that a plurality of local ports at a local fibre channel switch are compatible. Identifiers for the plurality of local ports are sent to a remote fibre channel switch. The remote fibre channel switch determines if a plurality of remote ports are compat-ible, the plurality of remote ports corresponding to the plurality of local ports. An indication that one or more of the remote physical ports are compatible is received. A port channel including one or more of the local ports correspond-ing to the compatible remote ports is created.”) Ghosh at [0010] (“In another embodiment, a fibre channel network is described. The fibre channel network includes a local fibre channel switch and a remote fibre channel switch. The local fibre channel switch aggregates a compatible subset of the plurality of local ports and sends identifiers for the compat-ible subset of the plurality of local ports to the remote fibre channel switch. The remote fibre channel switch determines if a subset of the plurality of remote ports are compatible. The subset of the plurality of remote ports corresponds to the compatible subset of the plurality of local ports.”) Ghosh at [0022] (“Switches in a fibre channel network are typically interconnected using multiple physical links. The physical links connecting a pair of switches allows transmission of

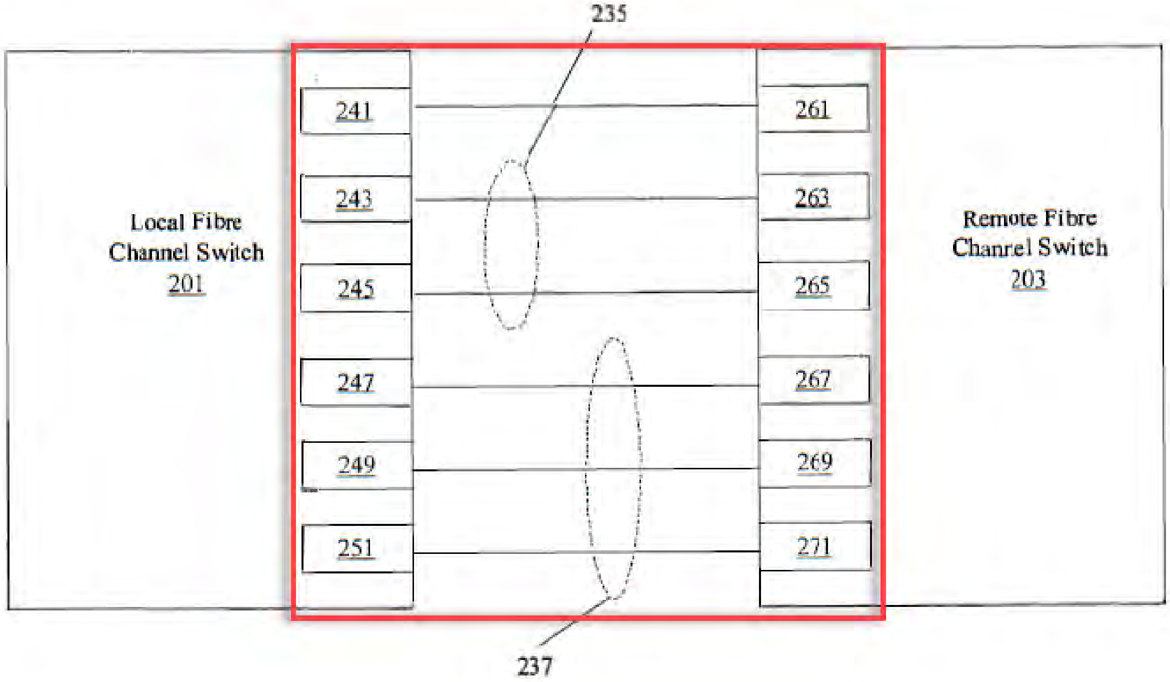
No.	’740 Patent Claim 4	Cisco EtherSwitch System
		data and control signals. In some instances, it is useful to aggregate multiple physical links into a logical link. Physi-cal links are also referred to herein as physical interfaces and channels while logical links are also referred to herein as logical interfaces and port channels. For example, a local switch may be connected to a remote switch through four physical links. Instead of having to transmit data through a particular physical link, the physical links can be aggregated to form one or more logical links. In one example, all four physical links are aggregated into a single logical link. Instead of having data transmitted through a particular physical link, the data can merely be transmitted over a particular logical link without regard to the particular physi-cal interface used. Aggregating physical links into a logical link allows for higher aggregated bandwidth, load balancing, and link redundancy. For example, if a particular physical link fails or is overloaded, data can still be transmitted over the logical link.”) Ghosh at [0029] (“FIG. 2 is a diagrammatic representation showing links between two switches, such as two fibre channel switches shown in FIG. 1. A local fibre channel switch 201 includes local ports 241, 243, 245, 247, 249, and 251. A remote fibre channel switch 203 includes remote ports 261, 263, 265, 267, 269, and 271. Local port 241 is coupled to remote port 261 through an individual physical link or channel. Connected ports are also referred to herein as peer ports. Local port 243 is coupled to remote port 263 and local port 245 is coupled to remote port 265. The two resulting physical links are aggregated to form port channel 235. Local ports 247, 249, and 251 are coupled to remote ports 267, 269, and 271 respectively. The three resulting physical links are aggregated to form port channel 237.”) Ghosh at [0030] (“According to various embodiments, local fibre channel switch 201 and remote fibre channel switch both have associated identifiers. In some examples, the identifiers are globally unique identifiers such as a global switch world wide names (WWNs). Each local port 241, 243, 245, 247, 249, and 251 and each remote port 261,263,265,267, 269, and 271 can also be associated with identifiers. In some examples, the identifiers are port WWNs. The port WWNs are typically used for debugging or identifying the peer port in alert or warning messages. However, according to various embodiments, the techniques of the present invention use WWNs as globally unique identifiers to aggregate ports instead of using compatibility keys

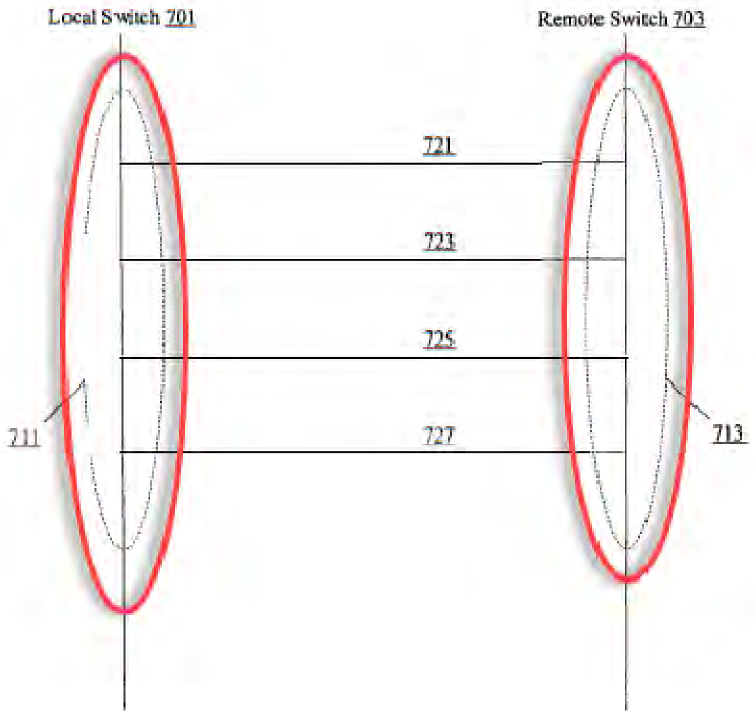
No.	’740 Patent Claim 4	Cisco EtherSwitch System
		which are only locally unique. Compatibility keys are mechanisms typically used by other protocols such as Ethernet for aggregation.”) Ghosh at [0033] (“A variety of parameters can be used to aggregate physical ports. FIG. 3 is a flow process diagram showing one technique for aggregating physical ports into a logical port. And 301, it is determined if auto create functionality is enabled. According to various embodiments, auto create functionality allows automatic configuration and detection of compatible physical ports as well as aggregation into one or more logical ports. Auto creation does not require user intervention. In other examples, administrators can manu-ally arrange ports for aggregation.”) Ghosh at [0037] (“FIG. 4 is an exchange diagram showing one example of a bring up procedure used for a port creating a new port channel. A local switch 401 is coupled to a remote switch 403. The local switch 401 includes a physical port Al coupled to physical port Bl included in remote switch 403. When two peer ports Al and Bl are being aggregated into a port channel, the peer switches 401 and 403 typically already know the world wide names of the individual physical peer ports. However, the peer switches only know the world wide name of their own logical port or port channel. That is, both switches have the individual physical link configured, but the link is not yet part of a port channel. At 421, a local switch 401 sends a synchronize (sync) message 411 to the remote switch 403 to begin the process of creating a port channel including ports Al and Bl.”) Ghosh at [0038] (“In some examples, the sync message 411 includes a local port channel identifier and a remote port channel identifier. In one particular example, the local port channel identifier is set to the world wide name of the local port channel assigned by the local switch 401. The remote port channel identifier is left blank to indicate that the port Al is being aggregated as part of a new port channel. The sync message 411 can also include other parameters such as channel status, channel model, or channel intent.”) Ghosh at [0042] (“When two peer ports A2 and B2 are aggregated into a port channel Cl, the peer switches 501 and 503 typically already know the world wide names of the indi-vidual

No.	’740 Patent Claim 4	Cisco EtherSwitch System
		physical peer ports A2 and B2 as well as the world wide name information of the port channel Cl. Conse-quently, the port channel is already successfully established. According to various embodiments, local switch 501 and remote switch 503 perform parameter checking to ensure that the new physical port A2 and B2 can be safely added to the existing port channel Cl. At 521, a local switch can check configuration parameters to ensure that physical ports Al and A2 at the local switch 501 are compatible. The compatibility checking can be performed anytime. In some examples, compatibility checking is checked before a local switch 501 sends a synchronize (sync) message 511 to the remote switch 503 to begin the process of aggregating ports A2 and B2 into the port channel.) Ghosh at [0043] (“In some examples, the sync message 511 includes local port channel identifier and a remote port channel identifier. In one particular example, the local port channel identifier is set to the world wide name of the local port channel assigned by the local switch 501. The remote port channel identifier is filled with the existing port channel identifier to indicate that the port A2 is being aggregated into existing port channel C2. The sync message 511 can also include other parameters such as channel status, channel model, or channel intent.”) Ghosh at [0044] (“At 531, remote switch 503 uses the information received from the local switch 501 to verify port B2 is compatible with other port in port channel C2. In one example, configuration parameters associated with B2 are checked against configuration parameters associated with Bl. The remote switch 503 can also check if the port B2 is already assigned to a different port channel. If the port B2 is compatible with port Bl, the remote switch 503 can proceed and send a sync accept message 513 in response to the sync message 511 to indicate that the port B2 can be aggregated into the port channel. The sync accept message indicates that a port channel can now be modified. At 523, local switch 501 uses the information to update its own port channel database. However, the port channel may not yet be fully operational until the hardware configuration is completed. The local switch 501 continues hardware configuration such as line card configuration to make the port A2 part of the port channel Cl. An acknowledgment 527 is sent and received by remote switch 503 at 529. In some examples, the local switch 501 sends a commit signal 515 when hardware configuration is complete.”)

No.	’740 Patent Claim 4	Cisco EtherSwitch System
		Ghosh at [0045] (“The remote switch 503 receives the commit signal at 533 and begins its own hardware configuration. On completion of its hardware configuration, remote switch 503 sends out a commit accept signal 517 to indicate to local switch 501 that hardware configuration is completed. According to various embodiments, local switch 501 receives the commit accept signal 517 and notifies relevant applications that the port channel is now fully operational at 525 and that port A2 has been aggregated into port channel Cl. The local switch 501 can also send out an acknowledge message 519. When the remote switch 503 receives the acknowledge, it notifies relevant applications that the port channel is operational at 535 and that port B2 has been aggregated into port channel Cl. In one embodiments, the techniques of the present invention contemplate using a two phase SYNC and COMMIT mechanism similar to the mechanism used in EPP.”) Ghosh at [0046] (“FIGS. 4 and 5 show examples of ports being aggregated into a port channel. At a particular switch, ports can be selected for aggregation into a port channel in a variety of manners. FIG. 6 is an exchange diagram showing automatic selection of ports at a switch for aggregation into a port channel. A local switch 601 is coupled to a remote switch 603. In one example, the local switch 601 includes physical ports Al, A2, A3, and A4 while remote switch 603 includes physical ports Bl, B2, B3, and B4. No port chan-nels have been formed.”) Ghosh at [0049] (“At 631, remote switch 603 uses the information received from the local switch 601 to verify that the peer ports of Al, A2, and A4 are compatible. That is, ports Bl, B2, and B4 are checked for compatibility. In one example, only ports Bl and B2 may be compatible, and consequently only ports Al, A2, Bl, and B2 can be included in the port channel. In another example, ports Bl, B2, and B4 are compatible, so ports Al, A2, A4, Bl, B2, and B4 can be aggregated into port channel Cl. According to various embodiments, if the port B2 is compatible with port Bl, the remote switch 603 can proceed and send a sync accept message 613 in response to the sync message 611 to indicate that the port B2 can be aggregated into the port channel. It should be noted that remote switch 603 can send a list indicating that ports B2 and B4 are compatible with Bl. However, the remote switch 603 sends only one compatible port B2 back for several reasons, and in the process of selection compatible port channels get priority over com-patible individual ports.”)

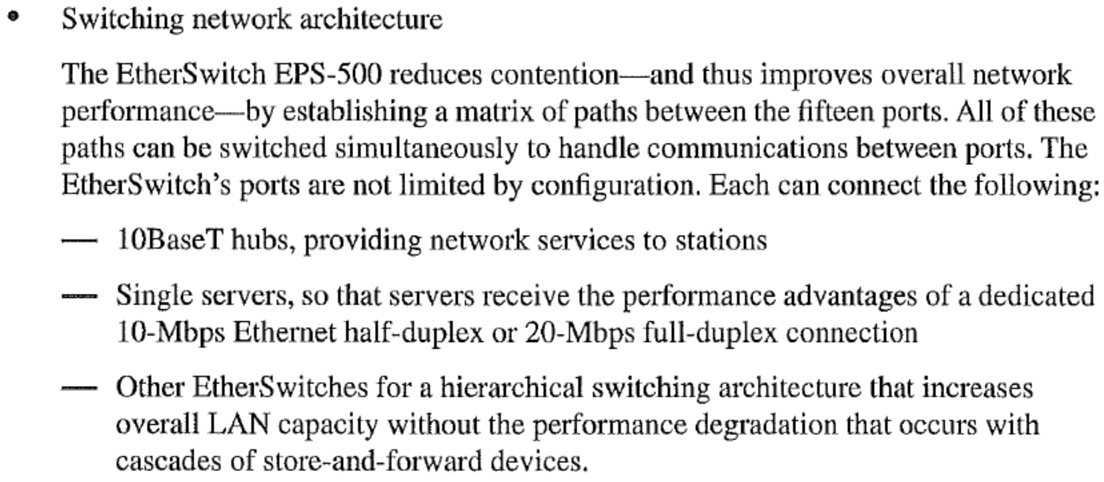
No.	’740 Patent Claim 4	Cisco EtherSwitch System
		Ghosh at [0050] (“One reason is that aggregation mechanisms and techniques can be implemented more elegantly by handling ports on an individual basis. Any individual port will either start a new port channel, be added to an existing port channel, or operate stand alone. There is no need to keep track of groups of ports to be aggregated. Another reason is that fewer ports need to be locked if only a single port is being aggregated at any one time. The sync accept message indicates that a port channel can now be modified. At 623, local switch 601 receives the information and recognizes that Al and A2 can now be aggregated into port channel Cl. However, the port channel may not yet be fully operational until the hardware configuration is completed. An acknowl-edgment 627 is sent and received by remote switch 603 at 629. In some examples, the local switch 601 sends a commit signal 615 when hardware configuration is complete.”) Ghosh at [0051] (“The remote switch 603 receives the commit signal at 633 to create port channel Cl including ports Bl and B2. Hardware configuration can now be performed. On comple-tion of its hardware configuration, remote switch 603 sends out a commit accept signal 617 to indicate to local switch 601 that hardware configuration is completed. According to various embodiments, local switch 601 receives the commit accept signal 617 and notifies relevant applications that the port channel is now fully operational at 625 and that ports Al and A2 have been aggregated into port channel Cl. The local switch 601 can also send out an acknowledge message 619. When the remote switch 603 receives the acknowledge, it notifies relevant applications that the port channel is fully operational at 635 and that ports Bl and B2 have been aggregated into port channel Cl.”) Ghosh at [0053] (“FIG. 7 is a diagrammatic representation showing synchronous aggregation of ports into a port channel. A local switch 701 is coupled to a remote switch 703 through links 721, 723, 725, and 727. According to various embodiments, the links are being aggregated into port channel 711 at the local switch 701 and port channel 713 at the remote switch 703 in a synchronous manner. That is the peer ports corre-sponding to each link are brought up in the same order at both the local switch 701 and the remote switch 703.”)

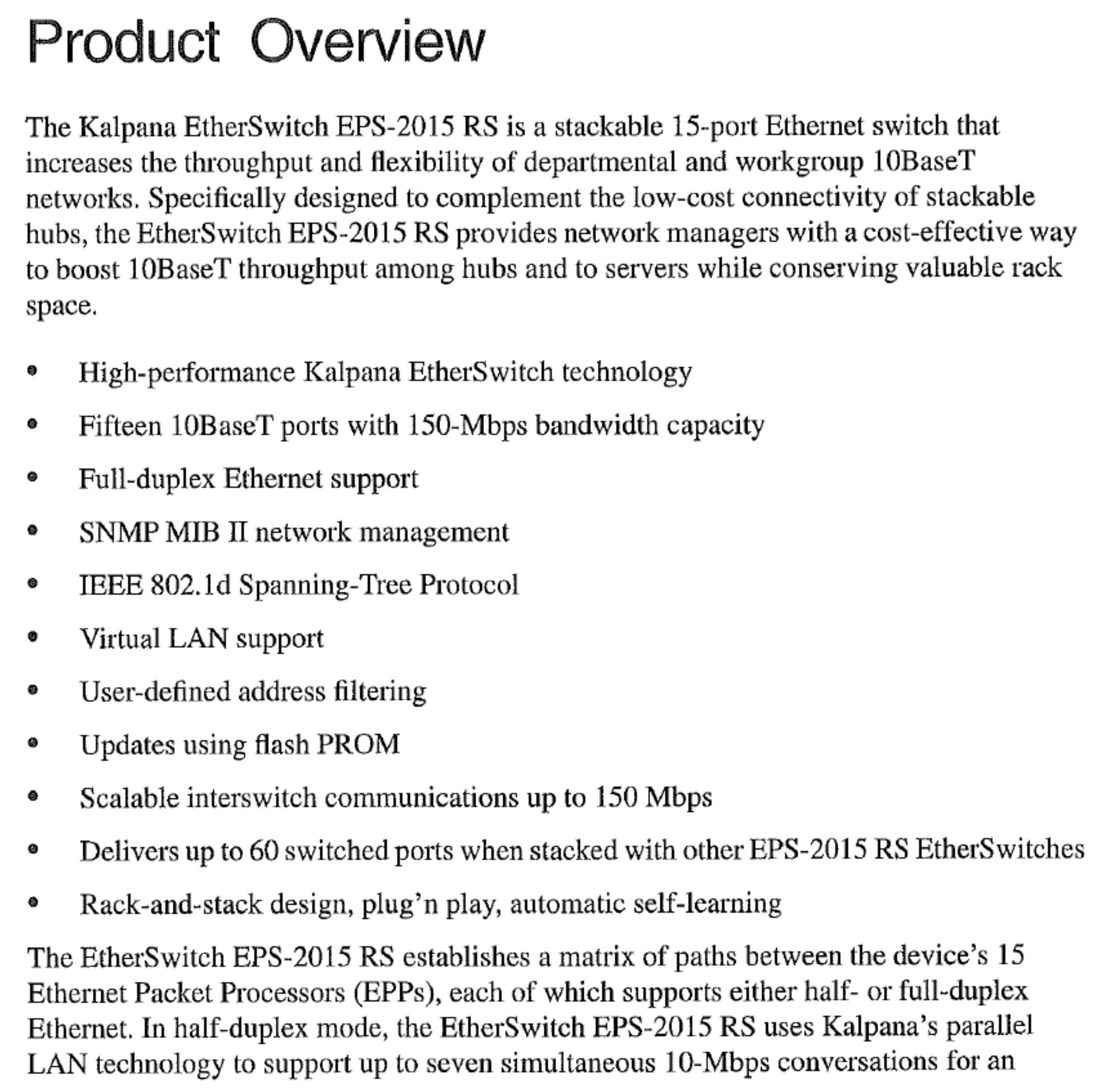
No.	’740 Patent Claim 4	Cisco EtherSwitch System
		Ghosh at Figure 2 (annotation added)  235 241 243 245 247 249 251 261 263 265 267 269 271 Local Fibre Channel Switch 201 Remote Fibre Chanrel Switch 203 237 Figure 2 Ghosh at Figure 7 (annotation added)

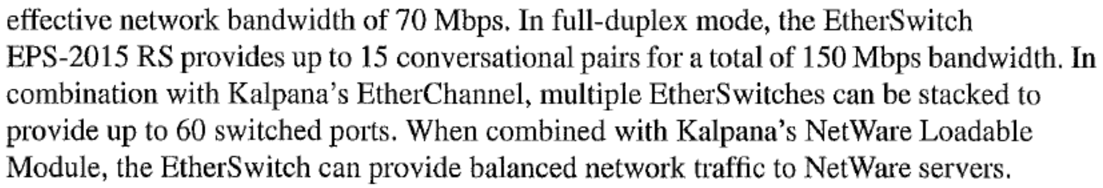
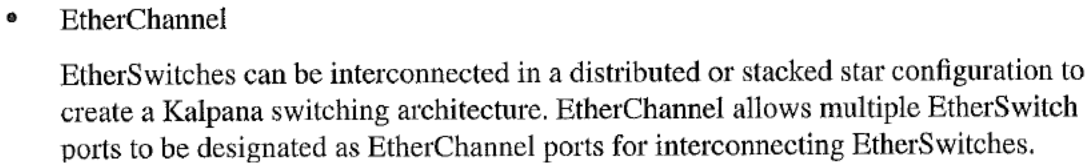
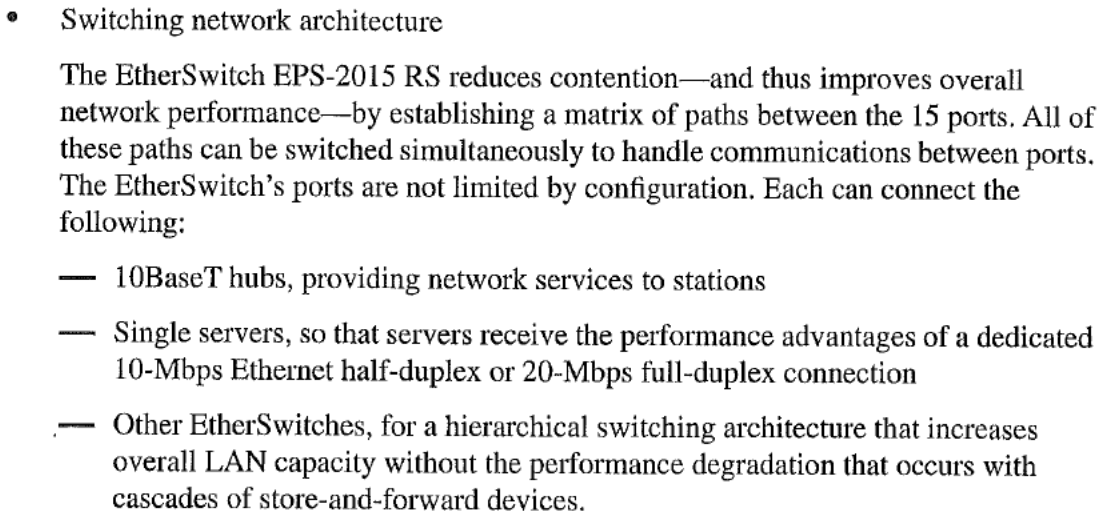
No.	’740 Patent Claim 4	Cisco EtherSwitch System
		 Local Switch 701 Remote Switch 703 721 723 725 727 711 713 Figure 7

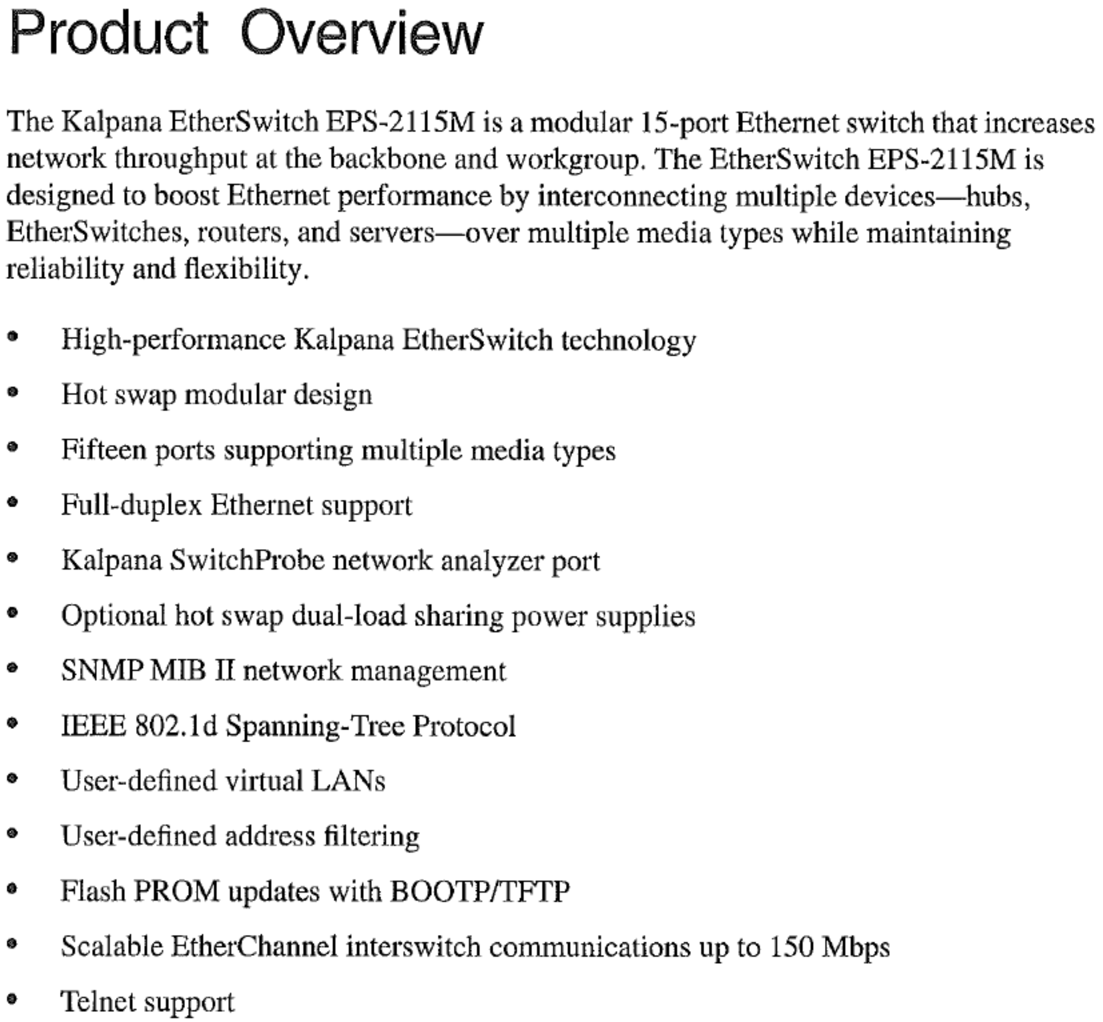
No.	’740 Patent Claim 5	Cisco EtherSwitch System
5[preamble]	A method for communication, comprising:	Cisco EtherSwitch System discloses a method for communication. *See supra at* 1[preamble].

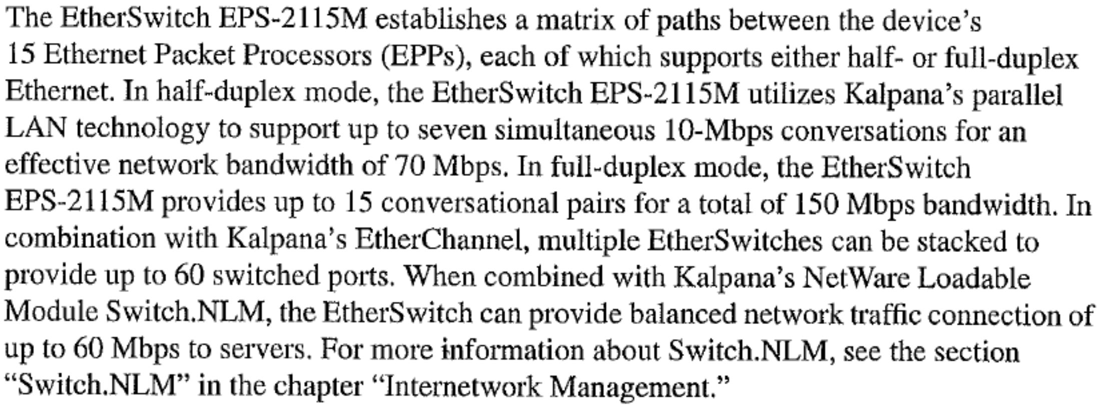
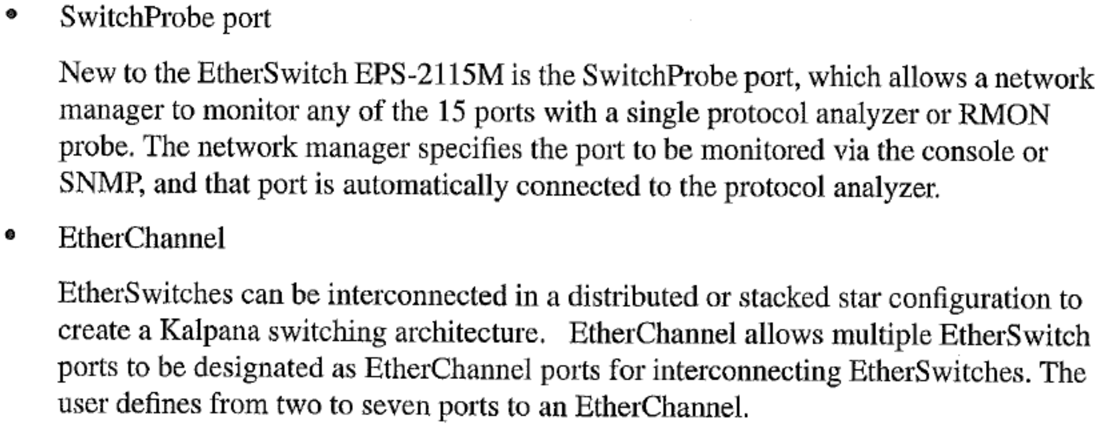
No.	’740 Patent Claim 5	Cisco EtherSwitch System
5[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	Cisco EtherSwitch System discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].
5[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Cisco EtherSwitch System discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra at* 1[c].
5[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	Cisco EtherSwitch System discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra at* 1[e].
5[d]	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	Cisco EtherSwitch System discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. *See supra at* 1[f].
5[e]	sending the data frame over the selected first	Cisco EtherSwitch System discloses sending the data frame over the selected first and second physical links.

No.	’740 Patent Claim 5	Cisco EtherSwitch System
	and second physical links,	*See supra at* 1[g].
5[f]	coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node.	Cisco EtherSwitch System discloses coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node. For example, Cisco EtherSwitch System discloses EtherChannel functionality, which allows multiple ports to be aggregated into an external EtherChannel for interconnecting multiple EtherSwitches with increased throughput capacity. Cisco EtherSwitch System Catalog at 114  • Switching network architecture The EtherSwitch EPS-500 reduces contention—and thus improves overall network performance—by establishing a matrix of paths between the fifteen ports. All of these paths can be switched simultaneously to handle communications between ports. The EtherSwitch’s ports are not limited by configuration. Each can connect the following: — 10BaseT hubs, providing network services to stations — Single servers, so that servers receive the performance advantages of a dedicated 10-Mbps Ethernet half-duplex or 20-Mbps full-duplex connection — Other EtherSwitches for a hierarchical switching architecture that increases overall LAN capacity without the performance degradation that occurs with cascades of store-and-forward devices. Cisco EtherSwitch System Catalog at 115

No.	’740 Patent Claim 5	Cisco EtherSwitch System
		 Product Overview The Kalpana EtherSwitch EPS-2015 RS is a stackable 15-port Ethernet switch that increases the throughput and flexibility of departmental and workgroup 10BaseT networks. Specifically designed to complement the low-cost connectivity of stackable hubs, the EtherSwitch EPS-2015 RS provides network managers with a cost-effective way to boost 10BaseT throughput among hubs and to servers while conserving valuable rack space. • High-performance Kalpana EtherSwitch technology • Fifteen 10BaseT ports with 150-Mbps bandwidth capacity • Full-duplex Ethernet support • SNMP MIB II network management • IEEE 802.1d Spanning-Tree Protocol • Virtual LAN support • User-defined address filtering • Updates using flash PROM • Scalable interswitch communications up to 150 Mbps • Delivers up to 60 switched ports when stacked with other EPS-2015 RS EtherSwitches • Rack-and-stack design, plug’n play, automatic self-learning The EtherSwitch EPS-2015 RS establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2015 RS uses Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an

No.	’740 Patent Claim 5	Cisco EtherSwitch System
		 effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2015 RS provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module, the EtherSwitch can provide balanced network traffic to NetWare servers. Cisco EtherSwitch System Catalog at 117  • EtherChannel EtherSwitches can be interconnected in a distributed or stacked star configuration to create a Kalpana switching architecture. EtherChannel allows multiple EtherSwitch ports to be designated as EtherChannel ports for interconnecting EtherSwitches. Cisco EtherSwitch System Catalog at 118  • Switching network architecture The EtherSwitch EPS-2015 RS reduces contention—and thus improves overall network performance—by establishing a matrix of paths between the 15 ports. All of these paths can be switched simultaneously to handle communications between ports. The EtherSwitch’s ports are not limited by configuration. Each can connect the following: — 10BaseT hubs, providing network services to stations — Single servers, so that servers receive the performance advantages of a dedicated 10-Mbps Ethernet half-duplex or 20-Mbps full-duplex connection — Other EtherSwitches, for a hierarchical switching architecture that increases overall LAN capacity without the performance degradation that occurs with cascades of store-and-forward devices.

No.	’740 Patent Claim 5	Cisco EtherSwitch System
		Cisco EtherSwitch System Catalog at 119  Product Overview The Kalpana EtherSwitch EPS-2115M is a modular 15-port Ethernet switch that increases network throughput at the backbone and workgroup. The EtherSwitch EPS-2115M is designed to boost Ethernet performance by interconnecting multiple devices—hubs, EtherSwitches, routers, and servers—over multiple media types while maintaining reliability and flexibility. • High-performance Kalpana EtherSwitch technology • Hot swap modular design • Fifteen ports supporting multiple media types • Full-duplex Ethernet support • Kalpana SwitchProbe network analyzer port • Optional hot swap dual-load sharing power supplies • SNMP MIB II network management • IEEE 802.1d Spanning-Tree Protocol • User-defined virtual LANs • User-defined address filtering • Flash PROM updates with BOOTP/TFTP • Scalable EtherChannel interswitch communications up to 150 Mbps • Telnet support Cisco EtherSwitch System Catalog at 120

No.	’740 Patent Claim 5	Cisco EtherSwitch System
		 The EtherSwitch EPS-2115M establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2115M utilizes Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2115M provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module Switch.NLM, the EtherSwitch can provide balanced network traffic connection of up to 60 Mbps to servers. For more information about Switch.NLM, see the section “Switch.NLM” in the chapter “Internetwork Management.” Cisco EtherSwitch System Catalog at 122  - SwitchProbe port New to the EtherSwitch EPS-2115M is the SwitchProbe port, which allows a network manager to monitor any of the 15 ports with a single protocol analyzer or RMON probe. The network manager specifies the port to be monitored via the console or SNMP, and that port is automatically connected to the protocol analyzer. - EtherChannel EtherSwitches can be interconnected in a distributed or stacked star configuration to create a Kalpana switching architecture. EtherChannel allows multiple EtherSwitch ports to be designated as EtherChannel ports for interconnecting EtherSwitches. The user defines from two to seven ports to an EtherChannel.

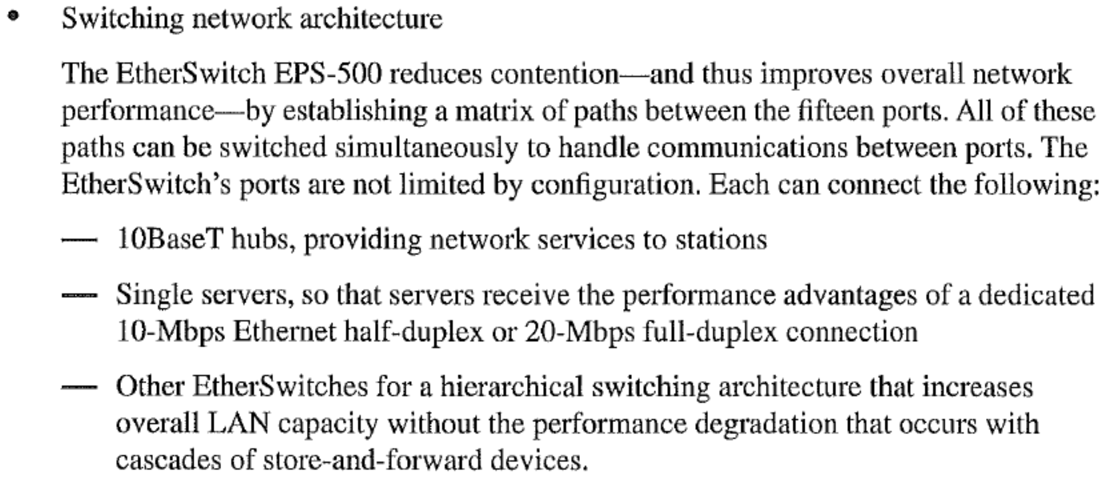
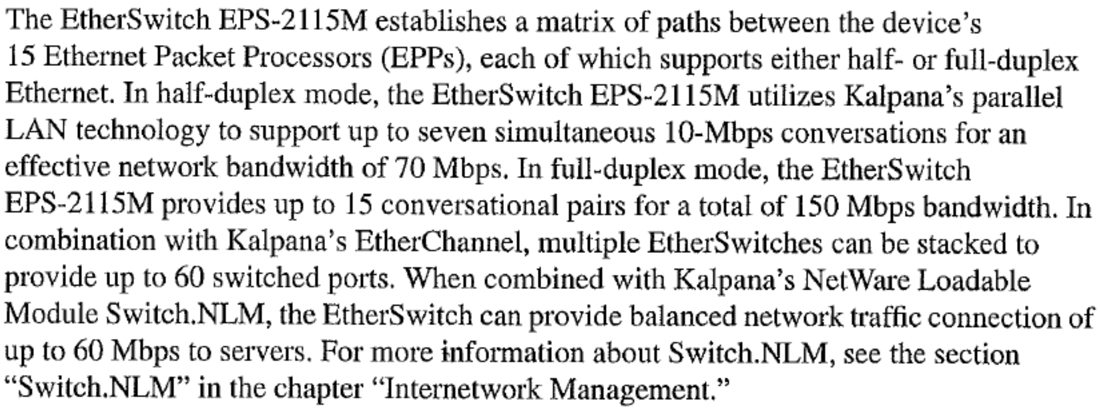
No.	’740 Patent Claim 6	Cisco EtherSwitch System
6	The method according to claim 1, wherein coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node.	Cisco EtherSwitch System discloses the method according to claim 1, wherein coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node. On information and belief, Cisco EtherSwitch System discloses a switch device with standard internal architecture, including a multiplexer and demultiplexer. A person of ordinary skill in the art would understand that the multiple paths for communicating packets between network devices in full duplex mode would be facilitated by a multiplexer/demultiplexer, a standard piece of switch architecture. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Cisco EtherSwitch System is found to not meet this limitation, wherein coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node would have been obvious to a person having ordinary skill in the art, as explained below. *See supra* Claim 1. Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Cisco EtherSwitch System in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 6 of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Wiher discloses multiplexing and demultiplexing circuitry to transmit and receive ATM data cells over a data link between the ATM network and network access equipment, i.e., multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node.

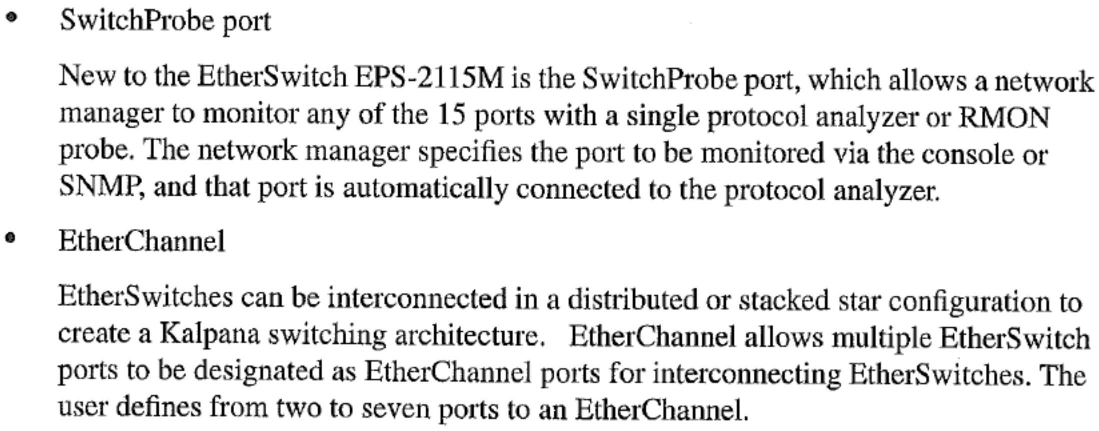
No.	’740 Patent Claim 6	Cisco EtherSwitch System
		Wiher at 3:43-65 (“In general, in another aspect, the invention features an apparatus for communicating data cells between a data link and a backplane. The apparatus includes transceiver circuitry to transmit and receive data cells over a data link and a plurality of backplane interfaces each including at least one cell signal terminal. Each of the backplane interface is coupled to a backplane interconnection circuit. Each backplane interconnection circuit transmits and receives cells over the cell signal terminals of its associated backplane interface. The apparatus also includes de-multiplexing circuitry coupling the transceiver circuitry to each of the backplane interconnection circuits. The de-multiplexing circuitry receives a data cell from the transceiver circuitry, select a backplane interconnection circuit associated with the data cell, and provide the data cell to the selected backplane interconnection circuit for transmission over the cell signal terminals of the associated backplane interface. The apparatus also includes multiplexing circuitry coupling the plurality of backplane interconnection circuits to the transceiver circuitry. The multiplexing circuitry receives data cells from each of the backplane interconnection circuits and provide the received data cells to the transceiver circuitry.”) Wiher at 3:66-4:22 (“Implementations of the invention may include one or more of the following features. The backplane interconnection circuits may independently receive and transmit data cells over the plurality of backplane interfaces. The de-multiplexing circuitry may select a backplane interface based on data in the header field of the data cell. The apparatus may include header translation circuitry to alter header data in cells sent between the plurality of backplane interfaces and the transceiver circuitry. Each of the plurality of backplane interfaces may include separate terminals to receive cells and separate terminals to transmit cells. The terminals to transmit cells may include a first and second control terminal and at least one outgoing cell data terminal. A backplane interface's backplane interconnection circuitry may accepts a signal on the first control terminal as indicating that a cell may be sent over the interface, asserts a 15 signal on the second control terminal to indicate that a cell is being transmitted, and transmits data bits of the cell on the outgoing cell data terminal. Each backplane interface may include a single outgoing cell data terminal and each bit of the cell may be serially transmitted over the single outgoing cell data terminal.

No.	’740 Patent Claim 6	Cisco EtherSwitch System
		Each backplane interface may include multiple outgoing cell data terminals and bits of the cell may be sent in parallel over the eight outgoing cell data terminals.”) For example, Lebizay discloses an optical add/drop multiplexer that multiplexers and demultiplexes data packets sent between the network boards and network, i.e., multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node. Lebizay at [0043] (“InfiniBand offers link layer Virtual Lanes (VLs) to support multiple logical channels (i.e. multiplexing) on the same physical link. Infiniband offers up to 16 virtual lanes per link. VLs provide a mechanism to avoid head-of-line blocking and the ability to support Quality of Service (QoS). The difference between a Virtual Lane and a Service Level (SL) is that a Virtual Lane is the actual logical lane (mul-tiplexed) used on a given point-to-point link. The Service Level stays constant as a packet traverses the fabric, and specifies the desired service level within a subnet. The SL (AF, EF or BE) is included in the link header, and each switch maps the SL to a VL supported by the destination link. A switch supporting a limited number of virtual lanes will map the SL field to a VL it supports. Without preserving the SL, the desired SL (AF, EF or BE) would be lost in this mapping, and later in the path, a switch supporting more VLs would be unable to recover finer granularity of SLs between two packets mapped to the same VL.”) Lebizay at [0050] (“The issue with using a ring, however, is how to map the addressing of multiple boards across these fibers. One solution is to employ Wavelength Division Multiplex-ing (WDM). A WDM optical mesh defines a meshed-topology in the wavelength space as opposed to the physical fiber space. By utilizing multiple discrete lambda-waves as optical carriers such that by meshing dedicated optical wavelengths between every two boards, layer 2 protocols are eliminated, thereby creating a dramatic improvement in the efficiency of the transport. Today, every packet transport requires a protocol that allows the end point (and interme-diate points) to decipher the intended path (or consumer) of the packet. This protocol increases the amount of overhead required in the packet bus, allowing less room for actual data to be sent. By moving the protocol into the wavelength of the actual

No.	’740 Patent Claim 6	Cisco EtherSwitch System
		optical signal, the destination is implied by the wavelength and no additional bandwidth needs to be sur-rendered on the signal to provide this information. This makes the efficiency of the transport better and also speeds the routing of the packet through the network. In addition, the use of optical interconnects in a backplane environment greatly increases chassis bandwidth as well as reducing electrical radiation that often accompanies copper intercon-nects. The components involved include an optical back-plane in a physical ring topology, and the necessary trans-mitters and receivers for the size of the installation (i.e., number of slots in the chassis). In addition, optical add/drop multiplexer devices are required.”)

No.	’740 Patent Claim 7	Cisco EtherSwitch System
7	The method according to claim 1, wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the first and second physical links.	Cisco EtherSwitch System discloses the method according to claim 1, wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the first and second physical links. For example, Cisco EtherSwitch System discloses increasing bandwidth and balancing increased network traffic connections among the matrix of paths, including ports and internal connections, used to transmit data packets. *See supra* Claim 1 Cisco EtherSwitch System Catalog at 114

No.	’740 Patent Claim 7	Cisco EtherSwitch System
		 • Switching network architecture The EtherSwitch EPS-500 reduces contention—and thus improves overall network performance—by establishing a matrix of paths between the fifteen ports. All of these paths can be switched simultaneously to handle communications between ports. The EtherSwitch’s ports are not limited by configuration. Each can connect the following: — 10BaseT hubs, providing network services to stations — Single servers, so that servers receive the performance advantages of a dedicated 10-Mbps Ethernet half-duplex or 20-Mbps full-duplex connection — Other EtherSwitches for a hierarchical switching architecture that increases overall LAN capacity without the performance degradation that occurs with cascades of store-and-forward devices. Cisco EtherSwitch System Catalog at 120  The EtherSwitch EPS-2115M establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2115M utilizes Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2115M provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module Switch.NLM, the EtherSwitch can provide balanced network traffic connection of up to 60 Mbps to servers. For more information about Switch.NLM, see the section “Switch.NLM” in the chapter “Internetwork Management.” Cisco EtherSwitch System Catalog at 122

No.	’740 Patent Claim 7	Cisco EtherSwitch System
		 • SwitchProbe port New to the EtherSwitch EPS-2115M is the SwitchProbe port, which allows a network manager to monitor any of the 15 ports with a single protocol analyzer or RMON probe. The network manager specifies the port to be monitored via the console or SNMP, and that port is automatically connected to the protocol analyzer. • EtherChannel EtherSwitches can be interconnected in a distributed or stacked star configuration to create a Kalpana switching architecture. EtherChannel allows multiple EtherSwitch ports to be designated as EtherChannel ports for interconnecting EtherSwitches. The user defines from two to seven ports to an EtherChannel.

No.	’740 Patent Claim 8	Cisco EtherSwitch System
8	The method according to claim 1, wherein selecting the first and second physical links comprises applying a mapping function to the at least one of the frame attributes.	Cisco EtherSwitch System discloses the method according to claim 1, wherein selecting the first and second physical links comprises applying a mapping function to the at least one of the frame attributes. For example, Cisco EtherSwitch System discloses switching in which the matrix of paths over which packets can be sent are determined by reading and processing packet information, such as the packet’s destination address. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Cisco EtherSwitch System is found to not meet this limitation, wherein selecting the first and second physical links comprises applying a mapping function to the at least one of the frame attributes would have been obvious to a person having ordinary skill in the art, as explained below. *See supra* Claim 1 Cisco EtherSwitch System Catalog at 113

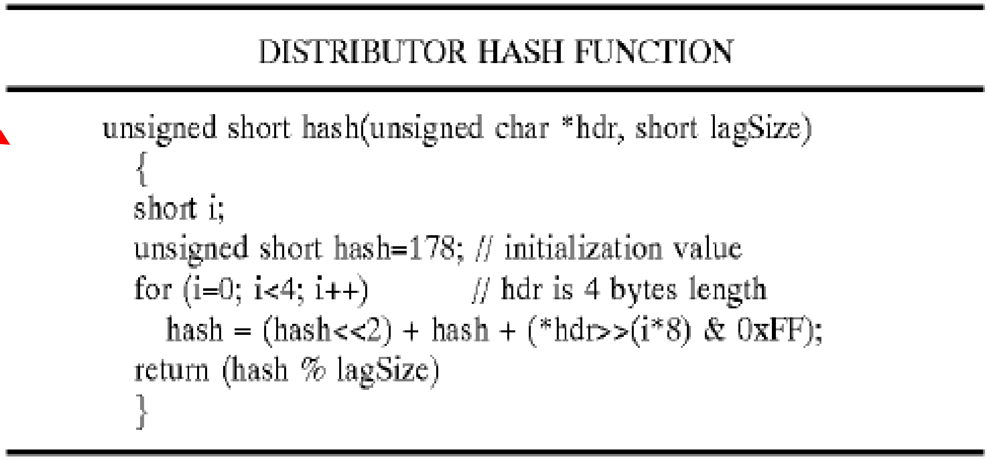
		• Two models available A managed version, the EtherSwitch EPS-500-SNMP, provides Simple Network Management Protocol with Management Information Base (MIB) II support. Kalpana's SNMP implementation allows Kalpana's SwitchVision Network Management System application to collect system and port information from the EtherSwitch EPS-500. The entry-level version of the EtherSwitch EPS-500, which can be upgraded with SNMP, supports out-of-band network management via an EIA/TIA-232 connector. • Easy to install Designed to be a plug-and-go device, the EtherSwitch EPS-500 requires no configuration for installation. The EtherSwitch EPS-500 automatically learns and keeps track of the logical addresses of up to 4,000 network nodes. Address-to-port references are learned at power up and automatically relearned any time the network configuration changes. The EtherSwitch EPS-500 ages addresses so that the system's address table is populated only by the more frequently used addresses. • On-the-fly packet switching With Kalpana's "on-the-fly" switching technology, a packet appears at the output port before it has finished entering the input port. During normal operation, packets moving through the EtherSwitch EPS-500 are delayed no more than 40 microseconds, which is 20 times less than that of bridges and routers using store-and-forward technology. The EtherSwitch EPS-500 does this by reading and processing the six-byte destination address immediately, instead of waiting for the entire packet to arrive first. The shorter delay is particularly important for delay-sensitive applications or direct host connections. • Ethernet throughput enhancement The principal building blocks for the corporate internetwork architecture have been bridges, routers, and structured wiring hubs. Now there is a new network building block—the Ethernet switch. Ethernet switching offers a new design element to complement existing hubs, internetworking devices and high-speed networks. Kalpana's EtherSwitch was designed with a simple purpose in mind—to enhance Ethernet network throughput. Routers are the building blocks of the extended internetwork—designed specifically to interconnect LANs and WANs and provide LAN security. Routing and EtherSwitch are complementary devices intended for specific purposes.

No.	’740 Patent Claim 8	Cisco EtherSwitch System
		Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Cisco EtherSwitch System in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 8 of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses distributing data frames over specific ports via physical links and traces based on a distribution algorithm using frame information. Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the

No.	’740 Patent Claim 8	Cisco EtherSwitch System
		headers to select a respective one of the physical links over which to transmit each of the data frames.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”)

<table>
<tr><th>No.</th><th>’740 Patent Claim 8</th><th>Cisco EtherSwitch System</th></tr>
<tr><td></td><td></td><td>

Bruckman at Table 1 (annotated)



hashing function “mapping function”

DISTRIBUTOR HASH FUNCTION

<pre>
unsigned short hash(unsigned char *hdr, short lagSize)
  {
  short i;
  unsigned short hash=178; // initialization value
  for (i=0; i<4; i++)   // hdr is 4 bytes length
     hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
  return (hash % lagSize)
  }
</pre>

Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)

As another example, Solomon discloses a mapping function including a hashing function to the packet label.

Solomon at [0024] (“In another embodiment, switching the data packets includes mapping the data packets to the selected port responsively to the label. Additionally or alternatively, map-ping the data packets includes applying a hashing function to the label so as to determine a number of the selected port, and choosing the label includes applying an inverse of the hashing function to the number of the selected port.”)

</td></tr>
</table>

No.	’740 Patent Claim 8	Cisco EtherSwitch System
		Solomon at [0048] (“The mapping function typically uses MPLS label 52 for mapping, since the MPLS label uniquely identifies MPLS tunnel 28, and it is required that all MPLS packets belonging to the same tunnel be switched through the same physical port 24. Additionally or alternatively, the mapping function uses a "PW" label (pseudo wire label, formerly known as a virtual connection, or VC label), which is optionally added to MPLS header 50. The PW label com-prises information that the egress node requires for deliver-ing the packet to its destination, and is optionally added during the encapsulation of MPLS packets. Additional details regarding the VC label can be found in an IETF draft by Martini et al. entitled "Encapsulation Methods for Trans-port of Ethernet Frames Over IP/MPLS Networks" (IETF draft-ietf-pwe3-ethernet-encap-07.txt, May, 2004), which is incorporated herein by reference. In some embodiments, mapper 34 applies a hashing function to the MPLS and/or PW label, as will be described below.”) Solomon at [0059] (“In this method, the mapping function used by mapper 34 of switch A is a hashing function. Various hashing functions are known in the art, and any suitable hashing function may be used in mapper 34. Since the hashing operation is performed for each packet, it is desirable to have a hashing function that is computationally simple.”) Solomon at [0060] (“As mentioned above, the hashing function typically hashes the value of MPLS label 52 to determine the selected physical port, as the MPLS label uniquely identifies tunnel 28. For example, the following hashing function may be used by mapper 34: Selected port number=l+((MPLS label) mod N), wherein N denotes the number of physical Ethernet ports in LAG group 25, and "mod" denotes the modulus operator. Assuming the values of MPLS labels are distrib-uted uniformly over a certain range, this function achieves a uniform distribution of port allocations for the different MPLS labels. It can also be seen that all packets carrying the same MPLS label (in other words-belonging to the same MPLS tunnel) will be mapped to the same physical port.”) Solomon at [0065] (“Mapper 34 of switch A maps each received packet to the selected physical port of LAG group 25 using the hashing function, at a hashing step 90. Mapper 34 extracts the MPLS label from each received packet and uses the hashing function to calculate the serial number of the selected physical port, which was selected by the CAC processor at step 82.

No.	’740 Patent Claim 8	Cisco EtherSwitch System
		Following the numerical example given above, the mapper extracts MPLS label=65647 from the packet. Substituting this value and N=3 into the hashing function gives: Selected port number=1+(65647 mod 3)=2, which is indeed the port number selected in the example above.”)

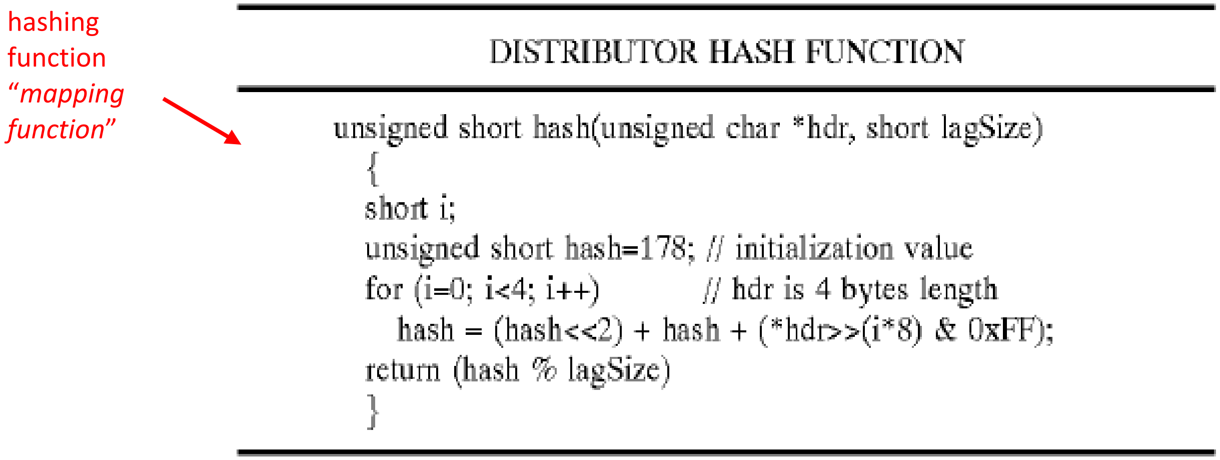
No.	’740 Patent Claim 9	Cisco EtherSwitch System
9	The method according to claim 8, wherein applying the mapping function comprises applying a hashing function.	Cisco EtherSwitch System discloses the method according to claim 8, wherein applying the mapping function comprises applying a hashing function. For example, Cisco EtherSwitch System discloses determining which of the matrix of paths to send a data packet over by reading and processing packet information. A person of ordinary skill in the art would understand that this determining includes standard functions and algorithms, including a hash function, which is a regularly employed function in route selection. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Cisco EtherSwitch System is found to not meet this limitation, wherein applying the mapping function comprises applying a hashing function would have been obvious to a person having ordinary skill in the art, as explained below. *See supra* Claim 8 Cisco EtherSwitch System Catalog at 1113

		• Two models available A managed version, the EtherSwitch EPS-500-SNMP, provides Simple Network Management Protocol with Management Information Base (MIB) II support. Kalpana's SNMP implementation allows Kalpana's SwitchVision Network Management System application to collect system and port information from the EtherSwitch EPS-500. The entry-level version of the EtherSwitch EPS-500, which can be upgraded with SNMP, supports out-of-band network management via an EIA/TIA-232 connector. • Easy to install Designed to be a plug-and-go device, the EtherSwitch EPS-500 requires no configuration for installation. The EtherSwitch EPS-500 automatically learns and keeps track of the logical addresses of up to 4,000 network nodes. Address-to-port references are learned at power up and automatically relearned any time the network configuration changes. The EtherSwitch EPS-500 ages addresses so that the system's address table is populated only by the more frequently used addresses. • On-the-fly packet switching With Kalpana's "on-the-fly" switching technology, a packet appears at the output port before it has finished entering the input port. During normal operation, packets moving through the EtherSwitch EPS-500 are delayed no more than 40 microseconds, which is 20 times less than that of bridges and routers using store-and-forward technology. The EtherSwitch EPS-500 does this by reading and processing the six-byte destination address immediately, instead of waiting for the entire packet to arrive first. The shorter delay is particularly important for delay-sensitive applications or direct host connections. • Ethernet throughput enhancement The principal building blocks for the corporate internetwork architecture have been bridges, routers, and structured wiring hubs. Now there is a new network building block—the Ethernet switch. Ethernet switching offers a new design element to complement existing hubs, internetworking devices and high-speed networks. Kalpana's EtherSwitch was designed with a simple purpose in mind—to enhance Ethernet network throughput. Routers are the building blocks of the extended internetwork—designed specifically to interconnect LANs and WANs and provide LAN security. Routing and EtherSwitch are complementary devices intended for specific purposes.

No.	’740 Patent Claim 9	Cisco EtherSwitch System
		Cisco EtherSwitch in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 9 of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses applying a distributor hash function. Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the

No.	’740 Patent Claim 9	Cisco EtherSwitch System
		headers to select a respective one of the physical links over which to transmit each of the data frames.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”)

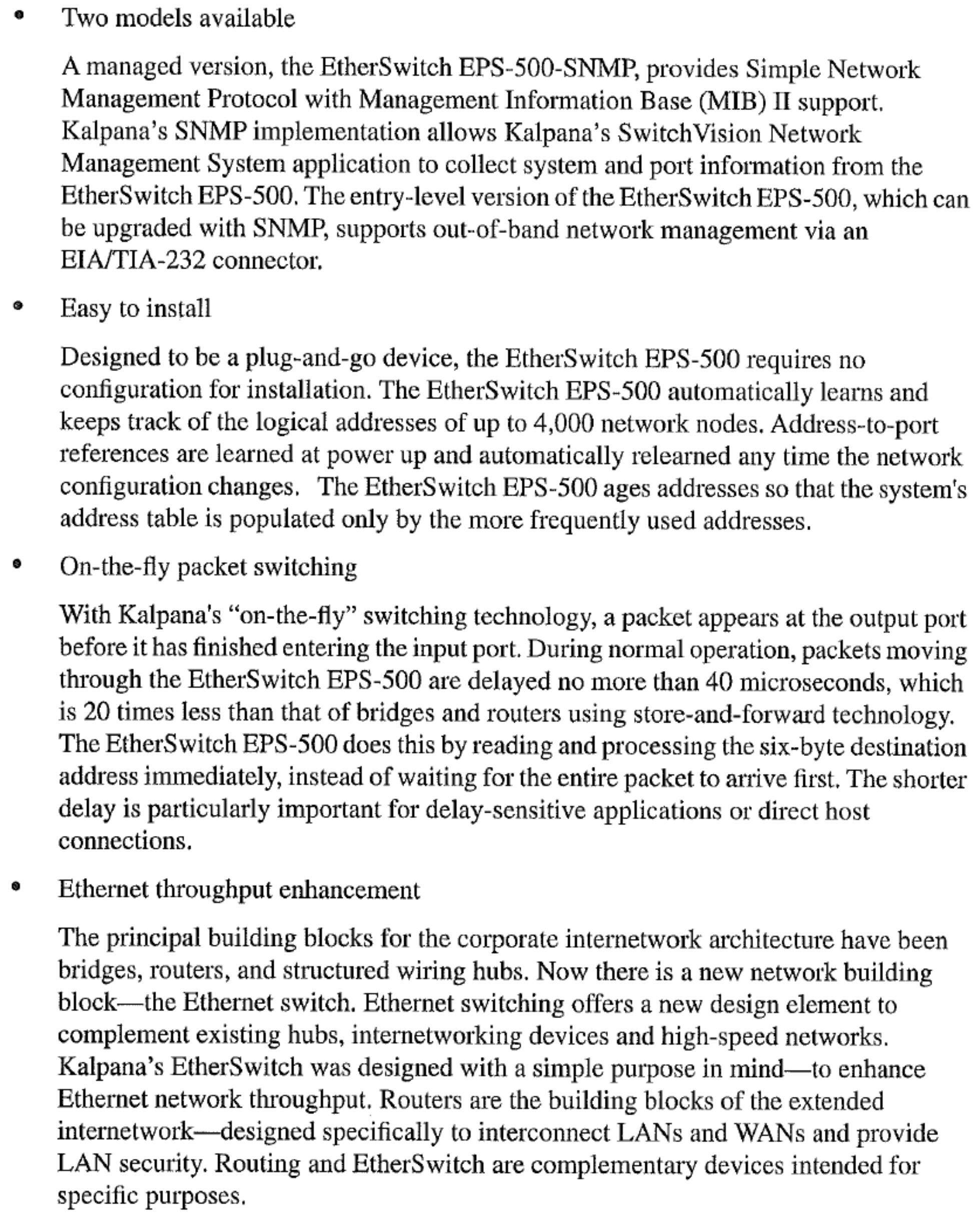
No.	’740 Patent Claim 9	Cisco EtherSwitch System

		Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:
		
		Bruckman at Table 1 (annotated)
		
		 hashing
		function
		“mapping
		function”
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		  hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		
		As another example, Solomon discloses applying a mapping function which comprises a hashing function performed by the mapper.
		
		Solomon at [0024] (“In another embodiment, switching the data packets includes mapping the data packets to the selected port responsively to the label. Additionally or alternatively, map-ping the data packets includes applying a hashing function to the label so as to determine a

No.	’740 Patent Claim 9	Cisco EtherSwitch System
		number of the selected port, and choosing the label includes applying an inverse of the hashing function to the number of the selected port.”) Solomon at [0048] (“The mapping function typically uses MPLS label 52 for mapping, since the MPLS label uniquely identifies MPLS tunnel 28, and it is required that all MPLS packets belonging to the same tunnel be switched through the same physical port 24. Additionally or alternatively, the mapping function uses a "PW" label (pseudo wire label, formerly known as a virtual connection, or VC label), which is optionally added to MPLS header 50. The PW label com-prises information that the egress node requires for deliver-ing the packet to its destination, and is optionally added during the encapsulation of MPLS packets. Additional details regarding the VC label can be found in an IETF draft by Martini et al. entitled "Encapsulation Methods for Trans-port of Ethernet Frames Over IP/MPLS Networks" (IETF draft-ietf-pwe3-ethernet-encap-07.txt, May, 2004), which is incorporated herein by reference. In some embodiments, mapper 34 applies a hashing function to the MPLS and/or PW label, as will be described below.”) Solomon at [0059] (“In this method, the mapping function used by mapper 34 of switch A is a hashing function. Various hashing functions are known in the art, and any suitable hashing function may be used in mapper 34. Since the hashing operation is performed for each packet, it is desirable to have a hashing function that is computationally simple.”) Solomon at [0060] (“As mentioned above, the hashing function typically hashes the value of MPLS label 52 to determine the selected physical port, as the MPLS label uniquely identifies tunnel 28. For example, the following hashing function may be used by mapper 34: Selected port number=l+((MPLS label) mod N), wherein N denotes the number of physical Ethernet ports in LAG group 25, and "mod" denotes the modulus operator. Assuming the values of MPLS labels are distrib-uted uniformly over a certain range, this function achieves a uniform distribution of port allocations for the different MPLS labels. It can also be seen that all packets carrying the same MPLS label (in other words-belonging to the same MPLS tunnel) will be mapped to the same physical port.”)

No.	’740 Patent Claim 9	Cisco EtherSwitch System
		Solomon at [0065] (“Mapper 34 of switch A maps each received packet to the selected physical port of LAG group 25 using the hashing function, at a hashing step 90. Mapper 34 extracts the MPLS label from each received packet and uses the hashing function to calculate the serial number of the selected physical port, which was selected by the CAC processor at step 82. Following the numerical example given above, the mapper extracts MPLS label=65647 from the packet. Substituting this value and N=3 into the hashing function gives: Selected port number=1+(65647 mod 3)=2, which is indeed the port number selected in the example above.”)

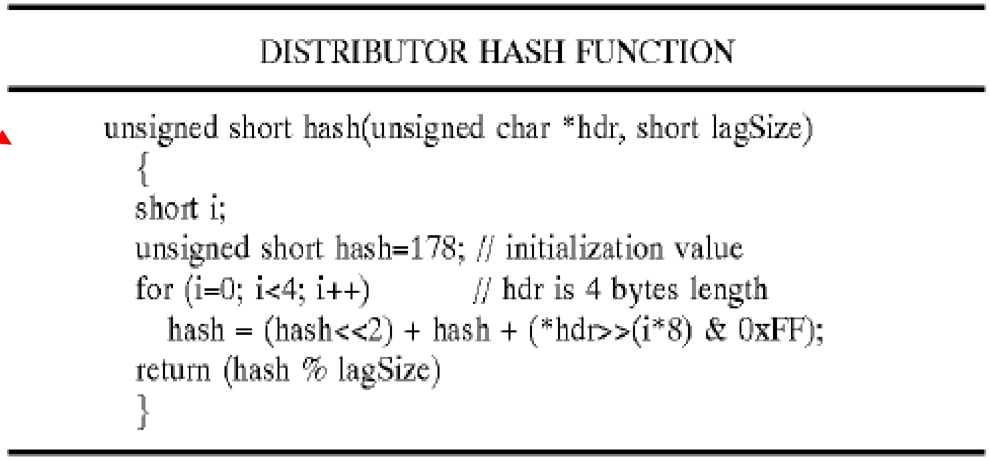
No.	’740 Patent Claim 10	Cisco EtherSwitch System
10[a]	The method according to claim 9, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second physical links,	Cisco EtherSwitch System discloses the method according to claim 9, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second physical links. For example, Cisco EtherSwitch System discloses determining which of the matrix of paths to send a data packet over by reading and processing packet information. A person of ordinary skill in the art would understand that this determining includes standard functions and algorithms, including a hash function, which is a regularly employed function in route selection, using the number of ports, paths, and internal connections. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Cisco EtherSwitch System is found to not meet this limitation, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second physical links would have been obvious to a person having ordinary skill in the art, as explained below. *See supra at* Claim 9. Cisco EtherSwitch System Catalog at 113

		 • Two models available A managed version, the EtherSwitch EPS-500-SNMP, provides Simple Network Management Protocol with Management Information Base (MIB) II support. Kalpana's SNMP implementation allows Kalpana's SwitchVision Network Management System application to collect system and port information from the EtherSwitch EPS-500. The entry-level version of the EtherSwitch EPS-500, which can be upgraded with SNMP, supports out-of-band network management via an EIA/TIA-232 connector. • Easy to install Designed to be a plug-and-go device, the EtherSwitch EPS-500 requires no configuration for installation. The EtherSwitch EPS-500 automatically learns and keeps track of the logical addresses of up to 4,000 network nodes. Address-to-port references are learned at power up and automatically relearned any time the network configuration changes. The EtherSwitch EPS-500 ages addresses so that the system's address table is populated only by the more frequently used addresses. • On-the-fly packet switching With Kalpana's "on-the-fly" switching technology, a packet appears at the output port before it has finished entering the input port. During normal operation, packets moving through the EtherSwitch EPS-500 are delayed no more than 40 microseconds, which is 20 times less than that of bridges and routers using store-and-forward technology. The EtherSwitch EPS-500 does this by reading and processing the six-byte destination address immediately, instead of waiting for the entire packet to arrive first. The shorter delay is particularly important for delay-sensitive applications or direct host connections. • Ethernet throughput enhancement The principal building blocks for the corporate internetwork architecture have been bridges, routers, and structured wiring hubs. Now there is a new network building block—the Ethernet switch. Ethernet switching offers a new design element to complement existing hubs, internetworking devices and high-speed networks. Kalpana's EtherSwitch was designed with a simple purpose in mind—to enhance Ethernet network throughput. Routers are the building blocks of the extended internetwork—designed specifically to interconnect LANs and WANs and provide LAN security. Routing and EtherSwitch are complementary devices intended for specific purposes.

No.	’740 Patent Claim 10	Cisco EtherSwitch System
		Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Cisco EtherSwitch System in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[a] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses applying a distributor hash function to the frame information which includes determining a number of the plurality of physical links. Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”)

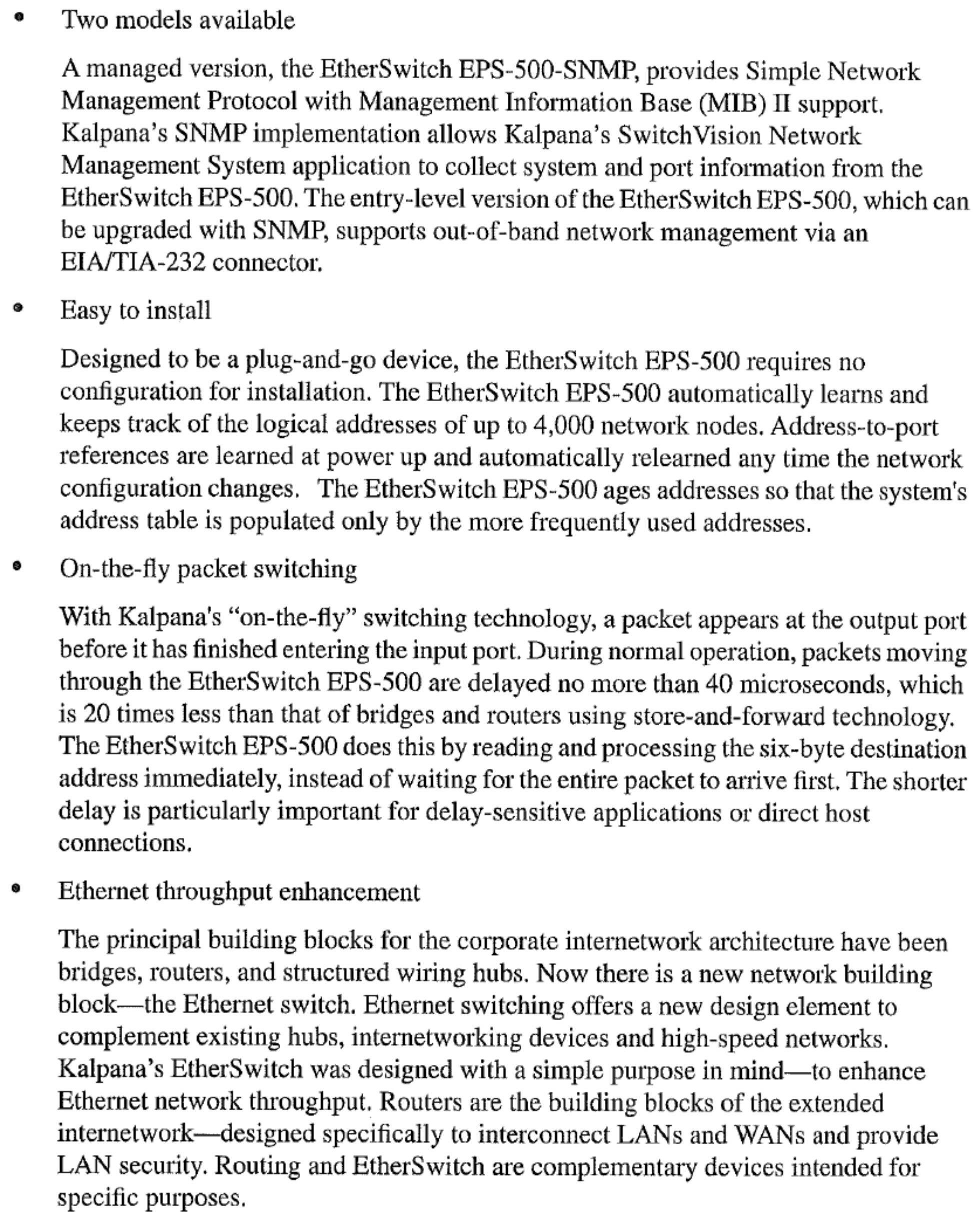
No.	’740 Patent Claim 10	Cisco EtherSwitch System
		Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values

No.	’740 Patent Claim 10	Cisco EtherSwitch System

		[0062] The hash function generates very different hash values for similar data.”)
		Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:
		Bruckman at Table 1 (annotated)
		hashing function *“mapping function”*
		
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		    hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		For example, Solomon discloses applying a distributor hash function to the frame information which includes determining a number of the plurality of physical links.

No.	’740 Patent Claim 10	Cisco EtherSwitch System
		Solomon at [0024] (“In another embodiment, switching the data packets includes mapping the data packets to the selected port responsively to the label. Additionally or alternatively, map-ping the data packets includes applying a hashing function to the label so as to determine a number of the selected port, and choosing the label includes applying an inverse of the hashing function to the number of the selected port.”) Solomon at [0048] (“The mapping function typically uses MPLS label 52 for mapping, since the MPLS label uniquely identifies MPLS tunnel 28, and it is required that all MPLS packets belonging to the same tunnel be switched through the same physical port 24. Additionally or alternatively, the mapping function uses a "PW" label (pseudo wire label, formerly known as a virtual connection, or VC label), which is optionally added to MPLS header 50. The PW label com-prises information that the egress node requires for deliver-ing the packet to its destination, and is optionally added during the encapsulation of MPLS packets. Additional details regarding the VC label can be found in an IETF draft by Martini et al. entitled "Encapsulation Methods for Trans-port of Ethernet Frames Over IP/MPLS Networks" (IETF draft-ietf-pwe3-ethernet-encap-07.txt, May, 2004), which is incorporated herein by reference. In some embodiments, mapper 34 applies a hashing function to the MPLS and/or PW label, as will be described below.”) Solomon at [0059] (“In this method, the mapping function used by mapper 34 of switch A is a hashing function. Various hashing functions are known in the art, and any suitable hashing function may be used in mapper 34. Since the hashing operation is performed for each packet, it is desirable to have a hashing function that is computationally simple.”) Solomon at [0060] (“As mentioned above, the hashing function typically hashes the value of MPLS label 52 to determine the selected physical port, as the MPLS label uniquely identifies tunnel 28. For example, the following hashing function may be used by mapper 34: Selected port number=l+((MPLS label) mod N), wherein N denotes the number of physical Ethernet ports in LAG group 25, and "mod" denotes the modulus operator. Assuming the values of MPLS labels are distrib-uted uniformly over a certain range, this function achieves a uniform distribution of port allocations for the different MPLS labels. It can also be seen that all packets

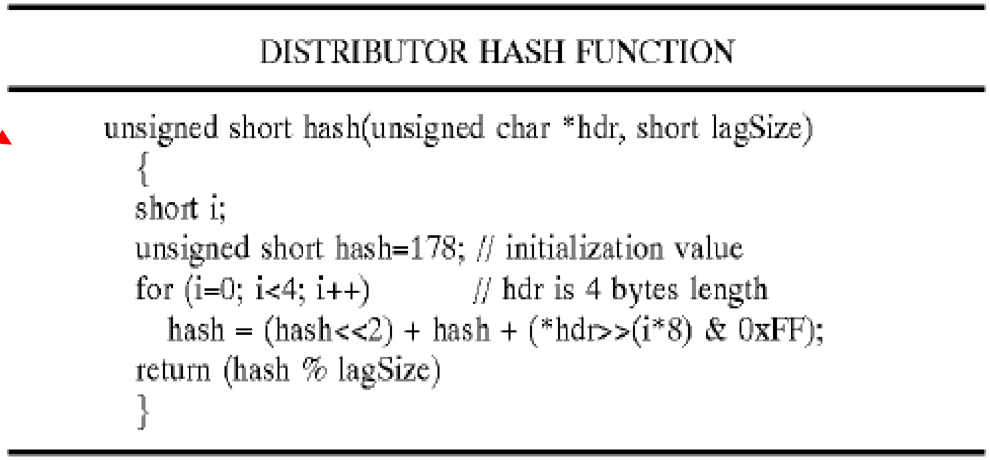
No.	’740 Patent Claim 10	Cisco EtherSwitch System
		carrying the same MPLS label (in other words-belonging to the same MPLS tunnel) will be mapped to the same physical port.”) Solomon at [0065] (“Mapper 34 of switch A maps each received packet to the selected physical port of LAG group 25 using the hashing function, at a hashing step 90. Mapper 34 extracts the MPLS label from each received packet and uses the hashing function to calculate the serial number of the selected physical port, which was selected by the CAC processor at step 82. Following the numerical example given above, the mapper extracts MPLS label=65647 from the packet. Substituting this value and N=3 into the hashing function gives: Selected port number=1+(65647 mod 3)=2, which is indeed the port number selected in the example above.”)
10[b]	applying the hashing function to the at least one of the frame attributes to produce a hashing key,	Cisco EtherSwitch System discloses applying the hashing function to the at least one of the frame attributes to produce a hashing key. For example, Cisco EtherSwitch System discloses determining which of the matrix of paths to send a data packet over by reading and processing packet information. A person of ordinary skill in the art would understand that this determination includes standard functions and algorithms, including a hash function, which is a regularly employed function in route selection, using the packet information, such as a packet destination address, to produce a routing value. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Cisco EtherSwitch System is found to not meet this limitation, applying the hashing function to the at least one of the frame attributes to produce a hashing key would have been obvious to a person having ordinary skill in the art, as explained below. Cisco EtherSwitch System Catalog at 113

		 • Two models available A managed version, the EtherSwitch EPS-500-SNMP, provides Simple Network Management Protocol with Management Information Base (MIB) II support. Kalpana's SNMP implementation allows Kalpana's SwitchVision Network Management System application to collect system and port information from the EtherSwitch EPS-500. The entry-level version of the EtherSwitch EPS-500, which can be upgraded with SNMP, supports out-of-band network management via an EIA/TIA-232 connector. • Easy to install Designed to be a plug-and-go device, the EtherSwitch EPS-500 requires no configuration for installation. The EtherSwitch EPS-500 automatically learns and keeps track of the logical addresses of up to 4,000 network nodes. Address-to-port references are learned at power up and automatically relearned any time the network configuration changes. The EtherSwitch EPS-500 ages addresses so that the system's address table is populated only by the more frequently used addresses. • On-the-fly packet switching With Kalpana's "on-the-fly" switching technology, a packet appears at the output port before it has finished entering the input port. During normal operation, packets moving through the EtherSwitch EPS-500 are delayed no more than 40 microseconds, which is 20 times less than that of bridges and routers using store-and-forward technology. The EtherSwitch EPS-500 does this by reading and processing the six-byte destination address immediately, instead of waiting for the entire packet to arrive first. The shorter delay is particularly important for delay-sensitive applications or direct host connections. • Ethernet throughput enhancement The principal building blocks for the corporate internetwork architecture have been bridges, routers, and structured wiring hubs. Now there is a new network building block—the Ethernet switch. Ethernet switching offers a new design element to complement existing hubs, internetworking devices and high-speed networks. Kalpana's EtherSwitch was designed with a simple purpose in mind—to enhance Ethernet network throughput. Routers are the building blocks of the extended internetwork—designed specifically to interconnect LANs and WANs and provide LAN security. Routing and EtherSwitch are complementary devices intended for specific purposes.

No.	’740 Patent Claim 10	Cisco EtherSwitch System
		Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Cisco EtherSwitch System in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[b] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses applying a distributor hash function to the frame information which includes determining a number of the plurality of physical links. Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the

No.	’740 Patent Claim 10	Cisco EtherSwitch System
		headers to select a respective one of the physical links over which to transmit each of the data frames.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”)

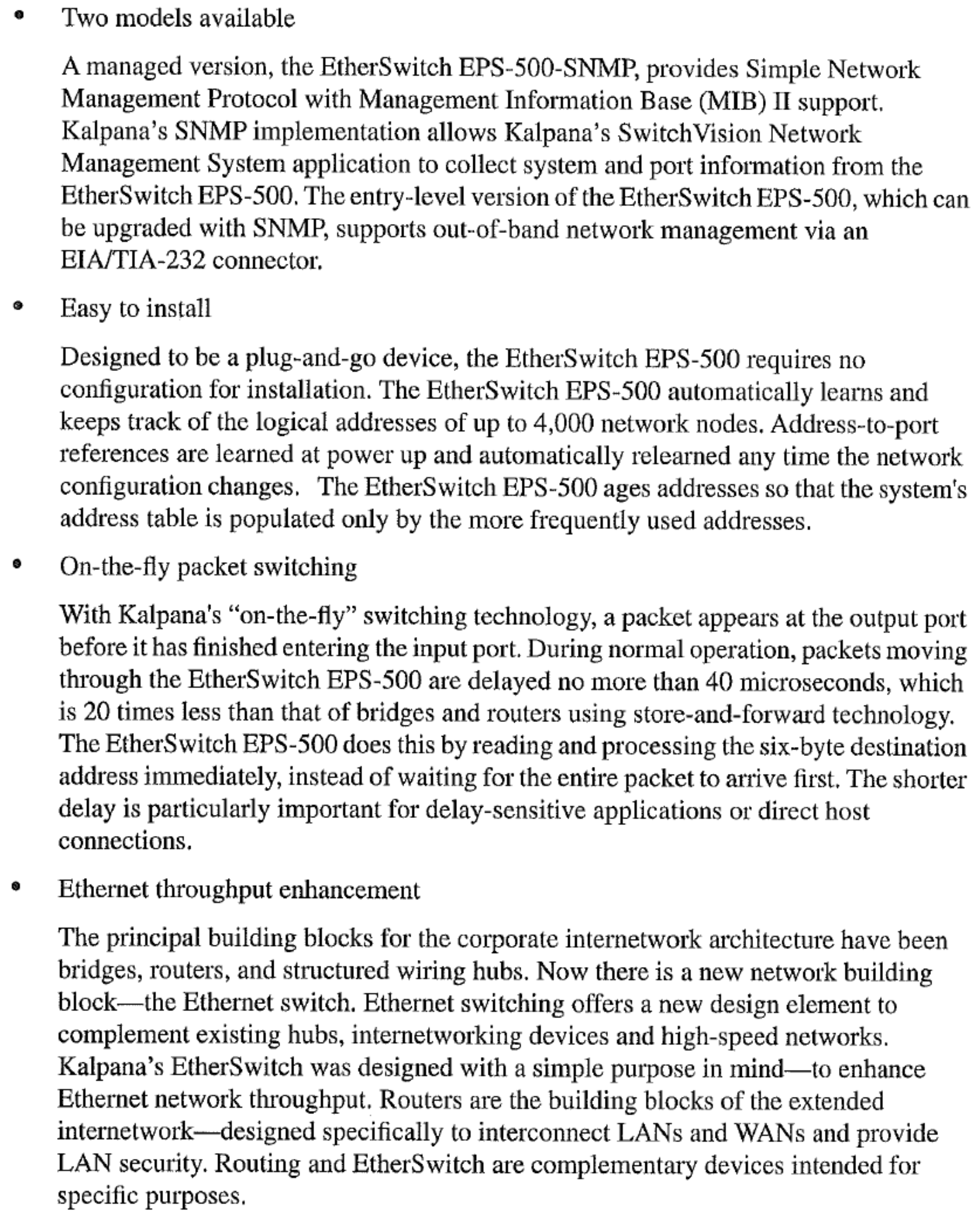
No.	’740 Patent Claim 10	Cisco EtherSwitch System

		Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:
		Bruckman at Table 1 (annotated)
		hashing function
		“mapping function”
		
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		    hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		For example, Alexander discloses applying a distributor hash function to the frame information which includes determining a hash key based on packet information.
		Alexander at 3:1-40 (“The hash function is preferably selected such that suc-cessive application of the hash function to all source and destination addresses expected to be seen by the Ethernet

No.	’740 Patent Claim 10	Cisco EtherSwitch System
		switch will produce a lowest value hash key, a highest value hash key, and a group of hash keys having intermediate values distributed evenly between the lowest and highest values. The distribution table contains a separate port identifier look-up table for each aggregated grouping of outgoing ports. Advantageously, the hash key is an N bit hash key; and, each port identifier look-up table contains 2^N entries occupying 2^N consecutive locations, with each entry being an identifier of a particular one of the physical outgoing ports. Identifiers for particular outgoing ports are retrieved from the distribution table by extracting first and second N bit hash keys which form part of the retrieved destination and source address contexts respectively. The hash keys are combined to form an N bit connection identifier. The port identifier look-up table corresponding to the aggregated grouping represented by the retrieved destination address is selected, and the entry at the table location corresponding to the value of the N bit connection identifier is retrieved. If the address look-up table does not contain a destination address corresponding to the extracted destination address then first and second hash keys are produced by applying a hash function to the extracted source and destination addresses respectively. The hash keys are combined to form an N bit connection identifier. The incoming port on which the packet containing the extracted source address was received is identified. All of the aggregated groupings are scanned to identify all outgoing ports to which packets may be directed from the incoming port on which the packet was received. For each one of those outgoing ports, the port identifier look-up table corresponding to the aggregated grouping containing that outgoing port is selected, the entry at the table location corresponding to the value of the N bit connection identifier is retrieved, and the received packet is queued for outgoing transmission on the outgoing port corresponding to the retrieved entry.”) Alexander at 5:10-35 (“If a packet arrives bearing a source Ethernet MAC address that was not found in look-up table 12 by address resolution unit 10, learning function 16 is invoked to update look-up table 12 with the new address (i.e. processing branches along the "No" exit from FIG. 2, block 36). Learning function 16 first computes a hash function on the source Ethernet MAC address, generating an N-bit hash key ("partial connection identifier") from the 48-bit

No.	’740 Patent Claim 10	Cisco EtherSwitch System
		MAC address, where N is some small integer in the range of 3 to 8 (FIG. 2, block 38). The physical port on which the packet arrived is then determined. If the physical port is found to be associated with an aggregate group (i.e., it is one of a set of ports that have been bound into a single logical port), then the logical identifier assigned to the aggregate group is also determined. The hash key is then stored into address look-up table 12 in conjunction with the actual Ethernet MAC address and the port identifier (FIG. 2, block 40). The physical port identifier is used if the port is not part of an aggregate group (i.e. if processing branched along the "No" exit from block 30 and through block 32), while the logical identifier is used for ports that have been aggregated (i.e. if processing branched along the "Yes" exit from block 30 and through block 34). The hash key and port identifier are considered to form the "context" for the given MAC address.”) Alexander at 5:36-46 (“The hash function should be selected to ensure an even distribution of hash key values over the range of MAC addresses that are expected to be seen by the Ethernet switch. As a specific example, the EXACT™ Ethernet switch system employs an exclusive-OR based hash function, wherein the 48-bit MAC address is divided into 16-bit blocks, which are then exclusive-ORed together to form a single 16-bit number; the 3 least significant bits (LSBs) of this number are taken to produce a 3-bit hash key. Other schemes such as CRC-based or checksum-based hashes may also be used.”) Alexander at 6:49-65 (“If the context information for the destination address indicates, however, that the target is an aggregate group (i.e. if processing branches along the "Yes" exit from FIG. 2, block 42) then the logical identifier assigned to the aggregate group is retrieved and is used to select the proper look-up table contained within the distribution table data structure. The hash keys (partial connection identifiers) stored into the contexts for the source and destination MAC addresses are obtained from address resolution unit 10 and combined to generate a "connection identifier" with the same number of bits (FIG. 2, block 44). (In the EXACT™ Ethernet switch, a Boolean exclusive-OR operation is used to combine the hash keys without increasing the number of bits.) This connection identifier is then used to index into the selected look-up table, and finally retrieve an actual physical port index on which the packet must be transmitted (FIG. 2, block 46).”)

No.	’740 Patent Claim 10	Cisco EtherSwitch System
10[c]	calculating a modulo of a division operation of the hashing key by the hashing size, and	Cisco EtherSwitch System discloses calculating a modulo of a division operation of the hashing key by the hashing size. For example, Cisco EtherSwitch System discloses determining which of the matrix of paths to send a data packet over by reading and processing packet information. A person of ordinary skill in the art would understand that this determination includes standard functions and algorithms, including a hash function, which is a regularly employed function in route selection, using standard mathematical operations, like division. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Cisco EtherSwitch System is found to not meet this limitation, calculating a modulo of a division operation of the hashing key by the hashing size would have been obvious to a person having ordinary skill in the art, as explained below. Cisco EtherSwitch System Catalog at 113

		 • Two models available A managed version, the EtherSwitch EPS-500-SNMP, provides Simple Network Management Protocol with Management Information Base (MIB) II support. Kalpana's SNMP implementation allows Kalpana's SwitchVision Network Management System application to collect system and port information from the EtherSwitch EPS-500. The entry-level version of the EtherSwitch EPS-500, which can be upgraded with SNMP, supports out-of-band network management via an EIA/TIA-232 connector. • Easy to install Designed to be a plug-and-go device, the EtherSwitch EPS-500 requires no configuration for installation. The EtherSwitch EPS-500 automatically learns and keeps track of the logical addresses of up to 4,000 network nodes. Address-to-port references are learned at power up and automatically relearned any time the network configuration changes. The EtherSwitch EPS-500 ages addresses so that the system's address table is populated only by the more frequently used addresses. • On-the-fly packet switching With Kalpana's "on-the-fly" switching technology, a packet appears at the output port before it has finished entering the input port. During normal operation, packets moving through the EtherSwitch EPS-500 are delayed no more than 40 microseconds, which is 20 times less than that of bridges and routers using store-and-forward technology. The EtherSwitch EPS-500 does this by reading and processing the six-byte destination address immediately, instead of waiting for the entire packet to arrive first. The shorter delay is particularly important for delay-sensitive applications or direct host connections. • Ethernet throughput enhancement The principal building blocks for the corporate internetwork architecture have been bridges, routers, and structured wiring hubs. Now there is a new network building block—the Ethernet switch. Ethernet switching offers a new design element to complement existing hubs, internetworking devices and high-speed networks. Kalpana's EtherSwitch was designed with a simple purpose in mind—to enhance Ethernet network throughput. Routers are the building blocks of the extended internetwork—designed specifically to interconnect LANs and WANs and provide LAN security. Routing and EtherSwitch are complementary devices intended for specific purposes.

No.	’740 Patent Claim 10	Cisco EtherSwitch System
		Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Cisco EtherSwitch System in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[c] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses distributing data frames over physical links and traces based on a hash function involving a division operation (%). Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0025] (“Typically, setting the protection policy includes determining a maximum number of the physical links that may fail while the logical link continues to provide at least the guaranteed bandwidth for the connection. In one embodiment, the guaranteed bandwidth is a bandwidth B, and the plurality of physical links consists of N links, and the maximum number is an integer P, and the link bandwidth allocated to each of the links is no less than B/(N-P). Conveying the data may further include managing the transmission of the data responsively to an actual number X of the physical links that have failed so that the guaranteed bandwidth on each of the links is limited to B/(N-X), X P, and an excess bandwidth on the physical links over the guaranteed bandwidth is available for other connections.”) Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link

<table>
<tr><th>No.</th><th>’740 Patent Claim 10</th><th>Cisco EtherSwitch System</th></tr>
<tr><td></td><td></td><td>

include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”)

Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:

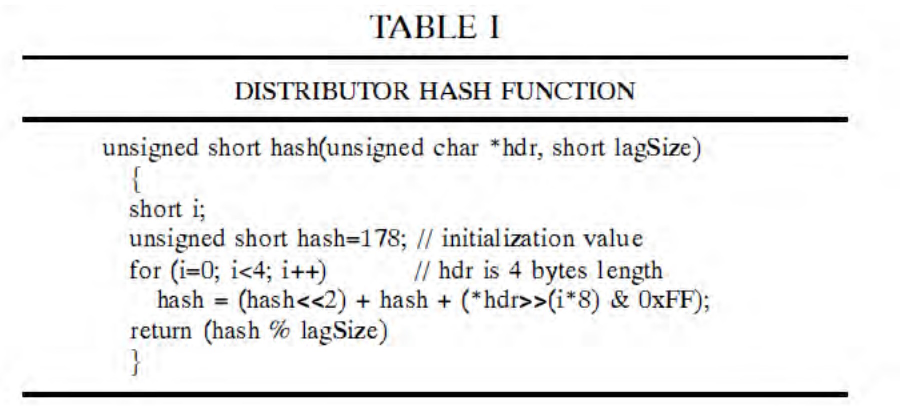


TABLE I

DISTRIBUTOR HASH FUNCTION

```
unsigned short hash(unsigned char *hdr, short lagSize)
  {
  short i;
  unsigned short hash=178; // initialization value
  for (i=0; i<4; i++)          // hdr is 4 bytes length
     hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
  return (hash % lagSize)
  }
```

”)

Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)

Bruckman at [0067] (“A similar problem may arise if there is a failure in a link in an aggregation group or in one of a number of line cards serving the aggregation group. In this case, to maintain the bandwidth allocation B made by CAC 44, each of the remaining links in the group must now carry, on average, B/(N-M) traffic, wherein M is the number of links in the

</td></tr>
</table>

No.	’740 Patent Claim 10	Cisco EtherSwitch System
		group that are out of service. If only BIN has been allocated to each link, the remaining active links may not have sufficient bandwidth to continue to provide the bandwidth that has been guaranteed to the connections that they are required to carry. A similar problem arises with respect to loading of traces 52. For example, if there is a failure in LC2 or in one of links 30 in group 36 that connect to LC2, the trace connecting the multiplexer 50 in LCl will have to carry a substantially larger share of the bandwidth, or even all of the bandwidth, that is allocated to the connection in ques-tion.”) Bruckman at [0068] (“FIG. 3 is a flow chart that schematically illustrates a method for dealing with these problems of fluctuating bandwidth requirements, in accordance with an embodiment of the present invention. In order to provide sufficient bandwidth for failure protection, CAC 44 uses a safety margin based on a protection parameter P, which is assigned at a protection setting step 60. P represents the maximum number of links in the group that can be out of service while still permitting the aggregation group to provide a given connection with the bandwidth that has been guaranteed to the connection. CAC 44 will then allocate at least B/(N-P) bandwidth to each link in the group, so that if P links fail, the group still provides total bandwidth of (N-P)*B/(N-P)= B. Setting P=l is equivalent to 1:N protection, so that the group will be unaffected by failure of a single link. In the example of group 36, shown in FIG. 2, setting P=2 will give both facility and equipment protection, i.e., the group will be unaffected not only by failure of a link, but also by failure of one of line cards 34. In the extreme case, in which P=N-1, CAC 44 will allocate the full bandwidth B on each link in the group.”) As another example, Singh discloses determining a ratio between the number of ingress and egress links and the number of links carrying data to the backplane and using a modulo to correspond to the channel’s link number. Singh at 9:30-43 (“The ratio between the number of line ingress links and the number of links carrying data to the backplane gives the backplane speedup for the system. In this example, there are 10 ingress links into the MS and 20 links (2 backplane channels) carrying that data to the backplane. This gives a backplane speedup of 2x. As another example, with 8 ingress links and 12 backplane links, there is a speedup of 1.5x. It should be noted that in addition to the

No.	’740 Patent Claim 10	Cisco EtherSwitch System
		backplane speedup, there is also an ingress/egress speedup. With 10 ingress links capable of carrying 2 Gbps each of raw data, this presents a 20 Gbps interface to the MS. An OC-192 only has approximately 10 Gbps worth of data. Taking into account cell overhead and cell quantization inefficiencies, there still remains excess capacity in the links.”) Singh at 11:29-38 (“FIG. 9 is a diagram illustrating link to channel assignments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a channel, whether it is a backplane channel or an ingress/egress channel. There is no logical relationship between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.”) Singh at 15:15-39 (“The number of crossbars that are required in a system is dependent on how many links are being used to create the backplane channels. There should be an even number of crossbars and they would be divided evenly across the switch cards. The following equation, for most cases, pro-vides the correct number of crossbars: # of Crossbars=(# links per ingress channelx# of ingress channels per portx# of port cardsx speedup)/32. For the 8x8 configuration, the # of crossbars should be multiplied by (4x# of iMS)/(# backplane channels per port card). The number of port cards should be rounded up to the nearest supported configuration, i.e. 8, 16, or 32. The speedup in the case of crossbars should be the fractional speedup that is desired. Example to determine the number of arbiters and cross-bars for the following system: 4 channel port cards (40 Gbps) 8 links per channel 16 port cards

No.	’740 Patent Claim 10	Cisco EtherSwitch System
		Speedup=l.5 # of arbiters=(4x2x2)/2=8 # of crossbars=(8x4x16xl.5)/32=24. This would give 3crossbars per arbiter.”) Singh at 16:28-44 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 16x16 and 32x32 is the organization of the switchplane. The port card remains the same. Backplane channels 1 and 2 are used for the backplane connectivity. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 16, correspond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”) Singh at 17:31-49 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 8x8 and 16x16 is the organi-zation of the switchplane. The port card remains the same. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Links 0-7 and 24-31 on the arbiters would not be used and would be powered off. Links 0-7 and 24-31 on the crossbars would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Backplane channels 1 and 2 are used for the backplane connectivity. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 8, corre-spond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”)

No.	’740 Patent Claim 10	Cisco EtherSwitch System
10[d]	selecting the first and second physical links responsively to the modulo.	Cisco EtherSwitch System discloses selecting the first and second physical links responsively to the modulo. For example, Cisco EtherSwitch System discloses determining which of the matrix of paths to send a data packet over by reading and processing packet information. A person of ordinary skill in the art would understand that this determination includes standard functions and algorithms, including a hash function, which is a regularly employed function in route selection. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Cisco EtherSwitch System is found to not meet this limitation, selecting the first and second physical links responsively to the modulo would have been obvious to a person having ordinary skill in the art, as explained below. Cisco EtherSwitch System Catalog at 113

- Two models available

  A managed version, the EtherSwitch EPS-500-SNMP, provides Simple Network Management Protocol with Management Information Base (MIB) II support. Kalpana's SNMP implementation allows Kalpana's SwitchVision Network Management System application to collect system and port information from the EtherSwitch EPS-500. The entry-level version of the EtherSwitch EPS-500, which can be upgraded with SNMP, supports out-of-band network management via an EIA/TIA-232 connector.

- Easy to install

  Designed to be a plug-and-go device, the EtherSwitch EPS-500 requires no configuration for installation. The EtherSwitch EPS-500 automatically learns and keeps track of the logical addresses of up to 4,000 network nodes. Address-to-port references are learned at power up and automatically relearned any time the network configuration changes. The EtherSwitch EPS-500 ages addresses so that the system's address table is populated only by the more frequently used addresses.

- On-the-fly packet switching

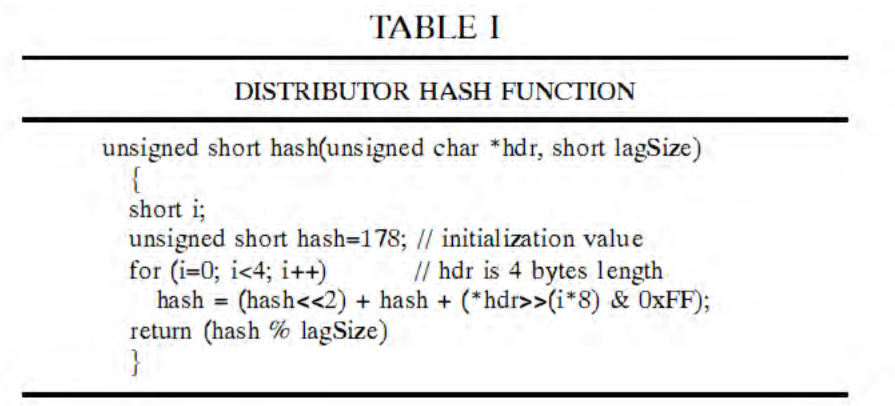
  With Kalpana's "on-the-fly" switching technology, a packet appears at the output port before it has finished entering the input port. During normal operation, packets moving through the EtherSwitch EPS-500 are delayed no more than 40 microseconds, which is 20 times less than that of bridges and routers using store-and-forward technology. The EtherSwitch EPS-500 does this by reading and processing the six-byte destination address immediately, instead of waiting for the entire packet to arrive first. The shorter delay is particularly important for delay-sensitive applications or direct host connections.

- Ethernet throughput enhancement

  The principal building blocks for the corporate internetwork architecture have been bridges, routers, and structured wiring hubs. Now there is a new network building block—the Ethernet switch. Ethernet switching offers a new design element to complement existing hubs, internetworking devices and high-speed networks. Kalpana's EtherSwitch was designed with a simple purpose in mind—to enhance Ethernet network throughput. Routers are the building blocks of the extended internetwork—designed specifically to interconnect LANs and WANs and provide LAN security. Routing and EtherSwitch are complementary devices intended for specific purposes.

No.	’740 Patent Claim 10	Cisco EtherSwitch System
		Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Cisco EtherSwitch System in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[d] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses distributing data frames over physical links and traces based on a hash function involving a division operation (%). Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0025] (“Typically, setting the protection policy includes determining a maximum number of the physical links that may fail while the logical link continues to provide at least the guaranteed bandwidth for the connection. In one embodiment, the guaranteed bandwidth is a bandwidth B, and the plurality of physical links consists of N links, and the maximum number is an integer P, and the link bandwidth allocated to each of the links is no less than B/(N-P). Conveying the data may further include managing the transmission of the data responsively to an actual number X of the physical links that have failed so that the guaranteed bandwidth on each of the links is limited to B/(N-X), X P, and an excess bandwidth on the physical links over the guaranteed bandwidth is available for other connections.”) Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link

No.	’740 Patent Claim 10	Cisco EtherSwitch System

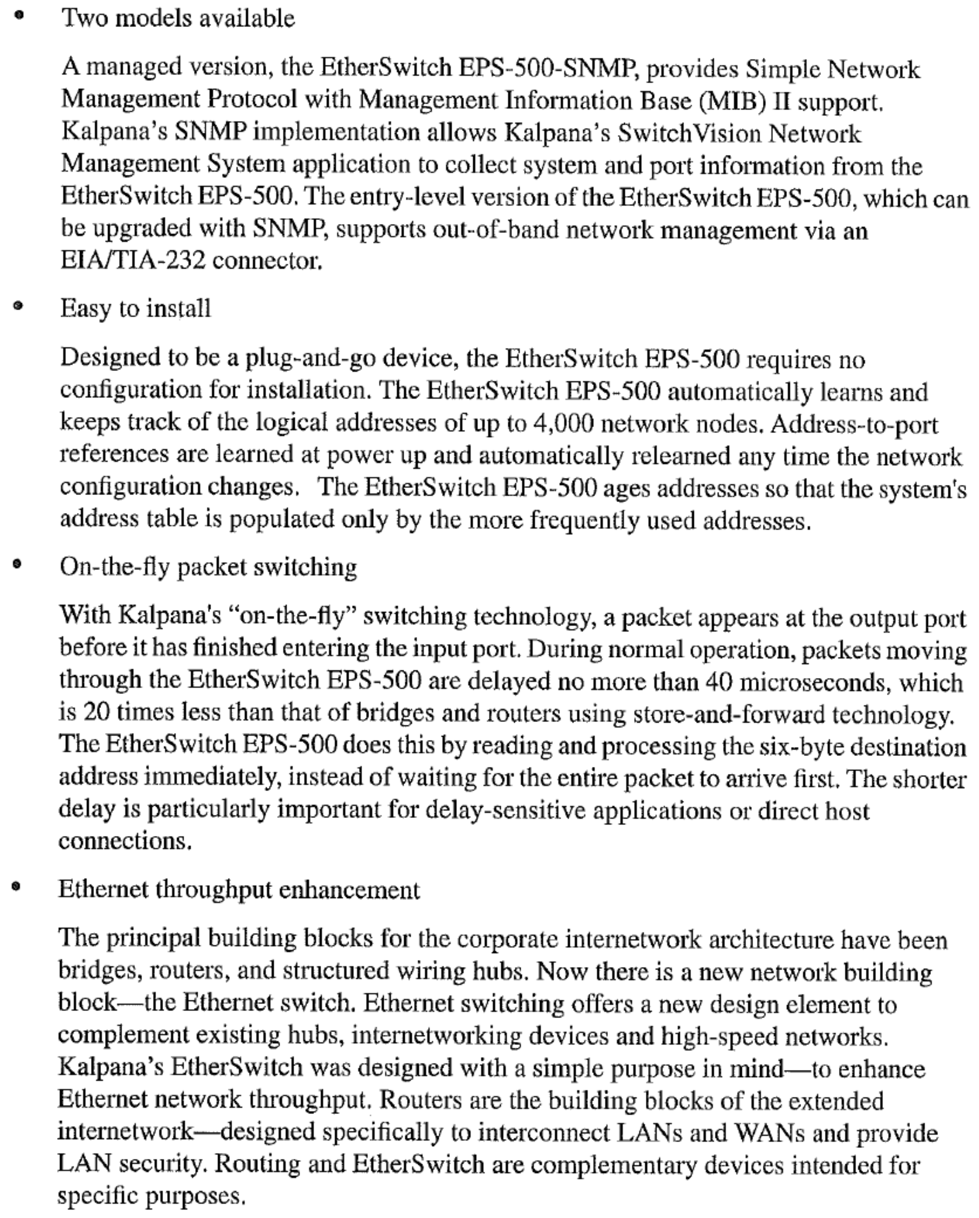
		include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”)
		
		Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:
		
		
		TABLE I
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		”)
		
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		
		Bruckman at [0067] (“A similar problem may arise if there is a failure in a link in an aggregation group or in one of a number of line cards serving the aggregation group. In this case, to maintain the bandwidth allocation B made by CAC 44, each of the remaining links in the group must now carry, on average, B/(N-M) traffic, wherein M is the number of links in the

No.	’740 Patent Claim 10	Cisco EtherSwitch System
		group that are out of service. If only BIN has been allocated to each link, the remaining active links may not have sufficient bandwidth to continue to provide the bandwidth that has been guaranteed to the connections that they are required to carry. A similar problem arises with respect to loading of traces 52. For example, if there is a failure in LC2 or in one of links 30 in group 36 that connect to LC2, the trace connecting the multiplexer 50 in LCl will have to carry a substantially larger share of the bandwidth, or even all of the bandwidth, that is allocated to the connection in ques-tion.”) Bruckman at [0068] (“FIG. 3 is a flow chart that schematically illustrates a method for dealing with these problems of fluctuating bandwidth requirements, in accordance with an embodiment of the present invention. In order to provide sufficient bandwidth for failure protection, CAC 44 uses a safety margin based on a protection parameter P, which is assigned at a protection setting step 60. P represents the maximum number of links in the group that can be out of service while still permitting the aggregation group to provide a given connection with the bandwidth that has been guaranteed to the connection. CAC 44 will then allocate at least B/(N-P) bandwidth to each link in the group, so that if P links fail, the group still provides total bandwidth of (N-P)*B/(N-P)= B. Setting P=l is equivalent to 1:N protection, so that the group will be unaffected by failure of a single link. In the example of group 36, shown in FIG. 2, setting P=2 will give both facility and equipment protection, i.e., the group will be unaffected not only by failure of a link, but also by failure of one of line cards 34. In the extreme case, in which P=N-1, CAC 44 will allocate the full bandwidth B on each link in the group.”) As another example, Singh discloses determining a ratio between the number of ingress and egress links and the number of links carrying data to the backplane and using a modulo to correspond to the channel’s link number. Singh at 9:30-43 (“The ratio between the number of line ingress links and the number of links carrying data to the backplane gives the backplane speedup for the system. In this example, there are 10 ingress links into the MS and 20 links (2 backplane channels) carrying that data to the backplane. This gives a backplane speedup of 2x. As another example, with 8 ingress links and 12 backplane links, there is a speedup of 1.5x. It should be noted that in addition to the

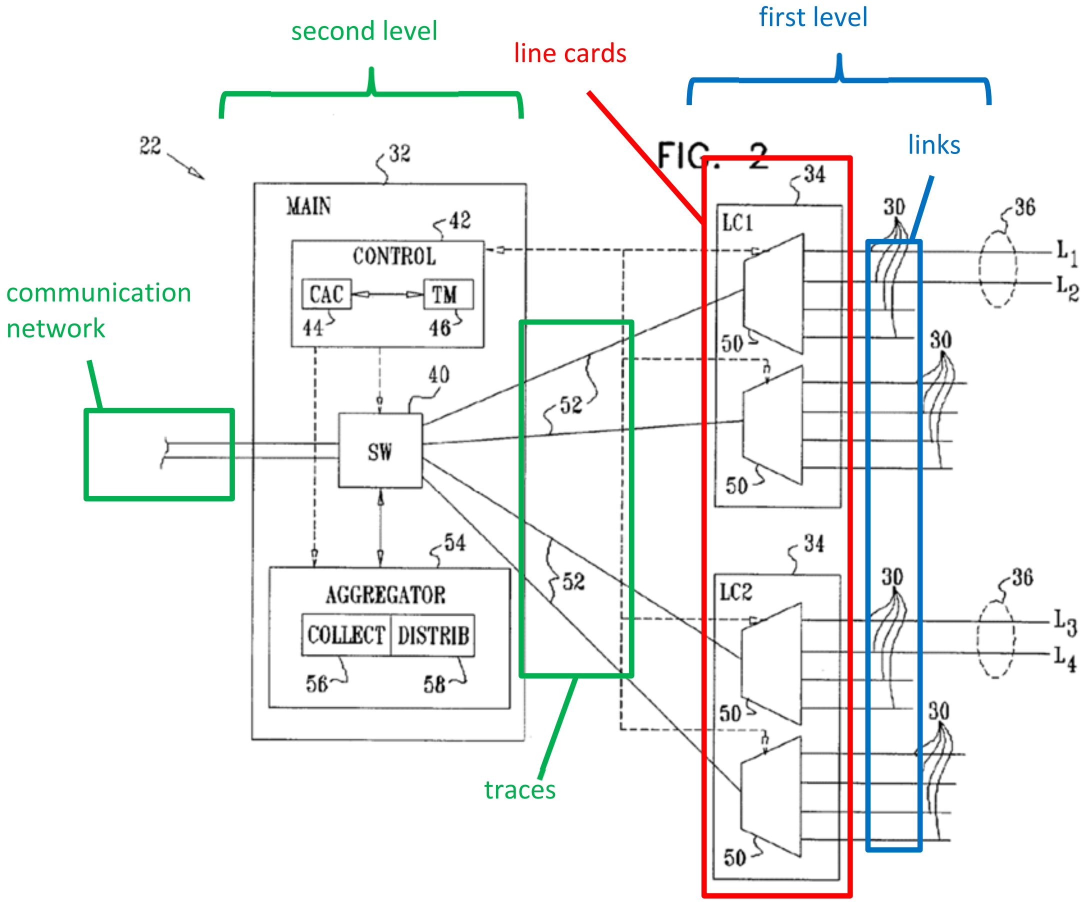
No.	’740 Patent Claim 10	Cisco EtherSwitch System
		backplane speedup, there is also an ingress/egress speedup. With 10 ingress links capable of carrying 2 Gbps each of raw data, this presents a 20 Gbps interface to the MS. An OC-192 only has approximately 10 Gbps worth of data. Taking into account cell overhead and cell quantization inefficiencies, there still remains excess capacity in the links.”) Singh at 11:29-38 (“FIG. 9 is a diagram illustrating link to channel assignments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a channel, whether it is a backplane channel or an ingress/egress channel. There is no logical relationship between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.”) Singh at 15:15-39 (“The number of crossbars that are required in a system is dependent on how many links are being used to create the backplane channels. There should be an even number of crossbars and they would be divided evenly across the switch cards. The following equation, for most cases, pro-vides the correct number of crossbars: # of Crossbars=(# links per ingress channelx# of ingress channels per portx# of port cardsx speedup)/32. For the 8x8 configuration, the # of crossbars should be multiplied by (4x# of iMS)/(# backplane channels per port card). The number of port cards should be rounded up to the nearest supported configuration, i.e. 8, 16, or 32. The speedup in the case of crossbars should be the fractional speedup that is desired. Example to determine the number of arbiters and cross-bars for the following system: 4 channel port cards (40 Gbps) 8 links per channel 16 port cards

No.	’740 Patent Claim 10	Cisco EtherSwitch System
		Speedup=l.5 # of arbiters=(4x2x2)/2=8 # of crossbars=(8x4x16xl.5)/32=24. This would give 3crossbars per arbiter.”) Singh at 16:28-44 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 16x16 and 32x32 is the organization of the switchplane. The port card remains the same. Backplane channels 1 and 2 are used for the backplane connectivity. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 16, correspond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”) Singh at 17:31-49 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 8x8 and 16x16 is the organi-zation of the switchplane. The port card remains the same. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Links 0-7 and 24-31 on the arbiters would not be used and would be powered off. Links 0-7 and 24-31 on the crossbars would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Backplane channels 1 and 2 are used for the backplane connectivity. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 8, corre-spond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”)

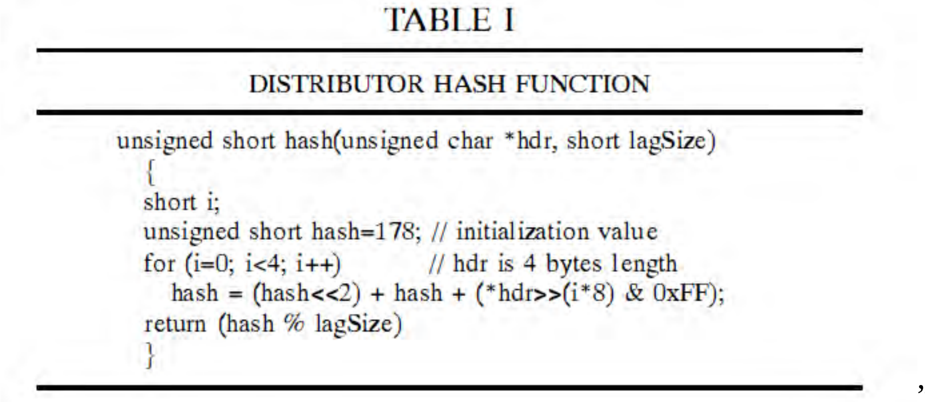
No.	’740 Patent Claim 11	Cisco EtherSwitch System
11	The method according to claim 10, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo.	Cisco EtherSwitch System discloses the method according to claim 10, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo. For example, Cisco EtherSwitch System discloses determining which of the matrix of paths to send a data packet over by reading and processing packet information. A person of ordinary skill in the art would understand that this determining includes standard functions and algorithms, including a hash function, which is a regularly employed function in route selection, using the number of ports, paths, and internal connections. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Cisco EtherChannel System is found to not meet this limitation, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo would have been obvious to a person having ordinary skill in the art, as explained below. *See supra* Claim 10 Cisco EtherSwitch System Catalog at 113

		 • Two models available A managed version, the EtherSwitch EPS-500-SNMP, provides Simple Network Management Protocol with Management Information Base (MIB) II support. Kalpana's SNMP implementation allows Kalpana's SwitchVision Network Management System application to collect system and port information from the EtherSwitch EPS-500. The entry-level version of the EtherSwitch EPS-500, which can be upgraded with SNMP, supports out-of-band network management via an EIA/TIA-232 connector. • Easy to install Designed to be a plug-and-go device, the EtherSwitch EPS-500 requires no configuration for installation. The EtherSwitch EPS-500 automatically learns and keeps track of the logical addresses of up to 4,000 network nodes. Address-to-port references are learned at power up and automatically relearned any time the network configuration changes. The EtherSwitch EPS-500 ages addresses so that the system's address table is populated only by the more frequently used addresses. • On-the-fly packet switching With Kalpana's "on-the-fly" switching technology, a packet appears at the output port before it has finished entering the input port. During normal operation, packets moving through the EtherSwitch EPS-500 are delayed no more than 40 microseconds, which is 20 times less than that of bridges and routers using store-and-forward technology. The EtherSwitch EPS-500 does this by reading and processing the six-byte destination address immediately, instead of waiting for the entire packet to arrive first. The shorter delay is particularly important for delay-sensitive applications or direct host connections. • Ethernet throughput enhancement The principal building blocks for the corporate internetwork architecture have been bridges, routers, and structured wiring hubs. Now there is a new network building block—the Ethernet switch. Ethernet switching offers a new design element to complement existing hubs, internetworking devices and high-speed networks. Kalpana's EtherSwitch was designed with a simple purpose in mind—to enhance Ethernet network throughput. Routers are the building blocks of the extended internetwork—designed specifically to interconnect LANs and WANs and provide LAN security. Routing and EtherSwitch are complementary devices intended for specific purposes.

No.	’740 Patent Claim 11	Cisco EtherSwitch System
		Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Cisco EtherSwitch System in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 11 of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses distributing data frames over physical links and traces based on a division operation in the hash function involving specific byte lengths of the frame information. Bruckman at Figure 2 (annotated)

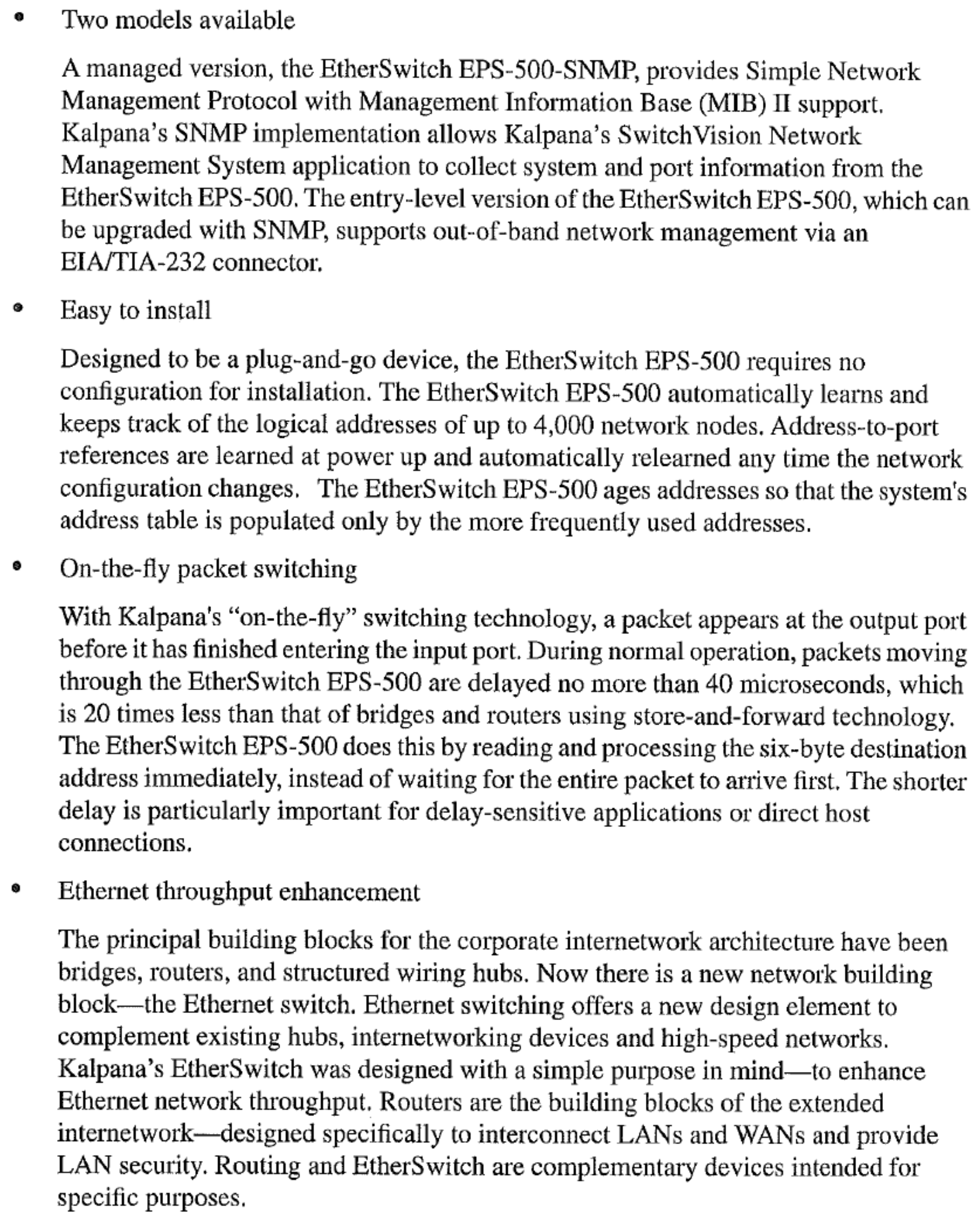
No.	’740 Patent Claim 11	Cisco EtherSwitch System
		 second level line cards first level links communication network traces 22 32 MAIN 42 CONTROL CAC TM 44 46 40 SW 52 52 54 AGGREGATOR COLLECT DISTRIB 56 58 34 LC1 50 50 34 LC2 50 50 30 30 30 30 36 36 L1 L2 L3 L4 FIG. 2 Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:

No.	’740 Patent Claim 11	Cisco EtherSwitch System

		
		TABLE I
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		”)
		
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		
		As another example, Solomon discloses using a subset of bits to encode for the selected physical port involving specific byte lengths of the frame information
		
		Solomon at [0054] (“Having selected a physical port, RSVP-TE proces-sor 30 of switch A now generates a suitable MPLS label, at a label generation step 64. The preceding node upstream of switch A will subsequently attach this MPLS label to all MPLS packets transmitted through tunnel 28 to switch A. The label is assigned, in conjunction with the mapping function of mapper 34, so as to ensure that all MPLS packets carrying this label are switched through the physical port that was selected for this tunnel at step 62. For this purpose, RSVP-TE processor 30 of switch A dedicates a sub-set of the bits of MPLS label 52 to encode the serial number of the selected physical port. For example, the four least-signifi-cant bits of MPLS label 52 may be used for encoding the selected port number. This configuration is suitable for representing LAG

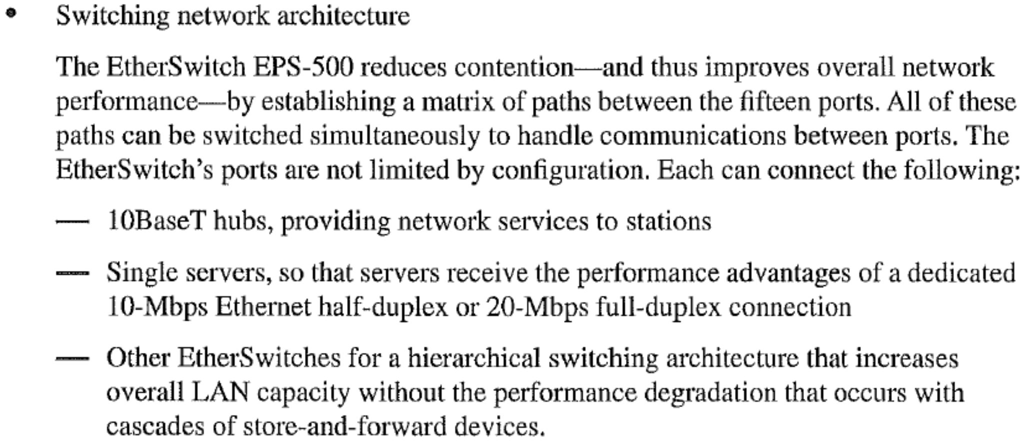
No.	’740 Patent Claim 11	Cisco EtherSwitch System
		groups having up to 16 physical ports (N<16). The remaining bits of MPLS label 52 may be chosen at random or using any suitable method known in the art.”) Solomon at [0056] (“Mapper 34 of switch A maps the received packets belonging to tunnel 28 to the selected physical Ethernet port at a mapping step 70. For this purpose, mapper 34 extracts the MPLS label from each received packet and decodes the selected physical port number from the dedicated sub-set of bits, such as the four LSB, as described in step 64 above. The decoded value is used for mapping the packet to the selected physical port, which was allocated by the CAC processor at step 62 above. In the four-bit example described above, the mapping function may be written explicitly as: Selected port number=((MPLS label) and (0x0000F)), wherein "and" denotes the "bitwise and" operator.”)

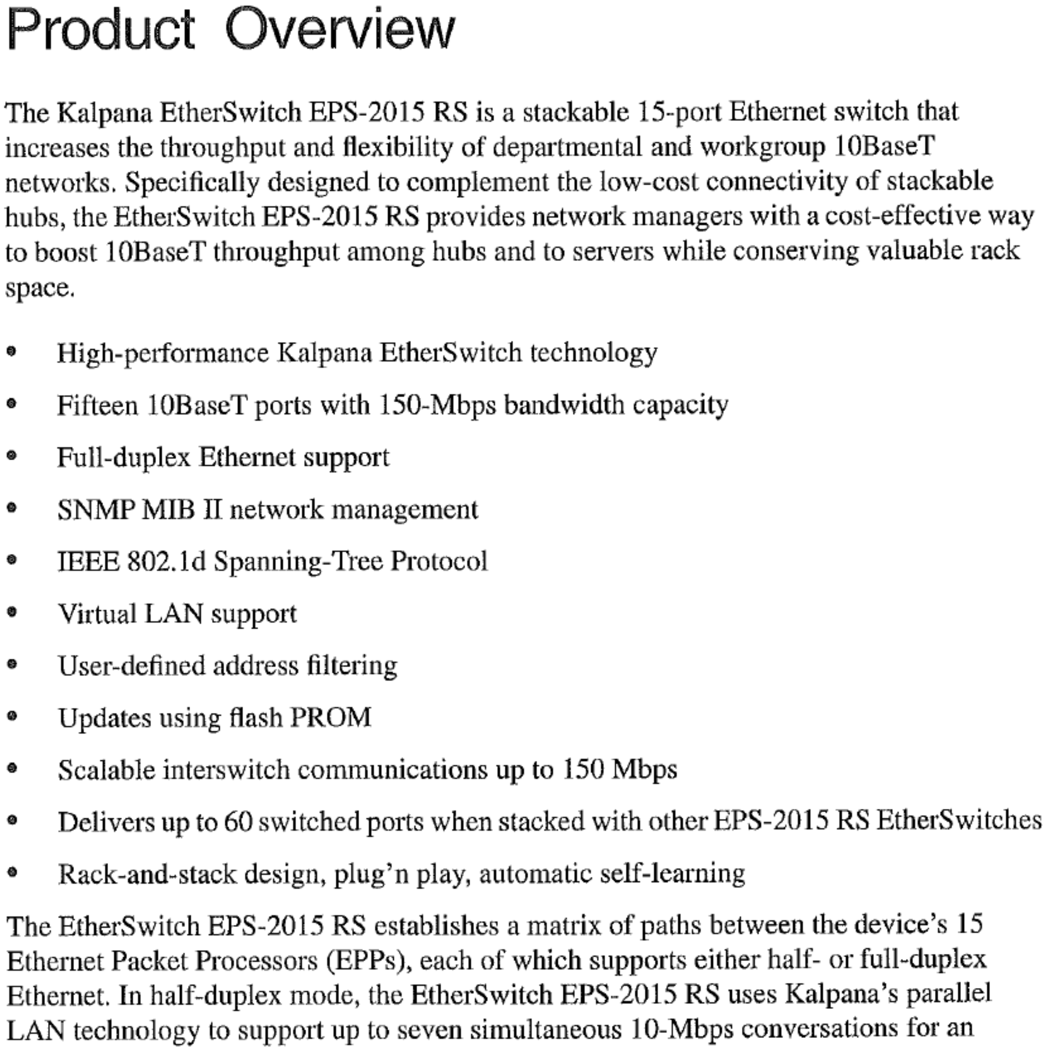
No.	’740 Patent Claim 12	Cisco EtherSwitch System
12	The method according to claim 1, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source	Cisco EtherSwitch System discloses the method according to claim 1, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port. For example, Cisco EtherSwitch System discloses reading and processing data packet information, such as destination addresses. A person of ordinary skill in the art would understand that the EtherSwitch could read and process other versions of data packet information as well. *See supra* Claim 1 Cisco EtherSwitch System Catalog at 113

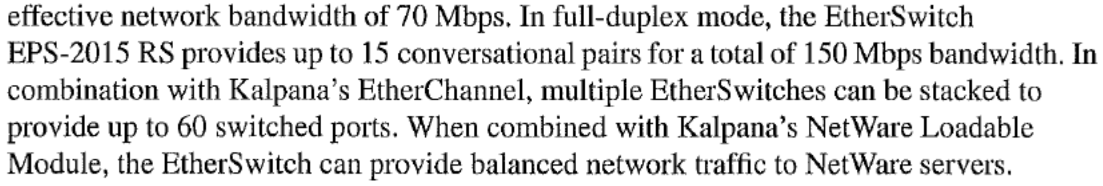
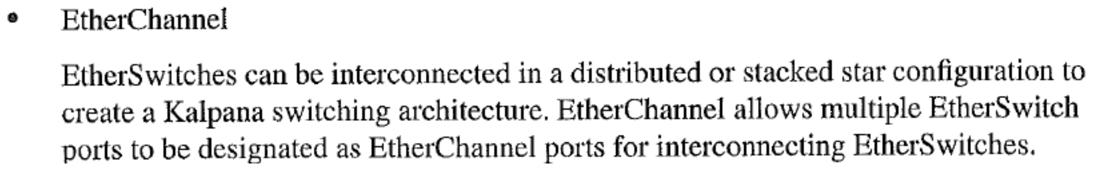
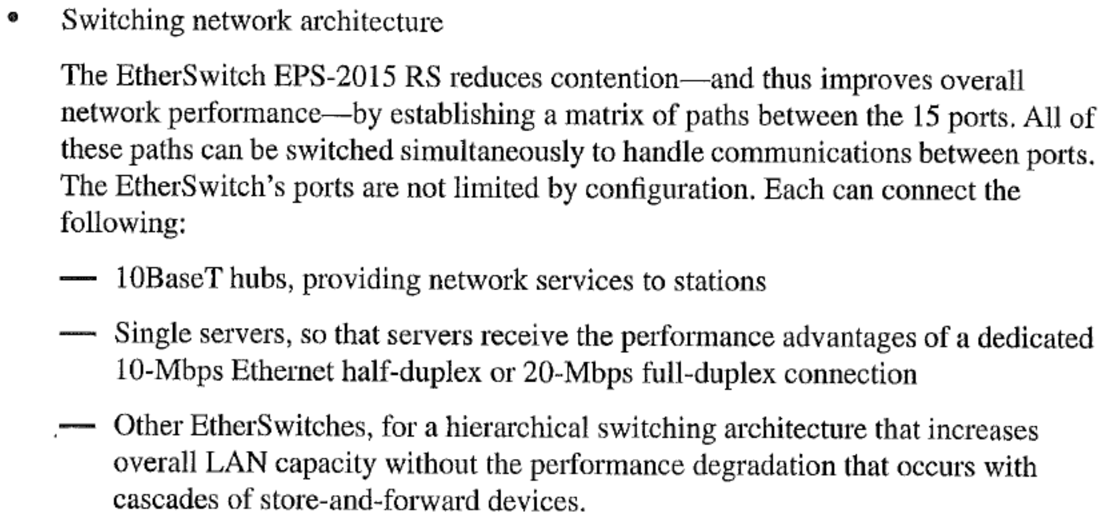
	medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port.	 • Two models available A managed version, the EtherSwitch EPS-500-SNMP, provides Simple Network Management Protocol with Management Information Base (MIB) II support. Kalpana's SNMP implementation allows Kalpana's SwitchVision Network Management System application to collect system and port information from the EtherSwitch EPS-500. The entry-level version of the EtherSwitch EPS-500, which can be upgraded with SNMP, supports out-of-band network management via an EIA/TIA-232 connector. • Easy to install Designed to be a plug-and-go device, the EtherSwitch EPS-500 requires no configuration for installation. The EtherSwitch EPS-500 automatically learns and keeps track of the logical addresses of up to 4,000 network nodes. Address-to-port references are learned at power up and automatically relearned any time the network configuration changes. The EtherSwitch EPS-500 ages addresses so that the system's address table is populated only by the more frequently used addresses. • On-the-fly packet switching With Kalpana's "on-the-fly" switching technology, a packet appears at the output port before it has finished entering the input port. During normal operation, packets moving through the EtherSwitch EPS-500 are delayed no more than 40 microseconds, which is 20 times less than that of bridges and routers using store-and-forward technology. The EtherSwitch EPS-500 does this by reading and processing the six-byte destination address immediately, instead of waiting for the entire packet to arrive first. The shorter delay is particularly important for delay-sensitive applications or direct host connections. • Ethernet throughput enhancement The principal building blocks for the corporate internetwork architecture have been bridges, routers, and structured wiring hubs. Now there is a new network building block—the Ethernet switch. Ethernet switching offers a new design element to complement existing hubs, internetworking devices and high-speed networks. Kalpana's EtherSwitch was designed with a simple purpose in mind—to enhance Ethernet network throughput. Routers are the building blocks of the extended internetwork—designed specifically to interconnect LANs and WANs and provide LAN security. Routing and EtherSwitch are complementary devices intended for specific purposes.

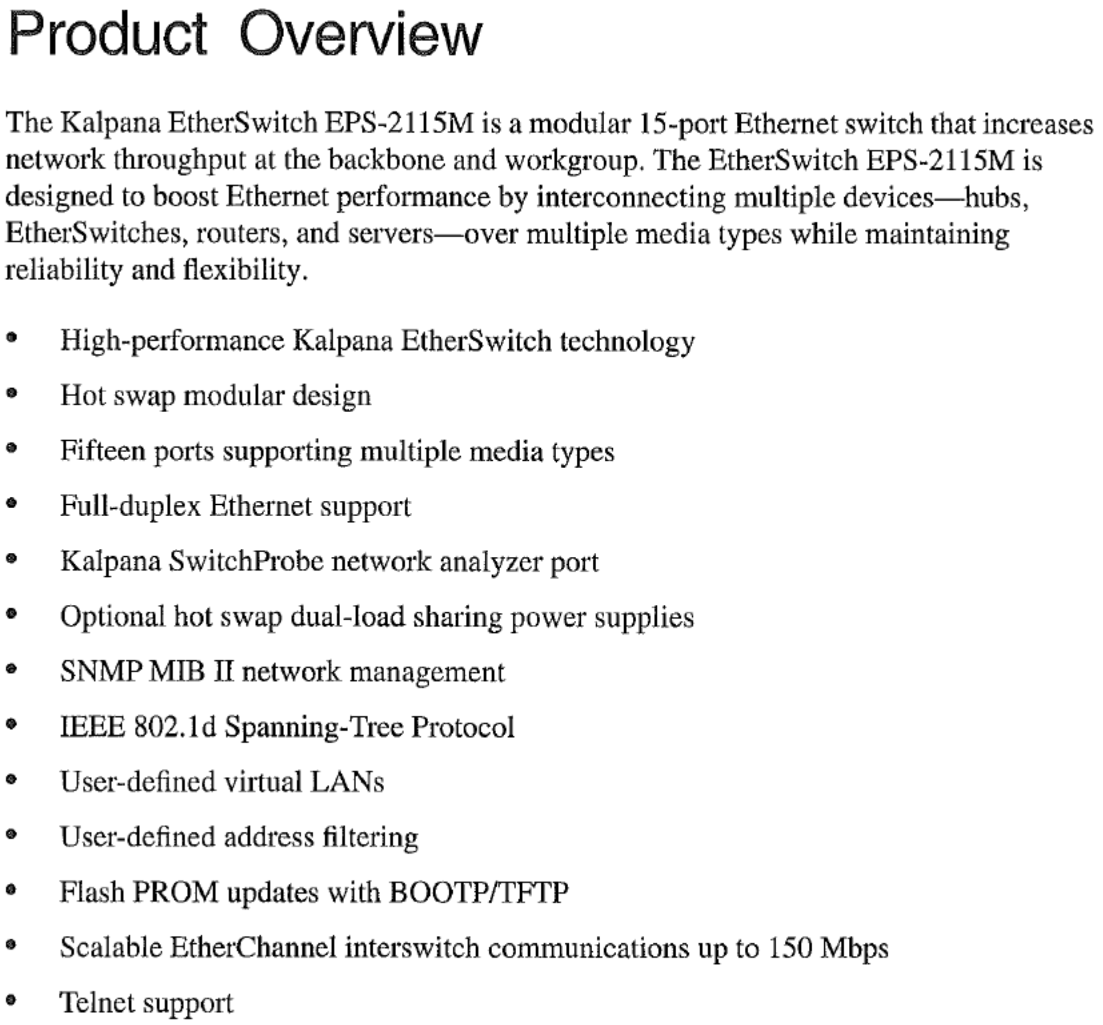
No.	’740 Patent Claim 13	Cisco EtherSwitch System
13[preamble]	A method for communication, comprising:	Cisco EtherSwitch System discloses a method for communication. *See supra at* 1[preamble].
13[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	Cisco EtherSwitch System discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].
13[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Cisco EtherSwitch System discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra at* 1[c].
13[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	Cisco EtherSwitch System discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra at* 1[e].

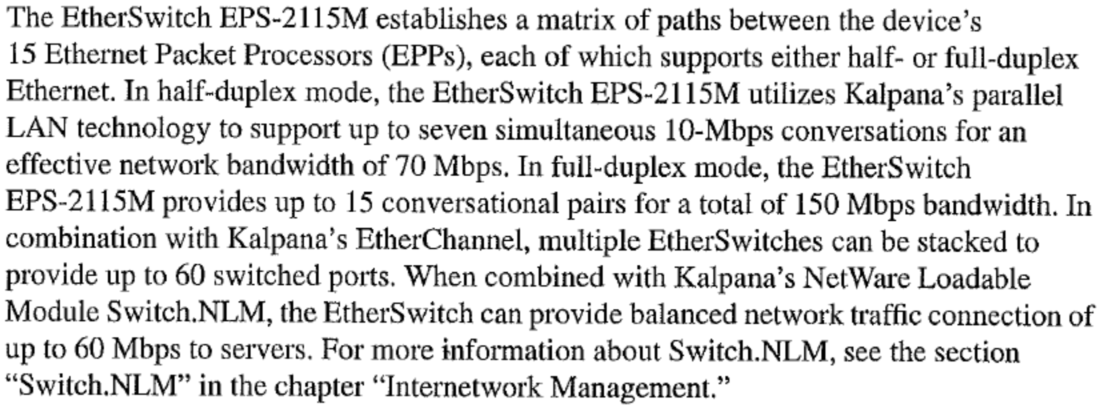
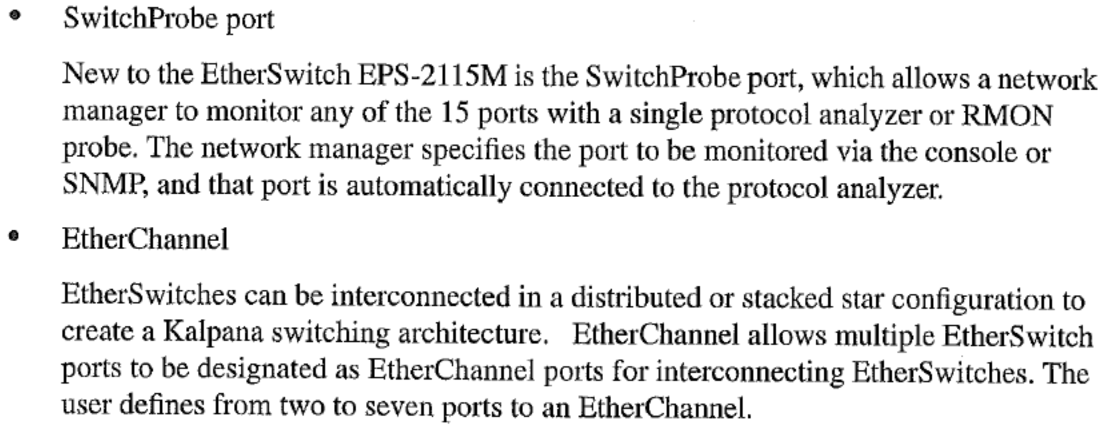
No.	’740 Patent Claim 13	Cisco EtherSwitch System
13[d]	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	Cisco EtherSwitch System discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. *See supra at* 1[f].
13[e]	sending the data frame over the selected first and second physical links,	Cisco EtherSwitch System discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].
13[f]	coupling the network node to the one or more interface modules and	Cisco EtherSwitch System discloses coupling the network node to the one or more interface modules. *See supra at* 1[a].
13[g]	coupling each of the one or more interface modules to the communication network comprising	Cisco EtherSwitch System discloses coupling each of the one or more interface modules to the communication network. *See supra at* 1[c].
13[h]	specifying bandwidth	Cisco EtherSwitch System discloses specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information

No.	’740 Patent Claim 13	Cisco EtherSwitch System
	requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node, and	rate (EIR) of a communication service provided by the communication network to the network node. For example, Cisco EtherSwitch System discloses specific bandwidth considerations that impact the communication service between network devices. A person of ordinary skill in the art would understand that such bandwidth requirements of the ports and connections of the EtherSwitch may be defined by a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR). Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Cisco EtherSwitch System is found to not meet this limitation, coupling each of the one or more interface modules to the communication network comprising specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node would have been obvious to a person having ordinary skill in the art, as explained below. Cisco EtherSwitch System Catalog at 114  • Switching network architecture The EtherSwitch EPS-500 reduces contention—and thus improves overall network performance—by establishing a matrix of paths between the fifteen ports. All of these paths can be switched simultaneously to handle communications between ports. The EtherSwitch’s ports are not limited by configuration. Each can connect the following: — 10BaseT hubs, providing network services to stations — Single servers, so that servers receive the performance advantages of a dedicated 10-Mbps Ethernet half-duplex or 20-Mbps full-duplex connection — Other EtherSwitches for a hierarchical switching architecture that increases overall LAN capacity without the performance degradation that occurs with cascades of store-and-forward devices. Cisco EtherSwitch System Catalog at 115

No.	’740 Patent Claim 13	Cisco EtherSwitch System
		 Product Overview The Kalpana EtherSwitch EPS-2015 RS is a stackable 15-port Ethernet switch that increases the throughput and flexibility of departmental and workgroup 10BaseT networks. Specifically designed to complement the low-cost connectivity of stackable hubs, the EtherSwitch EPS-2015 RS provides network managers with a cost-effective way to boost 10BaseT throughput among hubs and to servers while conserving valuable rack space. • High-performance Kalpana EtherSwitch technology • Fifteen 10BaseT ports with 150-Mbps bandwidth capacity • Full-duplex Ethernet support • SNMP MIB II network management • IEEE 802.1d Spanning-Tree Protocol • Virtual LAN support • User-defined address filtering • Updates using flash PROM • Scalable interswitch communications up to 150 Mbps • Delivers up to 60 switched ports when stacked with other EPS-2015 RS EtherSwitches • Rack-and-stack design, plug’n play, automatic self-learning The EtherSwitch EPS-2015 RS establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2015 RS uses Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an

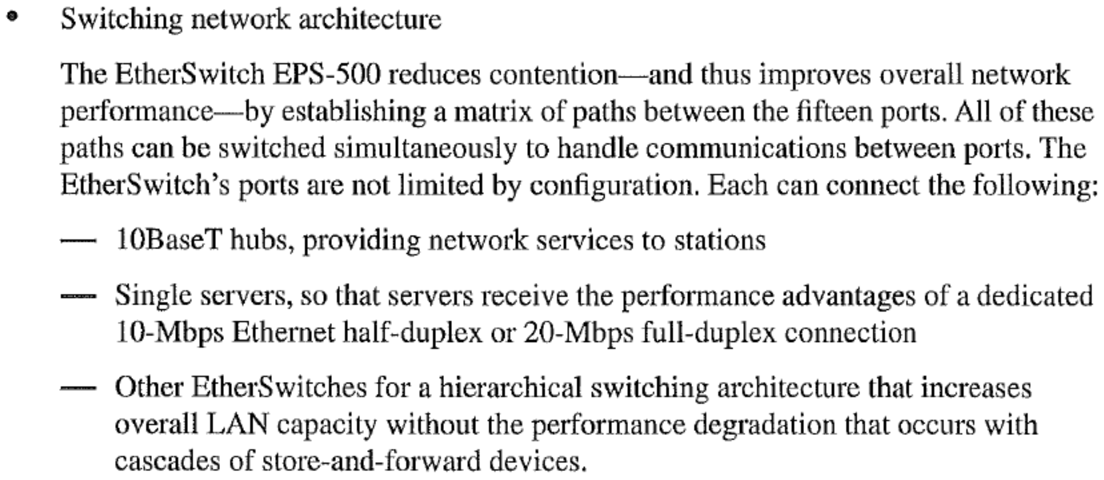
No.	’740 Patent Claim 13	Cisco EtherSwitch System
		 effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2015 RS provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module, the EtherSwitch can provide balanced network traffic to NetWare servers. Cisco EtherSwitch System Catalog at 117  • EtherChannel EtherSwitches can be interconnected in a distributed or stacked star configuration to create a Kalpana switching architecture. EtherChannel allows multiple EtherSwitch ports to be designated as EtherChannel ports for interconnecting EtherSwitches. Cisco EtherSwitch System Catalog at 118  • Switching network architecture The EtherSwitch EPS-2015 RS reduces contention—and thus improves overall network performance—by establishing a matrix of paths between the 15 ports. All of these paths can be switched simultaneously to handle communications between ports. The EtherSwitch’s ports are not limited by configuration. Each can connect the following: — 10BaseT hubs, providing network services to stations — Single servers, so that servers receive the performance advantages of a dedicated 10-Mbps Ethernet half-duplex or 20-Mbps full-duplex connection — Other EtherSwitches, for a hierarchical switching architecture that increases overall LAN capacity without the performance degradation that occurs with cascades of store-and-forward devices.

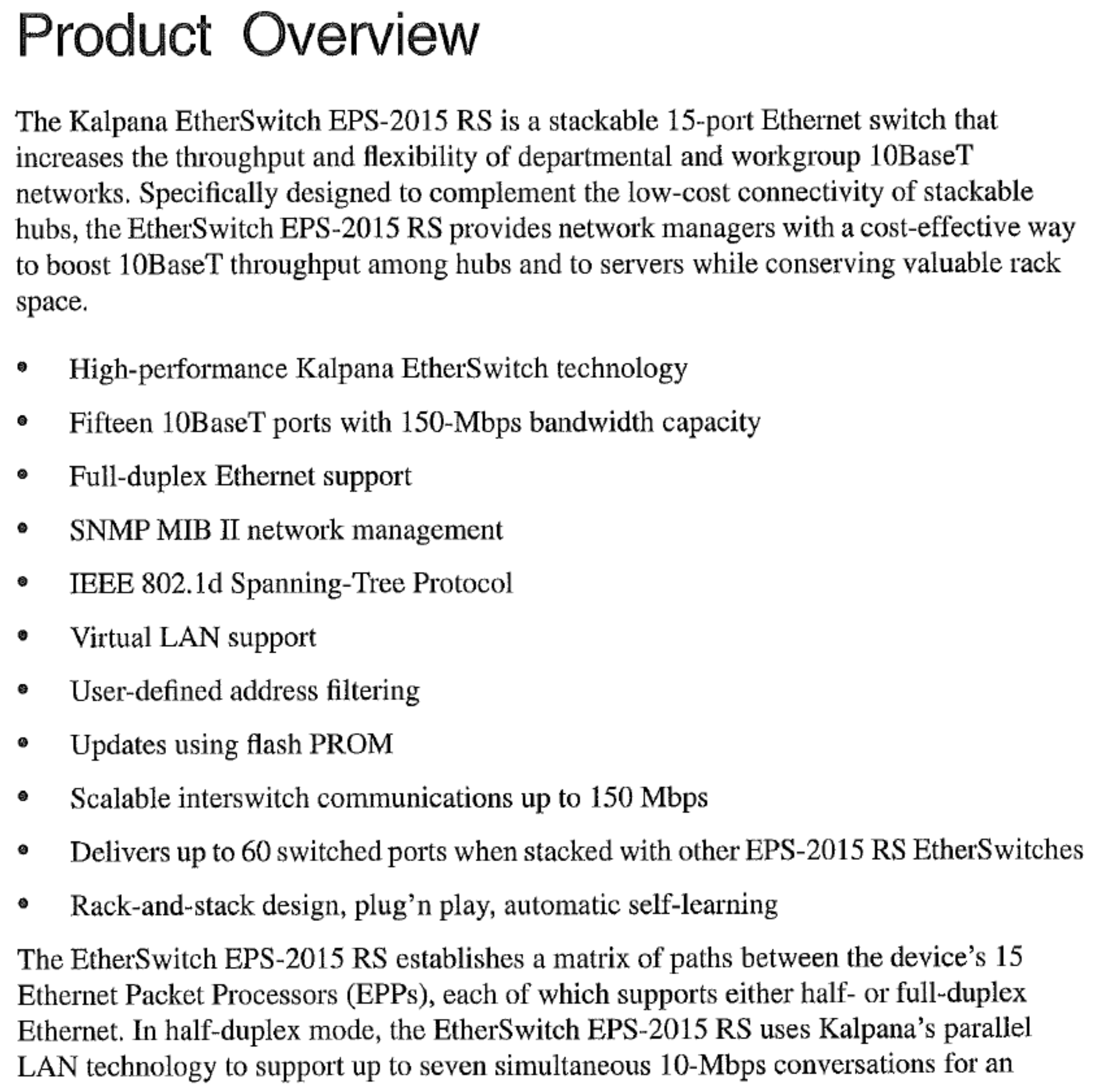
No.	’740 Patent Claim 13	Cisco EtherSwitch System
		Cisco EtherSwitch System Catalog at 119  Product Overview The Kalpana EtherSwitch EPS-2115M is a modular 15-port Ethernet switch that increases network throughput at the backbone and workgroup. The EtherSwitch EPS-2115M is designed to boost Ethernet performance by interconnecting multiple devices—hubs, EtherSwitches, routers, and servers—over multiple media types while maintaining reliability and flexibility. • High-performance Kalpana EtherSwitch technology • Hot swap modular design • Fifteen ports supporting multiple media types • Full-duplex Ethernet support • Kalpana SwitchProbe network analyzer port • Optional hot swap dual-load sharing power supplies • SNMP MIB II network management • IEEE 802.1d Spanning-Tree Protocol • User-defined virtual LANs • User-defined address filtering • Flash PROM updates with BOOTP/TFTP • Scalable EtherChannel interswitch communications up to 150 Mbps • Telnet support

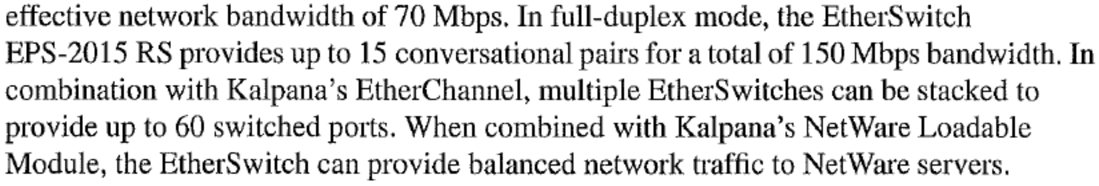
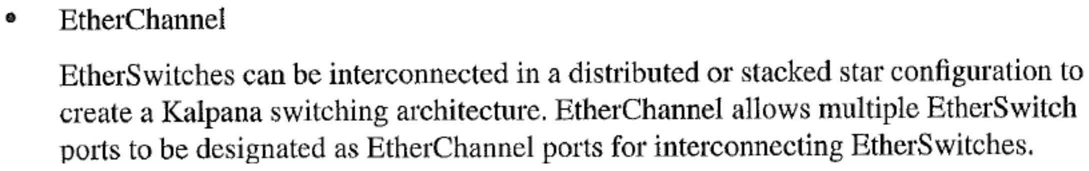
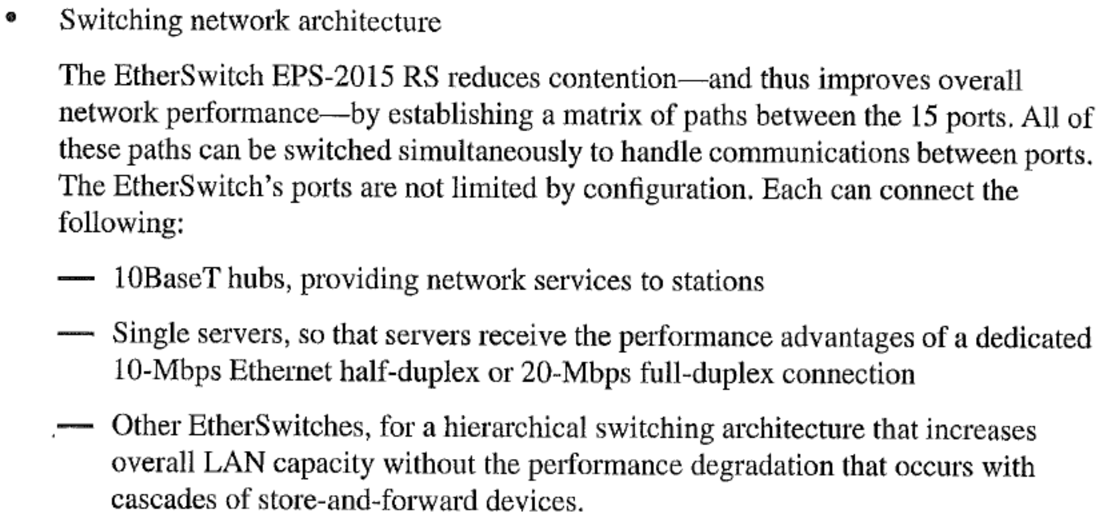
No.	’740 Patent Claim 13	Cisco EtherSwitch System
		Cisco EtherSwitch System Catalog at 120  The EtherSwitch EPS-2115M establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2115M utilizes Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2115M provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module Switch.NLM, the EtherSwitch can provide balanced network traffic connection of up to 60 Mbps to servers. For more information about Switch.NLM, see the section “Switch.NLM” in the chapter “Internetwork Management.” Cisco EtherSwitch System Catalog at 122  • SwitchProbe port New to the EtherSwitch EPS-2115M is the SwitchProbe port, which allows a network manager to monitor any of the 15 ports with a single protocol analyzer or RMON probe. The network manager specifies the port to be monitored via the console or SNMP, and that port is automatically connected to the protocol analyzer. • EtherChannel EtherSwitches can be interconnected in a distributed or stacked star configuration to create a Kalpana switching architecture. EtherChannel allows multiple EtherSwitch ports to be designated as EtherChannel ports for interconnecting EtherSwitches. The user defines from two to seven ports to an EtherChannel. Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Cisco EtherSwitch System in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together)

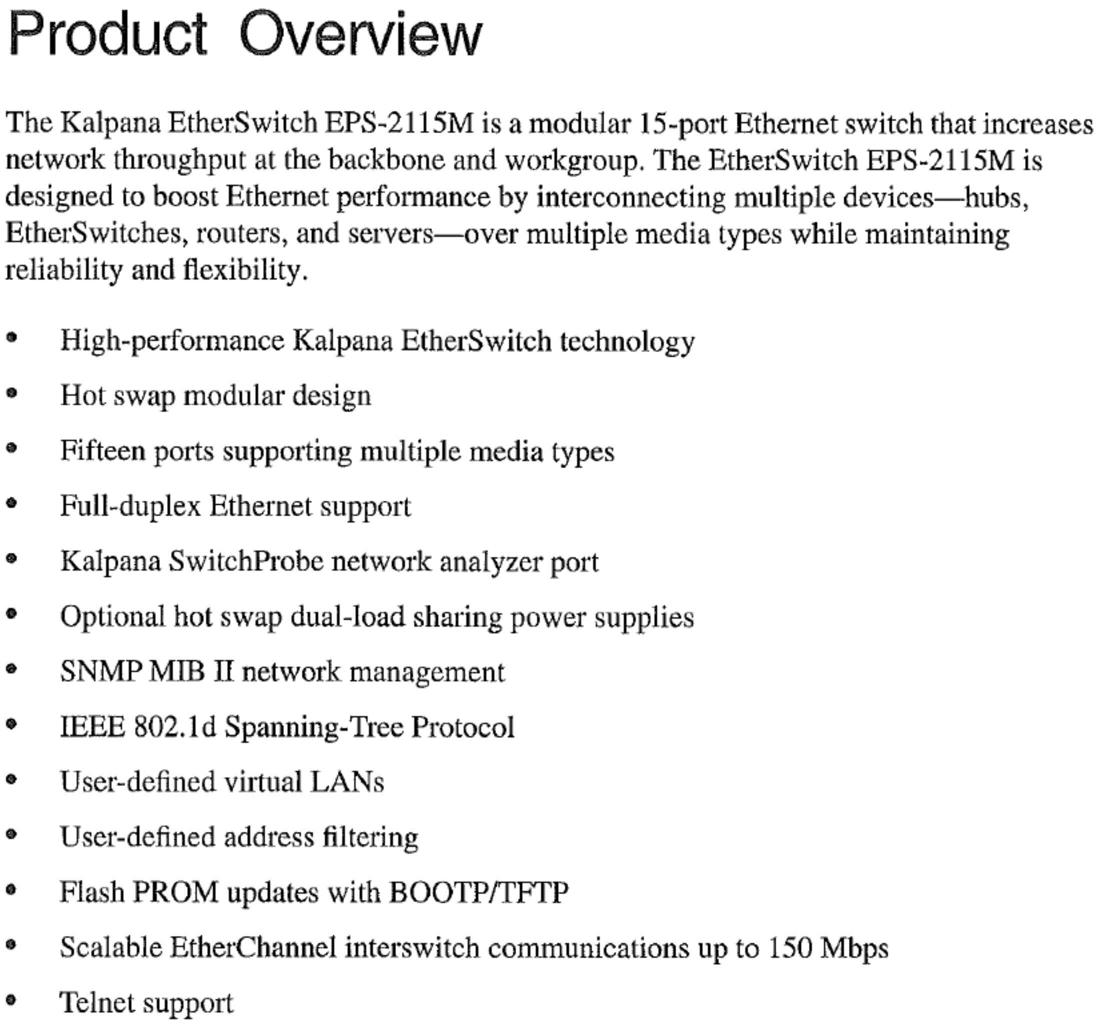
No.	’740 Patent Claim 13	Cisco EtherSwitch System
		of the references identified in element 13[h] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses specifying certain committed, excess, and guaranteed bandwidths, including CIR, EIR, and PIR, respectively. Bruckman at [0013] (“Service level agreements between network service providers and customers commonly specify a certain com-mitted bandwidth, or committed information rate (CIR), which the service provider guarantees to provide to the customer at all times, regardless of bandwidth stress on the network. Additionally or alternatively, the agreement may specify an excess bandwidth, which is available to the customer when network traffic permits. The excess band-width is typically used by customers for lower-priority services, which do not require committed bandwidth. The network service provider may guarantee the customer a certain minimum excess bandwidth, or excess information rate (EIR), in order to avoid starvation of such services in case of bandwidth stress. In general, the bandwidth guaran-teed by a service provider, referred to as the peak informa-tion rate (PIR), may include either CIR, or EIR, or both CIR and EIR (in which case PIR=CIR+EIR). The term "guaran-teed bandwidth," as used in the context of the present patent application and in the claims, includes all these types of guaranteed bandwidth.”) As another example, Solomon discloses a service property of a guaranteed bandwidth, sometimes denoted as CIR-Committed Information Rate and PIR-Peak Information Rate, i.e., specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node. Solomon at [0023] (“In another embodiment, establishing the path includes receiving an indication of a requested service property of the flow, and selecting the port includes assign-ing the port to the flow so as to comply with the requested service property. In a disclosed embodiment, the requested service property includes at least one of a guaranteed bandwidth, a peak bandwidth and a class-of-service. Addi-tionally or alternatively, assigning the port includes

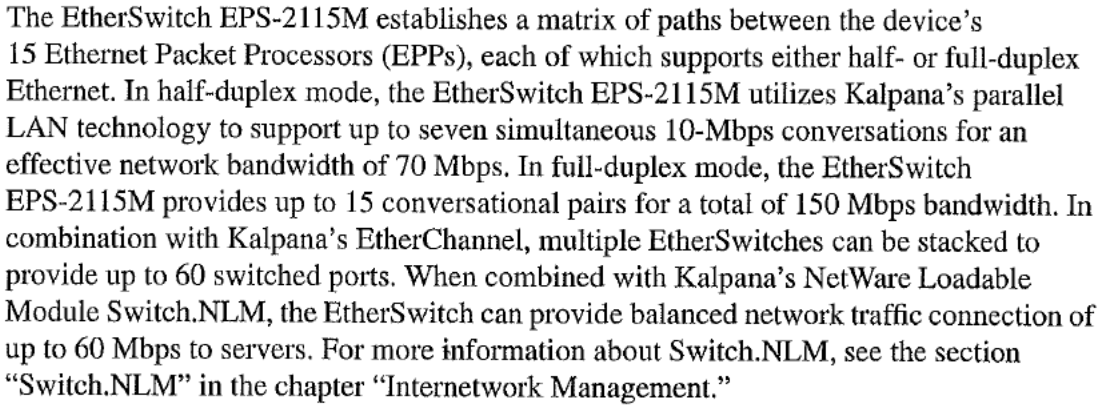
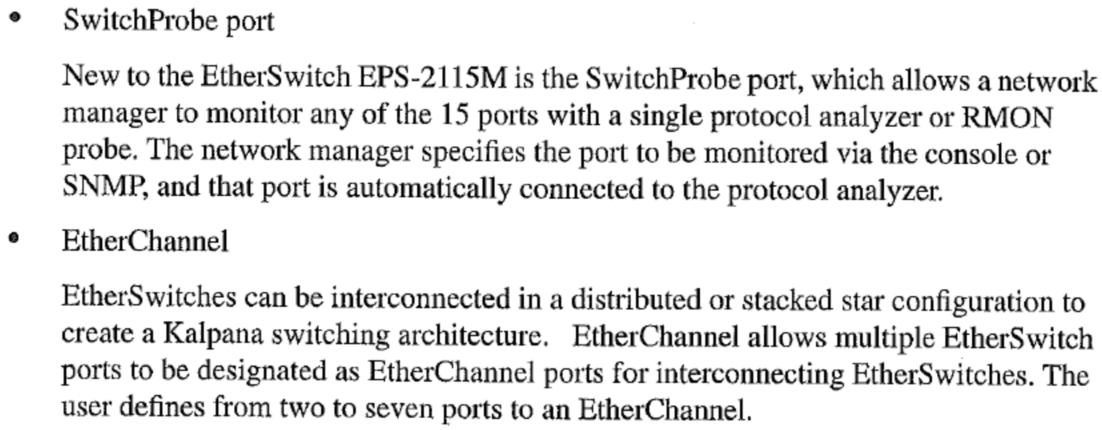
No.	’740 Patent Claim 13	Cisco EtherSwitch System
		selecting the port having a maximum available bandwidth out of the plurality of aggregated ports. Further additionally or alter-natively, assigning the port includes selecting the port hav-ing a minimum available bandwidth out of the plurality of aggregated ports, which is still greater than or equal to the guaranteed bandwidth.”) Solomon at [0050] (“The method of FIG. 3 begins when the preceding node asks to establish a part of tunnel 28 (comprising one or more hops) for sending MPLS packets to MPLS/LAG switch 26 A. The preceding node requests and then receives the MPLS label, which it will subsequently attach to all packets that are sent to MPLS/LAG switch 26 labeledA. The preceding node sends downstream an RSVP-TE PATH mes-sage augmented with a LABEL_REQUEST object, as defined by RSVP-TE, to MPLS/LAG switch A, at a label requesting step 60. The PATH message typically comprises information regarding service properties that are requested for tunnel 28. The service properties may comprise a guar-anteed bandwidth (sometimes denoted CIR-Committed Information Rate) and a peak bandwidth (sometimes denoted PIR-Peak Information Rate), as well as a requested CoS (Class of Service-a measure of packet priority).”)
13[i]	allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements.	Cisco EtherSwitch System discloses allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements. For example, Cisco EtherSwitch System discloses providing balanced network traffic across the ports and connections connecting network devices in an Ethernet network by increasing bandwidth capabilities and distributing packets over ports and connections in response to such bandwidth considerations. Cisco EtherSwitch System Catalog at 114

No.	’740 Patent Claim 13	Cisco EtherSwitch System
		 - Switching network architecture The EtherSwitch EPS-500 reduces contention—and thus improves overall network performance—by establishing a matrix of paths between the fifteen ports. All of these paths can be switched simultaneously to handle communications between ports. The EtherSwitch’s ports are not limited by configuration. Each can connect the following: — 10BaseT hubs, providing network services to stations — Single servers, so that servers receive the performance advantages of a dedicated 10-Mbps Ethernet half-duplex or 20-Mbps full-duplex connection — Other EtherSwitches for a hierarchical switching architecture that increases overall LAN capacity without the performance degradation that occurs with cascades of store-and-forward devices. Cisco EtherSwitch System Catalog at 115

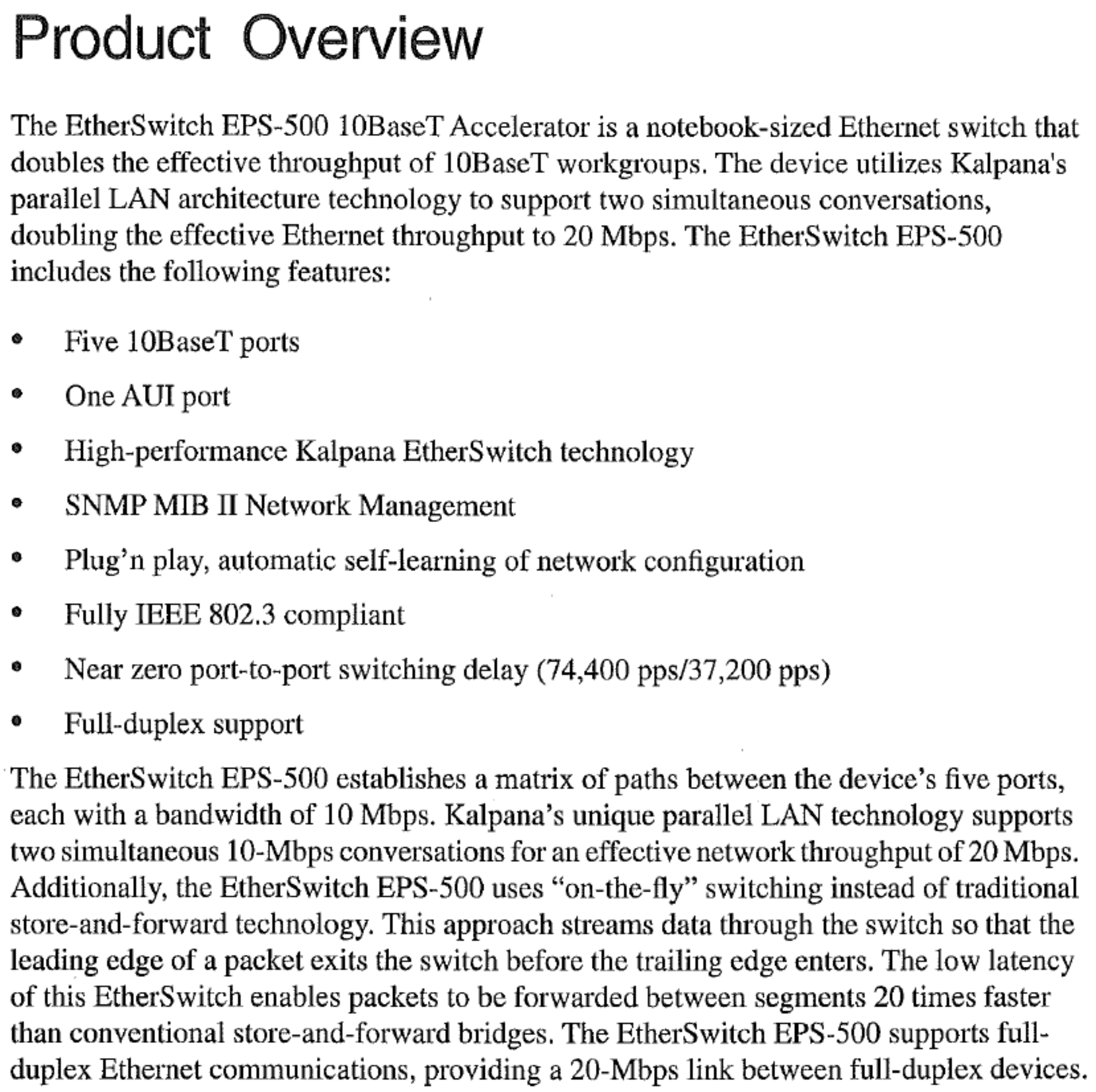
No.	’740 Patent Claim 13	Cisco EtherSwitch System
		 **Product Overview** The Kalpana EtherSwitch EPS-2015 RS is a stackable 15-port Ethernet switch that increases the throughput and flexibility of departmental and workgroup 10BaseT networks. Specifically designed to complement the low-cost connectivity of stackable hubs, the EtherSwitch EPS-2015 RS provides network managers with a cost-effective way to boost 10BaseT throughput among hubs and to servers while conserving valuable rack space. • High-performance Kalpana EtherSwitch technology • Fifteen 10BaseT ports with 150-Mbps bandwidth capacity • Full-duplex Ethernet support • SNMP MIB II network management • IEEE 802.1d Spanning-Tree Protocol • Virtual LAN support • User-defined address filtering • Updates using flash PROM • Scalable interswitch communications up to 150 Mbps • Delivers up to 60 switched ports when stacked with other EPS-2015 RS EtherSwitches • Rack-and-stack design, plug’n play, automatic self-learning The EtherSwitch EPS-2015 RS establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2015 RS uses Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an

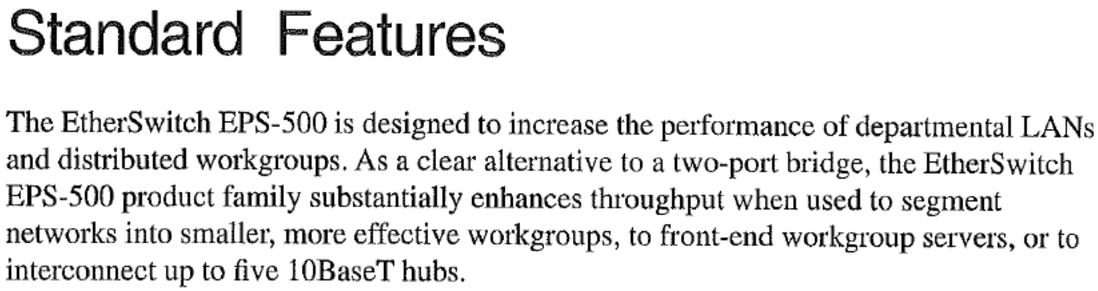
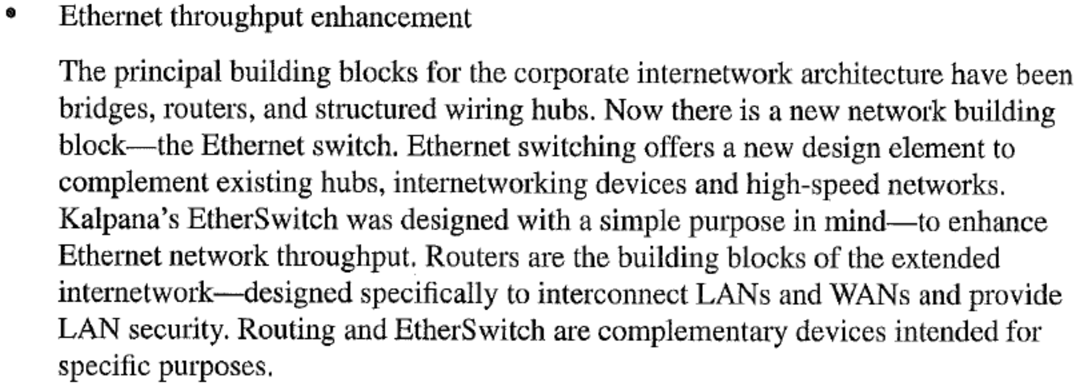
No.	’740 Patent Claim 13	Cisco EtherSwitch System
		 effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2015 RS provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module, the EtherSwitch can provide balanced network traffic to NetWare servers. Cisco EtherSwitch System Catalog at 117  • EtherChannel EtherSwitches can be interconnected in a distributed or stacked star configuration to create a Kalpana switching architecture. EtherChannel allows multiple EtherSwitch ports to be designated as EtherChannel ports for interconnecting EtherSwitches. Cisco EtherSwitch System Catalog at 118  • Switching network architecture The EtherSwitch EPS-2015 RS reduces contention—and thus improves overall network performance—by establishing a matrix of paths between the 15 ports. All of these paths can be switched simultaneously to handle communications between ports. The EtherSwitch’s ports are not limited by configuration. Each can connect the following: — 10BaseT hubs, providing network services to stations — Single servers, so that servers receive the performance advantages of a dedicated 10-Mbps Ethernet half-duplex or 20-Mbps full-duplex connection — Other EtherSwitches, for a hierarchical switching architecture that increases overall LAN capacity without the performance degradation that occurs with cascades of store-and-forward devices.

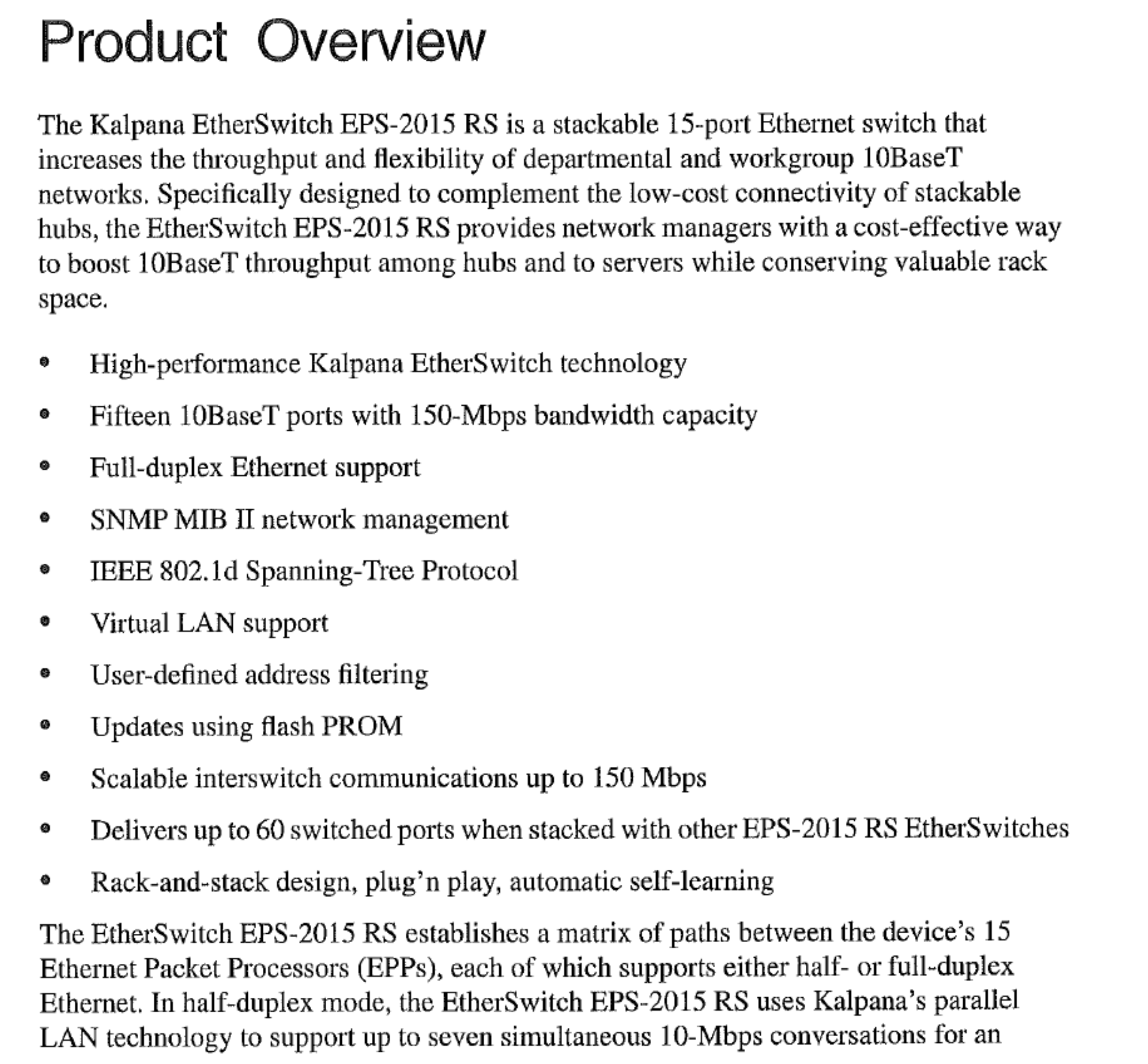
No.	’740 Patent Claim 13	Cisco EtherSwitch System
		Cisco EtherSwitch System Catalog at 119  Product Overview The Kalpana EtherSwitch EPS-2115M is a modular 15-port Ethernet switch that increases network throughput at the backbone and workgroup. The EtherSwitch EPS-2115M is designed to boost Ethernet performance by interconnecting multiple devices—hubs, EtherSwitches, routers, and servers—over multiple media types while maintaining reliability and flexibility. • High-performance Kalpana EtherSwitch technology • Hot swap modular design • Fifteen ports supporting multiple media types • Full-duplex Ethernet support • Kalpana SwitchProbe network analyzer port • Optional hot swap dual-load sharing power supplies • SNMP MIB II network management • IEEE 802.1d Spanning-Tree Protocol • User-defined virtual LANs • User-defined address filtering • Flash PROM updates with BOOTP/TFTP • Scalable EtherChannel interswitch communications up to 150 Mbps • Telnet support

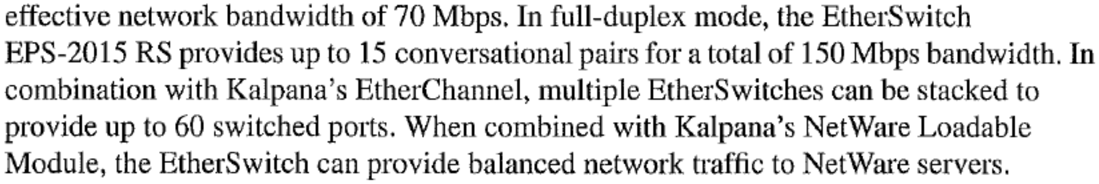
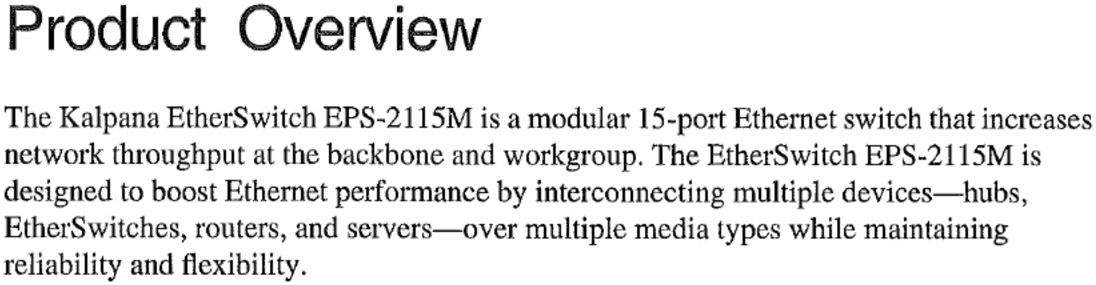
No.	’740 Patent Claim 13	Cisco EtherSwitch System
		Cisco EtherSwitch System Catalog at 120  The EtherSwitch EPS-2115M establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2115M utilizes Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2115M provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module Switch.NLM, the EtherSwitch can provide balanced network traffic connection of up to 60 Mbps to servers. For more information about Switch.NLM, see the section “Switch.NLM” in the chapter “Internetwork Management.” Cisco EtherSwitch System Catalog at 122  • SwitchProbe port New to the EtherSwitch EPS-2115M is the SwitchProbe port, which allows a network manager to monitor any of the 15 ports with a single protocol analyzer or RMON probe. The network manager specifies the port to be monitored via the console or SNMP, and that port is automatically connected to the protocol analyzer. • EtherChannel EtherSwitches can be interconnected in a distributed or stacked star configuration to create a Kalpana switching architecture. EtherChannel allows multiple EtherSwitch ports to be designated as EtherChannel ports for interconnecting EtherSwitches. The user defines from two to seven ports to an EtherChannel.

No.	’740 Patent Claim 14	Cisco EtherSwitch System
14[preamble]	A method for connecting user ports to a communication network, comprising:	Cisco EtherSwitch System discloses a method for connecting user ports to a communication network. For example, Cisco EtherSwitch System discloses an Ethernet switch that connects ports of interconnecting network devices to an Ethernet network for increased throughput of data traffic in networks. Cisco EtherSwitch System Catalog at 111

No.	’740 Patent Claim 14	Cisco EtherSwitch System
		 Product Overview The EtherSwitch EPS-500 10BaseT Accelerator is a notebook-sized Ethernet switch that doubles the effective throughput of 10BaseT workgroups. The device utilizes Kalpana's parallel LAN architecture technology to support two simultaneous conversations, doubling the effective Ethernet throughput to 20 Mbps. The EtherSwitch EPS-500 includes the following features: • Five 10BaseT ports • One AUI port • High-performance Kalpana EtherSwitch technology • SNMP MIB II Network Management • Plug’n play, automatic self-learning of network configuration • Fully IEEE 802.3 compliant • Near zero port-to-port switching delay (74,400 pps/37,200 pps) • Full-duplex support The EtherSwitch EPS-500 establishes a matrix of paths between the device’s five ports, each with a bandwidth of 10 Mbps. Kalpana’s unique parallel LAN technology supports two simultaneous 10-Mbps conversations for an effective network throughput of 20 Mbps. Additionally, the EtherSwitch EPS-500 uses “on-the-fly” switching instead of traditional store-and-forward technology. This approach streams data through the switch so that the leading edge of a packet exits the switch before the trailing edge enters. The low latency of this EtherSwitch enables packets to be forwarded between segments 20 times faster than conventional store-and-forward bridges. The EtherSwitch EPS-500 supports full-duplex Ethernet communications, providing a 20-Mbps link between full-duplex devices.

No.	’740 Patent Claim 14	Cisco EtherSwitch System
		Cisco EtherSwitch System Catalog at 113  Standard Features The EtherSwitch EPS-500 is designed to increase the performance of departmental LANs and distributed workgroups. As a clear alternative to a two-port bridge, the EtherSwitch EPS-500 product family substantially enhances throughput when used to segment networks into smaller, more effective workgroups, to front-end workgroup servers, or to interconnect up to five 10BaseT hubs. Cisco EtherSwitch System Catalog at 113  • Ethernet throughput enhancement The principal building blocks for the corporate internetwork architecture have been bridges, routers, and structured wiring hubs. Now there is a new network building block—the Ethernet switch. Ethernet switching offers a new design element to complement existing hubs, internetworking devices and high-speed networks. Kalpana’s EtherSwitch was designed with a simple purpose in mind—to enhance Ethernet network throughput. Routers are the building blocks of the extended internetwork—designed specifically to interconnect LANs and WANs and provide LAN security. Routing and EtherSwitch are complementary devices intended for specific purposes. Cisco EtherSwitch System Catalog at 115

No.	’740 Patent Claim 14	Cisco EtherSwitch System
		 Product Overview The Kalpana EtherSwitch EPS-2015 RS is a stackable 15-port Ethernet switch that increases the throughput and flexibility of departmental and workgroup 10BaseT networks. Specifically designed to complement the low-cost connectivity of stackable hubs, the EtherSwitch EPS-2015 RS provides network managers with a cost-effective way to boost 10BaseT throughput among hubs and to servers while conserving valuable rack space. • High-performance Kalpana EtherSwitch technology • Fifteen 10BaseT ports with 150-Mbps bandwidth capacity • Full-duplex Ethernet support • SNMP MIB II network management • IEEE 802.1d Spanning-Tree Protocol • Virtual LAN support • User-defined address filtering • Updates using flash PROM • Scalable interswitch communications up to 150 Mbps • Delivers up to 60 switched ports when stacked with other EPS-2015 RS EtherSwitches • Rack-and-stack design, plug’n play, automatic self-learning The EtherSwitch EPS-2015 RS establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2015 RS uses Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an

No.	’740 Patent Claim 14	Cisco EtherSwitch System
		 effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2015 RS provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module, the EtherSwitch can provide balanced network traffic to NetWare servers. Cisco EtherSwitch System Catalog at 119  Product Overview The Kalpana EtherSwitch EPS-2115M is a modular 15-port Ethernet switch that increases network throughput at the backbone and workgroup. The EtherSwitch EPS-2115M is designed to boost Ethernet performance by interconnecting multiple devices—hubs, EtherSwitches, routers, and servers—over multiple media types while maintaining reliability and flexibility.
14[a]	coupling the user ports to one or more user interface modules;	Cisco EtherSwitch System discloses coupling the user ports to one or more user interface modules. *See supra at* 1[a].
14[b]	coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel,	Cisco EtherSwitch System discloses coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel. *See supra at* 1[c], 3.
14[c]	at least one of said backplane traces being bi-directional	Cisco EtherSwitch System discloses at least one of said backplane traces being bi-directional and operative to communicate in both an upstream direction and a downstream direction.

No.	’740 Patent Claim 14	Cisco EtherSwitch System
	and operative to communicate in both an upstream direction and a downstream direction;	*See supra at* 14[b], 1[d].
14[d]	receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes;	Cisco EtherSwitch System discloses receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes. *See supra at* 14[a], 1[e].
14[e]	for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces; and	Cisco EtherSwitch System discloses for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces. *See supra at* 14[b], 1[f].
14[f]	sending the data frame over the selected backplane trace;	Cisco EtherSwitch System discloses sending the data frame over the selected backplane trace. *See supra at* 14[e], 1[g].
14[g]	said sending comprising communicating along said at least one of said backplane traces.	Cisco EtherSwitch System discloses said sending comprising communicating along said at least one of said backplane traces. *See supra at* 14[f], 1[h].

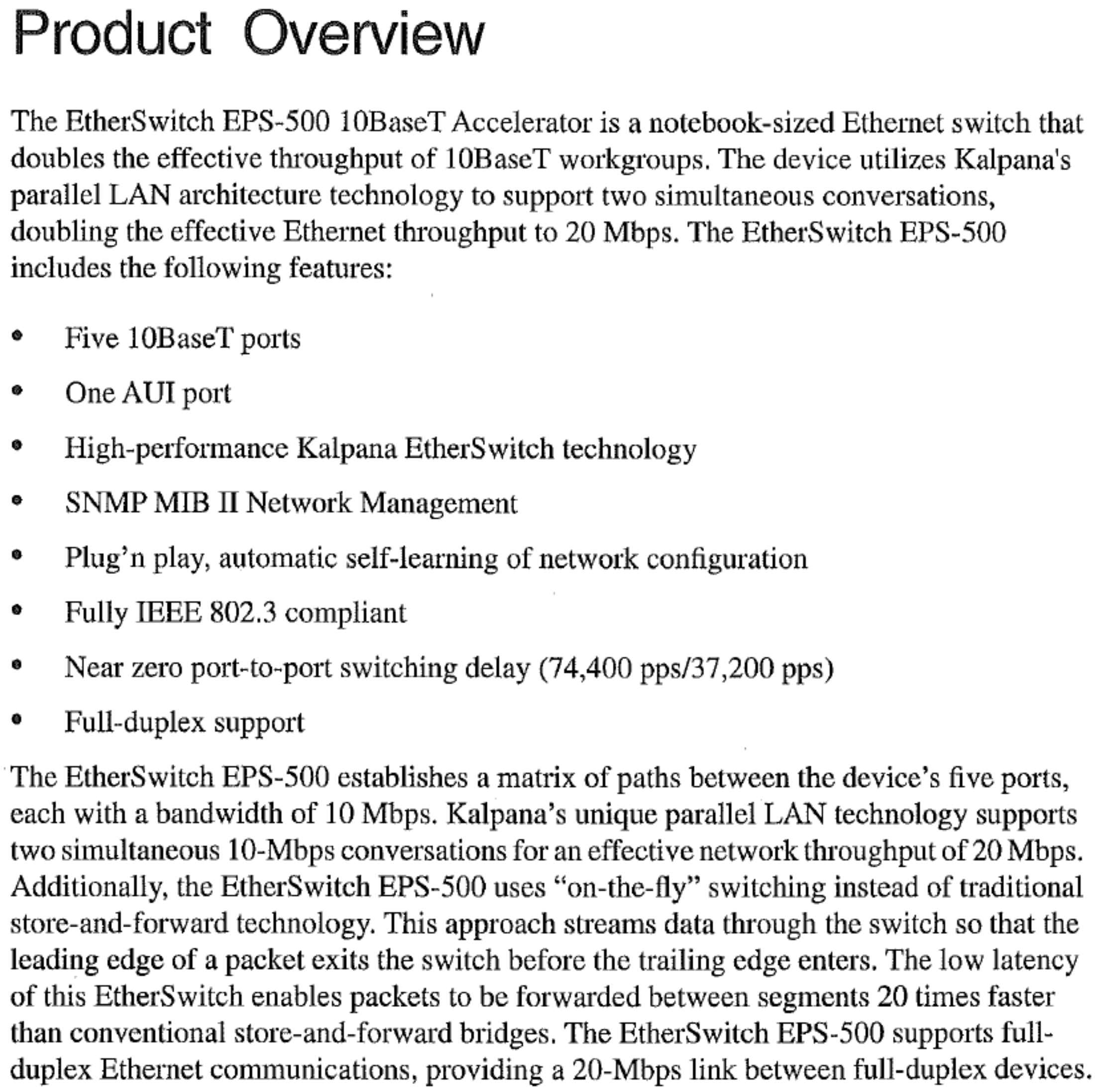
No.	’740 Patent Claim 15	Cisco EtherSwitch System
15[preamble]	A method for connecting user ports to a communication network, comprising:	Cisco EtherSwitch System discloses a method for connecting user ports to a communication network. *See supra at* 14[preamble].
15[a]	coupling the user ports to one or more user interface modules;	Cisco EtherSwitch System discloses coupling the user ports to one or more user interface modules. *See supra at* 14[a].
15[b]	coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel;	Cisco EtherSwitch System discloses coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel. *See supra at* 14[b].
15[c]	receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes;	Cisco EtherSwitch System discloses receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes. *See supra at* 14[d].
15[d]	for each data frame, selecting responsively to at least one of the respective frame attributes a backplane	Cisco EtherSwitch System discloses for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces. *See supra at* 14[e].

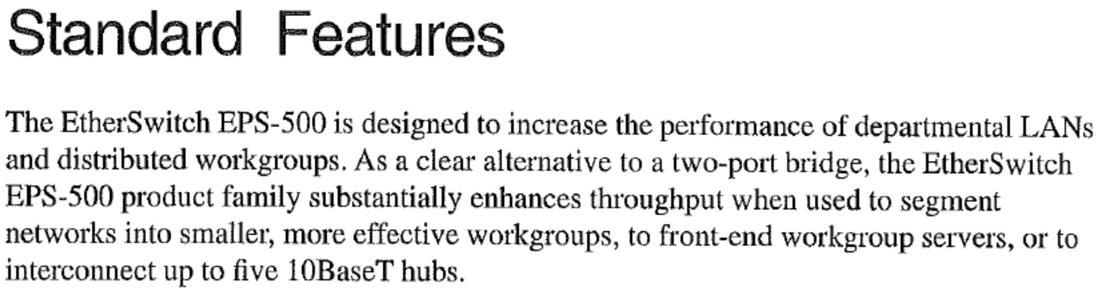
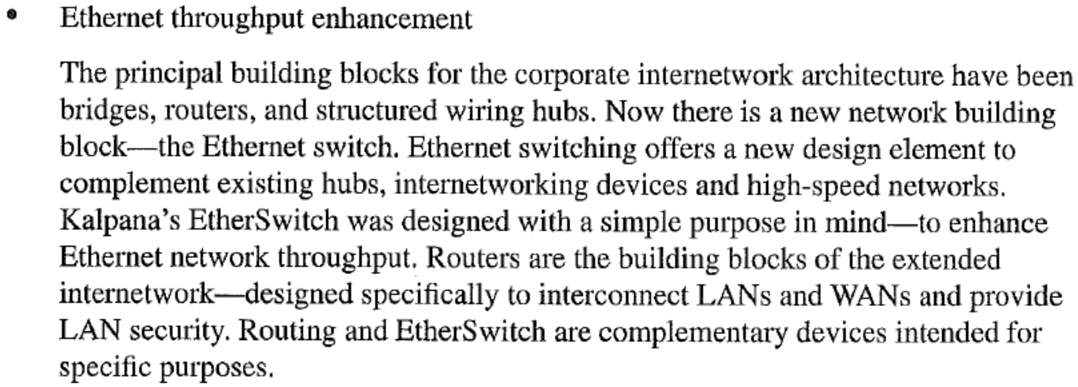
No.	’740 Patent Claim 15	Cisco EtherSwitch System
	trace from the two or more backplane traces; and	
15[e]	sending the data frame over the selected backplane trace,	Cisco EtherSwitch System discloses sending the data frame over the selected backplane trace. *See supra at* 14[f].
15[f]	at least some of the backplane traces being aggregated into an Ethernet link aggregation (LAG) group.	Cisco EtherSwitch System discloses at least some of the backplane traces being aggregated into an Ethernet link aggregation (LAG) group. *See supra at* 15[e], 4[f], 3.

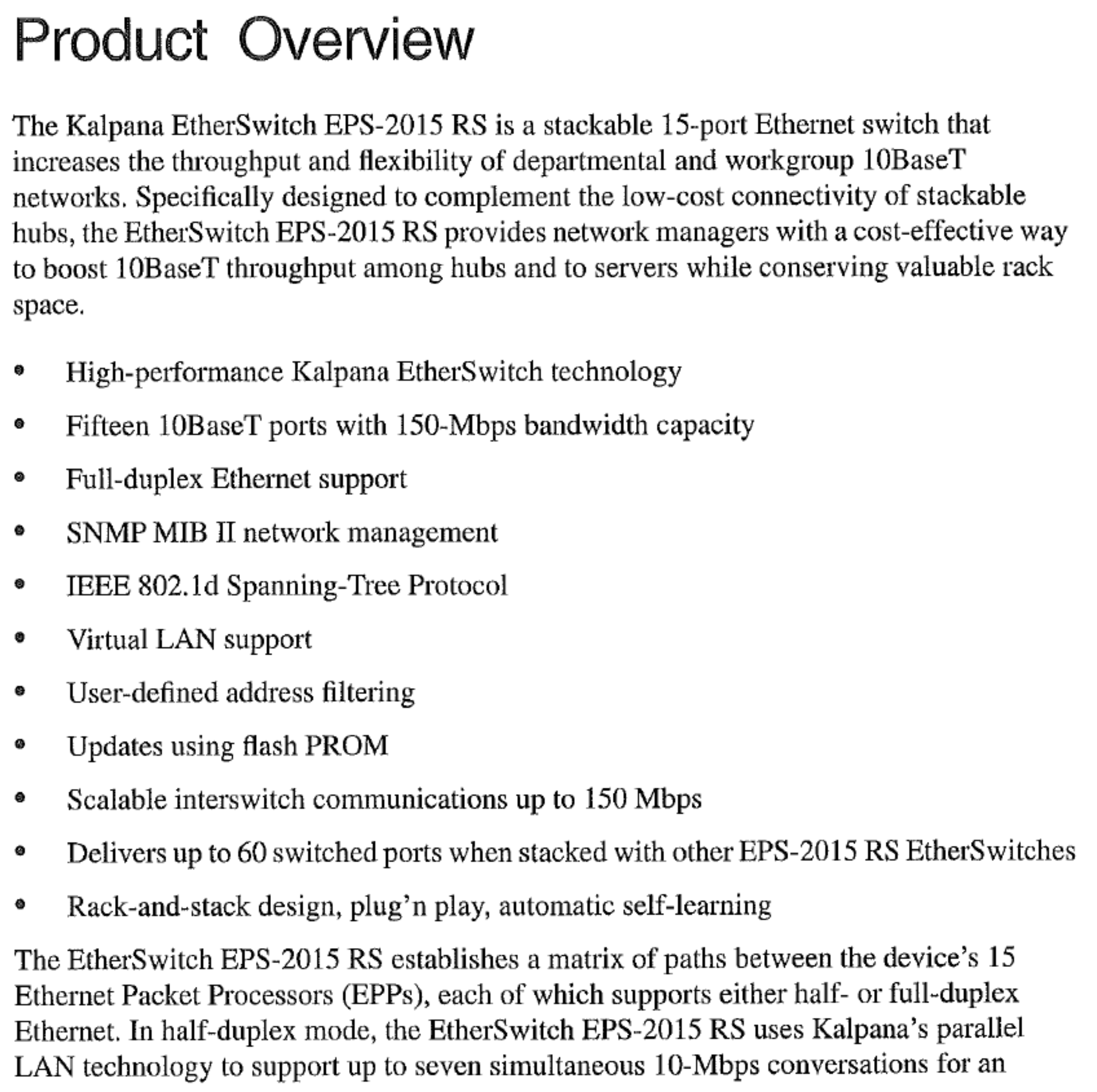
No.	’740 Patent Claim 16	Cisco EtherSwitch System
16	The method according to claim 14, wherein selecting the backplane trace comprises applying a hashing function to the at least one of the frame attributes.	Cisco EtherSwitch System discloses the method according to claim 14, wherein selecting the backplane trace comprises applying a hashing function to the at least one of the frame attributes. *See supra* at 14, 9, 8.

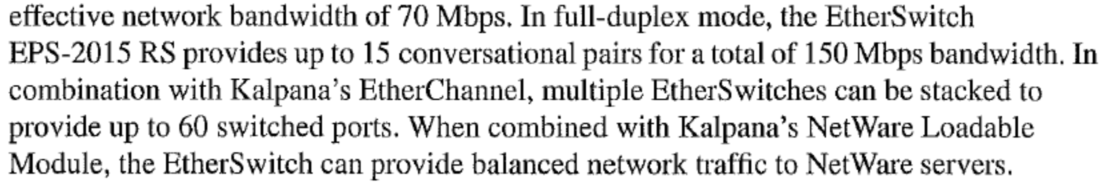
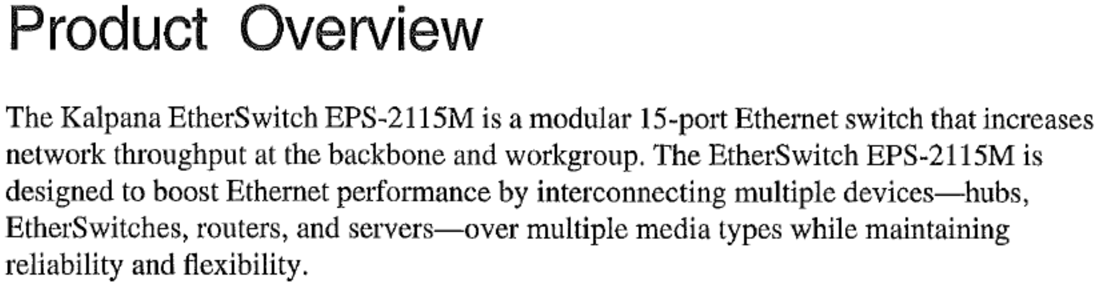
No.	’740 Patent Claim 17	Cisco EtherSwitch System
17[preamble]	Apparatus for connecting a network node with a	Cisco EtherSwitch System discloses apparatus for connecting a network node with a communication network.

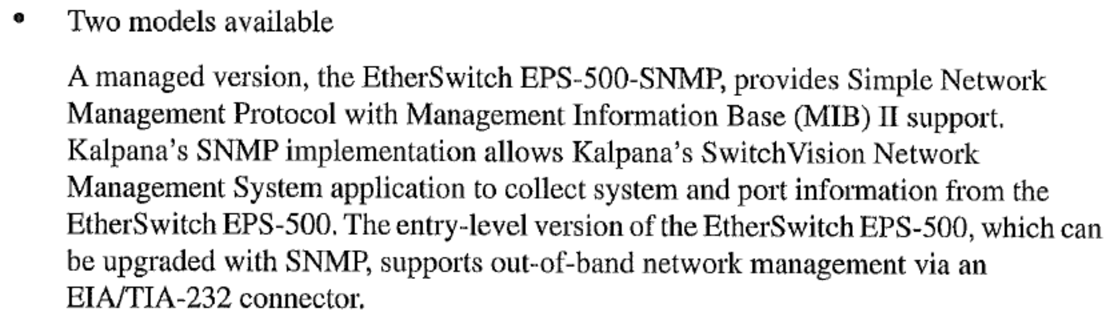
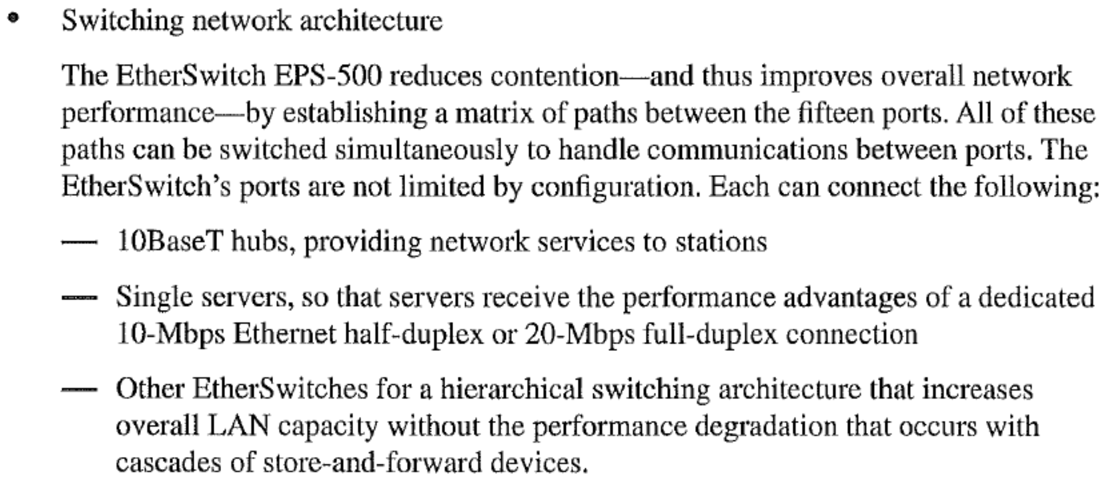
No.	’740 Patent Claim 17	Cisco EtherSwitch System
	communication network, comprising:	For example, Cisco EtherSwitch System discloses an Ethernet switch that connects interconnecting network devices to an Ethernet network for increased throughput of data traffic in networks. Cisco EtherSwitch System Catalog at 111

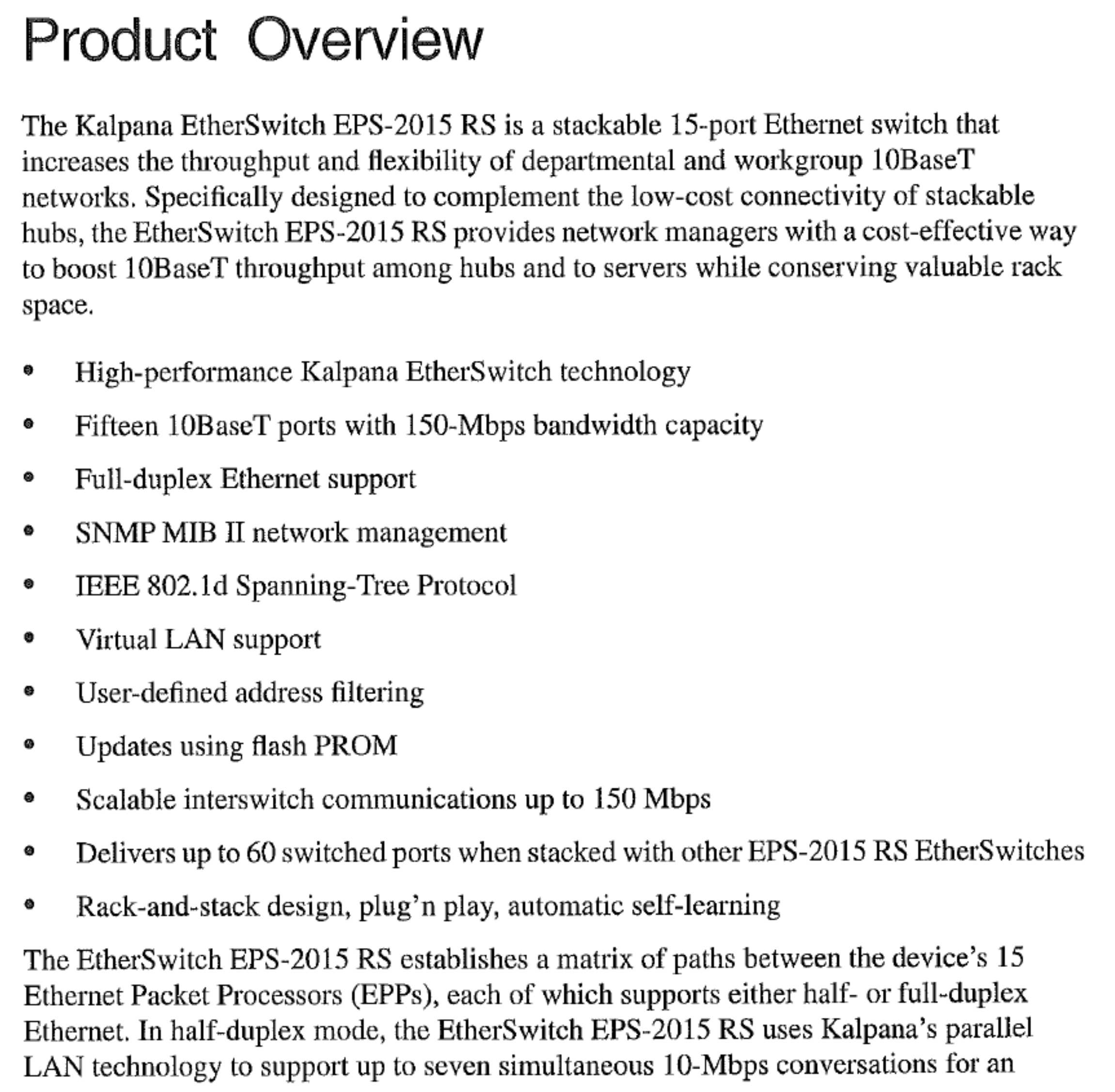
No.	’740 Patent Claim 17	Cisco EtherSwitch System
		 Product Overview The EtherSwitch EPS-500 10BaseT Accelerator is a notebook-sized Ethernet switch that doubles the effective throughput of 10BaseT workgroups. The device utilizes Kalpana's parallel LAN architecture technology to support two simultaneous conversations, doubling the effective Ethernet throughput to 20 Mbps. The EtherSwitch EPS-500 includes the following features: • Five 10BaseT ports • One AUI port • High-performance Kalpana EtherSwitch technology • SNMP MIB II Network Management • Plug’n play, automatic self-learning of network configuration • Fully IEEE 802.3 compliant • Near zero port-to-port switching delay (74,400 pps/37,200 pps) • Full-duplex support The EtherSwitch EPS-500 establishes a matrix of paths between the device’s five ports, each with a bandwidth of 10 Mbps. Kalpana’s unique parallel LAN technology supports two simultaneous 10-Mbps conversations for an effective network throughput of 20 Mbps. Additionally, the EtherSwitch EPS-500 uses “on-the-fly” switching instead of traditional store-and-forward technology. This approach streams data through the switch so that the leading edge of a packet exits the switch before the trailing edge enters. The low latency of this EtherSwitch enables packets to be forwarded between segments 20 times faster than conventional store-and-forward bridges. The EtherSwitch EPS-500 supports full-duplex Ethernet communications, providing a 20-Mbps link between full-duplex devices.

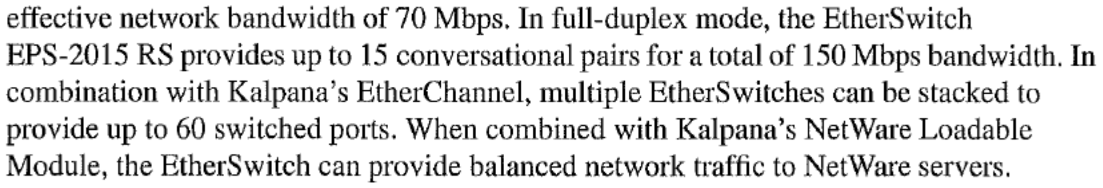
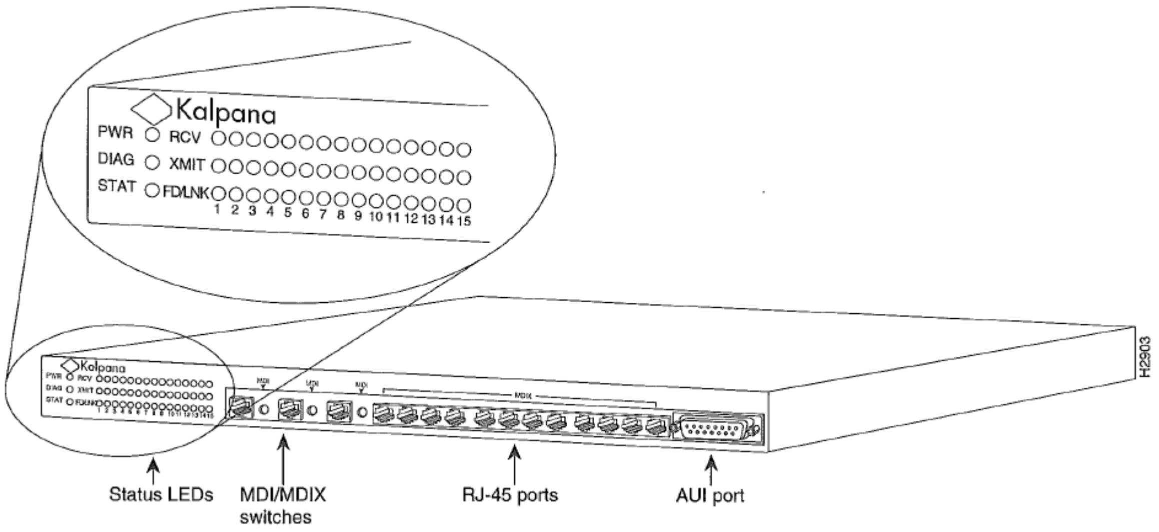
No.	’740 Patent Claim 17	Cisco EtherSwitch System
		Cisco EtherSwitch System Catalog at 113  **Standard Features** The EtherSwitch EPS-500 is designed to increase the performance of departmental LANs and distributed workgroups. As a clear alternative to a two-port bridge, the EtherSwitch EPS-500 product family substantially enhances throughput when used to segment networks into smaller, more effective workgroups, to front-end workgroup servers, or to interconnect up to five 10BaseT hubs. Cisco EtherSwitch System Catalog at 113  - Ethernet throughput enhancement The principal building blocks for the corporate internetwork architecture have been bridges, routers, and structured wiring hubs. Now there is a new network building block—the Ethernet switch. Ethernet switching offers a new design element to complement existing hubs, internetworking devices and high-speed networks. Kalpana’s EtherSwitch was designed with a simple purpose in mind—to enhance Ethernet network throughput. Routers are the building blocks of the extended internetwork—designed specifically to interconnect LANs and WANs and provide LAN security. Routing and EtherSwitch are complementary devices intended for specific purposes. Cisco EtherSwitch System Catalog at 115

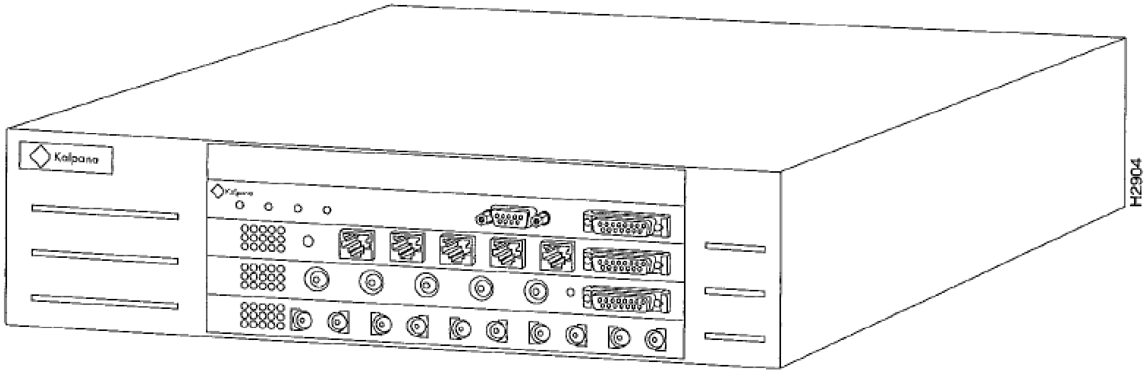
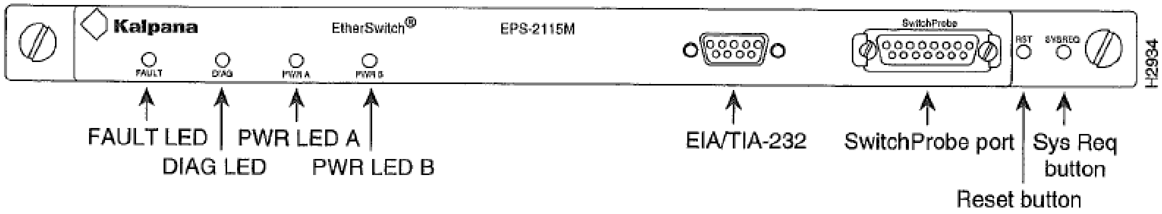
No.	’740 Patent Claim 17	Cisco EtherSwitch System
		 Product Overview The Kalpana EtherSwitch EPS-2015 RS is a stackable 15-port Ethernet switch that increases the throughput and flexibility of departmental and workgroup 10BaseT networks. Specifically designed to complement the low-cost connectivity of stackable hubs, the EtherSwitch EPS-2015 RS provides network managers with a cost-effective way to boost 10BaseT throughput among hubs and to servers while conserving valuable rack space. • High-performance Kalpana EtherSwitch technology • Fifteen 10BaseT ports with 150-Mbps bandwidth capacity • Full-duplex Ethernet support • SNMP MIB II network management • IEEE 802.1d Spanning-Tree Protocol • Virtual LAN support • User-defined address filtering • Updates using flash PROM • Scalable interswitch communications up to 150 Mbps • Delivers up to 60 switched ports when stacked with other EPS-2015 RS EtherSwitches • Rack-and-stack design, plug’n play, automatic self-learning The EtherSwitch EPS-2015 RS establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2015 RS uses Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an

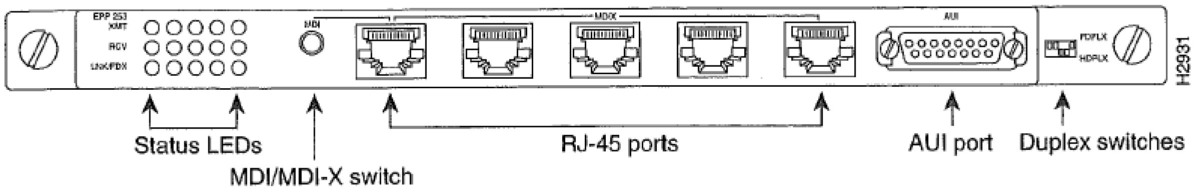
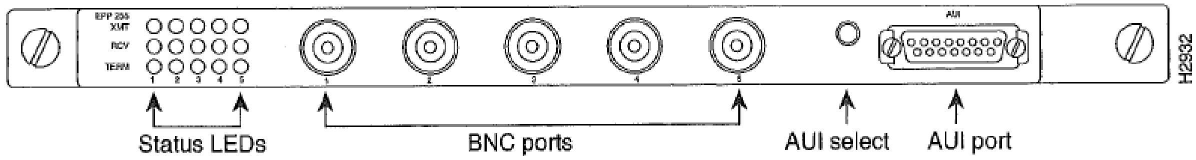
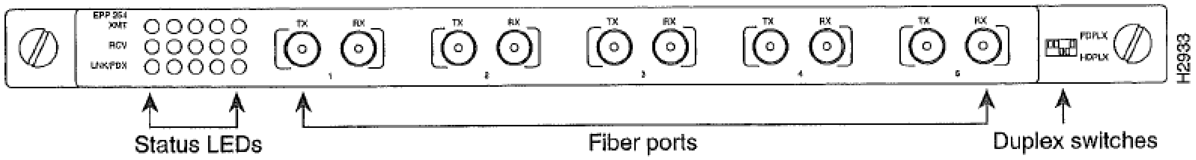
No.	’740 Patent Claim 17	Cisco EtherSwitch System
		 effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2015 RS provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module, the EtherSwitch can provide balanced network traffic to NetWare servers. Cisco EtherSwitch System Catalog at 119  **Product Overview** The Kalpana EtherSwitch EPS-2115M is a modular 15-port Ethernet switch that increases network throughput at the backbone and workgroup. The EtherSwitch EPS-2115M is designed to boost Ethernet performance by interconnecting multiple devices—hubs, EtherSwitches, routers, and servers—over multiple media types while maintaining reliability and flexibility.
17[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network,	Cisco EtherSwitch System discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. For example, Cisco EtherSwitch System discloses EtherSwitch network modules that read and process data packets with specific packet information, such as a destination address, that are sent between the interconnecting network devices in an Ethernet network. Cisco EtherSwitch System Catalog at 113

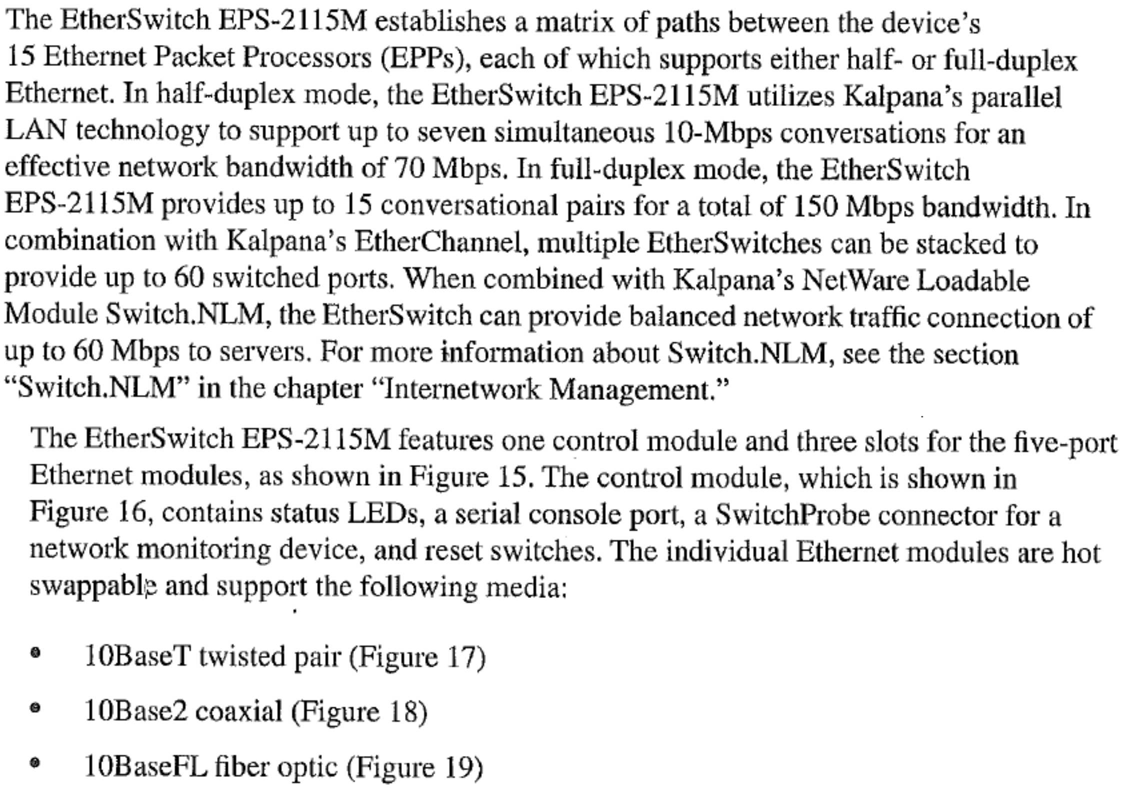
No.	’740 Patent Claim 17	Cisco EtherSwitch System
		 • Two models available A managed version, the EtherSwitch EPS-500-SNMP, provides Simple Network Management Protocol with Management Information Base (MIB) II support. Kalpana’s SNMP implementation allows Kalpana’s SwitchVision Network Management System application to collect system and port information from the EtherSwitch EPS-500. The entry-level version of the EtherSwitch EPS-500, which can be upgraded with SNMP, supports out-of-band network management via an EIA/TIA-232 connector. Cisco EtherSwitch System Catalog at 114  • Switching network architecture The EtherSwitch EPS-500 reduces contention—and thus improves overall network performance—by establishing a matrix of paths between the fifteen ports. All of these paths can be switched simultaneously to handle communications between ports. The EtherSwitch’s ports are not limited by configuration. Each can connect the following: — 10BaseT hubs, providing network services to stations — Single servers, so that servers receive the performance advantages of a dedicated 10-Mbps Ethernet half-duplex or 20-Mbps full-duplex connection — Other EtherSwitches for a hierarchical switching architecture that increases overall LAN capacity without the performance degradation that occurs with cascades of store-and-forward devices. Cisco EtherSwitch System Catalog at 115

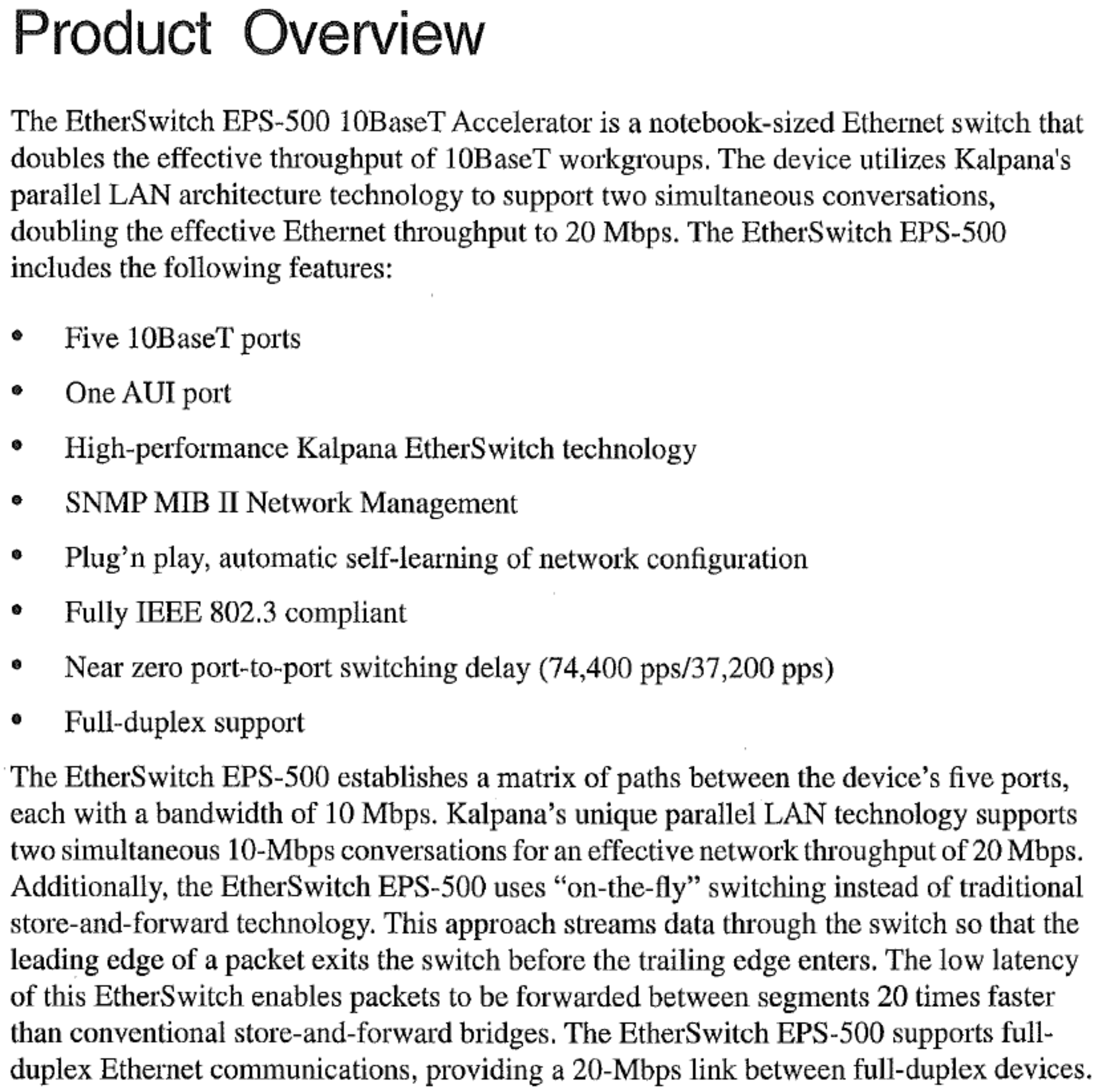
No.	’740 Patent Claim 17	Cisco EtherSwitch System
		 Product Overview The Kalpana EtherSwitch EPS-2015 RS is a stackable 15-port Ethernet switch that increases the throughput and flexibility of departmental and workgroup 10BaseT networks. Specifically designed to complement the low-cost connectivity of stackable hubs, the EtherSwitch EPS-2015 RS provides network managers with a cost-effective way to boost 10BaseT throughput among hubs and to servers while conserving valuable rack space. • High-performance Kalpana EtherSwitch technology • Fifteen 10BaseT ports with 150-Mbps bandwidth capacity • Full-duplex Ethernet support • SNMP MIB II network management • IEEE 802.1d Spanning-Tree Protocol • Virtual LAN support • User-defined address filtering • Updates using flash PROM • Scalable interswitch communications up to 150 Mbps • Delivers up to 60 switched ports when stacked with other EPS-2015 RS EtherSwitches • Rack-and-stack design, plug’n play, automatic self-learning The EtherSwitch EPS-2015 RS establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2015 RS uses Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an

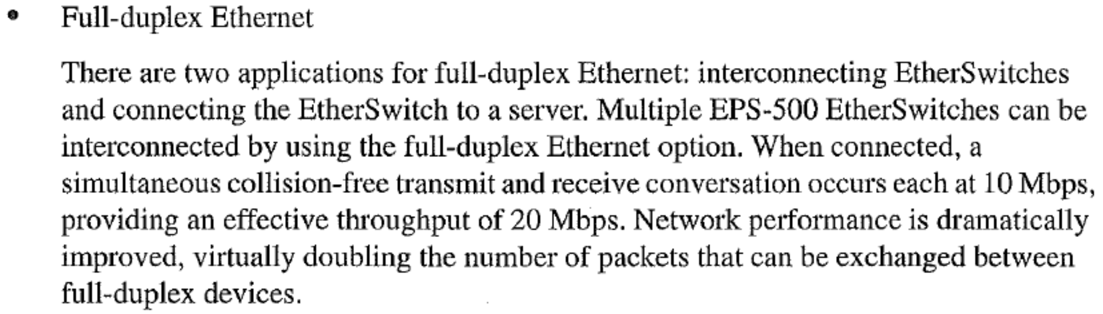
No.	’740 Patent Claim 17	Cisco EtherSwitch System
		 effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2015 RS provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module, the EtherSwitch can provide balanced network traffic to NetWare servers. Cisco EtherSwitch System Catalog at Figure 14 **Figure 14 EtherSwitch EPS-2015 RS Front Panel**  Kalpana PWR RCV DIAG XMIT STAT FD/LNK 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 MDI MDI MDI MDIX H2903 Status LEDs MDI/MDIX switches RJ-45 ports AUI port Cisco EtherSwitch System Catalog at Figures 15-19

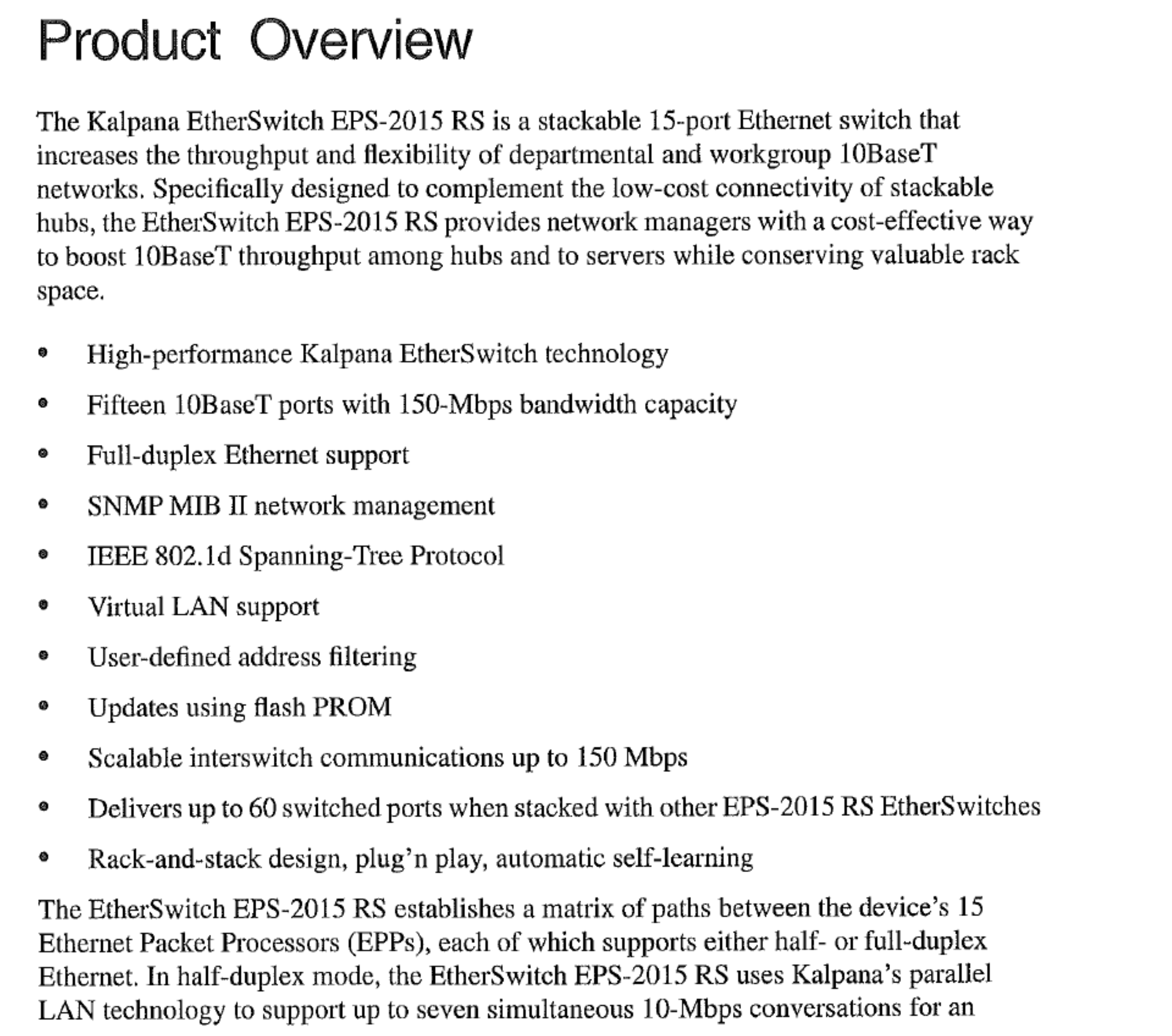
No.	’740 Patent Claim 17	Cisco EtherSwitch System
		Figure 15 EtherSwitch EPS-2115M Front Panel  Figure 16 EtherSwitch EPS-2115M Control Module  Kalpana EtherSwitch® EPS-2115M FAULT DIAG PWR A PWR B SwitchProbe RST SYSREQ H2934 FAULT LED DIAG LED PWR LED A PWR LED B EIA/TIA-232 SwitchProbe port Sys Req button Reset button

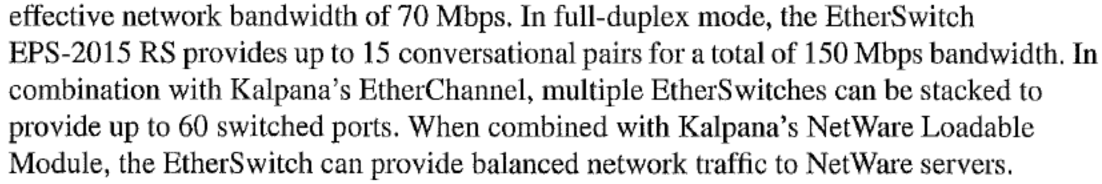
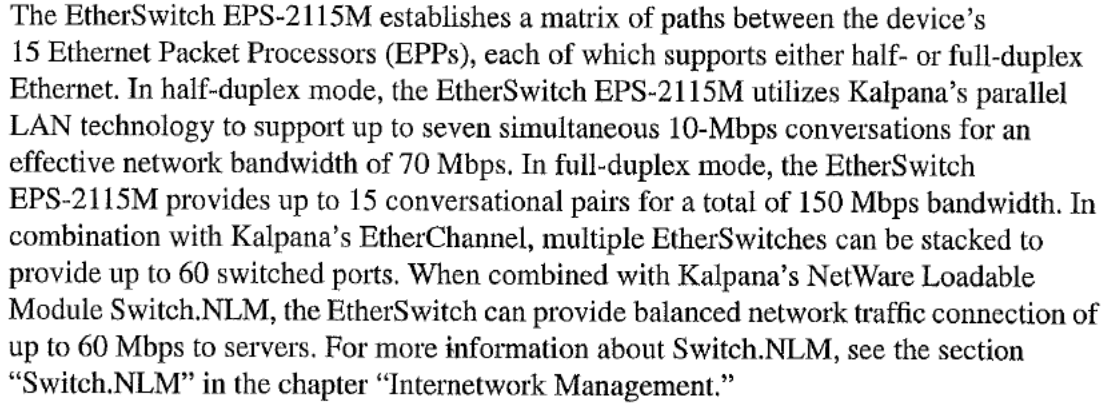
No.	’740 Patent Claim 17	Cisco EtherSwitch System
		Figure 17 EtherSwitch EPP-253 10BaseT Module  EPP 253 XMT RCV LNK/FDX MDI MDIX AUI FDPLX HDPLX H2931 Status LEDs MDI/MDI-X switch RJ-45 ports AUI port Duplex switches **Figure 18 EPP-255 10Base2 Module**  EPP 255 XMT RCV TERM 1 2 3 4 5 AUI H2932 Status LEDs BNC ports AUI select AUI port **Figure 19 EPP-254 10BaseFL Module**  EPP 254 XMT RCV LNK/FDX TX RX FDPLX HDPLX H2933 Status LEDs Fiber ports Duplex switches Cisco EtherSwitch System Catalog at 120

No.	’740 Patent Claim 17	Cisco EtherSwitch System
		 The EtherSwitch EPS-2115M establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2115M utilizes Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2115M provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module Switch.NLM, the EtherSwitch can provide balanced network traffic connection of up to 60 Mbps to servers. For more information about Switch.NLM, see the section “Switch.NLM” in the chapter “Internetwork Management.” The EtherSwitch EPS-2115M features one control module and three slots for the five-port Ethernet modules, as shown in Figure 15. The control module, which is shown in Figure 16, contains status LEDs, a serial console port, a SwitchProbe connector for a network monitoring device, and reset switches. The individual Ethernet modules are hot swappable and support the following media: • 10BaseT twisted pair (Figure 17) • 10Base2 coaxial (Figure 18) • 10BaseFL fiber optic (Figure 19)
17[b]	at least one of said interface modules being operative to communicate in both an upstream direction and a	Cisco EtherSwitch System discloses at least one of said interface modules being operative to communicate in both an upstream direction and a downstream direction. For example, Cisco EtherSwitch System discloses full duplex communication over the Ethernet network modules. A full duplex communication means inbound and outbound traffic are capable of being exchanged. Cisco EtherSwitch System Catalog at 111

No.	’740 Patent Claim 17	Cisco EtherSwitch System
	downstream direction;	 Product Overview The EtherSwitch EPS-500 10BaseT Accelerator is a notebook-sized Ethernet switch that doubles the effective throughput of 10BaseT workgroups. The device utilizes Kalpana's parallel LAN architecture technology to support two simultaneous conversations, doubling the effective Ethernet throughput to 20 Mbps. The EtherSwitch EPS-500 includes the following features: • Five 10BaseT ports • One AUI port • High-performance Kalpana EtherSwitch technology • SNMP MIB II Network Management • Plug’n play, automatic self-learning of network configuration • Fully IEEE 802.3 compliant • Near zero port-to-port switching delay (74,400 pps/37,200 pps) • Full-duplex support The EtherSwitch EPS-500 establishes a matrix of paths between the device’s five ports, each with a bandwidth of 10 Mbps. Kalpana’s unique parallel LAN technology supports two simultaneous 10-Mbps conversations for an effective network throughput of 20 Mbps. Additionally, the EtherSwitch EPS-500 uses “on-the-fly” switching instead of traditional store-and-forward technology. This approach streams data through the switch so that the leading edge of a packet exits the switch before the trailing edge enters. The low latency of this EtherSwitch enables packets to be forwarded between segments 20 times faster than conventional store-and-forward bridges. The EtherSwitch EPS-500 supports full-duplex Ethernet communications, providing a 20-Mbps link between full-duplex devices.

No.	’740 Patent Claim 17	Cisco EtherSwitch System
		Cisco EtherSwitch System Catalog at 114  • Full-duplex Ethernet There are two applications for full-duplex Ethernet: interconnecting EtherSwitches and connecting the EtherSwitch to a server. Multiple EPS-500 EtherSwitches can be interconnected by using the full-duplex Ethernet option. When connected, a simultaneous collision-free transmit and receive conversation occurs each at 10 Mbps, providing an effective throughput of 20 Mbps. Network performance is dramatically improved, virtually doubling the number of packets that can be exchanged between full-duplex devices. Cisco EtherSwitch System Catalog at 115

No.	’740 Patent Claim 17	Cisco EtherSwitch System
		 Product Overview The Kalpana EtherSwitch EPS-2015 RS is a stackable 15-port Ethernet switch that increases the throughput and flexibility of departmental and workgroup 10BaseT networks. Specifically designed to complement the low-cost connectivity of stackable hubs, the EtherSwitch EPS-2015 RS provides network managers with a cost-effective way to boost 10BaseT throughput among hubs and to servers while conserving valuable rack space. • High-performance Kalpana EtherSwitch technology • Fifteen 10BaseT ports with 150-Mbps bandwidth capacity • Full-duplex Ethernet support • SNMP MIB II network management • IEEE 802.1d Spanning-Tree Protocol • Virtual LAN support • User-defined address filtering • Updates using flash PROM • Scalable interswitch communications up to 150 Mbps • Delivers up to 60 switched ports when stacked with other EPS-2015 RS EtherSwitches • Rack-and-stack design, plug’n play, automatic self-learning The EtherSwitch EPS-2015 RS establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2015 RS uses Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an

No.	’740 Patent Claim 17	Cisco EtherSwitch System
		 effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2015 RS provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module, the EtherSwitch can provide balanced network traffic to NetWare servers. Cisco EtherSwitch System Catalog at 120  The EtherSwitch EPS-2115M establishes a matrix of paths between the device’s 15 Ethernet Packet Processors (EPPs), each of which supports either half- or full-duplex Ethernet. In half-duplex mode, the EtherSwitch EPS-2115M utilizes Kalpana’s parallel LAN technology to support up to seven simultaneous 10-Mbps conversations for an effective network bandwidth of 70 Mbps. In full-duplex mode, the EtherSwitch EPS-2115M provides up to 15 conversational pairs for a total of 150 Mbps bandwidth. In combination with Kalpana’s EtherChannel, multiple EtherSwitches can be stacked to provide up to 60 switched ports. When combined with Kalpana’s NetWare Loadable Module Switch.NLM, the EtherSwitch can provide balanced network traffic connection of up to 60 Mbps to servers. For more information about Switch.NLM, see the section “Switch.NLM” in the chapter “Internetwork Management.”
17[c]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Cisco EtherSwitch System discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 1[a].
17[d]	a second group of second physical links arranged in	Cisco EtherSwitch System discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 1[c].

No.	’740 Patent Claim 17	Cisco EtherSwitch System
	parallel so as to couple the one or more interface modules to the communication network; and	
17[e]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame;	Cisco EtherSwitch System discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 1[f].
17[f]	at least one of said first physical links	Cisco EtherSwitch System discloses at least one of said first physical links and at least one of said second links being bi-directional links operative to communicate in both said upstream direction and said downstream direction.

No.	’740 Patent Claim 17	Cisco EtherSwitch System
	and at least one of said second links being bi-directional links operative to communicate in both said upstream direction and said downstream direction.	*See supra at* 1[b], 1[d].

No.	’740 Patent Claim 18	Cisco EtherSwitch System
18[a]	The apparatus according to claim 17, and comprising a backplane to which the one or more interface modules are coupled,	Cisco EtherSwitch System discloses the apparatus according to claim 17, and comprising a backplane to which the one or more interface modules are coupled. *See supra at* 3, 17.
18[b]	wherein the second physical links comprise backplane traces formed on the backplane.	Cisco EtherSwitch System discloses wherein the second physical links comprise backplane traces formed on the backplane. *See supra at* 3, 17.

No.	’740 Patent Claim 19	Cisco EtherSwitch System
19[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Cisco EtherSwitch System discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
19[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	Cisco EtherSwitch System discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].
19[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Cisco EtherSwitch System discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].
19[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Cisco EtherSwitch System discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].

19[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,	Cisco EtherSwitch System discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].
19[e]	at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group.	Cisco EtherSwitch System discloses at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group. *See supra at* 4[f].

No.	’740 Patent Claim 20	Cisco EtherSwitch System
20[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Cisco EtherSwitch System discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
20[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	Cisco EtherSwitch System discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].
20[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Cisco EtherSwitch System discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].
20[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Cisco EtherSwitch System discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].

20[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,	Cisco EtherSwitch System discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].
20[e]	two or more of the first physical links being aggregated into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node.	Cisco EtherSwitch System discloses two or more of the first physical links being aggregated into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node. *See supra at* 19[e], 5[f].

No.	’740 Patent Claim 21	Cisco EtherSwitch System
21	The apparatus according to claim 17, and comprising a multiplexer, which is arranged to perform at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node.	Cisco EtherSwitch System discloses the apparatus according to claim 17, and comprising a multiplexer, which is arranged to perform at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node. *See supra at* 6, 17.

No.	’740 Patent Claim 22	Cisco EtherSwitch System
22	The apparatus according to claim 17, wherein the control module is arranged to balance a frame data rate among at least some of the first and second physical links.	Cisco EtherSwitch System discloses the apparatus according to claim 17, wherein the control module is arranged to balance a frame data rate among at least some of the first and second physical links. *See supra at* 7, 17.

No.	’740 Patent Claim 23	Cisco EtherSwitch System
23	The apparatus according to claim 17, wherein the control module is arranged to apply a mapping function to the at least one of the frame attributes so as to select the first and second physical links.	Cisco EtherSwitch System discloses the apparatus according to claim 17, wherein the control module is arranged to apply a mapping function to the at least one of the frame attributes so as to select the first and second physical links. *See supra at* 8, 17.

No.	’740 Patent Claim 24	Cisco EtherSwitch System
24	The apparatus according to claim 23, wherein the mapping function comprises a hashing function.	Cisco EtherSwitch System discloses the apparatus according to claim 23, wherein the mapping function comprises a hashing function. *See supra at* 9, 23.

No.	’740 Patent Claim 25	Cisco EtherSwitch System
25[a]	The apparatus according to claim 24, wherein the control module is arranged to determine a hashing size responsively to a number of at least some of the first and second physical links,	Cisco EtherSwitch System discloses the apparatus according to claim 24, wherein the control module is arranged to determine a hashing size responsively to a number of at least some of the first and second physical links. *See supra at* 10[a], 24.
25[b]	to apply the hashing function to the at least one of the frame attributes to produce a hashing key,	Cisco EtherSwitch System discloses to apply the hashing function to the at least one of the frame attributes to produce a hashing key. *See supra at* 10[b].
25[c]	to calculate a modulo of a division operation of the hashing key by the hashing size, and	Cisco EtherSwitch System discloses to calculate a modulo of a division operation of the hashing key by the hashing size. *See supra at* 10[c].
25[d]	to select the first and second physical links responsively to the modulo.	Cisco EtherSwitch System discloses to select the first and second physical links responsively to the modulo. *See supra at* 10[d].

No.	’740 Patent Claim 26	Cisco EtherSwitch System
26	The apparatus according to claim 25, wherein the control module is arranged to select the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo.	Cisco EtherSwitch System discloses the apparatus according to claim 25, wherein the control module is arranged to select the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo. *See supra at* 11, 25.

No.	’740 Patent Claim 27	Cisco EtherSwitch System
27	The apparatus according to claim 17, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port.	Cisco EtherSwitch System discloses the apparatus according to claim 17, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port. *See supra at* 12, 17.

No.	’740 Patent Claim 28	Cisco EtherSwitch System
28[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Cisco EtherSwitch System discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
28[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	Cisco EtherSwitch System discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].
28[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Cisco EtherSwitch System discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].
28[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Cisco EtherSwitch System discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].

28[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,	Cisco EtherSwitch System discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].
28[e]	the communication network being arranged to provide a communication service to the network node,	Cisco EtherSwitch System discloses the communication network being arranged to provide a communication service to the network node. *See supra at* 2[b].

28[f]	the service having specified bandwidth requirements comprising at least one of a committed information rate (CR), a peak information rate (PIR) and an excess information rate (EIR), and	Cisco EtherSwitch System discloses the service having specified bandwidth requirements comprising at least one of a committed information rate (CR), a peak information rate (PIR) and an excess information rate (EIR). *See supra at* 13[i].
28[g]	the first and second groups of physical links being dimensioned to provide an allocated bandwidth for the communication service responsively to the bandwidth requirements.	Cisco EtherSwitch System discloses the first and second groups of physical links being dimensioned to provide an allocated bandwidth for the communication service responsively to the bandwidth requirements. *See supra at* 13[j].

No.	’740 Patent Claim 29	Cisco EtherSwitch System
29[preamble]	Apparatus for connecting user ports to a communication network, comprising:	Cisco EtherSwitch System discloses apparatus for connecting user ports to a communication network. *See supra at* 17[preamble], 14[preamble].
29[a]	one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network,	Cisco EtherSwitch System discloses one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network. *See supra at* 17[a], 14[a].
29[b]	at least one of said user interface modules being bi-directional and operative to communicate in both an upstream direction and a downstream direction;	Cisco EtherSwitch System discloses at least one of said user interface modules being bi-directional and operative to communicate in both an upstream direction and a downstream direction. *See supra at* 17[b], 14[c].

29[c]	a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network,	Cisco EtherSwitch System discloses a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network. *See supra at* 14[b]-[e].
29[d]	at least one of said backplane traces being bi-directional and operative to communicate in both said upstream direction and said downstream direction; and	Cisco EtherSwitch System discloses at least one of said backplane traces being bi-directional and operative to communicate in both said upstream direction and said downstream direction. *See supra at* 14[c], 17[b].

29[e]	a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame.	Cisco EtherSwitch System discloses a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame. *See supra at* 14[e], 17[e].

No.	’740 Patent Claim 30	Cisco EtherSwitch System
30[preamble]	Apparatus for connecting user ports to a communication network, comprising:	Cisco EtherSwitch System discloses apparatus for connecting user ports to a communication network. *See supra at* 29[preamble].
30[a]	one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network;	Cisco EtherSwitch System discloses one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network. *See supra at* 29[a].
30[b]	a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network;	Cisco EtherSwitch System discloses a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network. *See supra at* 29[c].

30[c]	a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame;	Cisco EtherSwitch System discloses a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame. *See supra at* 29[e].
30[d]	at least some of the backplane traces are aggregated into an Ethernet link aggregation (LAG) group.	Cisco EtherSwitch System discloses at least some of the backplane traces are aggregated into an Ethernet link aggregation (LAG) group. *See supra at* 4[f], 15[f].

No.	**’740 Patent Claim 31**	**Cisco EtherSwitch System**
31	The apparatus according to claim 29, wherein the control module is arranged to apply a hashing function to the at least one of the frame attributes so as to select the backplane trace.	Cisco EtherSwitch System discloses the apparatus according to claim 29, wherein the control module is arranged to apply a hashing function to the at least one of the frame attributes so as to select the backplane trace. *See supra at* 16, 29, 30[c].

EXHIBIT C-6
Defendant’s Preliminary Invalidity Contentions
Orckit Corporation v. Cisco Systems, Inc., 2:22-cv-00276-JRG-RSP

<u>Chart for U.S. Patent 7,545,740 (“the ’740 Patent”)</u>
<u>U.S. Patent Publication No. 2004/0228278 to Bruckman et al. (“Bruckman”)</u>

As shown in the chart below, all Asserted Claims of the ’740 Patent are invalid under (1) 35 U.S.C. §§ 102 (a), (b), (e), (g), and (f)[1] because Bruckman meets each element of those claims, and/or (2) 35 U.S.C. § 103 because Bruckman renders those claims obvious either alone, or in combination with the knowledge of a person having ordinary skill in the art, and in further combination with the references specifically identified below and in the following claim chart and/or one or more references identified in Defendant’s Preliminary Invalidity Contentions. The following quotations and diagrams come from Bruckman titled “Bandwidth Allocation For Link Aggregation”, which was filed on May 13, 2003, and published on November 18, 2004.

Motivations to combine the disclosures in Bruckman with disclosures in other publications known in the art, as explained in this chart, include at least the similarity in subject matter between the references to the extent they concern methods of data communication systems, and specifically to methods and systems for link aggregation in a data communication network. Insofar as the references cite other patents or publications, or suggest additional changes, one of ordinary skill in the art would look beyond a single reference to other references in the field.

These invalidity contentions are based on Defendant’s present understanding of the asserted claims, and Orckit’s apparent construction of the claims in its November 3, 2022 Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1, and Orckit’s January 19, 2023 First Amended Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1 (Orckit’s “Infringement Disclosures”), which is deficient at least insofar as it fails to cite any documents or identify accused structures, acts, or materials in the Accused Products with particularity. Defendant does not agree with Orckit’s application of the claims, or that the claims satisfy the requirements of 35 U.S.C. § 112. Defendant’s contentions herein are not, and should in no way be seen as, admissions or adoptions as to any particular claim scope or construction, or as any admission that any particular element is met by any accused product in any particular way. Defendant objects to any attempt to imply claim construction from this chart. Defendant’s prior art invalidity

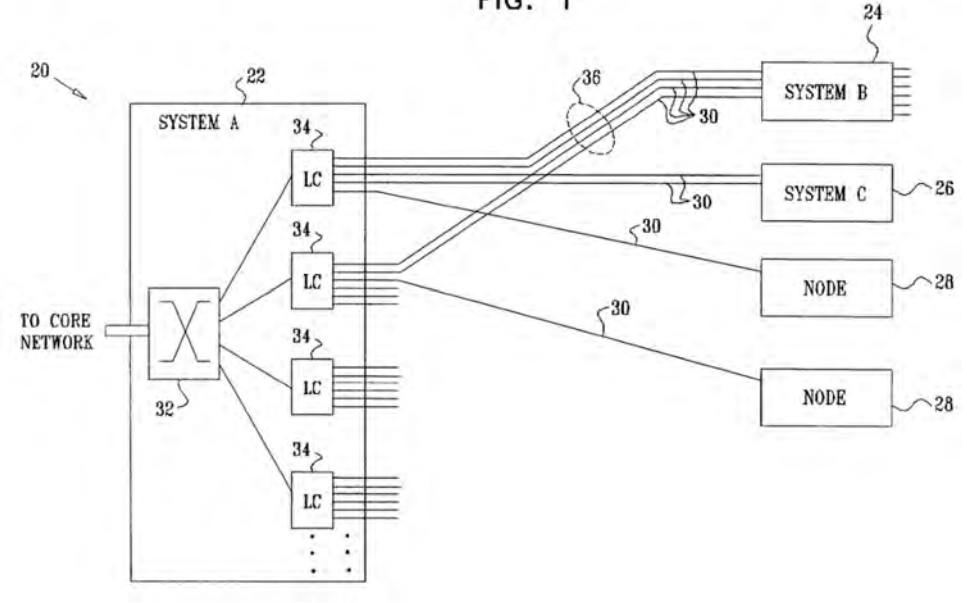
[1] How Bruckman invalidates the ’740 Patent under 35 U.S.C. §102(f) is explained in Section V of Cisco’s concurrently served Preliminary Invalidity Contention cover pleadings.

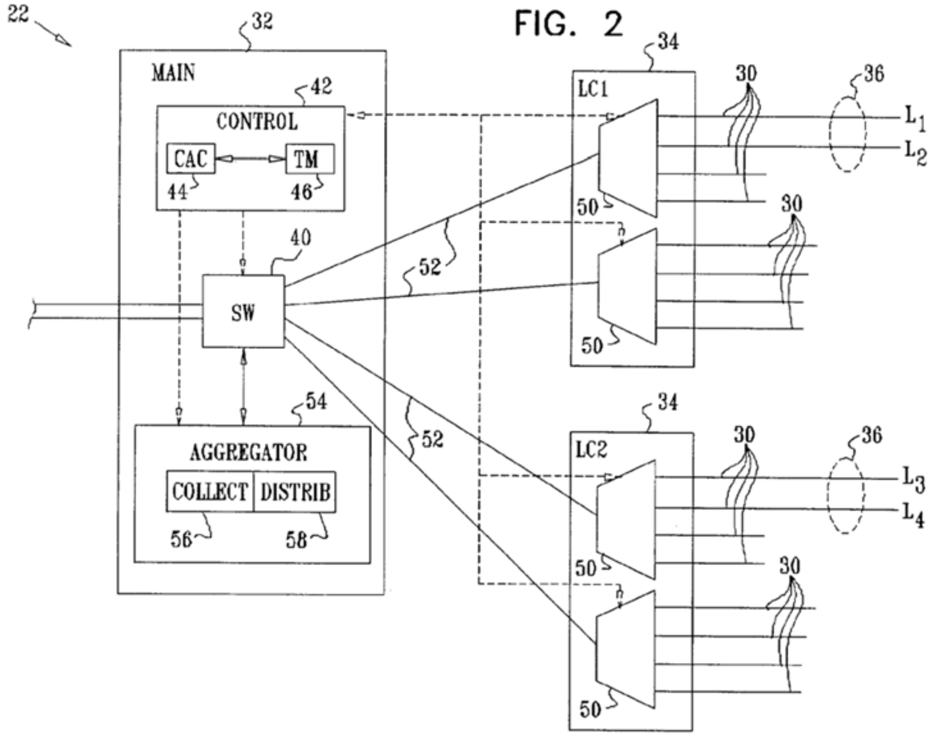
contentions are made in a variety of alternatives and do not represent Defendant's agreement or view as to the meaning, definiteness, written description support for, or enablement of any claim contained therein.

The following contentions are subject to revision and amendment pursuant to Federal Rule of Civil Procedure 26(e), the Local Rules, and the Orders of record in this matter subject to further investigation and discovery regarding the prior art and the Court's construction of the claims at issue.

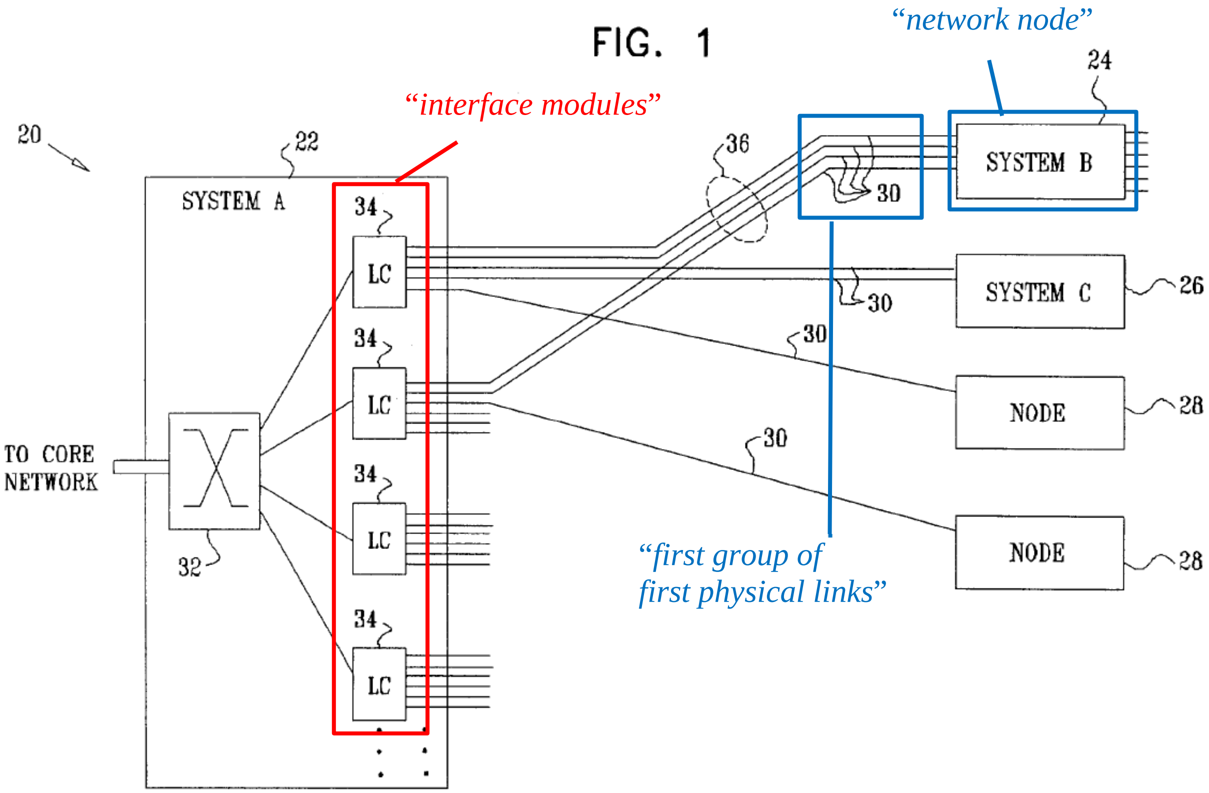
No.	'740 Patent Claim 1	Bruckman
1[preamble]	A method for communication, comprising:	Bruckman discloses a method for communication. For example, Bruckman discloses a method of transmitting data between endpoints. Bruckman at Abstract ("A method for establishing a connection with a guaranteed bandwidth for transmitting data over a logical link that includes a plurality of parallel physical links between first and second endpoints. A link bandwidth is allocated on each of the physical communication links so as to include a predefined safety margin, based on either a failure protection policy, or a measure of fluctuation that occurs in a rate of data transmission over the physical links, or both. A sum of the allocated link bandwidth over the plurality of the parallel physical links is substantially greater than the guaranteed bandwidth of the connection. The data are conveyed over the logical link by distributing the data for transmission among the physical links in accordance with the allocated link bandwidth.") Bruckman at [0001] ("The present invention relates generally to data communication systems, and specifically to methods and systems for link aggregation in a data communication net-work.") Bruckman at [0018]-[0022] ("There is therefore provided, in accordance with an embodiment of the present invention, a method for estab-lishing a connection with a guaranteed bandwidth for trans-mitting data between first and second endpoints, the method including: [0019] defining a logical link including a plurality of parallel physical links between the endpoints; [0020] setting a protection policy to be applied to the logical link; [0021] allocating a link bandwidth on each of the physical communication links for use in conveying the data between the endpoints such that the allocated link bandwidth includes a

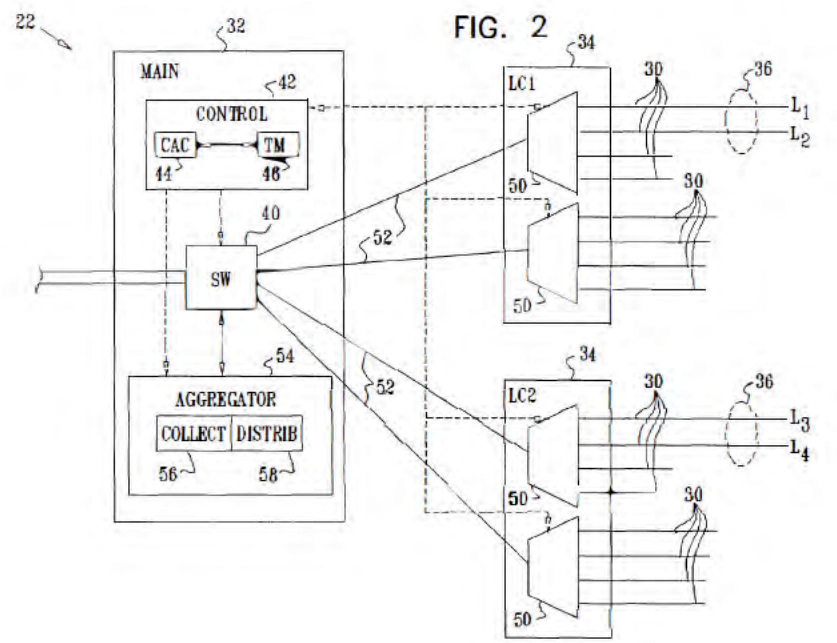
No.	’740 Patent Claim 1	Bruckman
		predefined safety margin based on the protection policy, so that a sum of the allocated link bandwidth over the plurality of the parallel physical links is substantially greater than the guaranteed bandwidth of the connection; and [0022] conveying the data over the logical link by distrib-uting the data for transmission among the physical links in accordance with the allocated link bandwidth.”) Bruckman at [0027]-[0031] (“There is also provided, in accordance with an embodiment of the present invention, a method for estab-lishing a connection with a guaranteed bandwidth for trans-mitting data between first and second endpoints, the method including: [0028] defining a logical link including a plurality of parallel physical links between the endpoints; [0029] determining a measure of fluctuation that occurs in a rate of transmission of the data over the physical links when the data are distributed for transmission among the physical links; [0030] allocating a link bandwidth on each of the physical communication links for use in conveying the data between the endpoints such that the allocated link bandwidth includes a predefined safety margin based on the measure of fluc-tuation, so that a sum of the allocated link bandwidth over the plurality of the parallel physical links is substantially greater than the guaranteed bandwidth of the connection; and [0031] conveying the data over the logical link by distrib-uting the data for transmission among the physical links in accordance with the allocated link bandwidth.”) Bruckman at [0055] (“Based on these records, CAC 44 decides whether to admit each request received by equipment 22 to set up a new connection, and allocates resources (such as bandwidth) to the connection accordingly. A "connection" is defined as a flow of data packets between two systems in a network, such as Systems A and B in FIG. 1. Such a flow may carry multiple conversations. All conversations on a given con-nection share the same bandwidth and are treated in a substantially identical manner by equipment 22. If a new connection requires more bandwidth than equipment 22 has available, the CAC rejects the request.”) Bruckman at Figure 1

No.	’740 Patent Claim 1	Bruckman
		FIG. 1  20 22 SYSTEM A 34 LC 34 LC 34 LC 34 LC TO CORE NETWORK 32 36 24 SYSTEM B 30 SYSTEM C 30 26 30 NODE 28 30 NODE 28 Bruckman at Figure 2

No.	’740 Patent Claim 1	Bruckman
		FIG. 2  22 32 34 MAIN 42 CONTROL CAC TM 44 46 LC1 30 36 L_1 L_2 50 30 40 SW 52 50 34 54 AGGREGATOR COLLECT DISTRIB 56 58 52 LC2 30 36 L_3 L_4 50 30 50
1[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel,	Bruckman discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. For example, Bruckman discloses connecting nodes to line cards having ports using physical links. Bruckman at Abstract (“A method for establishing a connection with a guaranteed bandwidth for transmitting data over a logical link that includes a plurality of parallel physical links between first and second endpoints. A link bandwidth is allocated on each of the physical communication links so as to include a predefined safety margin, based on either a failure protection policy, or a

No.	’740 Patent Claim 1	Bruckman
		measure of fluctuation that occurs in a rate of data transmission over the physical links, or both. A sum of the allocated link bandwidth over the plurality of the parallel physical links is substantially greater than the guaranteed bandwidth of the connection. The data are conveyed over the logical link by distributing the data for transmission among the physical links in accordance with the allocated link bandwidth.”) Bruckman at [0002] (“Link aggregation is a technique by which a group of parallel physical links between two endpoints in a data network can be joined together into a single logical link. Traffic transmitted between the endpoints is distributed among the physical links in a manner that is transparent to the clients that send and receive the traffic. Link aggregation offers benefits of increased bandwidth, as well as increased availability, since the logical link can continue to function (possibly with reduced bandwidth) even when one of the physical links fails or is taken out of service.”) Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”) Bruckman at [0047] (“FIG. 1 is a block diagram that schematically illus-trates elements of a communication system 20, in accor-dance with an embodiment of the present invention. In this example, central office equipment 22 communicates with customer nodes 24, 26, 28, ... , over physical links 30. Links 30 typically comprise full-duplex Ethernet links, such as lOBASE-n, lO0BASE-n or Gigabit Ethernet links, as are known in the art. (Alternatively, as noted above, other types of physical links may be used, such as ATM or PPP links.) Equipment 22 is configured

No.	’740 Patent Claim 1	Bruckman
		to convey packet data traffic between the customer nodes and a network (which may be a metro network, access network, or other type of core network, for example). For this purpose, equipment 22 comprises a main switching card 32, which is connected to multiple line cards 34 that serve links 30. Details of the structure and operation of equipment 22 are shown below in FIG. 2 and are described with reference thereto.”) Bruckman at Figure 1 (annotated) FIG. 1  “network node” “interface modules” 20 22 24 36 SYSTEM A SYSTEM B SYSTEM C 34 LC 30 26 28 NODE TO CORE NETWORK 32 “first group of first physical links”

No.	’740 Patent Claim 1	Bruckman
		Bruckman at [0049] (“FIG. 2 is a block diagram that schematically shows details of equipment 22, in accordance with an embodiment of the present invention. Main card 32 comprises a switching core 40, which switches traffic to and from line cards 34. Two line cards 34, labeled LCl and LC2, are shown in the figure. The operation of switch 40 is managed by a controller 42, typically an embedded microprocessor with suitable software for carrying out the functions described herein.”) Bruckman at Figure 2  FIG. 2 22 32 MAIN 42 CONTROL CAC TM 44 46 40 SW 54 AGGREGATOR COLLECT DISTRIB 56 58 52 52 34 LC1 30 36 L_1 L_2 50 30 50 34 LC2 30 36 L_3 L_4 50 30 50

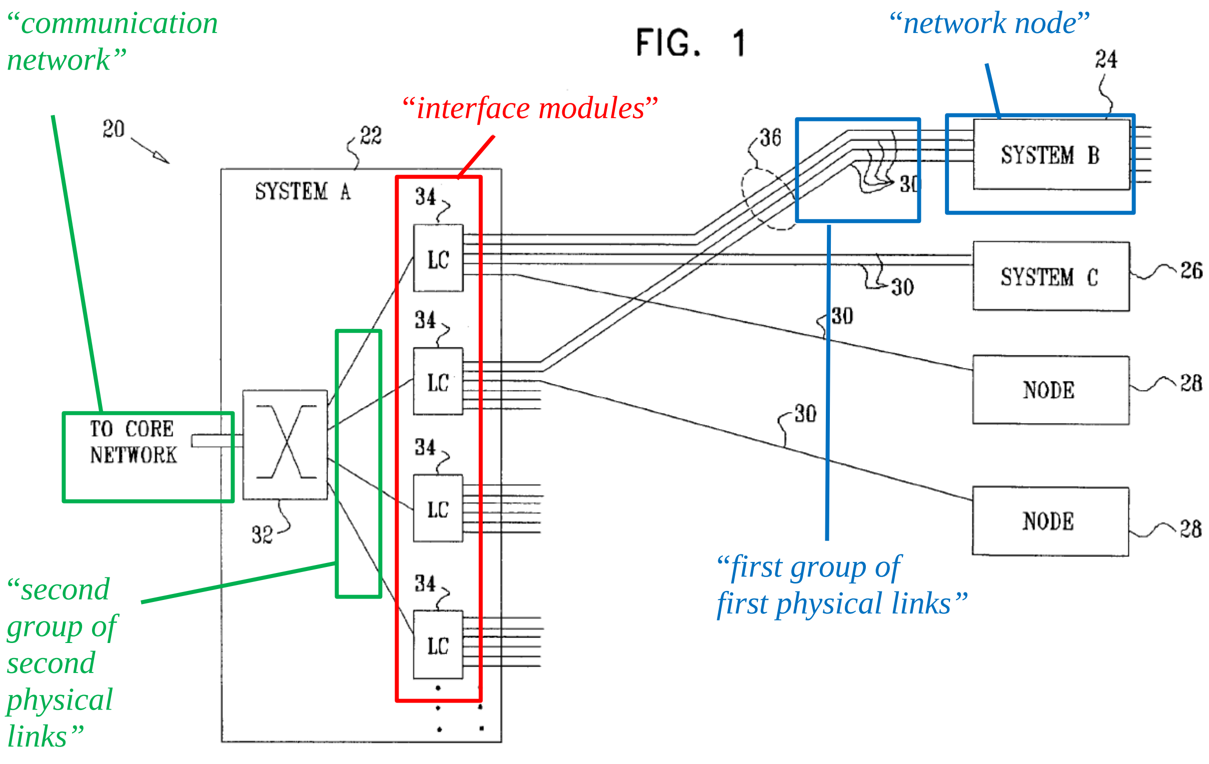
No.	’740 Patent Claim 1	Bruckman
		Bruckman at [0056] (“Each line card 34 comprises one or more concen-trators 50, which comprise multiple ports that serve respec-tive links 30. The concentrators multiplex data traffic between links 30 and traces 52, which connect the concen-trators to switching core 40. Typically, main card 32 and line cards 34 are arranged in a card rack and plug into a printed circuit back plane, (not shown) which comprises traces 52. The bandwidth of each trace 52 may be less than the total bandwidth available on links 30 that are connected to the respective concentrator 50, based on considerations of sta-tistical multiplexing. To prevent overloading of traces 52, concentrators 50 may limit the rate of incoming data admit-ted on each link 30 so that it remains between a predeter-mined minimum, which is determined by the guaranteed bandwidth of the connections on the link, and a maximum, which is determined by the peak bandwidth (guaranteed plus permitted excess bandwidth) of the connections on the link. A traffic manager 46, which may also be a software process on controller 42, receives information regarding the opera-tional status of links 30 (for example, link or equipment failures) and updates the data rate limits applied by concen-trators 50, based on the status information and the bandwidth allocations made by CAC 44.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”)

No.	’740 Patent Claim 1	Bruckman
1[b]	at least one of said first physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction	Bruckman discloses at least one of said first physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction. For example, Bruckman discloses physical links that both transmit and receive data. Bruckman at [0047] (“FIG. 1 is a block diagram that schematically illus-trates elements of a communication system 20, in accor-dance with an embodiment of the present invention. In this example, central office equipment 22 communicates with customer nodes 24, 26, 28, ... , over physical links 30. Links 30 typically comprise full-duplex Ethernet links, such as lOBASE-n, lO0BASE-n or Gigabit Ethernet links, as are known in the art. (Alternatively, as noted above, other types of physical links may be used, such as ATM or PPP links.) Equipment 22 is configured to convey packet data traffic between the customer nodes and a network (which may be a metro network, access network, or other type of core network, for example). For this purpose, equipment 22 comprises a main switching card 32, which is connected to multiple line cards 34 that serve links 30. Details of the structure and operation of equipment 22 are shown below in FIG. 2 and are described with reference thereto.”) Bruckman at [0056] (“Each line card 34 comprises one or more concen-trators 50, which comprise multiple ports that serve respec-tive links 30. The concentrators multiplex data traffic between links 30 and traces 52, which connect the concen-trators to switching core 40. Typically, main card 32 and line cards 34 are arranged in a card rack and plug into a printed circuit back plane, (not shown) which comprises traces 52. The bandwidth of each trace 52 may be less than the total bandwidth available on links 30 that are connected to the respective concentrator 50, based on considerations of sta-tistical multiplexing. To prevent overloading of traces 52, concentrators 50 may limit the rate of incoming data admit-ted on each link 30 so that it remains between a predeter-mined minimum, which is determined by the guaranteed bandwidth of the connections on the link, and a maximum, which is determined by the peak bandwidth (guaranteed plus permitted excess bandwidth) of the connections on the link. A traffic manager 46, which may also be a software process on controller 42, receives information regarding the opera-tional status of links 30 (for example, link or equipment failures) and updates the data rate limits applied by

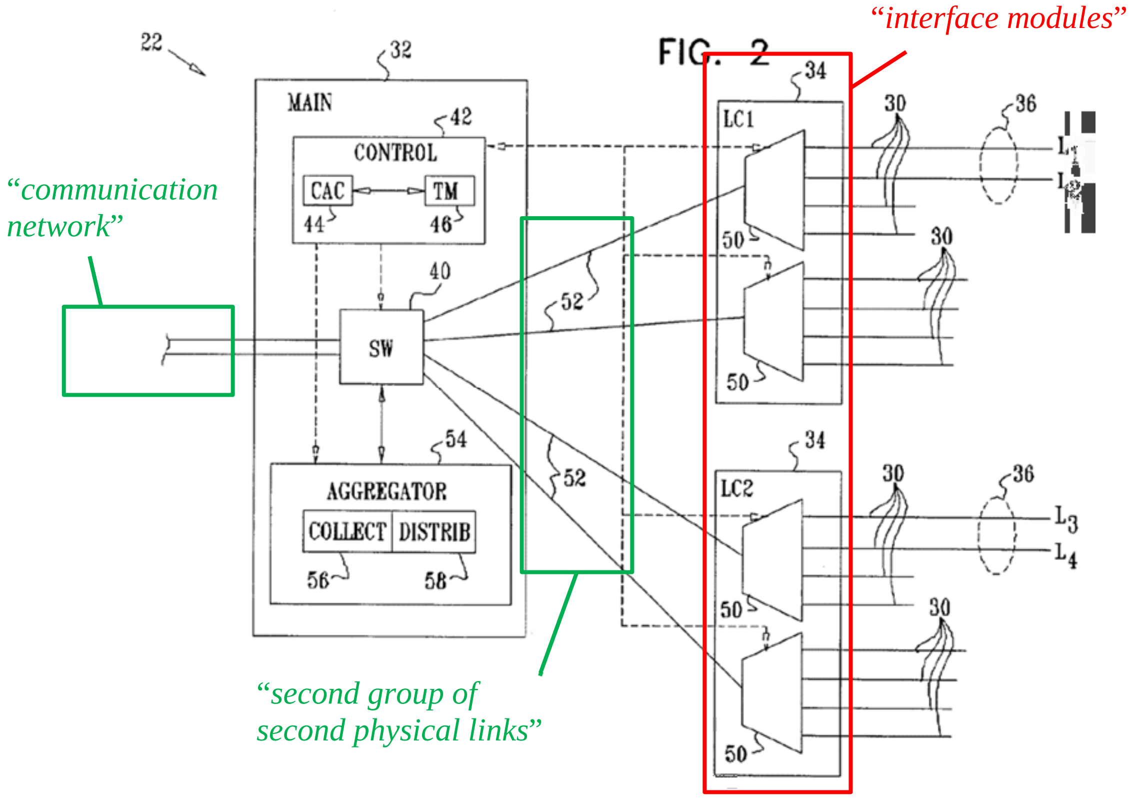
No.	’740 Patent Claim 1	Bruckman
		concen-trators 50, based on the status information and the bandwidth allocations made by CAC 44.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”) Bruckman at [0065] (“Aggregator 54 further comprises a collector 56, which collects data frames that were received over different links 30 in group 36, and arranges the frames back into a single traffic stream.”) Bruckman at [0073] (“Once CAC 44 has allocated bandwidth for a given connection on a link aggregation group, normal data trans-mission proceeds. The bandwidth allocations apply to the amount of guaranteed traffic carried on each link 30 in the group. (Note that different allocations and separate traffic management may apply to outgoing traffic generated by distributor 58 and incoming traffic, which is sent by nodes 24, 26, ... , and processed by collector 56.) The allocations also affect the bandwidth used on traces 52. The rate limiting function of concentrators 50 is set to allow for the traffic bandwidth that may be used on each of links 30 that feed the respective trace. As noted above, in allocating the band-width, CAC 44 ensures that the sum of the guaranteed bandwidth on all links sharing a given trace 52 is no greater than the trace bandwidth. The sum of the excess bandwidth allocated on the links, however, may exceed the trace bandwidth. In this case, the excess traffic is typically buff-ered as necessary, and is transmitted over the trace during

No.	’740 Patent Claim 1	Bruckman
		intervals in which one or more of the links are not trans-mitting their guaranteed traffic levels and the trace has bandwidth available, or dropped if the buffer capacity is exceeded.”) Bruckman at [0074] (“Normal data transmission over the connection con-tinues unless and until a failure is detected on one of links 30 or line cards 34, at a failure detection step 68. Traffic manager 46 is informed of the failure, and notifies distribu-tor 58 accordingly to modify its hash function so that outgoing traffic is distributed over the remaining links in the group. Use of the protection parameter P in setting the bandwidth allocation ensures that (as long as no more than P links are out of service) there is sufficient bandwidth available for the connection on the remaining links. It may also be necessary for the traffic manager to adjust the rate limiting function of concentrators 50, at a concentrator readjustment step 70, in order to deal with the increased incoming traffic on the remaining links. For example, if link L4 (FIG. 2) fails, the traffic on each of links Ll, L2 and L3 is expected to increase by 1/3, and the concentrator in LCl will have to deal with the resulting increase in traffic on the corresponding trace 52.”)
1[c]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel,	Bruckman discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. For example, Bruckman discloses connecting the line cards to the network using traces comprising a backplane. Bruckman at [0002] (“Link aggregation is a technique by which a group of parallel physical links between two endpoints in a data network can be joined together into a single logical link. Traffic transmitted between the endpoints is distributed among the physical links in a manner that is transparent to the clients that send and receive the traffic. Link aggregation offers benefits of increased bandwidth, as well as increased availability, since the logical link can continue to function (possibly with reduced bandwidth) even when one of the physical links fails or is taken out of service.”)

No.	’740 Patent Claim 1	Bruckman
		Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”) Bruckman at Figure 1 (annotated)

No.	’740 Patent Claim 1	Bruckman
		 “communication network” “network node” “interface modules” 20 22 24 26 28 28 30 30 30 30 32 34 34 34 34 36 SYSTEM A SYSTEM B SYSTEM C NODE NODE LC LC LC LC TO CORE NETWORK “second group of second physical links” “first group of first physical links” FIG. 1 Bruckman at [0056] (“Each line card 34 comprises one or more concen-trators 50, which comprise multiple ports that serve respec-tive links 30. The concentrators multiplex data traffic between links 30 and traces 52, which connect the concen-trators to switching core 40. Typically, main card 32 and line cards 34 are arranged in a card rack and plug into a printed circuit back plane, (not shown) which comprises traces 52. The bandwidth of each trace 52 may be less than the total bandwidth available on links 30 that are connected to the respective concentrator 50, based on considerations of sta-tistical multiplexing. To prevent overloading of traces 52, concentrators 50

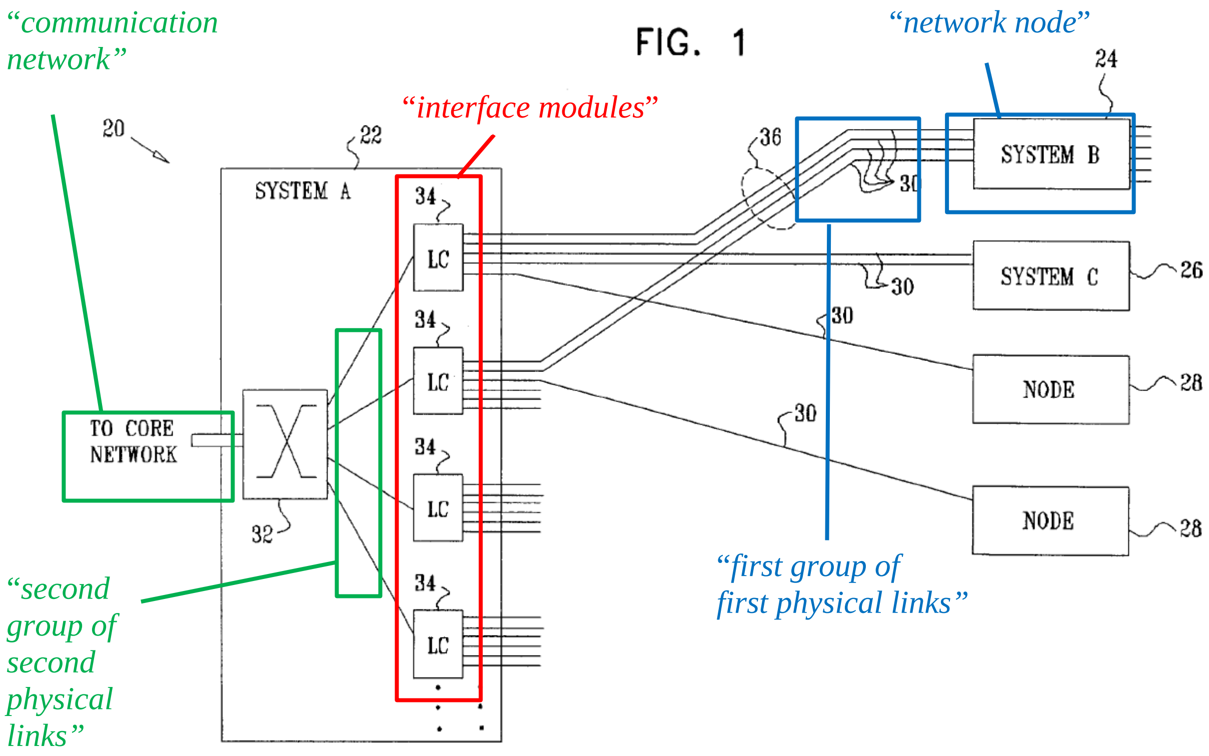
No.	’740 Patent Claim 1	Bruckman
		may limit the rate of incoming data admit-ted on each link 30 so that it remains between a predeter-mined minimum, which is determined by the guaranteed bandwidth of the connections on the link, and a maximum, which is determined by the peak bandwidth (guaranteed plus permitted excess bandwidth) of the connections on the link. A traffic manager 46, which may also be a software process on controller 42, receives information regarding the opera-tional status of links 30 (for example, link or equipment failures) and updates the data rate limits applied by concen-trators 50, based on the status information and the bandwidth allocations made by CAC 44.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at Figure 2 (annotated)

No.	’740 Patent Claim 1	Bruckman
		 FIG. 2 22 32 MAIN 42 CONTROL CAC TM 44 46 40 SW 54 AGGREGATOR COLLECT DISTRIB 56 58 52 52 34 LC1 50 50 30 30 36 L1 L2 34 LC2 30 30 36 L3 L4 50 50 *“interface modules”* *“communication network”* *“second group of second physical links”* Bruckman at [0066] (“When CAC 44 receives a request to open a con-nection with guaranteed bandwidth B over an aggregation group of N links, it might be assumed that the CAC should simply allocate bandwidth of BIN on each link. In practice, however, even if the hash function applied by distributor 58 meets the criteria outlined above, statistical variations in the traffic itself are likely to cause a larger portion of the traffic to be distributed to some of the links in the group than to others. In other words, some of the links may be required at times to carry group traffic

No.	’740 Patent Claim 1	Bruckman
		with bandwidth substantially greater than B/N. As a result, these links may not have sufficient capacity remaining to provide bandwidth that has been guaranteed to other connections that the CAC has committed to carry over these links. When an aggregation group extends over a number of concentrators 50 (as in the case of group 36), the traffic load on traces 52 may also be unbalanced. Overloading of traces 52 may likewise lead to a failure of system 22 to provide guaranteed bandwidth levels, in the distribution and/or the collection direction.”) Bruckman at [0067] (“A similar problem may arise if there is a failure in a link in an aggregation group or in one of a number of line cards serving the aggregation group. In this case, to maintain the bandwidth allocation B made by CAC 44, each of the remaining links in the group must now carry, on average, B/(N-M) traffic, wherein M is the number of links in the group that are out of service. If only BIN has been allocated to each link, the remaining active links may not have sufficient bandwidth to continue to provide the bandwidth that has been guaranteed to the connections that they are required to carry. A similar problem arises with respect to loading of traces 52. For example, if there is a failure in LC2 or in one of links 30 in group 36 that connect to LC2, the trace connecting the multiplexer 50 in LCl will have to carry a substantially larger share of the bandwidth, or even all of the bandwidth, that is allocated to the connection in ques-tion.”)
1[d]	at least one of said second physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction;	Bruckman discloses at least one of said second physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction. For example, Bruckman discloses traces that both transmit and receive data. Bruckman at [0056] (“Each line card 34 comprises one or more concen-trators 50, which comprise multiple ports that serve respec-tive links 30. The concentrators multiplex data traffic between links 30 and traces 52, which connect the concen-trators to switching core 40. Typically, main card 32 and line cards 34 are arranged in a card rack and plug into a printed circuit back plane, (not shown) which comprises traces 52. The bandwidth of each trace 52 may be less than the total bandwidth available on links 30 that are connected to the respective concentrator 50, based on

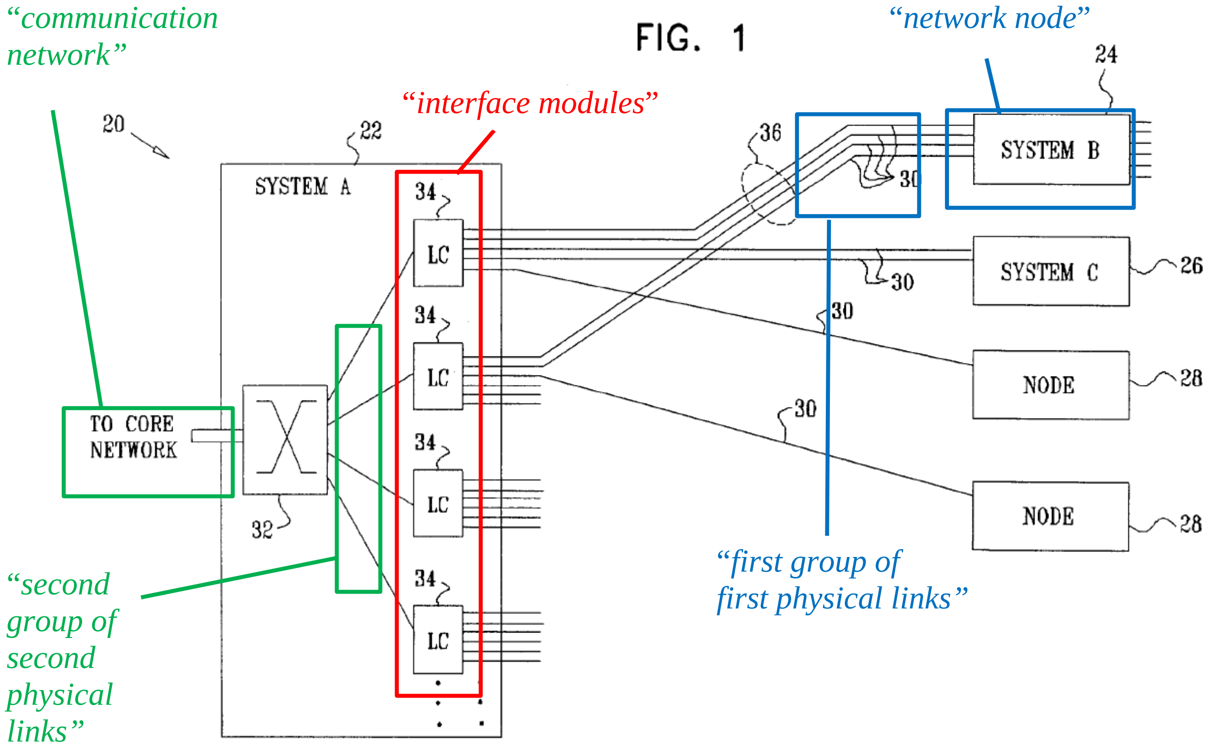
No.	’740 Patent Claim 1	Bruckman
		considerations of sta-tistical multiplexing. To prevent overloading of traces 52, concentrators 50 may limit the rate of incoming data admit-ted on each link 30 so that it remains between a predeter-mined minimum, which is determined by the guaranteed bandwidth of the connections on the link, and a maximum, which is determined by the peak bandwidth (guaranteed plus permitted excess bandwidth) of the connections on the link. A traffic manager 46, which may also be a software process on controller 42, receives information regarding the opera-tional status of links 30 (for example, link or equipment failures) and updates the data rate limits applied by concen-trators 50, based on the status information and the bandwidth allocations made by CAC 44.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria:

No.	’740 Patent Claim 1	Bruckman
		[0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”) Bruckman at [0065] (“Aggregator 54 further comprises a collector 56, which collects data frames that were received over different links 30 in group 36, and arranges the frames back into a single traffic stream.”) Bruckman at [0073] (“Once CAC 44 has allocated bandwidth for a given connection on a link aggregation group, normal data trans-mission proceeds. The bandwidth allocations apply to the amount of guaranteed traffic carried on each link 30 in the group. (Note that different allocations and separate traffic management may apply to outgoing traffic generated by distributor 58 and incoming traffic, which is sent by nodes 24, 26, ... , and processed by collector 56.) The allocations also affect the bandwidth used on traces 52. The rate limiting function of concentrators 50 is set to allow for the traffic bandwidth that may be used on each of links 30 that feed the respective trace. As noted above, in allocating the band-width, CAC 44 ensures that the sum of the guaranteed bandwidth on all links sharing a given trace 52 is no greater than the trace bandwidth. The sum of the excess bandwidth allocated on the links, however, may exceed the trace bandwidth. In this case, the excess traffic is typically buff-ered as necessary, and is transmitted over the trace during intervals in which one or more of the links are not trans-mitting their guaranteed traffic levels and the trace has bandwidth available, or dropped if the buffer capacity is exceeded.”) Bruckman at [0074] (“Normal data transmission over the connection con-tinues unless and until a failure is detected on one of links 30 or line cards 34, at a failure detection step 68. Traffic manager 46 is informed of the failure, and notifies distribu-tor 58 accordingly to modify its hash function so that outgoing traffic is distributed over the remaining links in the group. Use of the protection parameter P in setting the bandwidth allocation ensures that (as long as no more than P links are out of service) there is sufficient bandwidth available for the connection on the

No.	’740 Patent Claim 1	Bruckman
		remaining links. It may also be necessary for the traffic manager to adjust the rate limiting function of concentrators 50, at a concentrator readjustment step 70, in order to deal with the increased incoming traffic on the remaining links. For example, if link L4 (FIG. 2) fails, the traffic on each of links Ll, L2 and L3 is expected to increase by 1/3, and the concentrator in LCl will have to deal with the resulting increase in traffic on the corresponding trace 52.”) Bruckman at Figure 1 (annotated)  FIG. 1 “communication network” “network node” “interface modules” “second group of second physical links” “first group of first physical links” 20 22 24 26 28 28 30 30 30 30 30 32 34 34 34 34 36 SYSTEM A SYSTEM B SYSTEM C NODE NODE LC LC LC LC TO CORE NETWORK

No.	’740 Patent Claim 1	Bruckman
1[e]	receiving a data frame having frame attributes sent between the communication network and the network node:	Bruckman discloses receiving a data frame having frame attributes sent between the communication network and the network node. For example, Bruckman discloses data frames containing frame information sent between the network and node. Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”)

No.	’740 Patent Claim 1	Bruckman
		Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”) Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”) Bruckman at Figure 1 (annotations added)

No.	’740 Patent Claim 1	Bruckman
		 FIG. 1 “communication network” “network node” 24 “interface modules” 20 22 36 SYSTEM A 34 SYSTEM B 30 LC SYSTEM C 26 30 34 30 LC NODE 28 TO CORE NETWORK 30 34 LC NODE 28 32 34 LC “second group of second physical links” “first group of first physical links”
1[f]:	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and	Bruckman discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. For example, Bruckman discloses distributing data frames using a hash function based on frame information. Bruckman further discloses transmitting the distributed data frame over the selected specific physical link and the selected specific port. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met.

No.	’740 Patent Claim 1	Bruckman
	a second physical link out of the second group; and	Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”) Bruckman at [0026] (“Additionally or alternatively, the method may include determining a measure of fluctuation that occurs in a rate of data transmission over the physical links when the data are distributed for transmission among the physical links, wherein the safety margin is further based on the measure of fluctuation. A safety factor F may be set respon-sively to the measure of

No.	’740 Patent Claim 1	Bruckman
		fluctuation, wherein the link band-width allocated to each of the links is a minimum of B and F*B/(N-P).”) Bruckman at [0031] (“conveying the data over the logical link by distrib-uting the data for transmission among the physical links in accordance with the allocated link bandwidth.”) Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”) Bruckman at [0040] (“a controller, which is adapted to receive a measure of fluctuation that occurs in a rate of transmission of the data over the physical links when the data are distributed for transmission among the physical links, and to allocate a link bandwidth on each of the physical communication links for use in conveying the data between the endpoints such that the allocated link bandwidth includes a predefined safety margin based on the measure of fluctuation, so that a sum of the allocated link bandwidth over the plurality of the parallel physical links is substantially greater than the guaranteed bandwidth of the connection;”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are

No.	’740 Patent Claim 1	Bruckman
		connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at Figure 2 (annotated)

<table>
<tr><th>No.</th><th>’740 Patent Claim 1</th><th>Bruckman</th></tr>
<tr><td></td><td></td><td>

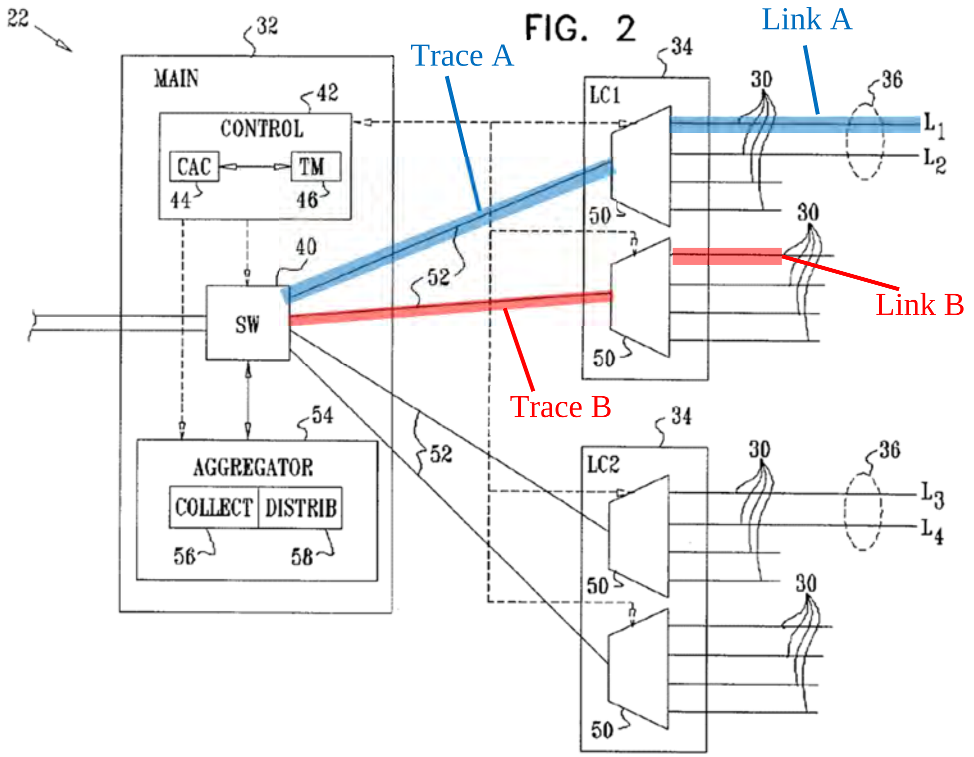
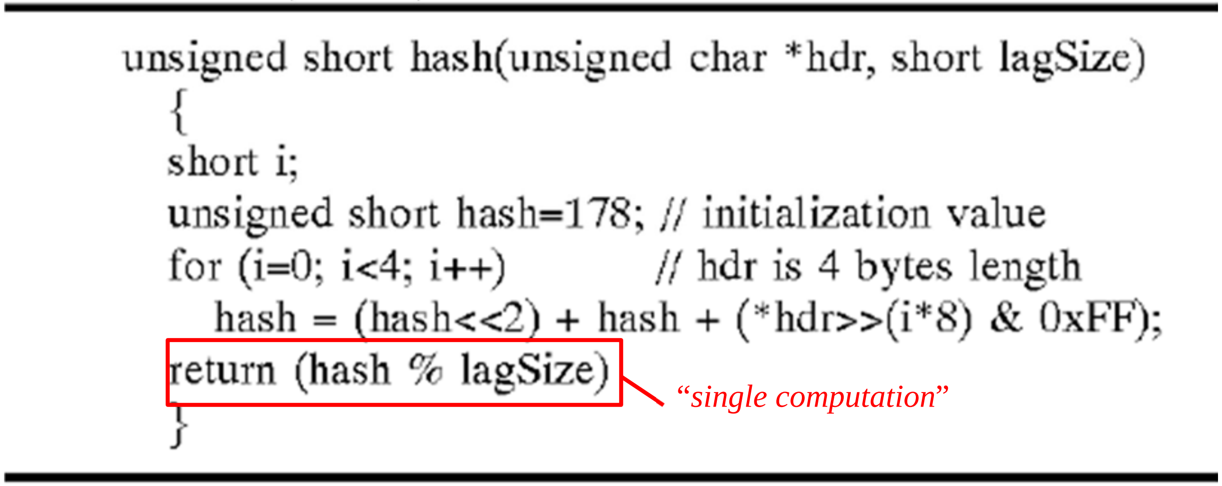


FIG. 2

Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria:

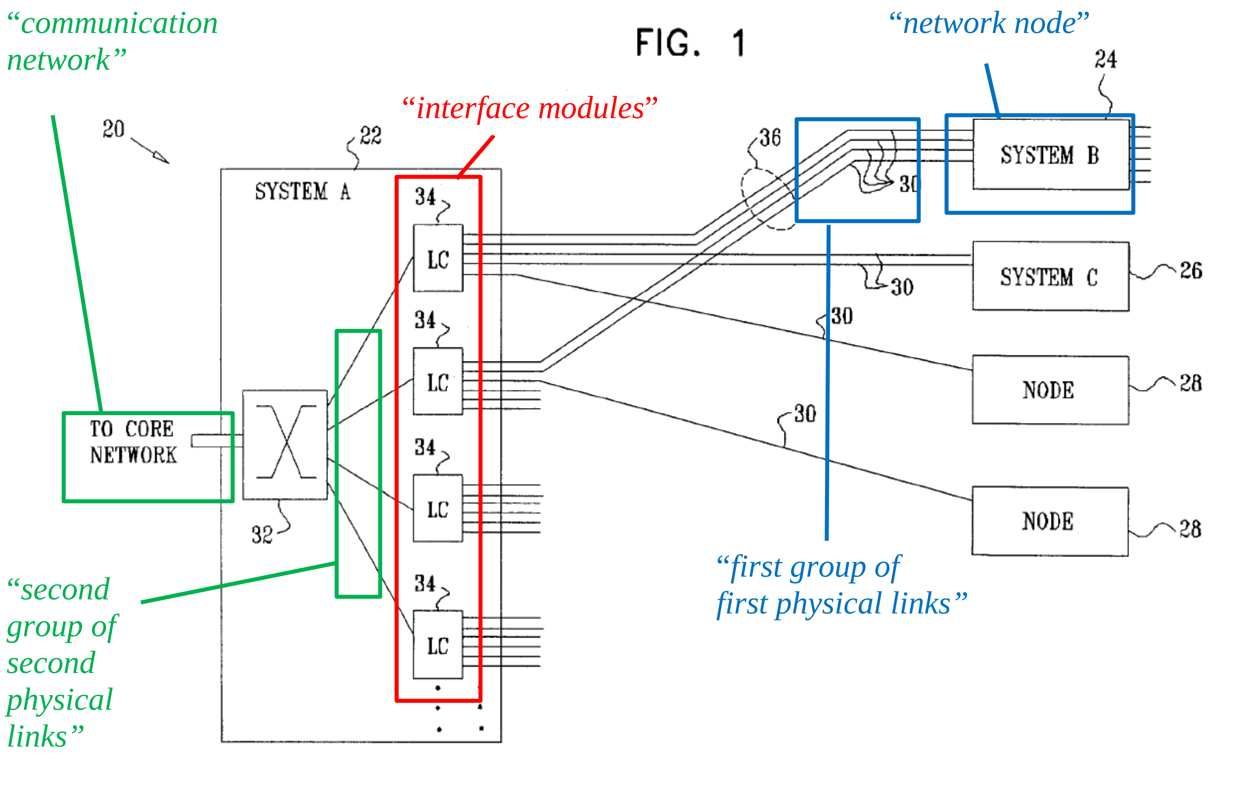
</td></tr>
</table>

No.	’740 Patent Claim 1	Bruckman

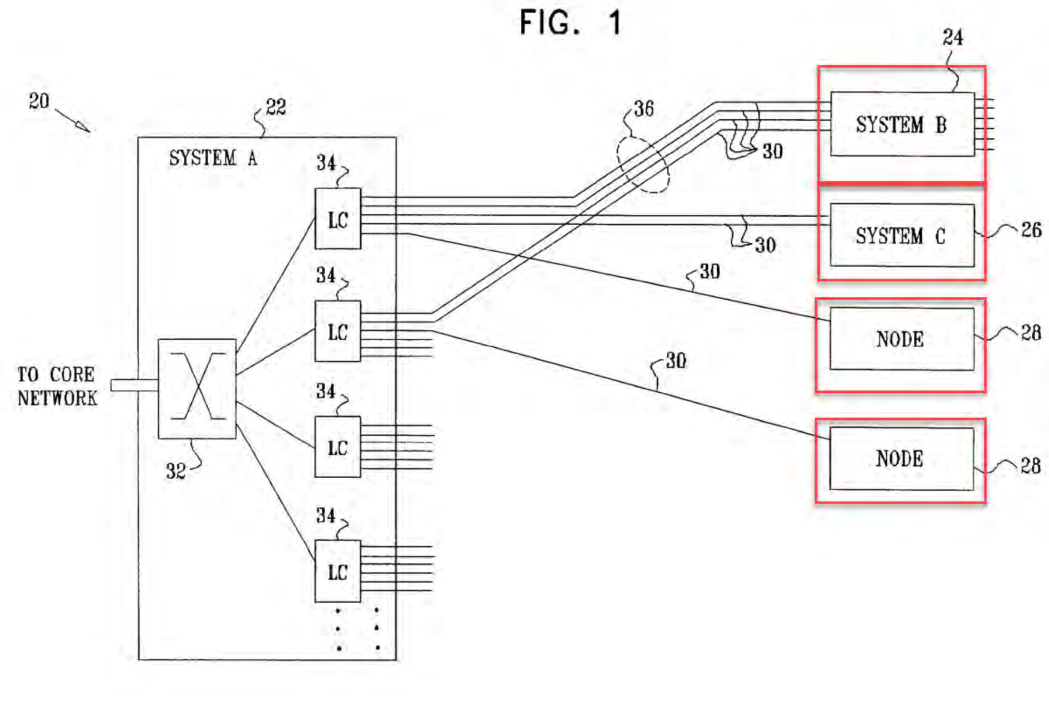
		[0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link.
		[0060] The hash function uses all the specified input data from the frame headers.
		[0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values
		[0062] The hash function generates very different hash values for similar data.”)
		
		Bruckman at Table 1 (annotated)
		
		`unsigned short hash(unsigned char *hdr, short lagSize)`
		`{`
		`short i;`
		`unsigned short hash=178; // initialization value`
		`for (i=0; i<4; i++)          // hdr is 4 bytes length`
		`   hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);`
		`return (hash % lagSize)`
		`}`

1[g]	sending the data frame over the selected first and second physical links,	Bruckman discloses sending the data frame over the selected first and second physical links. For example, Bruckman discloses establishing network connections and transmitting data frames over the determined ports via physical links and traces.

No.	’740 Patent Claim 1	Bruckman
		Bruckman at [0018]-[0022] (“There is therefore provided, in accordance with an embodiment of the present invention, a method for establishing a connection with a guaranteed bandwidth for trans-mitting data between first and second endpoints, the method including: [0019] defining a logical link including a plurality of parallel physical links between the endpoints; [0020] setting a protection policy to be applied to the logical link; [0021] allocating a link bandwidth on each of the physical communication links for use in conveying the data between the endpoints such that the allocated link bandwidth includes a predefined safety margin based on the protection policy, so that a sum of the allocated link bandwidth over the plurality of the parallel physical links is substantially greater than the guaranteed bandwidth of the connection; and [0022] conveying the data over the logical link by distrib-uting the data for transmission among the physical links in accordance with the allocated link bandwidth.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”) Bruckman at Figure 1 (annotations added)

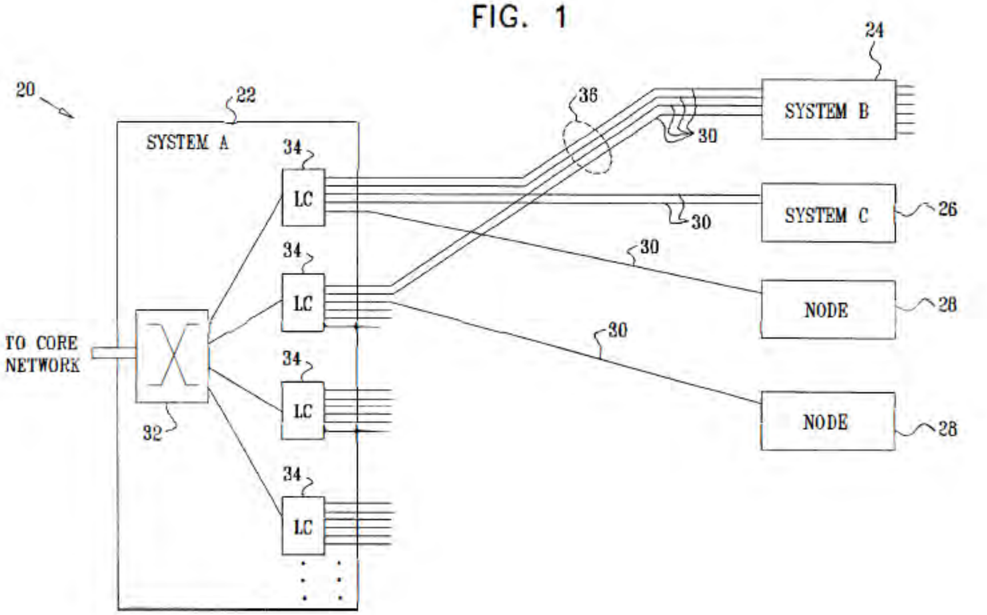
No.	’740 Patent Claim 1	Bruckman
		 FIG. 1 “communication network” “network node” “interface modules” “second group of second physical links” “first group of first physical links” 20 22 SYSTEM A 34 LC 34 LC 34 LC 34 LC TO CORE NETWORK 32 36 30 24 SYSTEM B SYSTEM C 26 30 30 NODE 28 30 NODE 28
1[h]	said sending comprising communicating along at least one of said bi-directional links.	Bruckman discloses said sending comprising communicating along at least one of said bi-directional links. *See supra at* 1[b], 1[d], 1[g].

No.	’740 Patent Claim 2	Bruckman
2[a]	The method according to claim 1, wherein the network node comprises a user node, and	Bruckman discloses the method according to claim 1, wherein the network node comprises a user node. For example, Bruckman discloses customer nodes connected in a network. Bruckman at [0013] (“Service level agreements between network service providers and customers commonly specify a certain com-mitted bandwidth, or committed information rate (CIR), which the service provider guarantees to provide to the customer at all times, regardless of bandwidth stress on the network. Additionally or alternatively, the agreement may specify an excess bandwidth, which is available to the customer when network traffic permits. The excess band-width is typically used by customers for lower-priority services, which do not require committed bandwidth. The network service provider may guarantee the customer a certain minimum excess bandwidth, or excess information rate (EIR), in order to avoid starvation of such services in case of bandwidth stress. In general, the bandwidth guaran-teed by a service provider, referred to as the peak informa-tion rate (PIR), may include either CIR, or EIR, or both CIR and EIR (in which case PIR=CIR+EIR). The term "guaran-teed bandwidth," as used in the context of the present patent application and in the claims, includes all these types of guaranteed bandwidth.”) Bruckman at [0047] (“FIG. 1 is a block diagram that schematically illus-trates elements of a communication system 20, in accor-dance with an embodiment of the present invention. In this example, central office equipment 22 communicates with customer nodes 24, 26, 28, ... , over physical links 30. Links 30 typically comprise full-duplex Ethernet links, such as lOBASE-n, lO0BASE-n or Gigabit Ethernet links, as are known in the art. (Alternatively, as noted above, other types of physical links may be used, such as ATM or PPP links.) Equipment 22 is configured to convey packet data traffic between the customer nodes and a network (which may be a metro network, access network, or other type of core network, for example). For this purpose, equipment 22 comprises a main switching card 32, which is connected to multiple line cards 34 that serve links 30. Details of the structure and operation of equipment 22 are shown below in FIG. 2 and are described with reference thereto.”) Bruckman at Figure 1

No.	’740 Patent Claim 2	Bruckman
		FIG. 1  20 22 SYSTEM A 34 LC 34 LC 34 LC 34 LC TO CORE NETWORK 32 36 24 SYSTEM B 30 30 SYSTEM C 26 30 NODE 28 30 NODE 28 Bruckman at [0048] (“Equipment 22 and certain customer nodes, such as nodes 24 and 26, are configured to serve as aggregation systems in accordance with the above-mentioned Clause 43 of the 802.3 standard. (Equipment 22 and nodes 24 and 26 are accordingly labeled as System A, B and C, respectively.) For example, an aggregation group 36 of four physical links is defined between equipment 22 and node 24. Another aggregation group of two physical links may be defined between equipment 22 and node 26. Each aggregation group (as well as each non-aggregated link 30) may serve multiple customer connections between the respective customer node and equipment 22.”) Bruckman at [0050] (“A Connection Admission Control entity (CAC) 44, typically a software process running on controller 42, man-ages the allocation of bandwidth in equipment 22. CAC

No.	’740 Patent Claim 2	Bruckman
		44 is responsible for ensuring that all connections between equipment 22 and customer nodes 24, 26, 28, ... , receive the amount of guaranteed bandwidth to which they are entitled, as well as for allocating any excess bandwidth available above the guaranteed minimum. For this purpose, CAC 44 maintains records that include:”) Bruckman at [0072] (“This is the bandwidth that the CAC allocates to each link in the link aggregation group. Traffic manager 46, however, may limit the actual data rate of each link to be no greater than B_{LINK}=min{B, F*B/(N-X)}, wherein Xis the number of failed links, X P. This latter limit prevents the link aggregation group from taking more than its fair share of bandwidth relative to other connections that share the same trace 52. In any case, the sum of guaranteed bandwidth on all connections sharing any given trace 52 may not exceed the trace capacity. CAC 44 may overbook the excess bandwidth remaining above the guaranteed limits, so that the total (peak) allocation exceeds the trace capacity. The connections on links 30, including any link aggregation groups, then compete for the remaining available bandwidth (typically in a weighted manner, based on the amount of excess bandwidth contracted for in the users' service level agreements, as is known in the art). By limiting the data rate of each link in the aggregation group to min{B, F*B/(N-X)}, rather than min{B, F*B/(N-P)}, traffic manager 46 leaves bandwidth available for other connections that share the same trace.”)
2[b]	wherein sending the data frame comprises establishing a communication service between the user node and the communication network.	Bruckman discloses wherein sending the data frame comprises establishing a communication service between the user node and the communication network. For example, Bruckman discloses a communication system in which data frames are sent between a network and customers nodes. Bruckman at Abstract (“A method for establishing a connection with a guaranteed bandwidth for transmitting data over a logical link that includes a plurality of parallel physical links between first and second endpoints. A link bandwidth is allocated on each of the physical communication links so as to include a predefined safety margin, based on either a failure protection policy, or a measure of fluctuation that occurs in a rate of data transmission over the physical links, or both. A sum of the allocated link bandwidth over the plurality of the

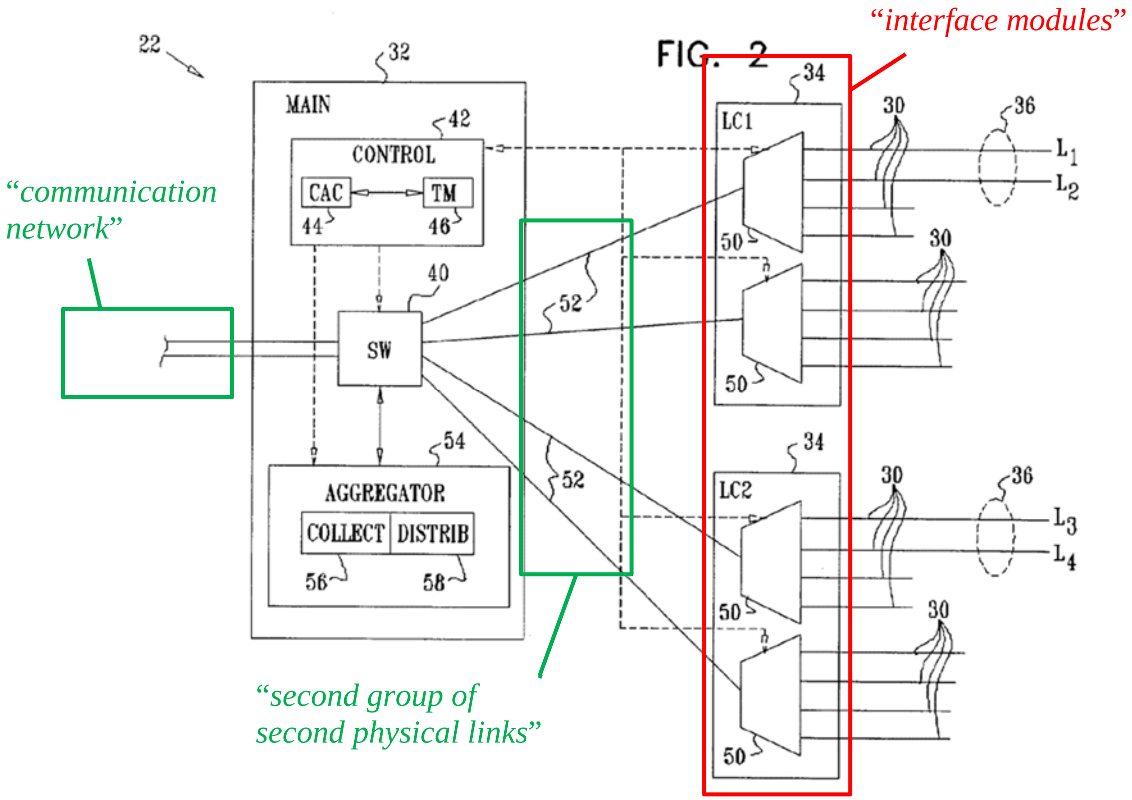
No.	’740 Patent Claim 2	Bruckman
		parallel physical links is substantially greater than the guaranteed bandwidth of the connection. The data are conveyed over the logical link by distributing the data for transmission among the physical links in accordance with the allocated link bandwidth.”) Bruckman at [0014] (“When aggregated links are used to serve a given customer, the service provider may allocate a certain frac-tion of the bandwidth on each of the physical links in the aggregation group so that the aggregated logical link pro-vides the total bandwidth guaranteed by the customer's service level agreement. Typically, however, the actual bandwidth consumed on each of the physical links fluctuates statistically due to the non-uniform distribution of load among the links in the aggregation group. Furthermore, if one of the physical links fails, the bandwidth consumed on the remaining links in the group will need to increase in order to maintain the minimum guaranteed total bandwidth on the aggregated logical link. Under these circumstances, the service provider may not be able to provide all customers with the minimum bandwidth guaranteed by their service level agreements.”) Bruckman at [0015] (“Embodiments of the present invention provide methods for bandwidth allocation in a link aggregation system to ensure that sufficient bandwidth will be available on the links in the group in order to meet service guarantees, notwithstanding load fluctuations and link failures. Safety margins are calculated, based on a measure of load fluctua-tion and on the level of protection to be provided (i.e., the worst-case number of link failures that must be tolerated by the system). These safety margins are applied in determining the bandwidth to be allocated for guaranteed services on each physical link in the aggregation group.”) Bruckman at [0047] (“FIG. 1 is a block diagram that schematically illus-trates elements of a communication system 20, in accor-dance with an embodiment of the present invention. In this example, central office equipment 22 communicates with customer nodes 24, 26, 28, ... , over physical links 30. Links 30 typically comprise full-duplex Ethernet links, such as lOBASE-n, lO0BASE-n or Gigabit Ethernet links, as are known in the art. (Alternatively, as noted above, other types of physical links may be used, such as ATM or PPP links.) Equipment 22 is configured to convey packet data traffic between the customer nodes and a network (which may be a metro network, access network, or other type of core network, for example). For this purpose, equipment 22 comprises a main switching card 32, which is connected to

No.	’740 Patent Claim 2	Bruckman
		multiple line cards 34 that serve links 30. Details of the structure and operation of equipment 22 are shown below in FIG. 2 and are described with reference thereto.”) FIG. 1  20 22 SYSTEM A 34 LC 34 LC 34 LC 34 LC TO CORE NETWORK 32 36 30 24 SYSTEM B 30 SYSTEM C 26 30 NODE 28 30 NODE 28 Bruckman at [0048] (“Equipment 22 and certain customer nodes, such as nodes 24 and 26, are configured to serve as aggregation systems in accordance with the above-mentioned Clause 43 of the 802.3 standard. (Equipment 22 and nodes 24 and 26 are accordingly labeled as System A, B and C, respectively.) For example, an aggregation group 36 of four physical links is defined between equipment 22 and node 24. Another aggregation group of two physical links may be defined between equipment 22 and node 26. Each aggregation group (as well as each non-aggregated link 30) may serve multiple customer connections between the respective customer node and equipment 22.”) Bruckman at [0050] (“A Connection Admission Control entity (CAC) 44, typically a software process running on controller 42, man-ages the allocation of bandwidth in equipment 22. CAC

No.	’740 Patent Claim 2	Bruckman
		44 is responsible for ensuring that all connections between equipment 22 and customer nodes 24, 26, 28, ... , receive the amount of guaranteed bandwidth to which they are entitled, as well as for allocating any excess bandwidth available above the guaranteed minimum. For this purpose, CAC 44 maintains records that include:”) Bruckman at [0073] (“Once CAC 44 has allocated bandwidth for a given connection on a link aggregation group, normal data trans-mission proceeds. The bandwidth allocations apply to the amount of guaranteed traffic carried on each link 30 in the group. (Note that different allocations and separate traffic management may apply to outgoing traffic generated by distributor 58 and incoming traffic, which is sent by nodes 24, 26, ... , and processed by collector 56.) The allocations also affect the bandwidth used on traces 52. The rate limiting function of concentrators 50 is set to allow for the traffic bandwidth that may be used on each of links 30 that feed the respective trace. As noted above, in allocating the band-width, CAC 44 ensures that the sum of the guaranteed bandwidth on all links sharing a given trace 52 is no greater than the trace bandwidth. The sum of the excess bandwidth allocated on the links, however, may exceed the trace bandwidth. In this case, the excess traffic is typically buff-ered as necessary, and is transmitted over the trace during intervals in which one or more of the links are not trans-mitting their guaranteed traffic levels and the trace has bandwidth available, or dropped if the buffer capacity is exceeded.”)

No.	’740 Patent Claim 3	Bruckman
3	The method according to claim 1, wherein the second physical links comprise backplane traces formed on a backplane to which the one or more	Bruckman discloses the method according to claim 1, wherein the second physical links comprise backplane traces formed on a backplane to which the one or more interface modules are coupled. For example, Bruckman discloses traces comprising a backplane which connect to line cards. *See supra* at Claim 1. Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the

No.	’740 Patent Claim 3	Bruckman
	interface modules are coupled.	line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”) Bruckman at [0049] (“FIG. 2 is a block diagram that schematically shows details of equipment 22, in accordance with an embodiment of the present invention. Main card 32 comprises a switching core 40, which switches traffic to and from line cards 34. Two line cards 34, labeled LCl and LC2, are shown in the figure. The operation of switch 40 is managed by a controller 42, typically an embedded microprocessor with suitable software for carrying out the functions described herein.”) Bruckman at Figure 2 (annotated)

No.	’740 Patent Claim 3	Bruckman
		 FIG. 2 “interface modules” “communication network” “second group of second physical links” 22 32 MAIN 42 CONTROL CAC TM 44 46 40 SW 52 52 54 AGGREGATOR COLLECT DISTRIB 56 58 34 LC1 34 LC2 30 36 L_1 L_2 L_3 L_4 50 Bruckman at [0056] (“Each line card 34 comprises one or more concen-trators 50, which comprise multiple ports that serve respec-tive links 30. The concentrators multiplex data traffic between links 30 and traces 52, which connect the concen-trators to switching core 40. Typically, main card 32 and line cards 34 are arranged in a card rack and plug into a printed circuit back plane, (not shown) which comprises traces 52. The bandwidth of each trace 52 may be less than

No.	’740 Patent Claim 3	Bruckman
		the total bandwidth available on links 30 that are connected to the respective concentrator 50, based on considerations of sta-tistical multiplexing. To prevent overloading of traces 52, concentrators 50 may limit the rate of incoming data admit-ted on each link 30 so that it remains between a predeter-mined minimum, which is determined by the guaranteed bandwidth of the connections on the link, and a maximum, which is determined by the peak bandwidth (guaranteed plus permitted excess bandwidth) of the connections on the link. A traffic manager 46, which may also be a software process on controller 42, receives information regarding the opera-tional status of links 30 (for example, link or equipment failures) and updates the data rate limits applied by concen-trators 50, based on the status information and the bandwidth allocations made by CAC 44.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”)

No.	’740 Patent Claim 4	Bruckman
4[preamble]	A method for communication, comprising:	Bruckman discloses a method for communication. *See supra* at 1[preamble].
4[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	Bruckman discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].
4[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Bruckman discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra* at 1[c].
4[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	Bruckman discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra* at 1[e].
4[d]	selecting, in a single computation based on at least one of the frame attributes,	Bruckman discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group.

No.	’740 Patent Claim 4	Bruckman
	a first physical link out of the first group and a second physical link out of the second group; and	*See supra* at 1[f].
4[e]	sending the data frame over the selected first and second physical links,	Bruckman discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].
4[f]	at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group.	Bruckman discloses at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group. For example, Bruckman discloses grouping the physical links connecting the ports and/or the traces into aggregated links in an Ethernet network. A person of ordinary skill in the art would understands other physical links could be aggregated as well. Bruckman at [0014] (“When aggregated links are used to serve a given customer, the service provider may allocate a certain frac-tion of the bandwidth on each of the physical links in the aggregation group so that the aggregated logical link pro-vides the total bandwidth guaranteed by the customer's service level agreement. Typically, however, the actual bandwidth consumed on each of the physical links fluctuates statistically due to the non-uniform distribution of load among the links in the aggregation group. Furthermore, if one of the physical links fails, the bandwidth consumed on the remaining links in the group will need to increase in order to maintain the minimum guaranteed total bandwidth on the aggregated logical link. Under these circumstances, the service provider may not be able to provide all customers with the minimum bandwidth guaranteed by their service level agreements.”) Bruckman at [0015] (“Embodiments of the present invention provide methods for bandwidth allocation in a link aggregation system to ensure that sufficient bandwidth will be available on

No.	’740 Patent Claim 4	Bruckman
		the links in the group in order to meet service guarantees, notwithstanding load fluctuations and link failures. Safety margins are calculated, based on a measure of load fluctua-tion and on the level of protection to be provided (i.e., the worst-case number of link failures that must be tolerated by the system). These safety margins are applied in determining the bandwidth to be allocated for guaranteed services on each physical link in the aggregation group.”) Bruckman at [0017] (“Although the embodiments described herein refer specifically to link aggregation in Ethernet (IEEE 802.3) networks, the principles of the present invention may simi-larly be used in other types of link aggregation, such as Inverse Multiplexing over ATM (IMA) and multi-link con-nections using the Point-to-Point (PPP) protocol.”) Bruckman at [0023] (“Typically, defining the logical link includes defin-ing a link aggregation group in accordance with IEEE standard 802.3. Alternatively, the physical links may include Asynchronous Transfer Mode (ATM) links, and defining the logical link may include grouping the physical links for Inverse Multiplexing over ATM (IMA) Further alternatively, defining the logical link may include defining a multi-link connection in accordance with a Point-to-Point (PPP) pro-tocol.”) Bruckman at [0048] (“Equipment 22 and certain customer nodes, such as nodes 24 and 26, are configured to serve as aggregation systems in accordance with the above-mentioned Clause 43 of the 802.3 standard. (Equipment 22 and nodes 24 and 26 are accordingly labeled as System A, B and C, respectively.) For example, an aggregation group 36 of four physical links is defined between equipment 22 and node 24. Another aggregation group of two physical links may be defined between equipment 22 and node 26. Each aggregation group (as well as each non-aggregated link 30) may serve multiple customer connections between the respective customer node and equipment 22.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be

No.	’740 Patent Claim 4	Bruckman
		carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”) Bruckman at [0072] (“This is the bandwidth that the CAC allocates to each link in the link aggregation group. Traffic manager 46, however, may limit the actual data rate of each link to be no greater than $B_{LINK=}$min{B, F*B/(N-X)}, wherein Xis the number of failed links, X P. This latter limit prevents the link aggregation group from taking more than its fair share of bandwidth relative to other connections that share the same trace 52. In any case, the sum of guaranteed bandwidth on all connections sharing any given trace 52 may not exceed the trace capacity. CAC 44 may overbook the excess bandwidth remaining above the guaranteed limits, so that the total

No.	’740 Patent Claim 4	Bruckman
		(peak) allocation exceeds the trace capacity. The connections on links 30, including any link aggregation groups, then compete for the remaining available bandwidth (typically in a weighted manner, based on the amount of excess bandwidth contracted for in the users' service level agreements, as is known in the art). By limiting the data rate of each link in the aggregation group to min{B, F*B/(N-X)}, rather than min{B, F*B/(N-P)}, traffic manager 46 leaves bandwidth available for other connections that share the same trace.”) Bruckman at [0075] (“Although the embodiments described above show a specific implementation of link aggregation bandwidth allocation and control in central office equipment 22, the methods used in this implementation may similarly be applied in a straightforward way in substantially any link aggregation system that operates in accordance with Clause 43 of the IEEE 802.3 standard. Furthermore, as noted above, the principles of the present invention may be applied, mu ta tis mutandis, in other types of link aggregation, such as Inverse Multiplexing over ATM (IMA) and multi-link con-nections using the Point-to-Point (PPP) protocol.”) Bruckman at Figure 1 (annotations added)

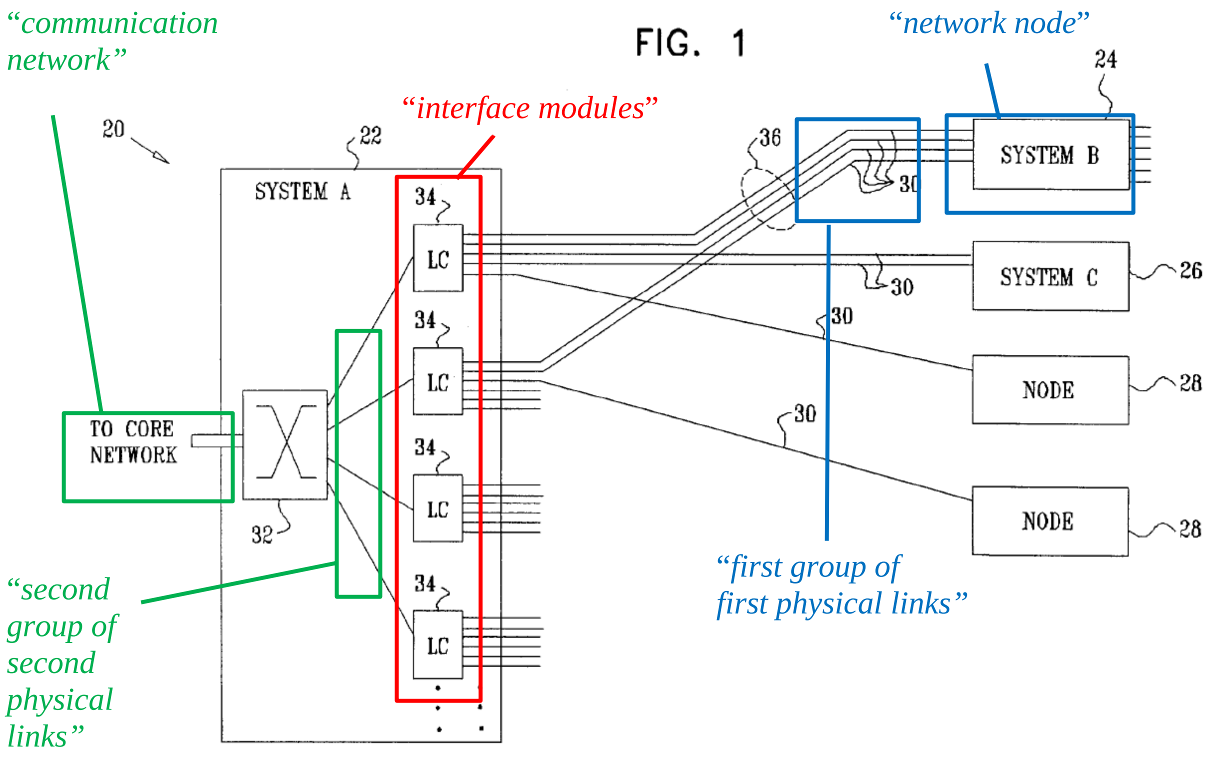
No.	’740 Patent Claim 4	Bruckman
		

FIG. 1

No.	’740 Patent Claim 5	Bruckman
5[preamble]	A method for communication, comprising:	Bruckman discloses a method for communication. *See supra at* 1[preamble].
5[a]	coupling a network node to one or more	Bruckman discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel.

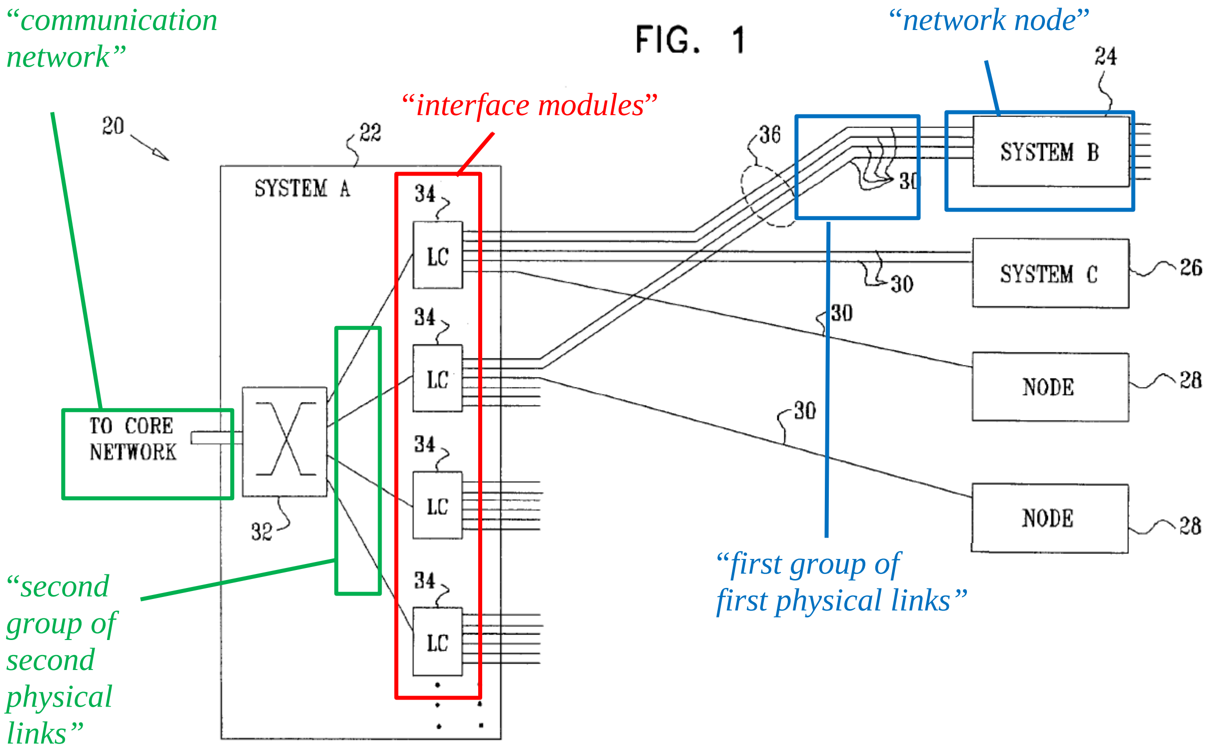
No.	’740 Patent Claim 5	Bruckman
	interface modules using a first group of first physical links arranged in parallel;	*See supra at* 1[a].
5[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Bruckman discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra at* 1[c].
5[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	Bruckman discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra at* 1[e].
5[d]	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	Bruckman discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. *See supra at* 1[f].

No.	’740 Patent Claim 5	Bruckman
5[e]	sending the data frame over the selected first and second physical links,	Bruckman discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].
5[f]	coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node.	Bruckman discloses coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node. For example, Bruckman discloses aggregating the physical links connecting line cards via ports to the node into aggregated links in an Ethernet network, in order to increase data bandwidth. Bruckman at [0002] (“Link aggregation is a technique by which a group of parallel physical links between two endpoints in a data network can be joined together into a single logical link. Traffic transmitted between the endpoints is distributed among the physical links in a manner that is transparent to the clients that send and receive the traffic. Link aggregation offers benefits of increased bandwidth, as well as increased availability, since the logical link can continue to function (possibly with reduced bandwidth) even when one of the physical links fails or is taken out of service.”) Bruckman at [0014] (“When aggregated links are used to serve a given customer, the service provider may allocate a certain frac-tion of the bandwidth on each of the physical links in the aggregation group so that the aggregated logical link pro-vides the total bandwidth guaranteed by the customer's service level agreement. Typically, however, the actual bandwidth consumed on each of the physical links fluctuates statistically due to the non-uniform distribution of load among the links in the aggregation group. Furthermore, if one of the physical links fails, the bandwidth consumed on the remaining links in the group will need to increase in order to maintain the minimum guaranteed total bandwidth on the aggregated logical link. Under these circumstances, the service provider may not be able to provide all customers with the minimum bandwidth guaranteed by their service level agreements.”)

No.	’740 Patent Claim 5	Bruckman
		Bruckman at [0015] (“Embodiments of the present invention provide methods for bandwidth allocation in a link aggregation system to ensure that sufficient bandwidth will be available on the links in the group in order to meet service guarantees, notwithstanding load fluctuations and link failures. Safety margins are calculated, based on a measure of load fluctua-tion and on the level of protection to be provided (i.e., the worst-case number of link failures that must be tolerated by the system). These safety margins are applied in determining the bandwidth to be allocated for guaranteed services on each physical link in the aggregation group.”) Bruckman at [0016] (“In other words, if the bandwidth guaranteed to a certain customer is B, and the customer is served by an aggregation group of N links, the minimum guaranteed bandwidth that could be allocated on each of the links would be B/N. The safety margins indicate the amount by which the bandwidth allocation must be increased above BIN in order to fulfill the guaranteed bandwidth requirement of the service level agreement. Any remaining excess bandwidth on the links in the aggregation group can be used for non-guaranteed, "best-effort" services.”) Bruckman at [0017] (“Although the embodiments described herein refer specifically to link aggregation in Ethernet (IEEE 802.3) networks, the principles of the present invention may simi-larly be used in other types of link aggregation, such as Inverse Multiplexing over ATM (IMA) and multi-link con-nections using the Point-to-Point (PPP) protocol.”) Bruckman at [0023] (“Typically, defining the logical link includes defin-ing a link aggregation group in accordance with IEEE standard 802.3. Alternatively, the physical links may include Asynchronous Transfer Mode (ATM) links, and defining the logical link may include grouping the physical links for Inverse Multiplexing over ATM (IMA) Further alternatively, defining the logical link may include defining a multi-link connection in accordance with a Point-to-Point (PPP) pro-tocol.”) Bruckman at [0048] (“Equipment 22 and certain customer nodes, such as nodes 24 and 26, are configured to serve as aggregation systems in accordance with the above-mentioned Clause 43 of the 802.3 standard. (Equipment 22 and nodes 24 and 26 are accordingly labeled as System A, B and C, respectively.) For example, an aggregation group 36 of four physical links is defined

No.	’740 Patent Claim 5	Bruckman
		between equipment 22 and node 24. Another aggregation group of two physical links may be defined between equipment 22 and node 26. Each aggregation group (as well as each non-aggregated link 30) may serve multiple customer connections between the respective customer node and equipment 22.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values

No.	’740 Patent Claim 5	Bruckman
		[0062] The hash function generates very different hash values for similar data.”) Bruckman at [0072] (“This is the bandwidth that the CAC allocates to each link in the link aggregation group. Traffic manager 46, however, may limit the actual data rate of each link to be no greater than $B_{LINK=}$min{B, F*B/(N-X)}, wherein Xis the number of failed links, X P. This latter limit prevents the link aggregation group from taking more than its fair share of bandwidth relative to other connections that share the same trace 52. In any case, the sum of guaranteed bandwidth on all connections sharing any given trace 52 may not exceed the trace capacity. CAC 44 may overbook the excess bandwidth remaining above the guaranteed limits, so that the total (peak) allocation exceeds the trace capacity. The connections on links 30, including any link aggregation groups, then compete for the remaining available bandwidth (typically in a weighted manner, based on the amount of excess bandwidth contracted for in the users' service level agreements, as is known in the art). By limiting the data rate of each link in the aggregation group to min{B, F*B/(N-X)}, rather than min{B, F*B/(N-P)}, traffic manager 46 leaves bandwidth available for other connections that share the same trace.”) Bruckman at [0075] (“Although the embodiments described above show a specific implementation of link aggregation bandwidth allocation and control in central office equipment 22, the methods used in this implementation may similarly be applied in a straightforward way in substantially any link aggregation system that operates in accordance with Clause 43 of the IEEE 802.3 standard. Furthermore, as noted above, the principles of the present invention may be applied, mu ta tis mutandis, in other types of link aggregation, such as Inverse Multiplexing over ATM (IMA) and multi-link con-nections using the Point-to-Point (PPP) protocol.”) Bruckman at Figure 1 (annotations added)

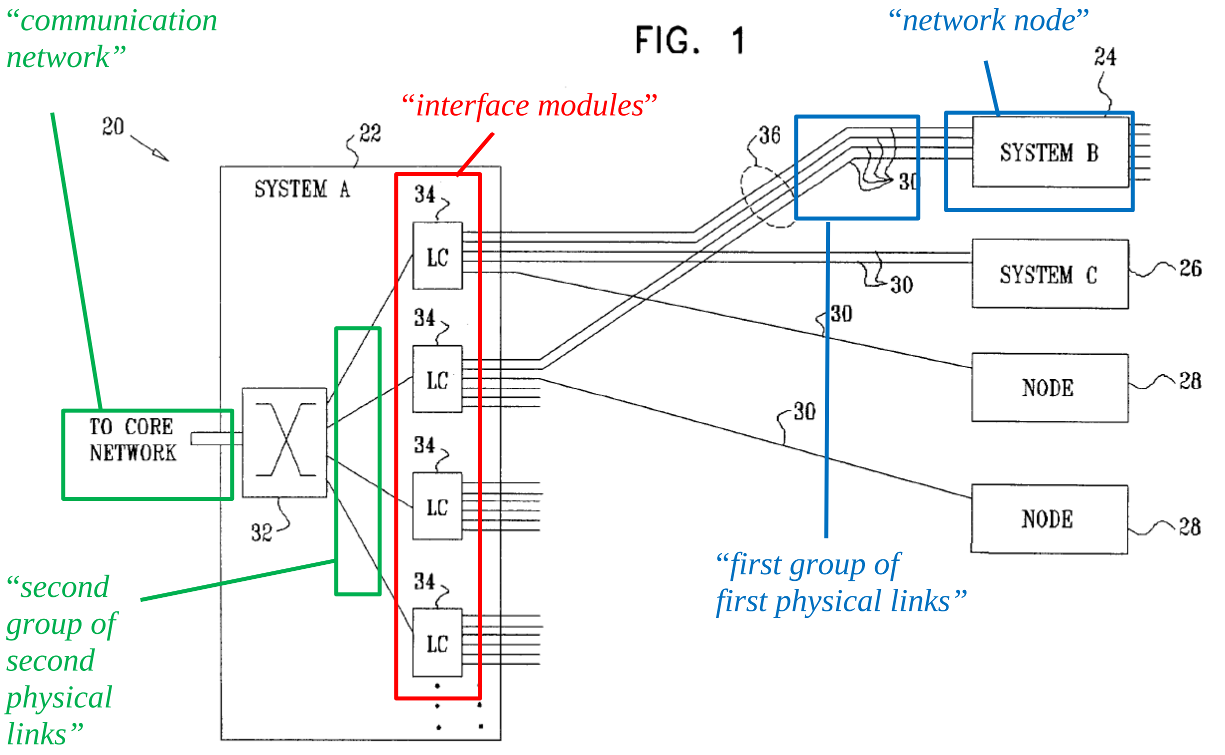
No.	’740 Patent Claim 5	Bruckman
		 FIG. 1 “communication network” “network node” 24 “interface modules” 20 22 36 SYSTEM A SYSTEM B 30 34 LC SYSTEM C 26 30 30 34 LC NODE 28 TO CORE NETWORK 30 34 LC NODE 28 32 “first group of first physical links” 34 “second group of second physical links” LC

No.	’740 Patent Claim 6	Bruckman
6	The method according to claim 1, wherein coupling each of the one or more interface	Bruckman discloses the method according to claim 1, wherein coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node.

No.	’740 Patent Claim 6	Bruckman
	modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node.	For example, Bruckman discloses a concentrator for multiplexing data between the physical links and traces and a distributor for distributing data frames arriving from the network to specific links and traces. *See supra* at Claim 1. Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”) Bruckman at [0056] (“Each line card 34 comprises one or more concen-trators 50, which comprise multiple ports that serve respec-tive links 30. The concentrators multiplex data traffic between links 30 and traces 52, which connect the concen-trators to switching core 40. Typically, main card 32 and line cards 34 are arranged in a card rack and plug into a printed circuit back plane, (not shown) which comprises traces 52. The bandwidth of each trace 52 may be less than the total bandwidth available on links 30 that are connected to the respective concentrator 50, based on considerations of sta-tistical multiplexing. To prevent overloading of traces 52, concentrators 50 may limit the rate of incoming data admit-ted on each link 30 so that it remains between a predeter-mined minimum, which is determined by the guaranteed bandwidth of the connections on the link, and a maximum, which is determined by the peak bandwidth (guaranteed plus permitted excess bandwidth) of the connections on the link. A traffic manager 46, which may also be a software process on controller 42, receives information regarding the opera-tional status of links 30 (for example, link or equipment failures) and updates the data rate limits applied by concen-trators 50, based on the status information and the bandwidth allocations made by CAC 44.”)

No.	’740 Patent Claim 6	Bruckman
		Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria:”) Bruckman at [0067] (“A similar problem may arise if there is a failure in a link in an aggregation group or in one of a number of line cards serving the aggregation group. In this case, to maintain the bandwidth allocation B made by CAC 44, each of the remaining links in the group must now carry, on average, B/(N-M) traffic, wherein M is the number of links in the group that are out of service. If only BIN has been allocated to each link, the remaining active links may not have sufficient bandwidth to continue to provide the bandwidth that has been guaranteed to the connections that they are required to carry. A similar problem arises with respect to loading of traces 52. For example, if there is a failure in LC2 or in one of links 30 in group 36 that connect to LC2, the trace connecting the multiplexer 50 in LCl will have to carry a substantially larger

No.	’740 Patent Claim 6	Bruckman
		share of the bandwidth, or even all of the bandwidth, that is allocated to the connection in ques-tion.”) Bruckman at [0073] (“Once CAC 44 has allocated bandwidth for a given connection on a link aggregation group, normal data trans-mission proceeds. The bandwidth allocations apply to the amount of guaranteed traffic carried on each link 30 in the group. (Note that different allocations and separate traffic management may apply to outgoing traffic generated by distributor 58 and incoming traffic, which is sent by nodes 24, 26, ... , and processed by collector 56.) The allocations also affect the bandwidth used on traces 52. The rate limiting function of concentrators 50 is set to allow for the traffic bandwidth that may be used on each of links 30 that feed the respective trace. As noted above, in allocating the band-width, CAC 44 ensures that the sum of the guaranteed bandwidth on all links sharing a given trace 52 is no greater than the trace bandwidth. The sum of the excess bandwidth allocated on the links, however, may exceed the trace bandwidth. In this case, the excess traffic is typically buff-ered as necessary, and is transmitted over the trace during intervals in which one or more of the links are not trans-mitting their guaranteed traffic levels and the trace has bandwidth available, or dropped if the buffer capacity is exceeded.”) Bruckman at Figure 1 (annotations added)

No.	’740 Patent Claim 6	Bruckman
		 FIG. 1

No.	’740 Patent Claim 7	Bruckman
7	The method according to claim 1, wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the	Bruckman discloses the method according to claim 1, wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the first and second physical links. For example, Bruckman discloses distributing data frames over physical links and traces including in response to fluctuations in data transmission rates.

No.	’740 Patent Claim 7	Bruckman
	first and second physical links.	*See supra* at Claim 1. Bruckman at Abstract (“A method for establishing a connection with a guaranteed bandwidth for transmitting data over a logical link that includes a plurality of parallel physical links between first and second endpoints. A link bandwidth is allocated on each of the physical communication links so as to include a predefined safety margin, based on either a failure protection policy, or a measure of fluctuation that occurs in a rate of data transmission over the physical links, or both. A sum of the allocated link bandwidth over the plurality of the parallel physical links is substantially greater than the guaranteed bandwidth of the connection. The data are conveyed over the logical link by distributing the data for transmission among the physical links in accordance with the allocated link bandwidth.”) Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.)

No.	’740 Patent Claim 7	Bruckman
		Bruckman at [0021] (“allocating a link bandwidth on each of the physical communication links for use in conveying the data between the endpoints such that the allocated link bandwidth includes a predefined safety margin based on the protection policy, so that a sum of the allocated link bandwidth over the plurality of the parallel physical links is substantially greater than the guaranteed bandwidth of the connection; and”) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”) Bruckman at [0026] (“Additionally or alternatively, the method may include determining a measure of fluctuation that occurs in a rate of data transmission over the physical links when the data are distributed for transmission among the physical links, wherein the safety margin is further based on the measure of fluctuation. A safety factor F may be set respon-sively to the measure of fluctuation, wherein the link band-width allocated to each of the links is a minimum of B and F*B/(N-P).”) Bruckman at [0029] (“determining a measure of fluctuation that occurs in a rate of transmission of the data over the physical links when the data are distributed for transmission among the physical links;”) Bruckman at [0031] (“conveying the data over the logical link by distrib-uting the data for transmission among the physical links in accordance with the allocated link bandwidth.”) Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first

No.	’740 Patent Claim 7	Bruckman
		and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”) Bruckman at [0040] (“a controller, which is adapted to receive a measure of fluctuation that occurs in a rate of transmission of the data over the physical links when the data are distributed for transmission among the physical links, and to allocate a link bandwidth on each of the physical communication links for use in conveying the data between the endpoints such that the allocated link bandwidth includes a predefined safety margin based on the measure of fluctuation, so that a sum of the allocated link bandwidth over the plurality of the parallel physical links is substantially greater than the guaranteed bandwidth of the connection;”) Bruckman at [0056] (“Each line card 34 comprises one or more concen-trators 50, which comprise multiple ports that serve respec-tive links 30. The concentrators multiplex data traffic between links 30 and traces 52, which connect the concen-trators to switching core 40. Typically, main card 32 and line cards 34 are arranged in a card rack and plug into a printed circuit back plane, (not shown) which comprises traces 52. The bandwidth of each trace 52 may be less than the total bandwidth available on links 30 that are connected to the respective concentrator 50, based on considerations of sta-tistical multiplexing. To prevent overloading of traces 52, concentrators 50 may limit the rate of incoming data admit-ted on each link 30 so that it remains between a predeter-mined minimum, which is determined by the guaranteed bandwidth of the connections on the link, and a maximum, which is determined by the peak bandwidth (guaranteed plus permitted excess bandwidth) of the connections on the link. A traffic manager 46, which may also be a software process on controller 42, receives information regarding the opera-tional status of links 30 (for example, link or equipment failures) and updates the data rate limits applied by concen-trators 50, based on the status information and the bandwidth allocations made by CAC 44.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded

No.	’740 Patent Claim 7	Bruckman
		processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”) Bruckman at [0066] (“When CAC 44 receives a request to open a con-nection with guaranteed bandwidth B over an aggregation group of N links, it might be assumed that the CAC should simply allocate bandwidth of BIN on each link. In practice, however, even if the hash function applied by distributor 58 meets the criteria outlined above, statistical variations in the traffic itself are likely to cause a larger portion of the traffic to be distributed to some of the links in the group than to others. In other words, some of the links may be required at times to carry group traffic with bandwidth substantially greater than B/N. As a result, these links may not

No.	’740 Patent Claim 7	Bruckman
		have sufficient capacity remaining to provide bandwidth that has been guaranteed to other connections that the CAC has committed to carry over these links. When an aggregation group extends over a number of concentrators 50 (as in the case of group 36), the traffic load on traces 52 may also be unbalanced. Overloading of traces 52 may likewise lead to a failure of system 22 to provide guaranteed bandwidth levels, in the distribution and/or the collection direction.”) Bruckman at [0072] (“This is the bandwidth that the CAC allocates to each link in the link aggregation group. Traffic manager 46, however, may limit the actual data rate of each link to be no greater than B_{LINK}=min{B, F*B/(N-X)}, wherein Xis the number of failed links, X P. This latter limit prevents the link aggregation group from taking more than its fair share of bandwidth relative to other connections that share the same trace 52. In any case, the sum of guaranteed bandwidth on all connections sharing any given trace 52 may not exceed the trace capacity. CAC 44 may overbook the excess bandwidth remaining above the guaranteed limits, so that the total (peak) allocation exceeds the trace capacity. The connections on links 30, including any link aggregation groups, then compete for the remaining available bandwidth (typically in a weighted manner, based on the amount of excess bandwidth contracted for in the users' service level agreements, as is known in the art). By limiting the data rate of each link in the aggregation group to min{B, F*B/(N-X)}, rather than min{B, F*B/(N-P)}, traffic manager 46 leaves bandwidth available for other connections that share the same trace.”) Bruckman at [0073] (“Once CAC 44 has allocated bandwidth for a given connection on a link aggregation group, normal data trans-mission proceeds. The bandwidth allocations apply to the amount of guaranteed traffic carried on each link 30 in the group. (Note that different allocations and separate traffic management may apply to outgoing traffic generated by distributor 58 and incoming traffic, which is sent by nodes 24, 26, ... , and processed by collector 56.) The allocations also affect the bandwidth used on traces 52. The rate limiting function of concentrators 50 is set to allow for the traffic bandwidth that may be used on each of links 30 that feed the respective trace. As noted above, in allocating the band-width, CAC 44 ensures that the sum of the guaranteed bandwidth on all links sharing a given trace 52 is no greater than the trace bandwidth. The sum of the excess bandwidth allocated on the links, however, may exceed the trace bandwidth. In this case, the excess traffic is typically buff-ered

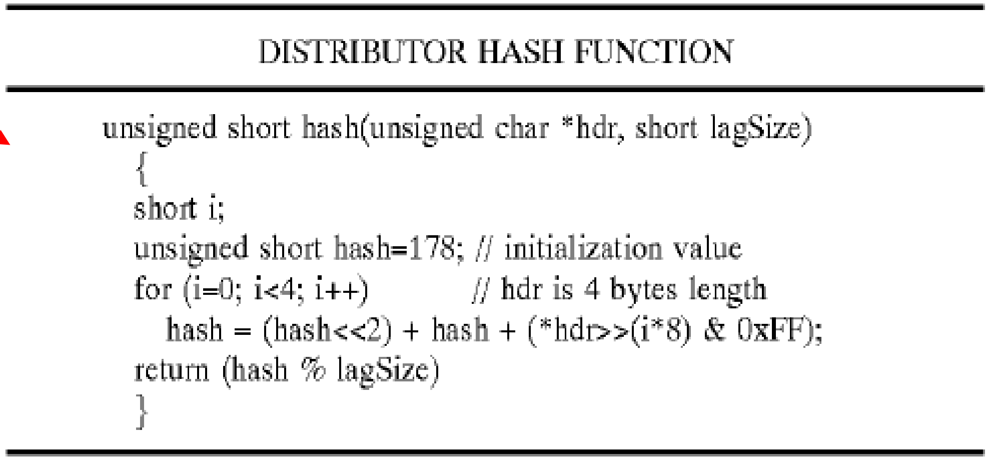
No.	’740 Patent Claim 7	Bruckman
		as necessary, and is transmitted over the trace during intervals in which one or more of the links are not trans-mitting their guaranteed traffic levels and the trace has bandwidth available, or dropped if the buffer capacity is exceeded.”) Bruckman at [0074] (“Normal data transmission over the connection con-tinues unless and until a failure is detected on one of links 30 or line cards 34, at a failure detection step 68. Traffic manager 46 is informed of the failure, and notifies distribu-tor 58 accordingly to modify its hash function so that outgoing traffic is distributed over the remaining links in the group. Use of the protection parameter P in setting the bandwidth allocation ensures that (as long as no more than P links are out of service) there is sufficient bandwidth available for the connection on the remaining links. It may also be necessary for the traffic manager to adjust the rate limiting function of concentrators 50, at a concentrator readjustment step 70, in order to deal with the increased incoming traffic on the remaining links. For example, if link L4 (FIG. 2) fails, the traffic on each of links Ll, L2 and L3 is expected to increase by 1/3, and the concentrator in LCl will have to deal with the resulting increase in traffic on the corresponding trace 52.”)

No.	’740 Patent Claim 8	Bruckman
8	The method according to claim 1, wherein selecting the first and second physical links comprises applying a mapping function to the at least one of the frame attributes.	Bruckman discloses the method according to claim 1, wherein selecting the first and second physical links comprises applying a mapping function to the at least one of the frame attributes. For example, Bruckman discloses distributing data frames over specific ports via physical links and traces based on a distribution algorithm using frame information. *See supra* at Claim 1. Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the

No.	’740 Patent Claim 8	Bruckman
		aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e.,

No.	’740 Patent Claim 8	Bruckman
		a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”) Bruckman at Table 1 (annotated)

No.	’740 Patent Claim 8	Bruckman

		hashing function “*mapping function*”
		
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		    hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)

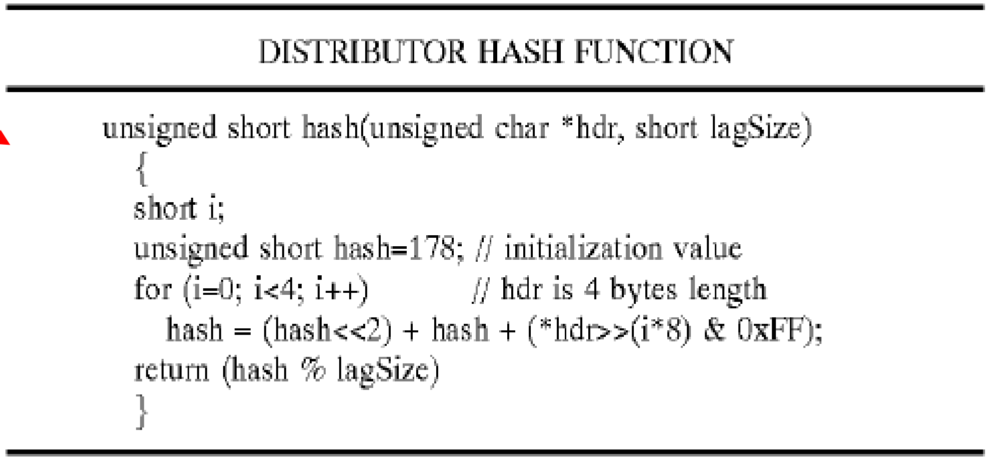
No.	’740 Patent Claim 9	Bruckman
9	The method according to claim 8, wherein applying the mapping function comprises	Bruckman discloses the method according to claim 8, wherein applying the mapping function comprises applying a hashing function. For example, Bruckman discloses applying a distributor hash function. *See supra* at Claim 1.

No.	’740 Patent Claim 9	Bruckman
	applying a hashing function.	Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In

No.	’740 Patent Claim 9	Bruckman
		the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”) Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I: Bruckman at Table 1 (annotated)

<table>
<tr><th>No.</th><th>’740 Patent Claim 9</th><th>Bruckman</th></tr>
<tr><td></td><td></td><td>

hashing function *“mapping function”*



DISTRIBUTOR HASH FUNCTION

```
unsigned short hash(unsigned char *hdr, short lagSize)
  {
  short i;
  unsigned short hash=178; // initialization value
  for (i=0; i<4; i++)          // hdr is 4 bytes length
     hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
  return (hash % lagSize)
  }
```

Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)

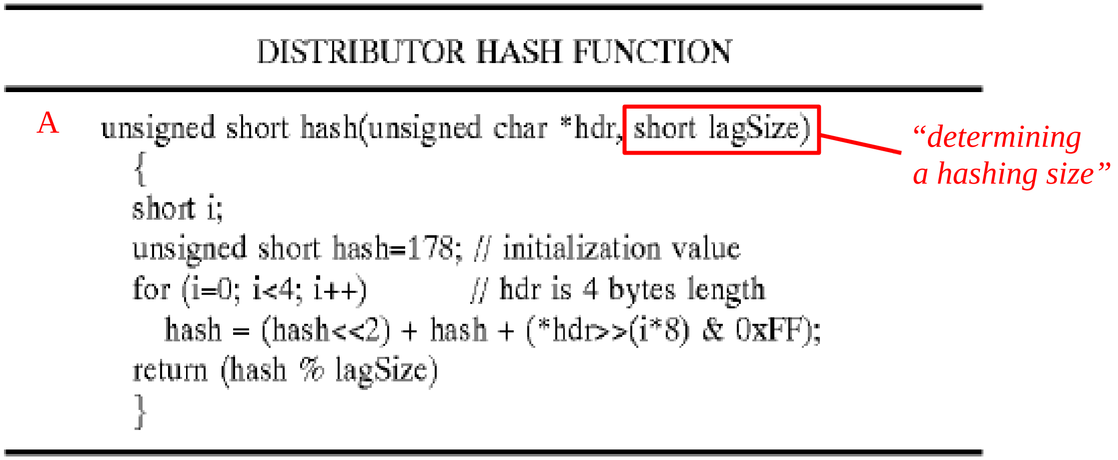
</td></tr>
</table>

No.	’740 Patent Claim 10	Bruckman
10[a]	The method according to claim 9, wherein applying the hashing function comprises determining a	Bruckman discloses the method according to claim 9, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second physical links. For example, Bruckman discloses applying a distributor hash function to the frame information which includes determining a number of the plurality of physical links.

No.	’740 Patent Claim 10	Bruckman
	hashing size responsively to a number of at least some of the first and second physical links,	*See supra at* Claim 9. Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0025] (“Typically, setting the protection policy includes determining a maximum number of the physical links that may fail while the logical link continues to provide at least the guaranteed bandwidth for the connection. In one embodiment, the guaranteed bandwidth is a bandwidth B, and the plurality of physical links consists of N links, and the maximum number is an integer P, and the link bandwidth allocated to each of the links is no less than B/(N-P). Conveying the data may further include managing the transmission of the data responsively to an actual number X of the physical links that have failed so that the guaranteed bandwidth on each of the links is limited to B/(N-X), X P, and an excess bandwidth on the physical links over the guaranteed bandwidth is available for other connections.”) Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”) Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:

<table>
<tr><th>No.</th><th>’740 Patent Claim 10</th><th>Bruckman</th></tr>
<tr><td></td><td></td><td>

Bruckman at Table 1 (annotated)



DISTRIBUTOR HASH FUNCTION

```
A   unsigned short hash(unsigned char *hdr, short lagSize)
      {
      short i;
      unsigned short hash=178; // initialization value
      for (i=0; i<4; i++)          // hdr is 4 bytes length
         hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
      return (hash % lagSize)
      }
```

“determining a hashing size”

Ex.1005, Table 1 (annotated); Ex.1003, *.**

Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)

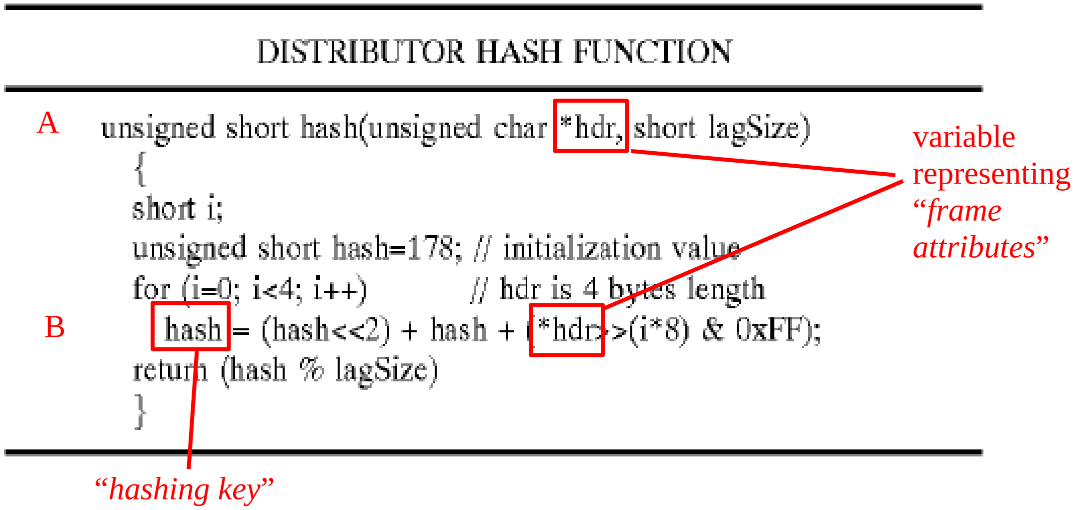
Bruckman at [0067] (“A similar problem may arise if there is a failure in a link in an aggregation group or in one of a number of line cards serving the aggregation group. In this case, to maintain the bandwidth allocation B made by CAC 44, each of the remaining links in the group must now carry, on average, B/(N-M) traffic, wherein M is the number of links in the group that are out of service. If only BIN has been allocated to each link, the remaining active links may not have sufficient bandwidth to continue to provide the bandwidth that has been guaranteed to the

</td></tr>
</table>

No.	’740 Patent Claim 10	Bruckman
		connections that they are required to carry. A similar problem arises with respect to loading of traces 52. For example, if there is a failure in LC2 or in one of links 30 in group 36 that connect to LC2, the trace connecting the multiplexer 50 in LCl will have to carry a substantially larger share of the bandwidth, or even all of the bandwidth, that is allocated to the connection in ques-tion.”) Bruckman at [0068] (“FIG. 3 is a flow chart that schematically illustrates a method for dealing with these problems of fluctuating bandwidth requirements, in accordance with an embodiment of the present invention. In order to provide sufficient bandwidth for failure protection, CAC 44 uses a safety margin based on a protection parameter P, which is assigned at a protection setting step 60. P represents the maximum number of links in the group that can be out of service while still permitting the aggregation group to provide a given connection with the bandwidth that has been guaranteed to the connection. CAC 44 will then allocate at least B/(N-P) bandwidth to each link in the group, so that if P links fail, the group still provides total bandwidth of (N-P)*B/(N-P)= B. Setting P=l is equivalent to 1:N protection, so that the group will be unaffected by failure of a single link. In the example of group 36, shown in FIG. 2, setting P=2 will give both facility and equipment protection, i.e., the group will be unaffected not only by failure of a link, but also by failure of one of line cards 34. In the extreme case, in which P=N-1, CAC 44 will allocate the full bandwidth B on each link in the group.”)
10[b]	applying the hashing function to the at least one of the frame attributes to produce a hashing key,	Bruckman discloses applying the hashing function to the at least one of the frame attributes to produce a hashing key. For example, Bruckman discloses applying a hash function to the frame “hdr” parameter to produce a hash value. Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is

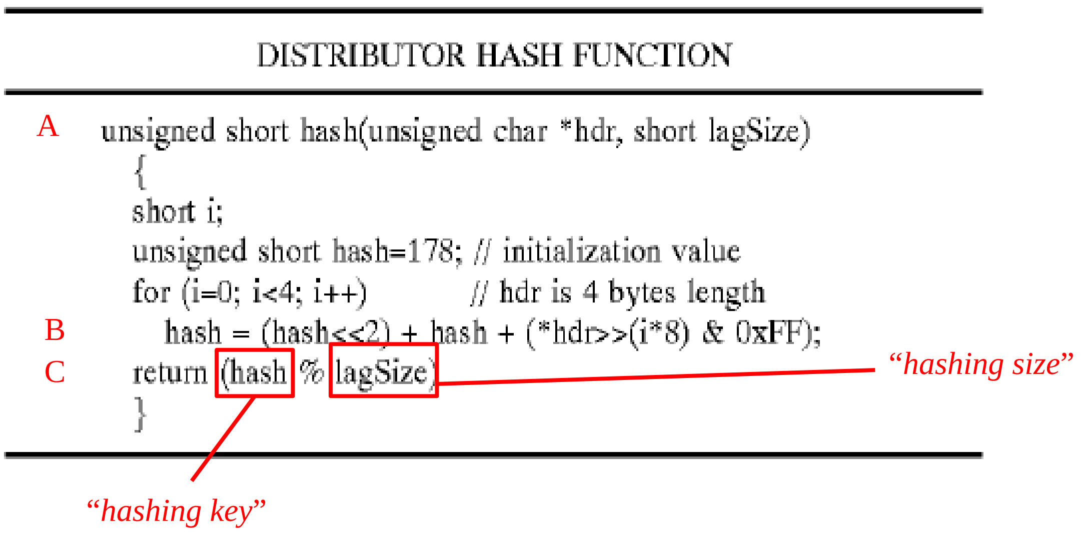
No.	’740 Patent Claim 10	Bruckman
		difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0025] (“Typically, setting the protection policy includes determining a maximum number of the physical links that may fail while the logical link continues to provide at least the guaranteed bandwidth for the connection. In one embodiment, the guaranteed bandwidth is a bandwidth B, and the plurality of physical links consists of N links, and the maximum number is an integer P, and the link bandwidth allocated to each of the links is no less than B/(N-P). Conveying the data may further include managing the transmission of the data responsively to an actual number X of the physical links that have failed so that the guaranteed bandwidth on each of the links is limited to B/(N-X), X P, and an excess bandwidth on the physical links over the guaranteed bandwidth is available for other connections.”) Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”) Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I: Bruckman at Table 1 (annotated)

No.	’740 Patent Claim 10	Bruckman

		
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		  hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		 A
		B
		variable representing “*frame attributes*”
		“*hashing key*”
		
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		
		Bruckman at [0067] (“A similar problem may arise if there is a failure in a link in an aggregation group or in one of a number of line cards serving the aggregation group. In this case, to maintain the bandwidth allocation B made by CAC 44, each of the remaining links in the group must now carry, on average, B/(N-M) traffic, wherein M is the number of links in the group that are out of service. If only BIN has been allocated to each link, the remaining active links may not have sufficient bandwidth to continue to provide the bandwidth that has been guaranteed to the connections that they are required to carry. A similar problem arises with respect to loading of traces 52. For example, if there is a failure in LC2 or in one of links 30 in group 36 that connect to LC2, the trace connecting the multiplexer 50 in LCl will have to carry a substantially larger

No.	’740 Patent Claim 10	Bruckman
		share of the bandwidth, or even all of the bandwidth, that is allocated to the connection in ques-tion.”) Bruckman at [0068] (“FIG. 3 is a flow chart that schematically illustrates a method for dealing with these problems of fluctuating bandwidth requirements, in accordance with an embodiment of the present invention. In order to provide sufficient bandwidth for failure protection, CAC 44 uses a safety margin based on a protection parameter P, which is assigned at a protection setting step 60. P represents the maximum number of links in the group that can be out of service while still permitting the aggregation group to provide a given connection with the bandwidth that has been guaranteed to the connection. CAC 44 will then allocate at least B/(N-P) bandwidth to each link in the group, so that if P links fail, the group still provides total bandwidth of (N-P)*B/(N-P)= B. Setting P=l is equivalent to 1:N protection, so that the group will be unaffected by failure of a single link. In the example of group 36, shown in FIG. 2, setting P=2 will give both facility and equipment protection, i.e., the group will be unaffected not only by failure of a link, but also by failure of one of line cards 34. In the extreme case, in which P=N-1, CAC 44 will allocate the full bandwidth B on each link in the group.”)
10[c]	calculating a modulo of a division operation of the hashing key by the hashing size, and	Bruckman discloses calculating a modulo of a division operation of the hashing key by the hashing size. For example, Bruckman discloses a division operation (%) applied to the hash value and lagSize to determine the distribution of data frames. Bruckman at [0015] (“Embodiments of the present invention provide methods for bandwidth allocation in a link aggregation system to ensure that sufficient bandwidth will be available on the links in the group in order to meet service guarantees, notwithstanding load fluctuations and link failures. Safety margins are calculated, based on a measure of load fluctua-tion and on the level of protection to be provided (i.e., the worst-case number of link failures that must be tolerated by the system). These safety margins are applied in determining the bandwidth to be allocated for guaranteed services on each physical link in the aggregation group.”)

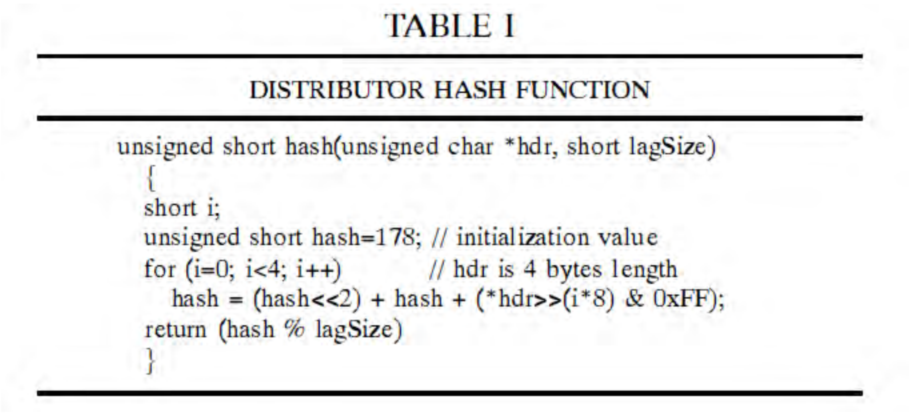
No.	’740 Patent Claim 10	Bruckman

		Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:
		Bruckman at Table 1 (annotated)
		
		DISTRIBUTOR HASH FUNCTION
		`unsigned short hash(unsigned char *hdr, short lagSize)`
		`{`
		`short i;`
		`unsigned short hash=178; // initialization value`
		`for (i=0; i<4; i++)          // hdr is 4 bytes length`
		`    hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);`
		`return (hash % lagSize)`
		`}`
		 A
		B
		C
		“hashing size”
		“hashing key”
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		Bruckman at [0068] (“FIG. 3 is a flow chart that schematically illustrates a method for dealing with these problems of fluctuating bandwidth requirements, in accordance with an embodiment of the present invention. In order to provide sufficient bandwidth for failure protection, CAC 44 uses a safety margin based on a protection parameter P, which is assigned at a protection setting

No.	’740 Patent Claim 10	Bruckman
		step 60. P represents the maximum number of links in the group that can be out of service while still permitting the aggregation group to provide a given connection with the bandwidth that has been guaranteed to the connection. CAC 44 will then allocate at least B/(N-P) bandwidth to each link in the group, so that if P links fail, the group still provides total bandwidth of (N-P)*B/(N-P)= B. Setting P=l is equivalent to 1:N protection, so that the group will be unaffected by failure of a single link. In the example of group 36, shown in FIG. 2, setting P=2 will give both facility and equipment protection, i.e., the group will be unaffected not only by failure of a link, but also by failure of one of line cards 34. In the extreme case, in which P=N-1, CAC 44 will allocate the full bandwidth B on each link in the group.”) Bruckman at [0069] (“In order to account for statistical fluctuations in the bandwidth consumed on the different links in the aggrega-tion group, a measure of these fluctuations is determined, at a deviation calculation step 62. For example, the standard deviation provides a useful a measure of the fluctuation of the actual bandwidth relative to the mean BIN (or B/(N-P)). It may be found by on-line measurement of the actual traffic flow on the links in the group or by off-line simulation or analytical calculation. Alternatively, the utilization of each link in the link aggregation group may be measured, and these measurements may be used to calculate the average utilization of the links and the actual maximum difference between the utilization of the most-loaded link and the average. In general, a connection characterized by long conversations will tend to have large fluctuations, since each conversation must be conveyed in its entirety over the same link. Connections carrying many short, different conversa-tions will generally have small fluctuations.”) Bruckman at [0070] (“To provide sufficient excess bandwidth for these statistical fluctuations, CAC 44 uses a safety margin based on a fluctuation factor F, which is assigned at a fluctuation setting step 64. F is calculated based on the standard deviation or other measure of fluctuation found at step 62. CAC 44 will then allocate at least F*BIN bandwidth to each link in the aggregation group. For example, for a given standard deviation a, the value F=1+3a will provide suffi-cient bandwidth to cover nearly all the statistical fluctuations on the links. As another example, F may be given by the actual, measured maximum difference between the utiliza-tion of the most-loaded link and the average utilization. Larger or smaller factors may be used, depending

No.	’740 Patent Claim 10	Bruckman
		on service level agreements and other constraints. Clearly, however, F N, since the total bandwidth allocated on any one of the links in the group need not be any greater than the guaran-teed total bandwidth B for the connection in question.”
10[d]	selecting the first and second physical links responsively to the modulo.	Bruckman discloses selecting the first and second physical links responsively to the modulo. For example, Bruckman discloses distributing data frames over physical links and traces based on a hash function involving a division operation (%). Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0025] (“Typically, setting the protection policy includes determining a maximum number of the physical links that may fail while the logical link continues to provide at least the guaranteed bandwidth for the connection. In one embodiment, the guaranteed bandwidth is a bandwidth B, and the plurality of physical links consists of N links, and the maximum number is an integer P, and the link bandwidth allocated to each of the links is no less than B/(N-P). Conveying the data may further include managing the transmission of the data responsively to an actual number X of the physical links that have failed so that the guaranteed bandwidth on each of the links is limited to B/(N-X), X P, and an excess bandwidth on the physical links over the guaranteed bandwidth is available for other connections.”) Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the

No.	’740 Patent Claim 10	Bruckman

		safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”)
		
		Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:
		
		TABLE I
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		”)
		
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		
		Bruckman at [0067] (“A similar problem may arise if there is a failure in a link in an aggregation group or in one of a number of line cards serving the aggregation group. In this case, to maintain the bandwidth allocation B made by CAC 44, each of the remaining links in the group must now carry, on average, B/(N-M) traffic, wherein M is the number of links in the group that are out of service. If only BIN has been allocated to each link, the remaining active links may not have sufficient bandwidth to continue to provide the bandwidth that has been guaranteed to the

No.	’740 Patent Claim 10	Bruckman
		connections that they are required to carry. A similar problem arises with respect to loading of traces 52. For example, if there is a failure in LC2 or in one of links 30 in group 36 that connect to LC2, the trace connecting the multiplexer 50 in LCl will have to carry a substantially larger share of the bandwidth, or even all of the bandwidth, that is allocated to the connection in ques-tion.”) Bruckman at [0068] (“FIG. 3 is a flow chart that schematically illustrates a method for dealing with these problems of fluctuating bandwidth requirements, in accordance with an embodiment of the present invention. In order to provide sufficient bandwidth for failure protection, CAC 44 uses a safety margin based on a protection parameter P, which is assigned at a protection setting step 60. P represents the maximum number of links in the group that can be out of service while still permitting the aggregation group to provide a given connection with the bandwidth that has been guaranteed to the connection. CAC 44 will then allocate at least B/(N-P) bandwidth to each link in the group, so that if P links fail, the group still provides total bandwidth of (N-P)*B/(N-P)= B. Setting P=l is equivalent to 1:N protection, so that the group will be unaffected by failure of a single link. In the example of group 36, shown in FIG. 2, setting P=2 will give both facility and equipment protection, i.e., the group will be unaffected not only by failure of a link, but also by failure of one of line cards 34. In the extreme case, in which P=N-1, CAC 44 will allocate the full bandwidth B on each link in the group.”)

No.	’740 Patent Claim 11	Bruckman
11	The method according to claim 10, wherein selecting the first and second physical links responsively to the modulo comprises	Bruckman discloses the method according to claim 10, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo. For example, Bruckman discloses distributing data frames over physical links and traces based on a division operation in the hash function, involving specific byte lengths of the frame information.

<table>
<tr><th>No.</th><th>’740 Patent Claim 11</th><th>Bruckman</th></tr>
<tr><td></td><td>selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo.</td><td>

See supra Claim 10.

Bruckman at Figure 2 (annotated)

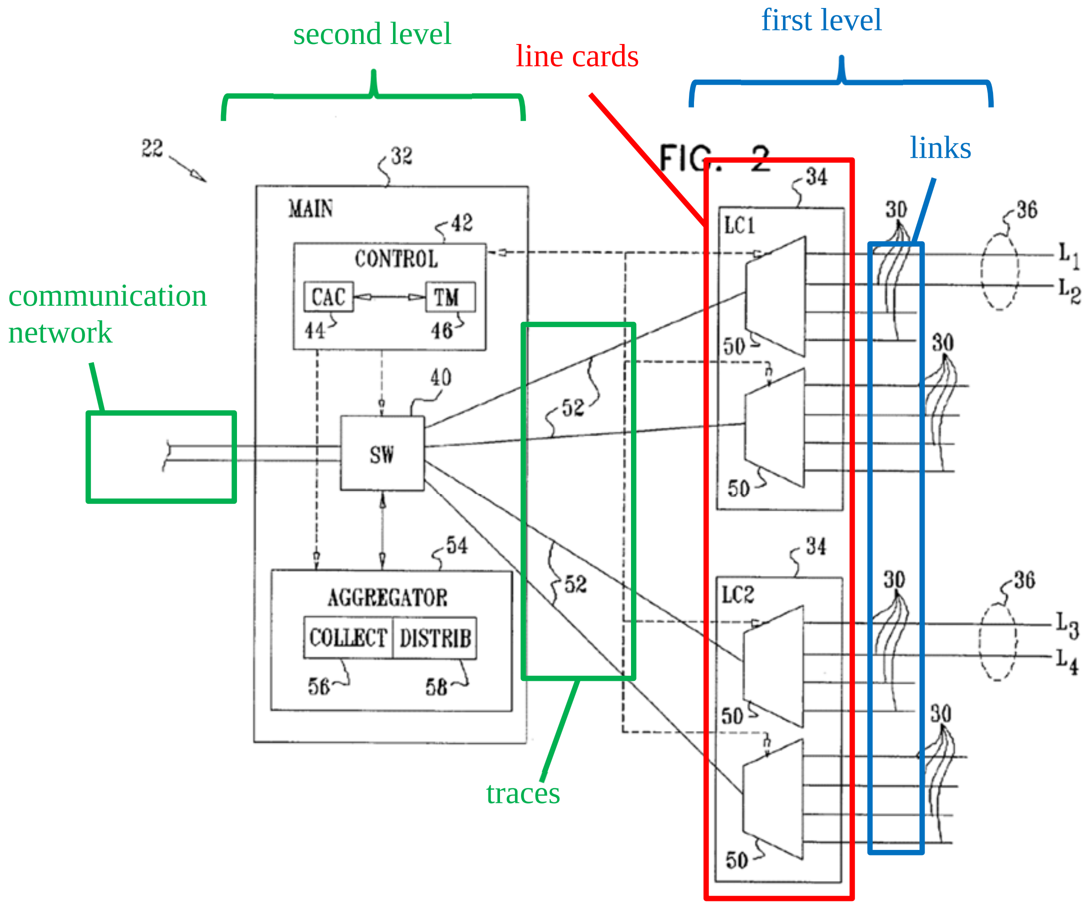
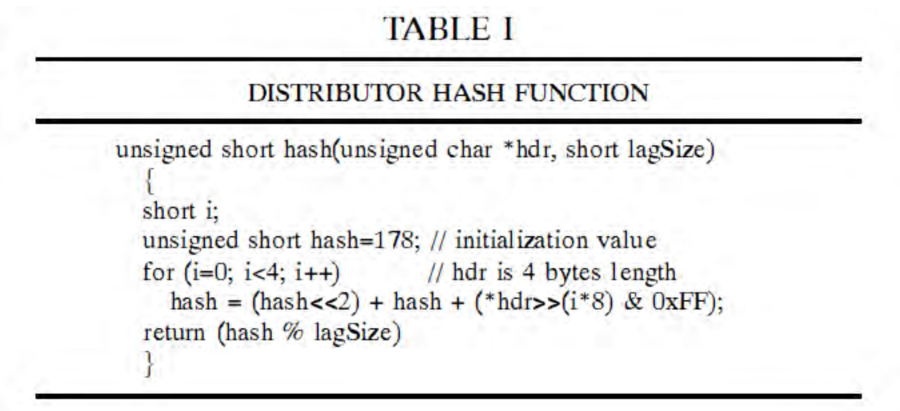


FIG. 2

</td></tr>
</table>

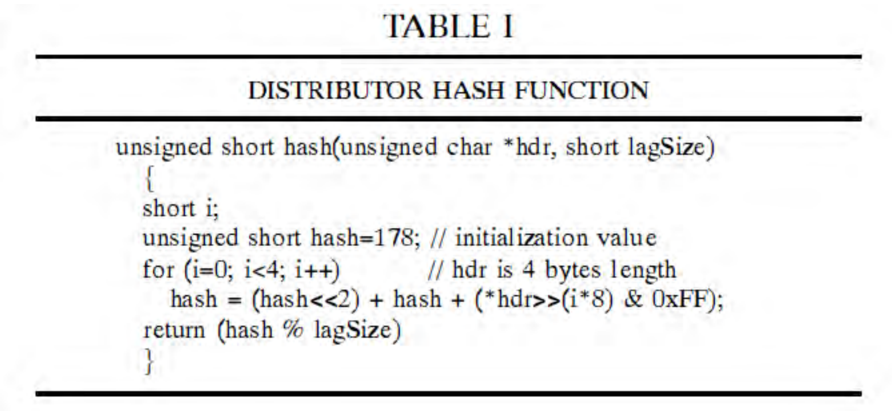
No.	’740 Patent Claim 11	Bruckman

		Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:
		
		TABLE I
		DISTRIBUTOR HASH FUNCTION
		`unsigned short hash(unsigned char *hdr, short lagSize)`
		`  {`
		`  short i;`
		`  unsigned short hash=178; // initialization value`
		`  for (i=0; i<4; i++)          // hdr is 4 bytes length`
		`    hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);`
		`  return (hash % lagSize)`
		`  }`
		”)
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)

No.	’740 Patent Claim 12	Bruckman
12	The method according to claim 1, wherein the at least one of the frame attributes comprises at least one of a layer 2	Bruckman discloses the method according to claim 1, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port.

No.	’740 Patent Claim 12	Bruckman
	header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port.	For example, Bruckman discloses frame information including header fields, source addresses, destination addresses, ports, etc. Bruckman specifically discloses Ethernet frame information including source MAC address, destination MAC address, reception port, type of destination address, Ethernet Length/Type value, and higher layer protocol information. *See supra* Claim 1. Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”)

No.	’740 Patent Claim 12	Bruckman

		Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria:
		[0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link.
		[0060] The hash function uses all the specified input data from the frame headers.
		[0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values
		[0062] The hash function generates very different hash values for similar data.”)
		
		Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:
		
		TABLE I
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		”)

No.	’740 Patent Claim 12	Bruckman
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)

No.	’740 Patent Claim 13	Bruckman
13[preamble]	A method for communication, comprising:	Bruckman discloses a method for communication. *See supra at* 1[preamble].
13[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	Bruckman discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].
13[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Bruckman discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra at* 1[c].

No.	’740 Patent Claim 13	Bruckman
13[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	Bruckman discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra at* 1[e].
13[d]	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	Bruckman discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. *See supra at* 1[f].
13[e]	sending the data frame over the selected first and second physical links,	Bruckman discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].
13[f]	coupling the network node to the one or more interface modules and	Bruckman discloses coupling the network node to the one or more interface modules. *See supra at* 1[a].
13[g]	coupling each of the one or more interface modules to the	Bruckman discloses coupling each of the one or more interface modules to the communication network. *See supra at* 1[c].

No.	’740 Patent Claim 13	Bruckman
	communication network comprising	
13[h]	specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node, and	Bruckman discloses specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node. For example, Bruckman discloses specifying certain committed, excess, and guaranteed bandwidths, including CIR, EIR, and PIR, respectively. Bruckman at [0013] (“Service level agreements between network service providers and customers commonly specify a certain com-mitted bandwidth, or committed information rate (CIR), which the service provider guarantees to provide to the customer at all times, regardless of bandwidth stress on the network. Additionally or alternatively, the agreement may specify an excess bandwidth, which is available to the customer when network traffic permits. The excess band-width is typically used by customers for lower-priority services, which do not require committed bandwidth. The network service provider may guarantee the customer a certain minimum excess bandwidth, or excess information rate (EIR), in order to avoid starvation of such services in case of bandwidth stress. In general, the bandwidth guaran-teed by a service provider, referred to as the peak informa-tion rate (PIR), may include either CIR, or EIR, or both CIR and EIR (in which case PIR=CIR+EIR). The term "guaran-teed bandwidth," as used in the context of the present patent application and in the claims, includes all these types of guaranteed bandwidth.”)
13[i]	allocating a bandwidth for the communication service over the first and second physical links	Bruckman discloses allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements. For examples, Bruckman discloses allocating bandwidth over the physical links and traces based on bandwidth requirements, margins, and fluctuations.

No.	’740 Patent Claim 13	Bruckman
	responsively to the bandwidth requirements.	Bruckman at [0014] (“When aggregated links are used to serve a given customer, the service provider may allocate a certain frac-tion of the bandwidth on each of the physical links in the aggregation group so that the aggregated logical link pro-vides the total bandwidth guaranteed by the customer's service level agreement. Typically, however, the actual bandwidth consumed on each of the physical links fluctuates statistically due to the non-uniform distribution of load among the links in the aggregation group. Furthermore, if one of the physical links fails, the bandwidth consumed on the remaining links in the group will need to increase in order to maintain the minimum guaranteed total bandwidth on the aggregated logical link. Under these circumstances, the service provider may not be able to provide all customers with the minimum bandwidth guaranteed by their service level agreements.”) Bruckman at [0015] (“Embodiments of the present invention provide methods for bandwidth allocation in a link aggregation system to ensure that sufficient bandwidth will be available on the links in the group in order to meet service guarantees, notwithstanding load fluctuations and link failures. Safety margins are calculated, based on a measure of load fluctua-tion and on the level of protection to be provided (i.e., the worst-case number of link failures that must be tolerated by the system). These safety margins are applied in determining the bandwidth to be allocated for guaranteed services on each physical link in the aggregation group.”) Bruckman at [0016] (In other words, if the bandwidth guaranteed to a certain customer is B, and the customer is served by an aggregation group of N links, the minimum guaranteed bandwidth that could be allocated on each of the links would be B/N. The safety margins indicate the amount by which the bandwidth allocation must be increased above BIN in order to fulfill the guaranteed bandwidth requirement of the service level agreement. Any remaining excess bandwidth on the links in the aggregation group can be used for non-guaranteed, "best-effort" services.”) Bruckman at [0021] (“allocating a link bandwidth on each of the physical communication links for use in conveying the data between the endpoints such that the allocated link bandwidth includes a predefined safety margin based on the protection policy, so that a sum of the allocated link bandwidth over the plurality of the parallel physical links is substantially greater than the guaranteed bandwidth of the connection; and”)

No.	’740 Patent Claim 13	Bruckman
		Bruckman at [0025] (“Typically, setting the protection policy includes determining a maximum number of the physical links that may fail while the logical link continues to provide at least the guaranteed bandwidth for the connection. In one embodiment, the guaranteed bandwidth is a bandwidth B, and the plurality of physical links consists of N links, and the maximum number is an integer P, and the link bandwidth allocated to each of the links is no less than B/(N-P). Conveying the data may further include managing the transmission of the data responsively to an actual number X of the physical links that have failed so that the guaranteed bandwidth on each of the links is limited to B/(N-X), X P, and an excess bandwidth on the physical links over the guaranteed bandwidth is available for other connections.”) Bruckman at [0027] (“There is also provided, in accordance with an embodiment of the present invention, a method for estab-lishing a connection with a guaranteed bandwidth for trans-mitting data between first and second endpoints, the method including:”) Bruckman at [0030] (“allocating a link bandwidth on each of the physical communication links for use in conveying the data between the endpoints such that the allocated link bandwidth includes a predefined safety margin based on the measure of fluc-tuation, so that a sum of the allocated link bandwidth over the plurality of the parallel physical links is substantially greater than the guaranteed bandwidth of the connection; and”) Bruckman at [0032] (“Typically, the guaranteed bandwidth is a bandwidth B, and the link bandwidth allocated to each of the links is no less than F*B/N, wherein F is a factor determined by the measure of fluctuation.”) Bruckman at [0035] (“a controller, which is adapted to allocate a link bandwidth on each of the physical communication links for use in conveying the data between the endpoints such that the link bandwidth includes a predefined safety margin based on the protection policy, so that a sum of the allocated link bandwidth over the plurality of the parallel physical links is substantially greater than the guaranteed bandwidth of the connection;”)

No.	’740 Patent Claim 13	Bruckman
		Bruckman at [0040] (“a controller, which is adapted to receive a measure of fluctuation that occurs in a rate of transmission of the data over the physical links when the data are distributed for transmission among the physical links, and to allocate a link bandwidth on each of the physical communication links for use in conveying the data between the endpoints such that the allocated link bandwidth includes a predefined safety margin based on the measure of fluctuation, so that a sum of the allocated link bandwidth over the plurality of the parallel physical links is substantially greater than the guaranteed bandwidth of the connection;”) Bruckman at [0050]-[0054] (“A Connection Admission Control entity (CAC) 44, typically a software process running on controller 42, man-ages the allocation of bandwidth in equipment 22. CAC 44 is responsible for ensuring that all connections between equipment 22 and customer nodes 24, 26, 28, ... , receive the amount of guaranteed bandwidth to which they are entitled, as well as for allocating any excess bandwidth available above the guaranteed minimum. For this purpose, CAC 44 maintains records that include: [0051] Throughput of equipment 22. [0052] Guaranteed and allocated excess bandwidth of each connection, as required by the applicable ser-vice level agreement. [0053] Overbooking ratio that the service provider who operates equipment 22 is prepared to use in allocating the available excess bandwidth. [0054] Safety factors to apply in determining bandwidth allocation on links in aggregation groups, as described below.”) Bruckman at [0066] (“When CAC 44 receives a request to open a con-nection with guaranteed bandwidth B over an aggregation group of N links, it might be assumed that the CAC should simply allocate bandwidth of BIN on each link. In practice, however, even if the hash function applied by distributor 58 meets the criteria outlined above, statistical variations in the traffic itself are likely to cause a larger portion of the traffic to be distributed to some of the links in the group than to others. In other words, some of the links may be required at times to carry group traffic with bandwidth substantially greater than B/N. As a result, these links may not have sufficient capacity remaining to provide bandwidth that has been guaranteed to other connections that the CAC has committed to carry over these links. When an aggregation group extends over a number

No.	’740 Patent Claim 13	Bruckman
		of concentrators 50 (as in the case of group 36), the traffic load on traces 52 may also be unbalanced. Overloading of traces 52 may likewise lead to a failure of system 22 to provide guaranteed bandwidth levels, in the distribution and/or the collection direction.”) Bruckman at [0071] (“Based on the safety margins determined at steps 60 and 64, CAC 44 allocates guaranteed bandwidth to each connection in a link aggregation group, at a bandwidth allocation step 66. To provide a shared safety margin for both failure protection and statistical bandwidth fluctuations, each link is preferably assigned a link bandwidth: B_{LINK}=min{B, F*B/(N-P)}”) Bruckman at [0072] (“This is the bandwidth that the CAC allocates to each link in the link aggregation group. Traffic manager 46, however, may limit the actual data rate of each link to be no greater than B_{LINK}=min {B, F*B/(N-X)}, wherein Xis the number of failed links, X P. This latter limit prevents the link aggregation group from taking more than its fair share of bandwidth relative to other connections that share the same trace 52. In any case, the sum of guaranteed bandwidth on all connections sharing any given trace 52 may not exceed the trace capacity. CAC 44 may overbook the excess bandwidth remaining above the guaranteed limits, so that the total (peak) allocation exceeds the trace capacity. The connections on links 30, including any link aggregation groups, then compete for the remaining available bandwidth (typically in a weighted manner, based on the amount of excess bandwidth contracted for in the users' service level agreements, as is known in the art). By limiting the data rate of each link in the aggregation group to min{B, F*B/(N-X)}, rather than min{B, F*B/(N-P)}, traffic manager 46 leaves bandwidth available for other connections that share the same trace.”) Bruckman at [0073] (“Once CAC 44 has allocated bandwidth for a given connection on a link aggregation group, normal data trans-mission proceeds. The bandwidth allocations apply to the amount of guaranteed traffic carried on each link 30 in the group. (Note that different allocations and separate traffic management may apply to outgoing traffic generated by distributor 58 and incoming traffic, which is sent by nodes 24, 26, ... , and processed by collector 56.) The allocations also affect the bandwidth used on traces 52. The rate limiting function of concentrators 50 is set to allow for the traffic bandwidth that may be used on each of links 30 that feed the

No.	’740 Patent Claim 13	Bruckman
		respective trace. As noted above, in allocating the band-width, CAC 44 ensures that the sum of the guaranteed bandwidth on all links sharing a given trace 52 is no greater than the trace bandwidth. The sum of the excess bandwidth allocated on the links, however, may exceed the trace bandwidth. In this case, the excess traffic is typically buff-ered as necessary, and is transmitted over the trace during intervals in which one or more of the links are not trans-mitting their guaranteed traffic levels and the trace has bandwidth available, or dropped if the buffer capacity is exceeded.”)

No.	’740 Patent Claim 14	Bruckman
14[preamble]	A method for connecting user ports to a communication network, comprising:	Bruckman discloses a method for connecting user ports to a communication network. For example, Bruckman discloses establishing a communication system by connecting a network transmitting with data with physical links to ports. Bruckman at Abstract (“A method for establishing a connection with a guaranteed bandwidth for transmitting data over a logical link that includes a plurality of parallel physical links between first and second endpoints. A link bandwidth is allocated on each of the physical communication links so as to include a predefined safety margin, based on either a failure protection policy, or a measure of fluctuation that occurs in a rate of data transmission over the physical links, or both. A sum of the allocated link bandwidth over the plurality of the parallel physical links is substantially greater than the guaranteed bandwidth of the connection. The data are conveyed over the logical link by distributing the data for transmission among the physical links in accordance with the allocated link bandwidth.”) Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the

No.	’740 Patent Claim 14	Bruckman
		distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”) Bruckman at [0056] (“Each line card 34 comprises one or more concen-trators 50, which comprise multiple ports that serve respec-tive links 30. The concentrators multiplex data traffic between links 30 and traces 52, which connect the concen-trators to switching core 40. Typically, main card 32 and line cards 34 are arranged in a card rack and plug into a printed circuit back plane, (not shown) which comprises traces 52. The bandwidth of each trace 52 may be less than the total bandwidth available on links 30 that are connected to the respective concentrator 50, based on considerations of sta-tistical multiplexing. To prevent overloading of traces 52, concentrators 50 may limit the rate of incoming data admit-ted on each link 30 so that it remains between a predeter-mined minimum, which is determined by the guaranteed bandwidth of the connections on the link, and a maximum, which is determined by the peak bandwidth (guaranteed plus permitted excess bandwidth) of the connections on the link. A traffic manager 46, which may also be a software process on controller 42, receives information regarding the opera-tional status of links 30 (for example, link or equipment failures) and updates the data rate limits applied by concen-trators 50, based on the status information and the bandwidth allocations made by CAC 44.”)
14[a]	coupling the user ports to one or more user interface modules;	Bruckman discloses coupling the user ports to one or more user interface modules. *See supra at* 1[a].
14[b]	coupling each user interface module to	Bruckman discloses coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel.

No.	’740 Patent Claim 14	Bruckman
	the communication network via a backplane using two or more backplane traces arranged in parallel,	*See supra at* 1[c], 3.
14[c]	at least one of said backplane traces being bi-directional and operative to communicate in both an upstream direction and a downstream direction;	Bruckman discloses at least one of said backplane traces being bi-directional and operative to communicate in both an upstream direction and a downstream direction. *See supra at* 14[b], 1[d].
14[d]	receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes;	Bruckman discloses receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes. *See supra at* 14[a], 1[e].
14[e]	for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces; and	Bruckman discloses for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces. *See supra at* 14[b], 1[f].

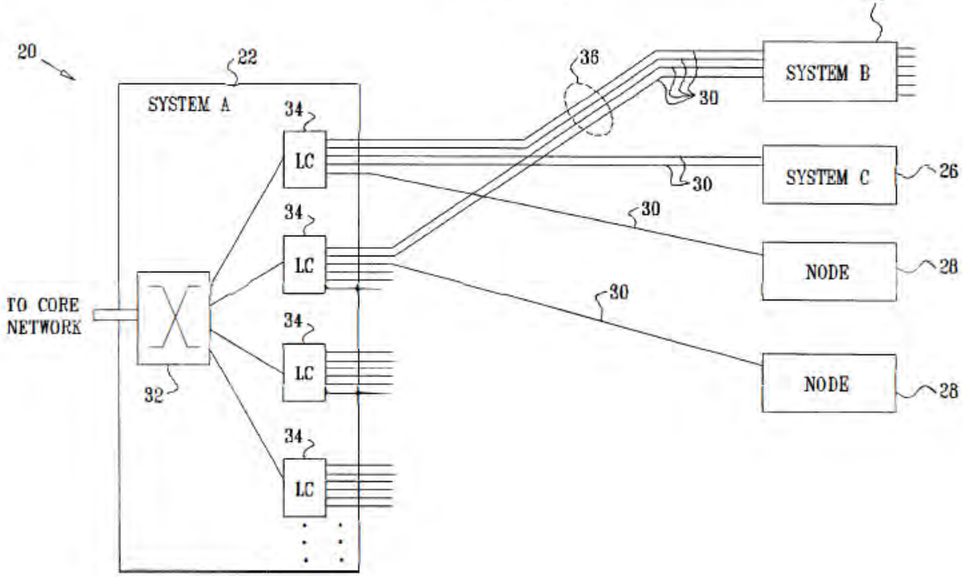
No.	’740 Patent Claim 14	Bruckman
14[f]	sending the data frame over the selected backplane trace;	Bruckman discloses sending the data frame over the selected backplane trace. *See supra at* 14[e], 1[g].
14[g]	said sending comprising communicating along said at least one of said backplane traces.	Bruckman discloses said sending comprising communicating along said at least one of said backplane traces. *See supra at* 14[f], 1[h].

No.	’740 Patent Claim 15	Bruckman
15[preamble]	A method for connecting user ports to a communication network, comprising:	Bruckman discloses a method for connecting user ports to a communication network. *See supra at* 14[preamble].
15[a]	coupling the user ports to one or more user interface modules;	Bruckman discloses coupling the user ports to one or more user interface modules. *See supra at* 14[a].
15[b]	coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel;	Bruckman discloses coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel. *See supra at* 14[b].
15[c]	receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes;	Bruckman discloses receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes. *See supra at* 14[d].

15[d]	for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces; and	Bruckman discloses for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces. *See supra at* 14[e].
15[e]	sending the data frame over the selected backplane trace,	Bruckman discloses sending the data frame over the selected backplane trace. *See supra at* 14[f].
15[f]	at least some of the backplane traces being aggregated into an Ethernet link aggregation (LAG) group.	Bruckman discloses at least some of the backplane traces being aggregated into an Ethernet link aggregation (LAG) group. *See supra at* 15[e], 4[f], 3.

No.	**’740 Patent Claim 16**	**Bruckman**
16	The method according to claim 14, wherein selecting the backplane trace comprises applying a hashing function to the at least one of the frame attributes.	Bruckman discloses the method according to claim 14, wherein selecting the backplane trace comprises applying a hashing function to the at least one of the frame attributes. *See supra* at 14, 9, 8.

No.	’740 Patent Claim 17	Bruckman
17[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Bruckman discloses apparatus for connecting a network node with a communication network. For example, Bruckman discloses equipment used to connect customer nodes with a network. Bruckman at [0014] (“When aggregated links are used to serve a given customer, the service provider may allocate a certain frac-tion of the bandwidth on each of the physical links in the aggregation group so that the aggregated logical link pro-vides the total bandwidth guaranteed by the customer's service level agreement. Typically, however, the actual bandwidth consumed on each of the physical links fluctuates statistically due to the non-uniform distribution of load among the links in the aggregation group. Furthermore, if one of the physical links fails, the bandwidth consumed on the remaining links in the group will need to increase in order to maintain the minimum guaranteed total bandwidth on the aggregated logical link. Under these circumstances, the service provider may not be able to provide all customers with the minimum bandwidth guaranteed by their service level agreements.”) [Bruckman at [0015] (“Embodiments of the present invention provide methods for bandwidth allocation in a link aggregation system to ensure that sufficient bandwidth will be available on the links in the group in order to meet service guarantees, notwithstanding load fluctuations and link failures. Safety margins are calculated, based on a measure of load fluctua-tion and on the level of protection to be provided (i.e., the worst-case number of link failures that must be tolerated by the system). These safety margins are applied in determining the bandwidth to be allocated for guaranteed services on each physical link in the aggregation group.”) Bruckman at [0047] (“FIG. 1 is a block diagram that schematically illus-trates elements of a communication system 20, in accor-dance with an embodiment of the present invention. In this example, central office equipment 22 communicates with customer nodes 24, 26, 28, ... , over physical links 30. Links 30 typically comprise full-duplex Ethernet links, such as lOBASE-n, lO0BASE-n or Gigabit Ethernet links, as are known in the art. (Alternatively, as noted above, other types of physical links may be used, such as ATM or PPP links.) Equipment 22 is configured to convey packet data traffic between the customer nodes and a network (which may be a metro network, access network, or other type of core network, for example). For this

No.	’740 Patent Claim 17	Bruckman
		purpose, equipment 22 comprises a main switching card 32, which is connected to multiple line cards 34 that serve links 30. Details of the structure and operation of equipment 22 are shown below in FIG. 2 and are described with reference thereto.”) Bruckman at Figure 1 FIG. 1  20 22 SYSTEM A 34 LC 34 LC 34 LC 34 LC TO CORE NETWORK 32 36 24 SYSTEM B 30 SYSTEM C 26 30 30 NODE 28 30 NODE 28 Bruckman at [0048] (“Equipment 22 and certain customer nodes, such as nodes 24 and 26, are configured to serve as aggregation systems in accordance with the above-mentioned Clause 43 of the 802.3 standard. (Equipment 22 and nodes 24 and 26 are accordingly labeled as System A, B and C, respectively.) For example, an aggregation group 36 of four physical links is defined between equipment 22 and node 24. Another aggregation group of two physical links may be defined between equipment 22 and node 26. Each aggregation group (as well as each

No.	’740 Patent Claim 17	Bruckman
		non-aggregated link 30) may serve multiple customer connections between the respective customer node and equipment 22.”) Bruckman at [0050] (“A Connection Admission Control entity (CAC) 44, typically a software process running on controller 42, man-ages the allocation of bandwidth in equipment 22. CAC 44 is responsible for ensuring that all connections between equipment 22 and customer nodes 24, 26, 28, ... , receive the amount of guaranteed bandwidth to which they are entitled, as well as for allocating any excess bandwidth available above the guaranteed minimum. For this purpose, CAC 44 maintains records that include:”) Bruckman at [0073] (“Once CAC 44 has allocated bandwidth for a given connection on a link aggregation group, normal data trans-mission proceeds. The bandwidth allocations apply to the amount of guaranteed traffic carried on each link 30 in the group. (Note that different allocations and separate traffic management may apply to outgoing traffic generated by distributor 58 and incoming traffic, which is sent by nodes 24, 26, ... , and processed by collector 56.) The allocations also affect the bandwidth used on traces 52. The rate limiting function of concentrators 50 is set to allow for the traffic bandwidth that may be used on each of links 30 that feed the respective trace. As noted above, in allocating the band-width, CAC 44 ensures that the sum of the guaranteed bandwidth on all links sharing a given trace 52 is no greater than the trace bandwidth. The sum of the excess bandwidth allocated on the links, however, may exceed the trace bandwidth. In this case, the excess traffic is typically buff-ered as necessary, and is transmitted over the trace during intervals in which one or more of the links are not trans-mitting their guaranteed traffic levels and the trace has bandwidth available, or dropped if the buffer capacity is exceeded.”)
17[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network	Bruckman discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. For example, Bruckman discloses line cards which process data frames with specific frame information that are transmitted between customer nodes and the network.

No.	’740 Patent Claim 17	Bruckman
	node and the communication network,	Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”) Bruckman at [0049] (“FIG. 2 is a block diagram that schematically shows details of equipment 22, in accordance with an embodiment of the present invention. Main card 32 comprises a switching core 40, which switches traffic to and from line cards 34. Two line cards 34, labeled LCl and LC2, are shown in the figure. The operation of switch 40 is managed by a controller 42, typically an embedded microprocessor with suitable software for carrying out the functions described herein.”)

No.	’740 Patent Claim 17	Bruckman
		Bruckman at [0056] (“Each line card 34 comprises one or more concen-trators 50, which comprise multiple ports that serve respec-tive links 30. The concentrators multiplex data traffic between links 30 and traces 52, which connect the concen-trators to switching core 40. Typically, main card 32 and line cards 34 are arranged in a card rack and plug into a printed circuit back plane, (not shown) which comprises traces 52. The bandwidth of each trace 52 may be less than the total bandwidth available on links 30 that are connected to the respective concentrator 50, based on considerations of sta-tistical multiplexing. To prevent overloading of traces 52, concentrators 50 may limit the rate of incoming data admit-ted on each link 30 so that it remains between a predeter-mined minimum, which is determined by the guaranteed bandwidth of the connections on the link, and a maximum, which is determined by the peak bandwidth (guaranteed plus permitted excess bandwidth) of the connections on the link. A traffic manager 46, which may also be a software process on controller 42, receives information regarding the opera-tional status of links 30 (for example, link or equipment failures) and updates the data rate limits applied by concen-trators 50, based on the status information and the bandwidth allocations made by CAC 44.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”)

No.	’740 Patent Claim 17	Bruckman
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
17[b]	at least one of said interface modules being operative to communicate in both an upstream direction and a downstream direction;	Bruckman discloses at least one of said interface modules being operative to communicate in both an upstream direction and a downstream direction. For example, Bruckman discloses line cards that both transmit and receive data frames in incoming and outgoing directions. Bruckman at [0056] (“Each line card 34 comprises one or more concen-trators 50, which comprise multiple ports that serve respec-tive links 30. The concentrators multiplex data traffic between links 30 and traces 52, which connect the concen-trators to switching core 40. Typically, main card 32 and line cards 34 are arranged in a card rack and plug into a printed circuit back plane, (not shown) which comprises traces 52. The bandwidth of each trace 52 may be less than the total bandwidth available on links 30 that are connected to the respective concentrator 50, based on considerations of sta-tistical multiplexing. To prevent overloading of traces 52, concentrators 50 may limit the rate of incoming data admit-ted on each link 30 so that it remains between a predeter-mined minimum, which is determined by the guaranteed bandwidth of the connections on the link, and a maximum, which is determined by the peak bandwidth (guaranteed plus permitted excess bandwidth) of the connections on the link. A traffic manager 46, which may also be a software process on controller 42, receives information regarding the opera-tional status of links 30 (for example, link or equipment failures) and updates the data rate limits applied by concen-trators 50, based on the status information and the bandwidth allocations made by CAC 44.”) Bruckman at [0073] (“Once CAC 44 has allocated bandwidth for a given connection on a link aggregation group, normal data trans-mission proceeds. The bandwidth allocations apply to the amount of guaranteed traffic carried on each link 30 in the group. (Note that different allocations and separate traffic management may apply to outgoing traffic generated by

No.	’740 Patent Claim 17	Bruckman
		distributor 58 and incoming traffic, which is sent by nodes 24, 26, ... , and processed by collector 56.) The allocations also affect the bandwidth used on traces 52. The rate limiting function of concentrators 50 is set to allow for the traffic bandwidth that may be used on each of links 30 that feed the respective trace. As noted above, in allocating the band-width, CAC 44 ensures that the sum of the guaranteed bandwidth on all links sharing a given trace 52 is no greater than the trace bandwidth. The sum of the excess bandwidth allocated on the links, however, may exceed the trace bandwidth. In this case, the excess traffic is typically buff-ered as necessary, and is transmitted over the trace during intervals in which one or more of the links are not trans-mitting their guaranteed traffic levels and the trace has bandwidth available, or dropped if the buffer capacity is exceeded.”) Bruckman at [0074] (“Normal data transmission over the connection con-tinues unless and until a failure is detected on one of links 30 or line cards 34, at a failure detection step 68. Traffic manager 46 is informed of the failure, and notifies distribu-tor 58 accordingly to modify its hash function so that outgoing traffic is distributed over the remaining links in the group. Use of the protection parameter P in setting the bandwidth allocation ensures that (as long as no more than P links are out of service) there is sufficient bandwidth available for the connection on the remaining links. It may also be necessary for the traffic manager to adjust the rate limiting function of concentrators 50, at a concentrator readjustment step 70, in order to deal with the increased incoming traffic on the remaining links. For example, if link L4 (FIG. 2) fails, the traffic on each of links Ll, L2 and L3 is expected to increase by 1/3, and the concentrator in LCl will have to deal with the resulting increase in traffic on the corresponding trace 52.”)
17[c]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Bruckman discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 1[a].

No.	’740 Patent Claim 17	Bruckman
17[d]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Bruckman discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 1[c].
17[e]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame;	Bruckman discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 1[f].
17[f]	at least one of said first physical links and at least one of said second links being bi-directional links operative to	Bruckman discloses at least one of said first physical links and at least one of said second links being bi-directional links operative to communicate in both said upstream direction and said downstream direction. *See supra at* 1[b], 1[d].

No.	’740 Patent Claim 17	Bruckman
	communicate in both said upstream direction and said downstream direction.	

No.	’740 Patent Claim 18	Bruckman
18[a]	The apparatus according to claim 17, and comprising a backplane to which the one or more interface modules are coupled,	Bruckman discloses the apparatus according to claim 17, and comprising a backplane to which the one or more interface modules are coupled. *See supra at* 3, 17.
18[b]	wherein the second physical links comprise backplane traces formed on the backplane.	Bruckman discloses wherein the second physical links comprise backplane traces formed on the backplane. *See supra at* 3, 17.

No.	’740 Patent Claim 19	Bruckman
19[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Bruckman discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
19[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	Bruckman discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].
19[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Bruckman discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].
19[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Bruckman discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].

19[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,	Bruckman discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].
19[e]	at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group.	Bruckman discloses at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group. *See supra at* 4[f].

No.	’740 Patent Claim 20	Bruckman
20[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Bruckman discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
20[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	Bruckman discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].
20[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Bruckman discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].
20[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Bruckman discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].

20[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,	Bruckman discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].
20[e]	two or more of the first physical links being aggregated into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node.	Bruckman discloses two or more of the first physical links being aggregated into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node. *See supra at* 19[e], 5[f].

No.	’740 Patent Claim 21	Bruckman
21	The apparatus according to claim 17, and comprising a multiplexer, which is arranged to perform at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node.	Bruckman discloses the apparatus according to claim 17, and comprising a multiplexer, which is arranged to perform at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node. *See supra at* 6, 17.

No.	’740 Patent Claim 22	Bruckman
22	The apparatus according to claim 17, wherein the control module is arranged to balance a frame data rate among at least some of the first and second physical links.	Bruckman discloses the apparatus according to claim 17, wherein the control module is arranged to balance a frame data rate among at least some of the first and second physical links. *See supra at* 7, 17.

No.	’740 Patent Claim 23	Bruckman
23	The apparatus according to claim 17, wherein the control module is arranged to apply a mapping function to the at least one of the frame attributes so as to select the first and second physical links.	Bruckman discloses the apparatus according to claim 17, wherein the control module is arranged to apply a mapping function to the at least one of the frame attributes so as to select the first and second physical links. *See supra at* 8, 17.

No.	’740 Patent Claim 24	Bruckman
24	The apparatus according to claim 23, wherein the mapping function comprises a hashing function.	Bruckman discloses the apparatus according to claim 23, wherein the mapping function comprises a hashing function. *See supra at* 9, 23.

No.	’740 Patent Claim 25	Bruckman
25[a]	The apparatus according to claim 24, wherein the control module is arranged to determine a hashing size responsively to a number of at least some of the first and second physical links,	Bruckman discloses the apparatus according to claim 24, wherein the control module is arranged to determine a hashing size responsively to a number of at least some of the first and second physical links. *See supra at* 10[a], 24.
25[b]	to apply the hashing function to the at least one of the frame attributes to produce a hashing key,	Bruckman discloses to apply the hashing function to the at least one of the frame attributes to produce a hashing key. *See supra at* 10[b].
25[c]	to calculate a modulo of a division operation of the hashing key by the hashing size, and	Bruckman discloses to calculate a modulo of a division operation of the hashing key by the hashing size. *See supra at* 10[c].
25[d]	to select the first and second physical links responsively to the modulo.	Bruckman discloses to select the first and second physical links responsively to the modulo. *See supra at* 10[d].

No.	’740 Patent Claim 26	Bruckman
26	The apparatus according to claim 25, wherein the control module is arranged to select the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo.	Bruckman discloses the apparatus according to claim 25, wherein the control module is arranged to select the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo. *See supra at* 11, 25.

No.	’740 Patent Claim 27	Bruckman
27	The apparatus according to claim 17, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port.	Bruckman discloses the apparatus according to claim 17, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port. *See supra at* 12, 17.

No.	’740 Patent Claim 28	Bruckman
28[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Bruckman discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
28[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	Bruckman discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].
28[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Bruckman discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].
28[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Bruckman discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].

28[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,	Bruckman discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].
28[e]	the communication network being arranged to provide a communication service to the network node,	Bruckman discloses the communication network being arranged to provide a communication service to the network node. *See supra at* 2[b].

28[f]	the service having specified bandwidth requirements comprising at least one of a committed information rate (CR), a peak information rate (PIR) and an excess information rate (EIR), and	Bruckman discloses the service having specified bandwidth requirements comprising at least one of a committed information rate (CR), a peak information rate (PIR) and an excess information rate (EIR). *See supra at* 13[i].
28[g]	the first and second groups of physical links being dimensioned to provide an allocated bandwidth for the communication service responsively to the bandwidth requirements.	Bruckman discloses the first and second groups of physical links being dimensioned to provide an allocated bandwidth for the communication service responsively to the bandwidth requirements. *See supra at* 13[j].

No.	’740 Patent Claim 29	Bruckman
29[preamble]	Apparatus for connecting user ports to a communication network, comprising:	Bruckman discloses apparatus for connecting user ports to a communication network. *See supra at* 17[preamble], 14[preamble].
29[a]	one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network,	Bruckman discloses one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network. *See supra at* 17[a], 14[a].
29[b]	at least one of said user interface modules being bi-directional and operative to communicate in both an upstream direction and a downstream direction;	Bruckman discloses at least one of said user interface modules being bi-directional and operative to communicate in both an upstream direction and a downstream direction. *See supra at* 17[b], 14[c].

29[c]	a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network,	Bruckman discloses a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network. *See supra at* 14[b]-[e].
29[d]	at least one of said backplane traces being bi-directional and operative to communicate in both said upstream direction and said downstream direction; and	Bruckman discloses at least one of said backplane traces being bi-directional and operative to communicate in both said upstream direction and said downstream direction. *See supra at* 14[c], 17[b].

29[e]	a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame.	Bruckman discloses a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame. *See supra at* 14[e], 17[e].

No.	’740 Patent Claim 30	Bruckman
30[preamble]	Apparatus for connecting user ports to a communication network, comprising:	Bruckman discloses apparatus for connecting user ports to a communication network. *See supra at* 29[preamble].
30[a]	one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network;	Bruckman discloses one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network. *See supra at* 29[a].
30[b]	a backplane having the one or more user interface modules coupled thereto and	Bruckman discloses a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network.

No.	’740 Patent Claim 30	Bruckman
	comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network;	*See supra at* 29[c].
30[c]	a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame;	Bruckman discloses a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame. *See supra at* 29[e].
30[d]	at least some of the backplane traces are aggregated into an Ethernet link aggregation (LAG) group.	Bruckman discloses at least some of the backplane traces are aggregated into an Ethernet link aggregation (LAG) group. *See supra at* 4[f], 15[f].

No.	’740 Patent Claim 31	Bruckman
31	The apparatus according to claim 29, wherein the control module is arranged to apply a hashing function to the at least one of the frame attributes so as to select the backplane trace.	Bruckman discloses the apparatus according to claim 29, wherein the control module is arranged to apply a hashing function to the at least one of the frame attributes so as to select the backplane trace. *See supra at* 16, 29, 30[c].

EXHIBIT C-7
Defendant’s Preliminary Invalidity Contentions
Orckit Corporation v. Cisco Systems, Inc., 2:22-cv-00276-JRG-RSP

Chart for U.S. Patent 7,545,740 (“the ’740 Patent”)
U.S. Patent Publication No. 2003/0210688 to Basso et al. (“Basso”)

As shown in the chart below, all Asserted Claims of the ’740 Patent are invalid under (1) 35 U.S.C. §§ 102 (a), (b), (e), and (g) because Basso meets each element of those claims, and/or (2) 35 U.S.C. § 103 because Basso renders those claims obvious either alone, or in combination with the knowledge of a person having ordinary skill in the art, and in further combination with the references specifically identified below and in the following claim chart and/or one or more references identified in Defendant’s Preliminary Invalidity Contentions. The following quotations and diagrams come from Basso titled “Logically Grouping Physical Ports Into Logical Interfaces To Expand Bandwidth”, which was filed on May 13, 2002, and published on November 13, 2003.

Motivations to combine the disclosures in Basso with disclosures in other publications known in the art, as explained in this chart, include at least the similarity in subject matter between the references to the extent they concern methods of data communication systems, and specifically to methods and systems for link aggregation in a data communication network. Insofar as the references cite other patents or publications, or suggest additional changes, one of ordinary skill in the art would look beyond a single reference to other references in the field.

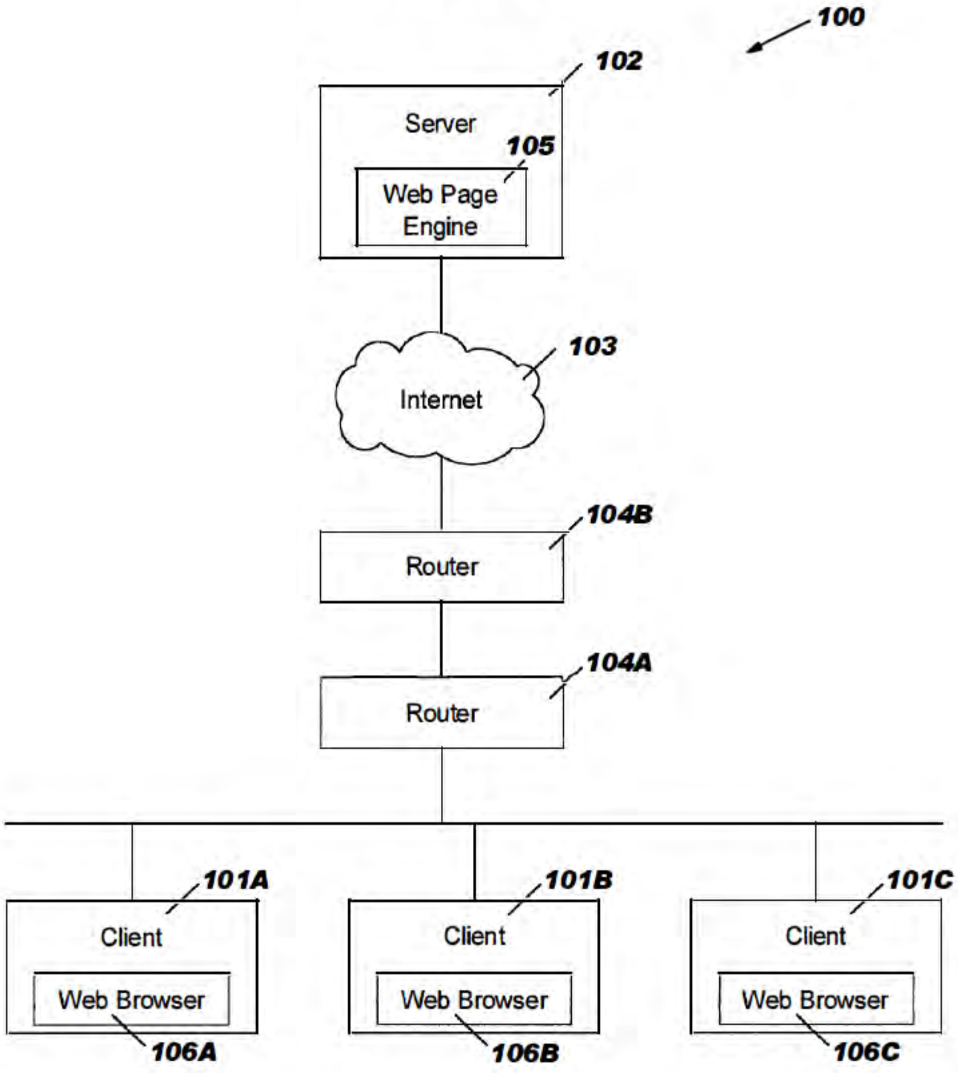
These invalidity contentions are based on Defendant’s present understanding of the asserted claims, and Orckit’s apparent construction of the claims in its November 3, 2022 Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1, and Orckit’s January 19, 2023 First Amended Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1 (Orckit’s “Infringement Disclosures”), which is deficient at least insofar as it fails to cite any documents or identify accused structures, acts, or materials in the Accused Products with particularity. Defendant does not agree with Orckit’s application of the claims, or that the claims satisfy the requirements of 35 U.S.C. § 112. Defendant’s contentions herein are not, and should in no way be seen as, admissions or adoptions as to any particular claim scope or construction, or as any admission that any particular element is met by any accused product in any particular way. Defendant objects to any attempt to imply claim construction from this chart. Defendant’s prior art invalidity contentions are made in a variety of alternatives and do not represent Defendant’s agreement or view as to the meaning, definiteness, written description support for, or enablement of any claim contained therein.

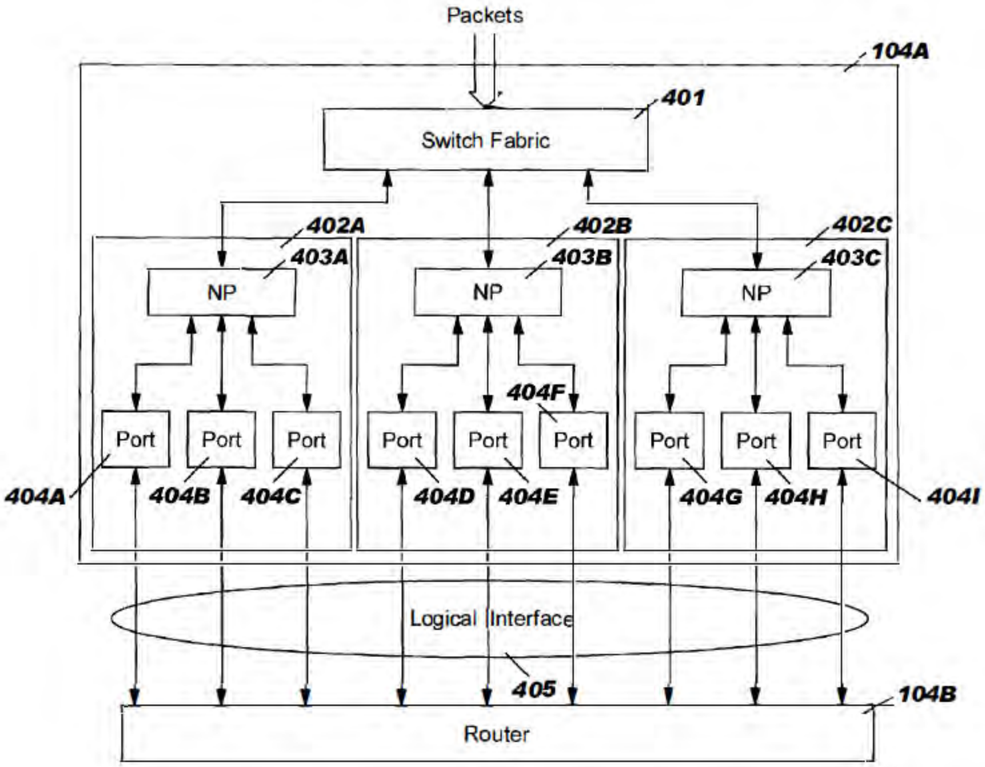
The following contentions are subject to revision and amendment pursuant to Federal Rule of Civil Procedure 26(e), the Local Rules, and the Orders of record in this matter subject to further investigation and discovery regarding the prior art and the Court's construction of the claims at issue.

No.	'740 Patent Claim 1	Basso
1[preamble]	A method for communication, comprising:	Basso discloses a method for communication. For example, Basso discloses a method for routing packets with a network device. Basso at Abstract ("A method, system and computer program product for rout-ing packets. A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more physical ports. The physical ports may be connected to another one or more network devices. A plurality of physical ports across one or more blades connected to the same network device may be logically mapped into a logical interface to that network device. By logically grouping a plurality of physical ports into a logical interface to a network device, a network processor may be able to transmit packets of data to that network device across multiple ports instead of one physical port.") Basso at [0008] ("In one embodiment of the present invention, a method for routing packets may comprise the step of map-ping a plurality of physical ports in a network device, e.g., router, connected to another network device, e.g., router, gateway, edge device, server, into a logical interface to that network device. By logically grouping a plurality of ports into a logical interface, a network processor in the network device, e.g., router, may be able to transmit packets of data across multiple physical ports to a particular network device instead of one physical port as discussed further below.") Basso at [0009] ("A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more ports. These ports may be connected to another one or more network devices. The switch fabric may be configured to direct incoming packets of data to particular blades where one or more of the network proces-sors in the recipient blade may be configured to process the received packets.")

No.	’740 Patent Claim 1	Basso
		Basso at [0021] (“FIG. 1 illustrates one embodiment of the present invention of a network system 100. Network system 100 may comprise one or more clients 101A-C connected to a server 102 via the Internet 103. A more detailed description of server 102 is provided further below in conjunction with FIG. 3. The Internet 103 may refer to a network of com-puters. Network system 100 may further comprise one or more routers, e.g., 104A-B, that may be coupled to one or more clients lOlA-C. Routers 104A-B may be configured to forward packets of information from the one or more clients lOlA-C to the Internet 103. Clients lOlA-C may collec-tively or individually be referred to as clients 101 or client 101, respectively. A more detailed description of client 101 is provided further below in conjunction with FIG. 2. Routers 104A-B may collectively or individually be referred to as routers 104 or router 104, respectively. Amore detailed description of router 104 implementing logical interface(s) is provided further below in conjunction with FIG. 4. It is noted that network system 100 may comprise any number of clients 101, any number of servers 102 as well as any number of routers 104 and that FIG. 1 is illustrative. It is further noted that network system 100 may comprise one or more routers (not shown) that may be coupled to server 102. These routers (not shown) may be configured to forward received packets of information to server 102. It is further noted that the connection between clients 101 and the Internet 103 may be any medium type, e.g., wireless, wired. It is further noted that client 101 may be any type of device, e.g., wireless, Personal Digital Assistant (PDA), cell phone, personal computer system, workstation, Internet appliance, configured with the capability of connecting to the Internet 103 and consequently communicating with server 102. It is further noted that FIG. 1 is not to be limited in scope to any one particular embodiment.”) Basso at [0029] (“FIG. 4 illustrates an embodiment of the present invention of router 104A (FIG. 1) implementing logical interface(s) as discussed further below. It is noted that even though the following discusses an embodiment of the present invention of router 104A that the description of router 104A applies to any router 104, e.g., router 104B, of network system 100 (FIG. 1).”) Basso at [0030] (“Returning to FIG. 4, router 104A may be config-ured to receive packets of data such as from client 101 (FIG. 1), e.g., client 101A, that may be directed to another

No.	’740 Patent Claim 1	Basso
		particular network device, e.g., router 104B, in network 100 (FIG. 1). Router 104A may comprise a switch fabric 401 configured to direct the incoming packets of data to particu-lar blades 402A-C coupled to switch fabric 401. Blade 402A may comprise a network processor 403A coupled with one or more ports 404A-C. Blade 402B may comprise a network processor 403B coupled with one or more ports 404D-F. Blade 402C may comprise a network processor 403C coupled with one or more ports 404G-I. Blades 402A-C may collectively or individually be referred to as blades 402 or blade 402, respectively. Network processors 403A-C may collectively or individually be referred to as network pro-cessors 403 or network processor 403, respectively. Ports 404A-I may collectively or individually be referred to as ports 404 or port 404, respectively. Each port 404 may be coupled to a particular network device, e.g., gateway, server, router such as router 104B, in network system 100. It is noted that some of ports 404 may be coupled to a separate network device and that FIG. 4 is illustrative. It is further noted that router 104A may comprise any number of blades 402 and each blade 402 may comprise any number of network processors 403 and ports 404.”) Basso at [0036] (“Network processor 403, e.g., network processor 403A, may be configured in one embodiment to comprise a memory (not shown), e.g., non-volatile memory, to store a program to perform the steps of a method for routing packets implementing logical interface(s) 405 as described below in conjunction with FIG. 5. The memory (not shown) in network processor 403 may further be configured to store a forwarding table and tables associated with logical inter-faces 405 as discussed below in FIG. 5. Network processor 403 may further comprise a processor (not shown), com-monly referred to as a packet processor, coupled to the memory (not shown). The packet processor (not shown) may be configured to execute the instructions of the program. It is further noted that the steps of the method performed by the program mentioned above may in an alternative embodi-ment be implemented in hardware such as in an Application Specific Integrated Circuit (ASIC).”) Basso at Figure 1

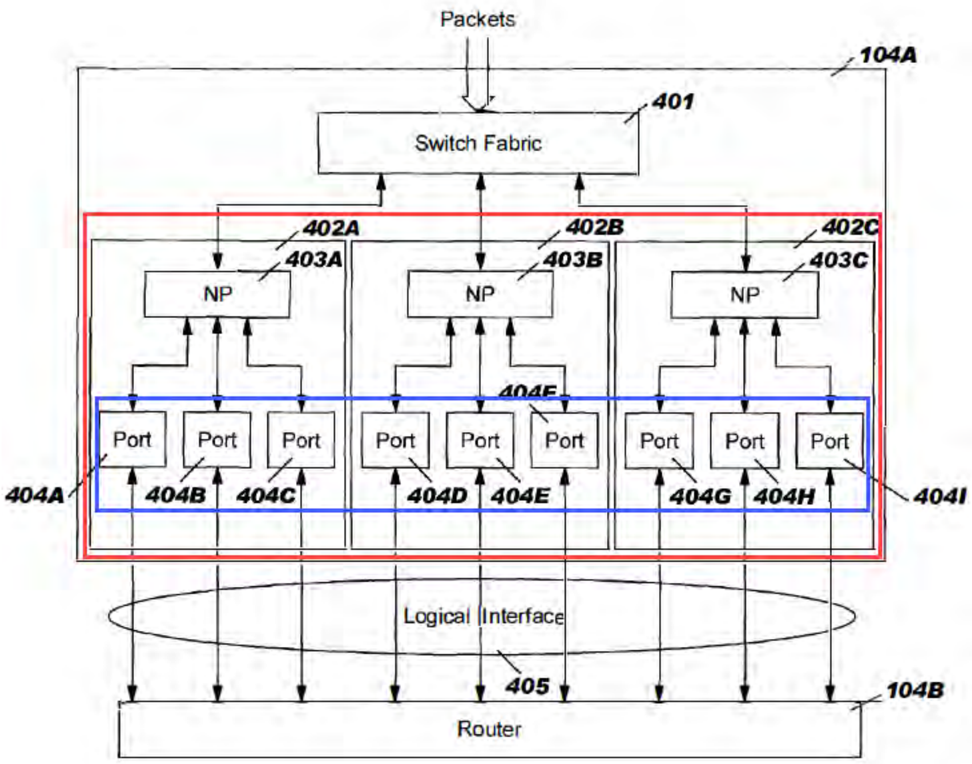
No.	’740 Patent Claim 1	Basso
		FIG. 1  100 102 Server 105 Web Page Engine 103 Internet 104B Router 104A Router 101A Client Web Browser 106A 101B Client Web Browser 106B 101C Client Web Browser 106C

No.	’740 Patent Claim 1	Basso
		Basso at Figure 4 FIG. 4  Packets 104A 401 Switch Fabric 402A 403A NP 402B 403B NP 402C 403C NP 404F Port Port Port Port Port Port Port Port Port 404A 404B 404C 404D 404E 404G 404H 404I Logical Interface 405 104B Router
1[a]	coupling a network node to one or more interface modules using a first group of	Basso discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. For example, Basso discloses coupling a plurality of blades to a network device, using a plurality of parallel ports.

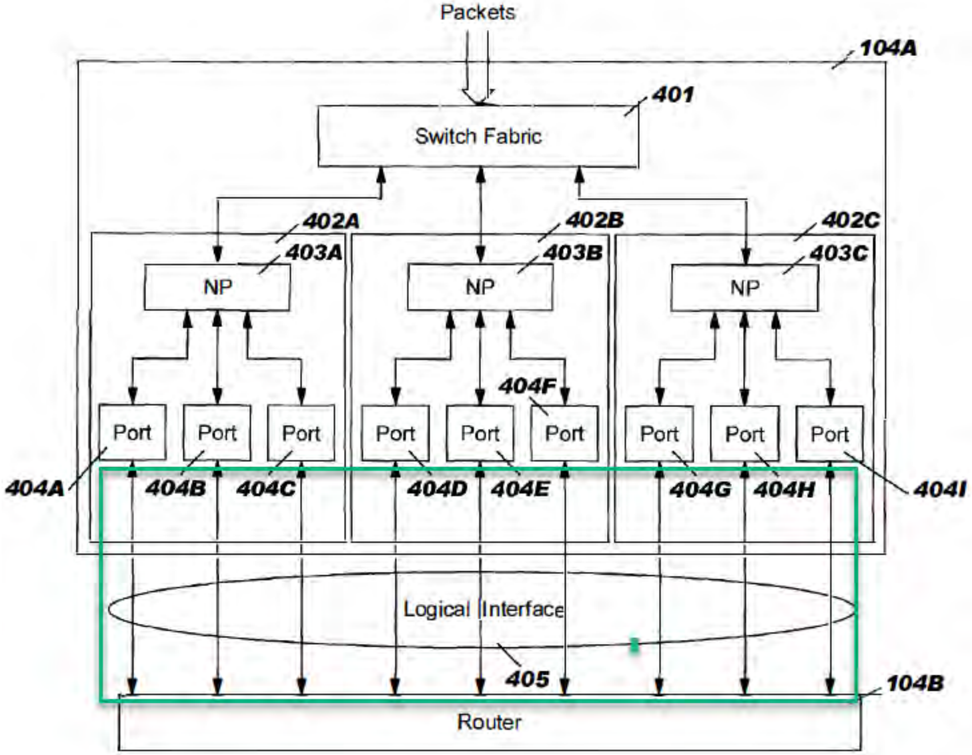
No.	’740 Patent Claim 1	Basso
	first physical links arranged in parallel,	Basso at Abstract (“A method, system and computer program product for rout-ing packets. A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more physical ports. The physical ports may be connected to another one or more network devices. A plurality of physical ports across one or more blades connected to the same network device may be logically mapped into a logical interface to that network device. By logically grouping a plurality of physical ports into a logical interface to a network device, a network processor may be able to transmit packets of data to that network device across multiple ports instead of one physical port.”) Basso at [0007] (“The problems outlined above may at least in part be solved in some embodiments by mapping a plurality of physical ports connected to a network device into a logical interface to that network device, e.g., router, gateway, edge device, server. That is, a plurality of physical ports may be logically grouped into a logical interface to a network device thereby enabling a network processor to transmit packets of data across multiple ports instead of one physical port.”) Basso at [0008] (“In one embodiment of the present invention, a method for routing packets may comprise the step of map-ping a plurality of physical ports in a network device, e.g., router, connected to another network device, e.g., router, gateway, edge device, server, into a logical interface to that network device. By logically grouping a plurality of ports into a logical interface, a network processor in the network device, e.g., router, may be able to transmit packets of data across multiple physical ports to a particular network device instead of one physical port as discussed further below.”) Basso at [0009] (“A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more ports. These ports may be connected to another one or more network devices. The switch fabric may be configured to direct incoming packets of data to particular blades where one or more of the network proces-sors in the recipient blade may be configured to process the received packets.”)

No.	’740 Patent Claim 1	Basso
		Basso at [0030] (“Returning to FIG. 4, router 104A may be config-ured to receive packets of data such as from client 101 (FIG. 1), e.g., client 101A, that may be directed to another particular network device, e.g., router 104B, in network 100 (FIG. 1). Router 104A may comprise a switch fabric 401 configured to direct the incoming packets of data to particu-lar blades 402A-C coupled to switch fabric 401. Blade 402A may comprise a network processor 403A coupled with one or more ports 404A-C. Blade 402B may comprise a network processor 403B coupled with one or more ports 404D-F. Blade 402C may comprise a network processor 403C coupled with one or more ports 404G-I. Blades 402A-C may collectively or individually be referred to as blades 402 or blade 402, respectively. Network processors 403A-C may collectively or individually be referred to as network pro-cessors 403 or network processor 403, respectively. Ports 404A-I may collectively or individually be referred to as ports 404 or port 404, respectively. Each port 404 may be coupled to a particular network device, e.g., gateway, server, router such as router 104B, in network system 100. It is noted that some of ports 404 may be coupled to a separate network device and that FIG. 4 is illustrative. It is further noted that router 104A may comprise any number of blades 402 and each blade 402 may comprise any number of network processors 403 and ports 404.”) Basso ay [0031] (“Referring to FIG. 4, network processor 403 may be configured to receive a packet of data from switch fabric 401. Upon receiving a packet, network processor 403 may be configured to process the packet of data. Processing may include but not limited to: determining what activities to be performed on the received packet, transmitting or discarding the received packet, determining which network device, e.g., router such as router 104B, server, edge device, gateway, to transmit the received packet, etc. Network processor 403 may then transmit the processed packet to a particular network device, e.g., router such as router 104B, through a port 404 connected to that network device.”) Basso at [0033] (“If, however, packets of data were able to be trans-mitted across multiple physical ports to a particular network device from a particular network processor, then the band-width capacity would be increased. That is, if packets of data were able to be transmitted across multiple physical ports to a particular network device by a network processor, then the rate packets of data are transmitted may be increased. It would therefore be desirable to enable

No.	’740 Patent Claim 1	Basso
		a network processor to transmit packets of data across multiple physical ports to a particular network device instead of one physical port.”) Basso at [0034] (“Network processor 403 may be configured to trans-mit packets of data across multiple physical ports 404 to a particular network device, e.g., router 104B, by logically grouping a plurality of physical ports 404, e.g., ports 404A-I, into a logical interface 405. That is, the physical connections between router 104A and a particular network device, e.g., router 104B, may be logically grouped into a logical inter-face 405. Each set of physical connections between router 104A and a particular network device may be logically grouped into a particular logical interface 405. For example, if ports 404A-F were physically connected to network device #1, then ports 404A-F may be logically grouped into logical interface 405#1. If ports 404G-I were physically connected to network device #2, then ports 404G-I may be logically grouped into logical interface 405#2. It is noted that physical ports 404 may be logically grouped into more than one logical interface 405 and that FIG. 4 is illustrative. It is further noted that each logical interface 405 may be associated with logically grouping a plurality of physical connections to a particular network device, e.g., router, gateway, edge device, server. It is further noted that a logical interface 405 may be associated with logically grouping a plurality of ports in either one blade 402 or across multiple blades 402.”) Basso at [0035] (“By logically grouping a plurality of ports 404 coupled to a particular network device into a logical inter-face 405, network processor 403 may be configured to transmit processed packets to that particular network device via any blade 402/port 404 combination grouped in that logical interface 405. For example, referring to FIG. 4, ports 404A-404I are physically connected to router 104B. If ports 404A-404I were logically grouped into logical interface 405, then a particular network processor 403, e.g., network processor 403A, may be configured to transmit processed packets that are determined to be transmitted to router 104B through any of ports 404A-404I in blades 402A-C, respec-tively. Network processor 403, e.g., network processor 403A, may be configured to transmit the processed packets to router 104B through ports 404, e.g., ports 404D-I, not in its blade 402, e.g., blade 402A, by forwarding the processed packets to switch fabric 401 which may then direct the processed packets to another appropriate physical blade 402/port 404 combination. Network processor 403, e.g., network processor 403A, may further be configured to transmit the processed packets to router 104B

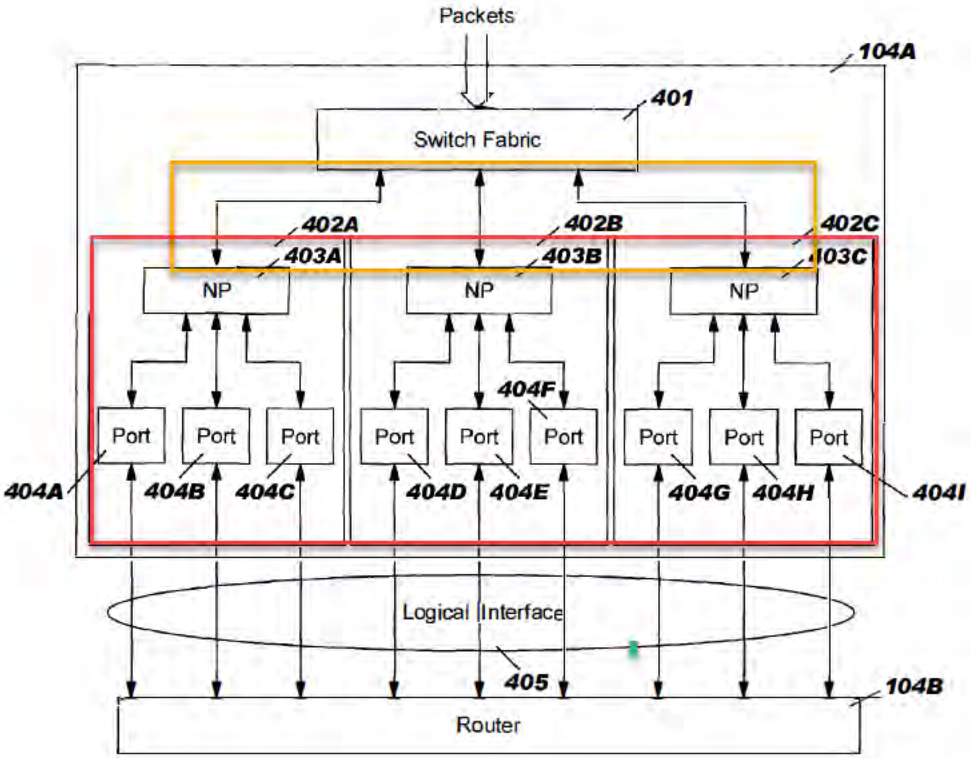
No.	’740 Patent Claim 1	Basso
		through any ports 404, e.g., ports 404A-C, in its blade 402, e.g., blade 402A, instead of just one physical port 404 in its blade 402, e.g., blade 402A. A more detailed description of routing packets implementing logical interface(s) 405 is provided below in FIG. 5.”) Basso at Figure 4 (annotation added; plurality of blades, i.e, interface modules, in red, physical ports, i.e., first group of first physical links arranged in parallel, in blue) FIG. 4  Packets 104A 401 Switch Fabric 402A 402B 402C 403A 403B 403C NP NP NP 404F Port Port Port Port Port Port Port Port Port 404A 404B 404C 404D 404E 404G 404H 404I Logical Interface 405 104B Router

No.	’740 Patent Claim 1	Basso
1[b]	at least one of said first physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction	Basso discloses at least one of said first physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction. For example, Basso discloses ports connecting a router to blades capable of both receiving and transmitting packets. Basso at [0030] (“Returning to FIG. 4, router 104A may be config-ured to receive packets of data such as from client 101 (FIG. 1), e.g., client 101A, that may be directed to another particular network device, e.g., router 104B, in network 100 (FIG. 1). Router 104A may comprise a switch fabric 401 configured to direct the incoming packets of data to particu-lar blades 402A-C coupled to switch fabric 401. Blade 402A may comprise a network processor 403A coupled with one or more ports 404A-C. Blade 402B may comprise a network processor 403B coupled with one or more ports 404D-F. Blade 402C may comprise a network processor 403C coupled with one or more ports 404G-I. Blades 402A-C may collectively or individually be referred to as blades 402 or blade 402, respectively. Network processors 403A-C may collectively or individually be referred to as network pro-cessors 403 or network processor 403, respectively. Ports 404A-I may collectively or individually be referred to as ports 404 or port 404, respectively. Each port 404 may be coupled to a particular network device, e.g., gateway, server, router such as router 104B, in network system 100. It is noted that some of ports 404 may be coupled to a separate network device and that FIG. 4 is illustrative. It is further noted that router 104A may comprise any number of blades 402 and each blade 402 may comprise any number of network processors 403 and ports 404.”) Basso at [0031] (“Referring to FIG. 4, network processor 403 may be configured to receive a packet of data from switch fabric 401. Upon receiving a packet, network processor 403 may be configured to process the packet of data. Processing may include but not limited to: determining what activities to be performed on the received packet, transmitting or discarding the received packet, determining which network device, e.g., router such as router 104B, server, edge device, gateway, to transmit the received packet, etc. Network processor 403 may then transmit the processed packet to a particular network device, e.g., router such as router 104B, through a port 404 connected to that network device.”)

No.	’740 Patent Claim 1	Basso
		Basso at Figure 4 (annotation added) FIG. 4  Packets 104A 401 Switch Fabric 402A 403A 402B 403B 402C 403C NP NP NP 404F Port Port Port Port Port Port Port Port Port 404A 404B 404C 404D 404E 404G 404H 404I Logical Interface 405 104B Router Basso at [0039] (“Referring to FIG. 5, in conjunction with FIGS. 1 and 4, in step 501, a plurality of physical ports 404 con-nected to a particular network device, e.g., router 104B, may be mapped into a logical interface 405. For example, refer-ring to FIG. 4, ports 404A-I of router 104A connected to router 104B may be logically grouped into a logical inter-face 405 thereby enabling network processor 403, e.g., network processor 403A, to transmit packets of

No.	’740 Patent Claim 1	Basso
		data across multiple physical ports 404 to a particular network device, e.g., router 104B, instead of one physical port 404 as discussed further below.”) Basso at [0040] (“In step 502, network processor 403, e.g., network processor 403A, may receive a packet of data from switch fabric 401. Upon receiving the packet of data, network processor 403, in step 503, may index into a table, com-monly referred to as a forwarding table, to determine the table associated with a particular logical interface 405 as well as the next destination address, i.e., the next hop address. The forwarding table may comprise a plurality of entries where each entry may comprise information indicat-ing a particular table associated with a particular logical interface 405 as well as the next destination address. Each logical interface 405 may be associated with a table storing a plurality of entries containing blade 402/port 404 combi-nations as discussed further below. In one embodiment, an entry may be indexed in the forwarding table using a destination address in the received packet header. It is noted that an entry may be indexed in the forwarding table using other means and that such means would be recognized by an artisan of ordinary skill in the art. It is further noted that embodiments implementing such means would fall within the scope of the present invention.”)
1[c]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel,	Basso discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. For example, Basso discloses coupling the blades to the communication network via link connections to the switch fabric. Basso at [0009] (“A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more ports. These ports may be connected to another one or more network devices. The switch fabric may be configured to direct incoming packets of data to particular blades where one or more of the network proces-sors in the recipient blade may be configured to process the received packets.”) Basso at [0030] (“Returning to FIG. 4, router 104A may be config-ured to receive packets of data such as from client 101 (FIG. 1), e.g., client 101A, that may be directed to another

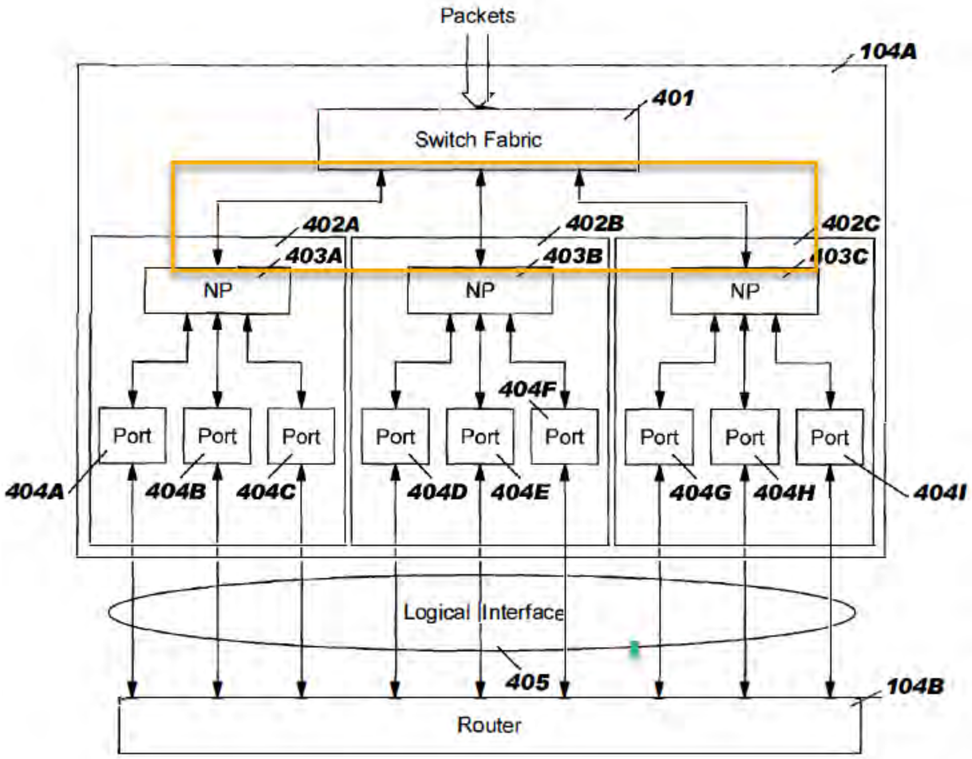
No.	’740 Patent Claim 1	Basso
		particular network device, e.g., router 104B, in network 100 (FIG. 1). Router 104A may comprise a switch fabric 401 configured to direct the incoming packets of data to particu-lar blades 402A-C coupled to switch fabric 401. Blade 402A may comprise a network processor 403A coupled with one or more ports 404A-C. Blade 402B may comprise a network processor 403B coupled with one or more ports 404D-F. Blade 402C may comprise a network processor 403C coupled with one or more ports 404G-I. Blades 402A-C may collectively or individually be referred to as blades 402 or blade 402, respectively. Network processors 403A-C may collectively or individually be referred to as network pro-cessors 403 or network processor 403, respectively. Ports 404A-I may collectively or individually be referred to as ports 404 or port 404, respectively. Each port 404 may be coupled to a particular network device, e.g., gateway, server, router such as router 104B, in network system 100. It is noted that some of ports 404 may be coupled to a separate network device and that FIG. 4 is illustrative. It is further noted that router 104A may comprise any number of blades 402 and each blade 402 may comprise any number of network processors 403 and ports 404.”) Basso at [0031] (“Referring to FIG. 4, network processor 403 may be configured to receive a packet of data from switch fabric 401. Upon receiving a packet, network processor 403 may be configured to process the packet of data. Processing may include but not limited to: determining what activities to be performed on the received packet, transmitting or discarding the received packet, determining which network device, e.g., router such as router 104B, server, edge device, gateway, to transmit the received packet, etc. Network processor 403 may then transmit the processed packet to a particular network device, e.g., router such as router 104B, through a port 404 connected to that network device.”) Basso at Figure 4 (annotation added; plurality of blades, i.e., interface modules, in red; links connecting the plurality of blades to the switch fabric, i.e., a second group of second physical links arranged in parallel)

No.	’740 Patent Claim 1	Basso
		FIG. 4  Packets 104A 401 Switch Fabric 402A 402B 402C 403A 403B 403C NP NP NP 404F Port Port Port Port Port Port Port Port Port 404A 404B 404C 404D 404E 404G 404H 404I Logical Interface 405 104B Router Basso at [0035] (“By logically grouping a plurality of ports 404 coupled to a particular network device into a logical inter-face 405, network processor 403 may be configured to transmit processed packets to that particular network device via any blade 402/port 404 combination grouped in that logical interface 405. For example, referring to FIG. 4, ports 404A-404I are physically connected to router 104B. If ports 404A-404I were logically grouped into logical interface 405, then a particular network processor 403, e.g., network processor 403A, may be

No.	’740 Patent Claim 1	Basso
		configured to transmit processed packets that are determined to be transmitted to router 104B through any of ports 404A-404I in blades 402A-C, respec-tively. Network processor 403, e.g., network processor 403A, may be configured to transmit the processed packets to router 104B through ports 404, e.g., ports 404D-I, not in its blade 402, e.g., blade 402A, by forwarding the processed packets to switch fabric 401 which may then direct the processed packets to another appropriate physical blade 402/port 404 combination. Network processor 403, e.g., network processor 403A, may further be configured to transmit the processed packets to router 104B through any ports 404, e.g., ports 404A-C, in its blade 402, e.g., blade402A, instead of just one physical port 404 in its blade 402, e.g., blade 402A. A more detailed description of routing packets implementing logical interface(s) 405 is provided below in FIG. 5.”) Basso at [0040] (“In step 502, network processor 403, e.g., network processor 403A, may receive a packet of data from switch fabric 401. Upon receiving the packet of data, network processor 403, in step 503, may index into a table, com-monly referred to as a forwarding table, to determine the table associated with a particular logical interface 405 as well as the next destination address, i.e., the next hop address. The forwarding table may comprise a plurality of entries where each entry may comprise information indicat-ing a particular table associated with a particular logical interface 405 as well as the next destination address. Each logical interface 405 may be associated with a table storing a plurality of entries containing blade 402/port 404 combi-nations as discussed further below. In one embodiment, an entry may be indexed in the forwarding table using a destination address in the received packet header. It is noted that an entry may be indexed in the forwarding table using other means and that such means would be recognized by an artisan of ordinary skill in the art. It is further noted that embodiments implementing such means would fall within the scope of the present invention.”) Basso at [0046] (“In step 507, the received packet may be transmitted through the blade 402/port 404 combination identified in step 506 to the destination identified in step 503. For example, referring to FIG. 4, if network processor 403A identified blade 402B/port 404D as being the blade 402/port 404 combination to transmit the processed packet, then network processor 403A may forward the processed packet to switch fabric 401 which may then direct the processed packet to blade 402B/port 404D.”)

No.	’740 Patent Claim 1	Basso
1[d]	at least one of said second physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction;	Basso discloses at least one of said second physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction. For example, Basso discloses connections between a plurality of blades and a switch fabric that are capable of both receiving and transmitting packets. Basso at [0009] (“A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more ports. These ports may be connected to another one or more network devices. The switch fabric may be configured to direct incoming packets of data to particular blades where one or more of the network proces-sors in the recipient blade may be configured to process the received packets.”) Basso at [0030] (“Returning to FIG. 4, router 104A may be config-ured to receive packets of data such as from client 101 (FIG. 1), e.g., client 101A, that may be directed to another particular network device, e.g., router 104B, in network 100 (FIG. 1). Router 104A may comprise a switch fabric 401 configured to direct the incoming packets of data to particu-lar blades 402A-C coupled to switch fabric 401. Blade 402A may comprise a network processor 403A coupled with one or more ports 404A-C. Blade 402B may comprise a network processor 403B coupled with one or more ports 404D-F. Blade 402C may comprise a network processor 403C coupled with one or more ports 404G-I. Blades 402A-C may collectively or individually be referred to as blades 402 or blade 402, respectively. Network processors 403A-C may collectively or individually be referred to as network pro-cessors 403 or network processor 403, respectively. Ports 404A-I may collectively or individually be referred to as ports 404 or port 404, respectively. Each port 404 may be coupled to a particular network device, e.g., gateway, server, router such as router 104B, in network system 100. It is noted that some of ports 404 may be coupled to a separate network device and that FIG. 4 is illustrative. It is further noted that router 104A may comprise any number of blades 402 and each blade 402 may comprise any number of network processors 403 and ports 404.”)

No.	’740 Patent Claim 1	Basso
		Basso at [0031] (“Referring to FIG. 4, network processor 403 may be configured to receive a packet of data from switch fabric 401. Upon receiving a packet, network processor 403 may be configured to process the packet of data. Processing may include but not limited to: determining what activities to be performed on the received packet, transmitting or discarding the received packet, determining which network device, e.g., router such as router 104B, server, edge device, gateway, to transmit the received packet, etc. Network processor 403 may then transmit the processed packet to a particular network device, e.g., router such as router 104B, through a port 404 connected to that network device.”) Basso at Figure 4 (annotation added)

No.	’740 Patent Claim 1	Basso
		FIG. 4  Packets 104A 401 Switch Fabric 402A 403A 402B 403B 402C 403C NP NP NP 404F Port Port Port Port Port Port Port Port Port 404A 404B 404C 404D 404E 404G 404H 404I Logical Interface 405 104B Router Basso at [0035] (“By logically grouping a plurality of ports 404 coupled to a particular network device into a logical inter-face 405, network processor 403 may be configured to transmit processed packets to that particular network device via any blade 402/port 404 combination grouped in that logical interface 405. For example, referring to FIG. 4, ports 404A-404I are physically connected to router 104B. If ports 404A-404I were logically grouped into logical interface 405, then a particular network processor 403, e.g., network processor 403A, may be

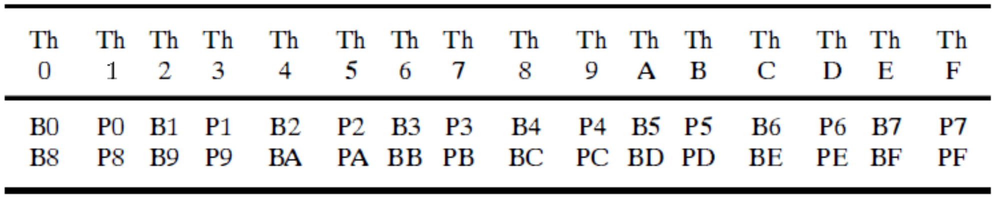
No.	’740 Patent Claim 1	Basso
		configured to transmit processed packets that are determined to be transmitted to router 104B through any of ports 404A-404I in blades 402A-C, respec-tively. Network processor 403, e.g., network processor 403A, may be configured to transmit the processed packets to router 104B through ports 404, e.g., ports 404D-I, not in its blade 402, e.g., blade 402A, by forwarding the processed packets to switch fabric 401 which may then direct the processed packets to another appropriate physical blade 402/port 404 combination. Network processor 403, e.g., network processor 403A, may further be configured to transmit the processed packets to router 104B through any ports 404, e.g., ports 404A-C, in its blade 402, e.g., blade402A, instead of just one physical port 404 in its blade 402, e.g., blade 402A. A more detailed description of routing packets implementing logical interface(s) 405 is provided below in FIG. 5.”)
1[e]	receiving a data frame having frame attributes sent between the communication network and the network node:	Basso discloses receiving a data frame having frame attributes sent between the communication network and the network node. For example, Basso discloses data packets with information, including headers, addresses, and ports sent between the communication network and network device. Basso at [0010] (“Upon a network processor receiving a packet of data, the network processor may index into a table, com-monly referred to as a forwarding table, to determine the table associated with a particular logical interface as well as the next destination address. The forwarding table may comprise a plurality of entries where each entry may com-prise information indicating a particular table associated with a particular logical interface as well as the next desti-nation address. Each logical interface may be associated with a table storing a plurality of entries containing blade/ port combinations as discussed further below. In one embodiment, an entry may be indexed in the forwarding table using a destination address in the received packet header.”) Basso at [0011] (“A hash function may then be performed on the received packet to generate a hash value. In one embodi-ment, a hash function may be performed on the source and destination address in the packet header to generate a hash value.”)

No.	’740 Patent Claim 1	Basso
		Basso at [0040] (“In step 502, network processor 403, e.g., network processor 403A, may receive a packet of data from switch fabric 401. Upon receiving the packet of data, network processor 403, in step 503, may index into a table, com-monly referred to as a forwarding table, to determine the table associated with a particular logical interface 405 as well as the next destination address, i.e., the next hop address. The forwarding table may comprise a plurality of entries where each entry may comprise information indicat-ing a particular table associated with a particular logical interface 405 as well as the next destination address. Each logical interface 405 may be associated with a table storing a plurality of entries containing blade 402/port 404 combi-nations as discussed further below. In one embodiment, an entry may be indexed in the forwarding table using a destination address in the received packet header. It is noted that an entry may be indexed in the forwarding table usingother means and that such means would be recognized by an artisan of ordinary skill in the art. It is further noted that embodiments implementing such means would fall within the scope of the present invention.”) Basso at [0041] (“In step 504, a hash function may be performed on the received packet to generate a hash value. In one embodi-ment, a hash function may be performed on the source and destination address in the packet header to generate a hash value. It is noted that in other embodiments a hash function may be performed on different fields, e.g., port, type of service, in the received packet to generate a hash value.”)
1[f]:	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	Basso discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. For example, Basso discloses using a hash function and index table to select for blade/port combinations over which to send the packet over a user port and a switch fabric link. Basso further discloses that this selection is performed based upon packet information. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. Basso at [0010] (“Upon a network processor receiving a packet of data, the network processor may index into a table, com-monly referred to as a forwarding table, to determine the table associated with a particular logical interface as well as the next destination address. The

No.	’740 Patent Claim 1	Basso
		forwarding table may comprise a plurality of entries where each entry may com-prise information indicating a particular table associated with a particular logical interface as well as the next desti-nation address. Each logical interface may be associated with a table storing a plurality of entries containing blade/ port combinations as discussed further below. In one embodiment, an entry may be indexed in the forwarding table using a destination address in the received packet header.”) Basso at [0011] (“A hash function may then be performed on the received packet to generate a hash value. In one embodi-ment, a hash function may be performed on the source and destination address in the packet header to generate a hash value.”) Basso at [0012] (“The hash value generated may be used to index into the table associated with a particular logical interface. Upon indexing into the table associated with the logical interface, an appropriate blade/port combination may be identified to transmit the received packet of data. In one embodiment, a blade/port combination may be selected in the indexed entry of the table associated with the logical interface by using a portion of the bits of the hashed value. The received packet may then be transmitted through the identified blade/port combination to the next destination (next destination previ-ously identified by the next destination address in the forwarding table).”) Basso at [0035] (“By logically grouping a plurality of ports 404 coupled to a particular network device into a logical inter-face 405, network processor 403 may be configured to transmit processed packets to that particular network device via any blade 402/port 404 combination grouped in that logical interface 405. For example, referring to FIG. 4, ports 404A-404I are physically connected to router 104B. If ports 404A-404I were logically grouped into logical interface 405, then a particular network processor 403, e.g., network processor 403A, may be configured to transmit processed packets that are determined to be transmitted to router 104B through any of ports 404A-404I in blades 402A-C, respec-tively. Network processor 403, e.g., network processor 403A, may be configured to transmit the processed packets to router 104B through ports 404, e.g., ports 404D-I, not in its blade 402, e.g., blade 402A, by forwarding the processed packets to switch fabric 401 which may then direct the processed packets to another appropriate physical blade 402/port 404 combination. Network processor 403, e.g., network

No.	’740 Patent Claim 1	Basso
		processor 403A, may further be configured to transmit the processed packets to router 104B through any ports 404, e.g., ports 404A-C, in its blade 402, e.g., blade402A, instead of just one physical port 404 in its blade 402, e.g., blade 402A. A more detailed description of routing packets implementing logical interface(s) 405 is provided below in FIG. 5.”) Basso at [0040] (“In step 502, network processor 403, e.g., network processor 403A, may receive a packet of data from switch fabric 401. Upon receiving the packet of data, network processor 403, in step 503, may index into a table, com-monly referred to as a forwarding table, to determine the table associated with a particular logical interface 405 as well as the next destination address, i.e., the next hop address. The forwarding table may comprise a plurality of entries where each entry may comprise information indicat-ing a particular table associated with a particular logical interface 405 as well as the next destination address. Each logical interface 405 may be associated with a table storing a plurality of entries containing blade 402/port 404 combi-nations as discussed further below. In one embodiment, an entry may be indexed in the forwarding table using a destination address in the received packet header. It is noted that an entry may be indexed in the forwarding table usingother means and that such means would be recognized by an artisan of ordinary skill in the art. It is further noted that embodiments implementing such means would fall within the scope of the present invention.”) Basso at [0041] (“In step 504, a hash function may be performed on the received packet to generate a hash value. In one embodi-ment, a hash function may be performed on the source and destination address in the packet header to generate a hash value. It is noted that in other embodiments a hash function may be performed on different fields, e.g., port, type of service, in the received packet to generate a hash value.”) Basso at [0042] (“In step 505, the hash value generated in step 504 may be used to index into the table associated with a particular logical interface 405 determined in step 503. Upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be identified in step 506 to transmit the received packet of data as explained below.”)

<table>
<tr><th>No.</th><th>’740 Patent Claim 1</th><th>Basso</th></tr>
<tr><td></td><td></td><td>Basso at [0043] (“As stated above, the table associated with a par-ticular logical interface 405 may comprise a plurality of entries where each entry may comprise a threshold value associated with a particular blade 402/port 404 combination. The threshold value may represent a percentage of the total number of packets received by router 104A that may be transmitted through the blade 402/port 404 combination associated with that threshold value. In one embodiment, the threshold value may be updated periodically by a user, e.g., system administrator, in control of router 104, e.g., router 104A. For example, the threshold value, e.g., twenty percent of the number of packets received by router 104A, associ-ated with a particular blade 402/port 404 combination may be updated by lowering the threshold value by one percent during each update. An example of an entry of the table associated with a particular logical interface 405 is shown in Table 1 below:

TABLE 1

<table>
<tr><th>Th
0</th><th>Th
1</th><th>Th
2</th><th>Th
3</th><th>Th
4</th><th>Th
5</th><th>Th
6</th><th>Th
7</th><th>Th
8</th><th>Th
9</th><th>Th
A</th><th>Th
B</th><th>Th
C</th><th>Th
D</th><th>Th
E</th><th>Th
F</th></tr>
<tr><td>B0
B8</td><td>P0
P8</td><td>B1
B9</td><td>P1
P9</td><td>B2
BA</td><td>P2
PA</td><td>B3
BB</td><td>P3
PB</td><td>B4
BC</td><td>P4
PC</td><td>B5
BD</td><td>P5
PD</td><td>B6
BE</td><td>P6
PE</td><td>B7
BF</td><td>P7
PF</td></tr>
</table>

Basso at [0044] (“Table 1 above illustrates an exemplary entry in the table associated with a particular logical interface 405. Each entry may comprise a plurality of threshold values (16 threshold values in exemplary Table 1) where each threshold value is associated with a particular blade 402/port 404 combination. For example, threshold value (Th0) is associ-ated with blade B0/port P0 combination where blade BO may refer to a particular blade 402, e.g., blade 402B, and port PO may refer to a particular port 404, e.g., port 404E. Threshold value (Thl) is associated with blade Bl/port Pl combination and so forth. As stated above, each threshold value may represent a percentage of the total number of packets received by router 104A that may be transmitted through the blade 402/port 404 combination associated with that threshold value. For example, threshold value (Th0) may represent a percentage of the total number of packets received by router 104A that may be transmitted through port PO in blade</td></tr>
</table>

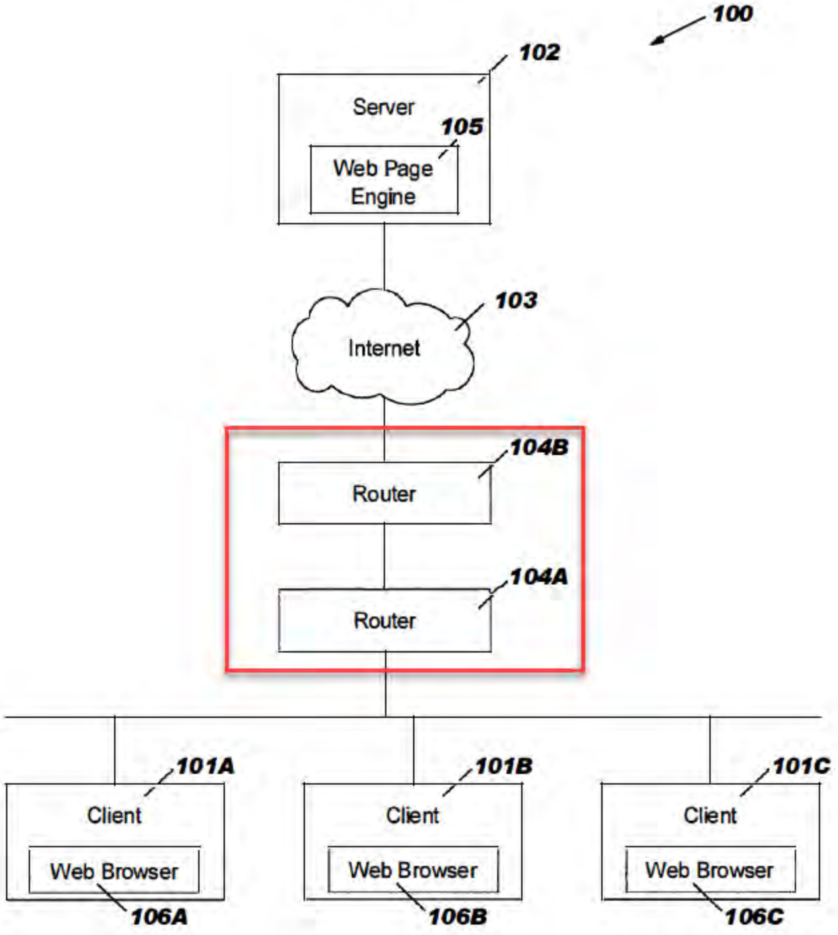
No.	’740 Patent Claim 1	Basso
		BO. If port PO refers to port 404D and blade BO refers to blade 402B, then if Th0 has a value of twenty percent, a maximum of twenty percent of the total packets received by router 104A may be transmitted through port 404D in blade 402B.”) Basso at [0045] (“As stated above, upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be identified in step 506 to transmit the received packet of data. In one embodiment, the hash value generated in step 504 may be used to select a particular threshold value and hence a blade 402/port 404 combination associated with the selected threshold value. In one embodiment, a portion of the bits of the hash value, e.g., most significant bits, may be used to select a particular threshold value in the entry indexed in step 505. For example, referring to Table 1, since there are 16 different threshold values in each entry of the table associated with logical interface 405, only four bits of the hash value generated in step 504 may be used to select a threshold value. Upon selecting a threshold value, the blade 402/port 404 combination associated with the selected threshold value may be used to transmit the received packet.”)
1[g]	sending the data frame over the selected first and second physical links,	Basso discloses sending the data frame over the selected first and second physical links. For example, Basso discloses transmitting the packet over the selected user port and the selected switch fabric link. Basso at [0012] (“The hash value generated may be used to index into the table associated with a particular logical interface. Upon indexing into the table associated with the logical interface, an appropriate blade/port combination may be identified to transmit the received packet of data. In one embodiment, a blade/port combination may be selected in the indexed entry of the table associated with the logical interface by using a portion of the bits of the hashed value. The received packet may then be transmitted through the identified blade/port combination to the next destination (next destination previ-ously identified by the next destination address In the forwarding table).”) Basso at [0035] (“By logically grouping a plurality of ports 404 coupled to a particular network device into a logical inter-face 405, network processor 403 may be configured to transmit

No.	’740 Patent Claim 1	Basso
		processed packets to that particular network device via any blade 402/port 404 combination grouped in that logical interface 405. For example, referring to FIG. 4, ports 404A-404I are physically connected to router 104B. If ports 404A-404I were logically grouped into logical interface 405, then a particular network processor 403, e.g., network processor 403A, may be configured to transmit processed packets that are determined to be transmitted to router 104B through any of ports 404A-404I in blades 402A-C, respec-tively. Network processor 403, e.g., network processor 403A, may be configured to transmit the processed packets to router 104B through ports 404, e.g., ports 404D-I, not in its blade 402, e.g., blade 402A, by forwarding the processed packets to switch fabric 401 which may then direct the processed packets to another appropriate physical blade 402/port 404 combination. Network processor 403, e.g., network processor 403A, may further be configured to transmit the processed packets to router 104B through any ports 404, e.g., ports 404A-C, in its blade 402, e.g., blade402A, instead of just one physical port 404 in its blade 402, e.g., blade 402A. A more detailed description of routing packets implementing logical interface(s) 405 is provided below in FIG. 5.”) Basso at [0045] (“As stated above, upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be identified in step 506 to transmit the received packet of data. In one embodiment, the hash value generated in step 504 may be used to select a particular threshold value and hence a blade 402/port 404 combination associated with the selected threshold value. In one embodiment, a portion of the bits of the hash value, e.g., most significant bits, may be used to select a particular threshold value in the entry indexed in step 505. For example, referring to Table 1, since there are 16 different threshold values in each entry of the table associated with logical interface 405, only four bits of the hash value generated in step 504 may be used to select a threshold value. Upon selecting a threshold value, the blade 402/port 404 combination associated with the selected threshold value may be used to transmit the received packet.”) Basso at [0046] (“In step 507, the received packet may be transmitted through the blade 402/port 404 combination identified in step 506 to the destination identified in step 503. For example, referring to FIG. 4, if network processor 403A identified blade 402B/port 404D as being the blade 402/port 404 combination to transmit the processed packet, then network

No.	’740 Patent Claim 1	Basso
		processor 403A may forward the processed packet to switch fabric 401 which may then direct the processed packet to blade 402B/port 404D.)
1[h]	said sending comprising communicating along at least one of said bi-directional links.	Basso discloses said sending comprising communicating along at least one of said bi-directional links. *See supra at* 1[b], 1[d], 1[g].

No.	’740 Patent Claim 2	Basso
2[a]	The method according to claim 1, wherein the network node comprises a user node, and	Basso discloses the method according to claim 1, wherein the network node comprises a user node. For example, Basso discloses a network device such as a router or switch. *See supra* at Claim 1. Basso at [0021] (“FIG. 1 illustrates one embodiment of the present invention of a network system 100. Network system 100 may comprise one or more clients 101A-C connected to a server 102 via the Internet 103. A more detailed description of server 102 is provided further below in conjunction with FIG. 3. The Internet 103 may refer to a network of com-puters. Network system 100 may further comprise one or more routers, e.g., 104A-B, that may be coupled to one or more clients lOlA-C. Routers 104A-B may be configured to forward packets of information from the one or more clients lOlA-C to the Internet 103. Clients lOlA-C may collec-tively or individually be referred to as clients 101 or client 101, respectively. A more detailed description of client 101 is provided further below in conjunction with FIG. 2. Routers 104A-B may collectively or individually be referred to as routers 104 or router 104, respectively. Amore detailed description of router 104 implementing logical interface(s) is provided further below in conjunction with FIG. 4. It is noted that network system 100 may comprise any number of clients 101, any number of servers 102 as well as any number of routers 104 and that FIG. 1 is illustrative. It is further noted that network system 100 may

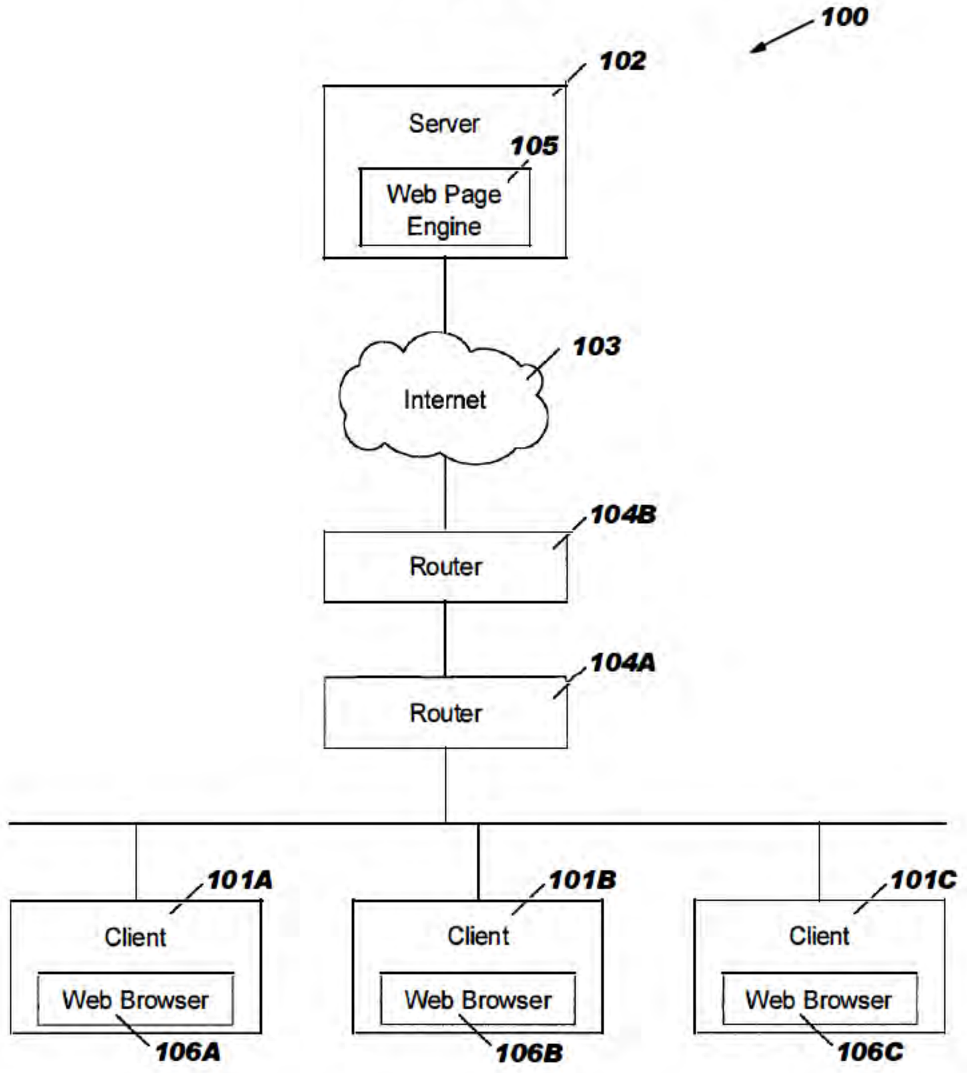
No.	’740 Patent Claim 2	Basso
		comprise one or more routers (not shown) that may be coupled to server 102. These routers (not shown) may be configured to forward received packets of information to server 102. It is further noted that the connection between clients 101 and the Internet 103 may be any medium type, e.g., wireless, wired. It is further noted that client 101 may be any type of device, e.g., wireless, Personal Digital Assistant (PDA), cell phone, personal computer system, workstation, Internet appliance, configured with the capability of connecting to the Internet 103 and consequently communicating with server 102. It is further noted that FIG. 1 is not to be limited in scope to any one particular embodiment.”) Basso at Figure 1 (annotation added)

No.	’740 Patent Claim 2	Basso
		FIG. 1  100 102 Server 105 Web Page Engine 103 Internet 104B Router 104A Router 101A Client Web Browser 106A 101B Client Web Browser 106B 101C Client Web Browser 106C

No.	’740 Patent Claim 2	Basso
		Basso at [0025] (“Referring to FIG. 2, client 101 may further com-prise a communications adapter 234 coupled to bus 212. Communications adapter 234 may enable client 101 to communicate with server 102 (FIG. 1), router 104 (FIG. 1). 1/0 devices may also be connected to system bus 212 via a user interface adapter 222 and a display adapter 236. Key-board 224, mouse 226 and speaker 230 may all be inter-connected to bus 212 through user interface adapter 222. Event data may be inputted to client 101 through any of these devices. A display monitor 238 may be connected to system bus 212 by display adapter 236. In this manner, a user is capable of inputting to client 101 through keyboard 224 or mouse 226 and receiving output from client 101 via display 238.”) Basso at [0029] (“FIG. 4 illustrates an embodiment of the present invention of router 104A (FIG. 1) implementing logical interface(s) as discussed further below. It is noted that even though the following discusses an embodiment of the present invention of router 104A that the description of router 104A applies to any router 104, e.g., router 104B, of network system 100 (FIG. 1).”) Basso at [0030] (“Returning to FIG. 4, router 104A may be config-ured to receive packets of data such as from client 101 (FIG. 1), e.g., client 101A, that may be directed to another particular network device, e.g., router 104B, in network 100 (FIG. 1). Router 104A may comprise a switch fabric 401 configured to direct the incoming packets of data to particu-lar blades 402A-C coupled to switch fabric 401. Blade 402A may comprise a network processor 403A coupled with one or more ports 404A-C. Blade 402B may comprise a network processor 403B coupled with one or more ports 404D-F. Blade 402C may comprise a network processor 403C coupled with one or more ports 404G-I. Blades 402A-C may collectively or individually be referred to as blades 402 or blade 402, respectively. Network processors 403A-C may collectively or individually be referred to as network pro-cessors 403 or network processor 403, respectively. Ports 404A-I may collectively or individually be referred to as ports 404 or port 404, respectively. Each port 404 may be coupled to a particular network device, e.g., gateway, server, router such as router 104B, in network system 100. It is noted that some of ports 404 may be coupled to a separate network device and that FIG. 4 is illustrative. It is further noted that router 104A may comprise any number of blades 402 and each blade 402 may comprise any number of network processors 403 and ports 404.”)

No.	’740 Patent Claim 2	Basso
		Basso at [0039] (“Referring to FIG. 5, in conjunction with FIGS. 1 and 4, in step 501, a plurality of physical ports 404 con-nected to a particular network device, e.g., router 104B, may be mapped into a logical interface 405. For example, refer-ring to FIG. 4, ports 404A-I of router 104A connected to router 104B may be logically grouped into a logical inter-face 405 thereby enabling network processor 403, e.g., network processor 403A, to transmit packets of data across multiple physical ports 404 to a particular network device, e.g., router 104B, instead of one physical port 404 as discussed further below.”)
2[b]	wherein sending the data frame comprises establishing a communication service between the user node and the communication network.	Basso discloses wherein sending the data frame comprises establishing a communication service between the user node and the communication network. For example, Basso discloses receiving and transmitting data packets, which establishes a communication system between the network device and network. Basso at [0021] (“FIG. 1 illustrates one embodiment of the present invention of a network system 100. Network system 100 may comprise one or more clients 101A-C connected to a server 102 via the Internet 103. A more detailed description of server 102 is provided further below in conjunction with FIG. 3. The Internet 103 may refer to a network of com-puters. Network system 100 may further comprise one or more routers, e.g., 104A-B, that may be coupled to one or more clients lOlA-C. Routers 104A-B may be configured to forward packets of information from the one or more clients lOlA-C to the Internet 103. Clients lOlA-C may collec-tively or individually be referred to as clients 101 or client 101, respectively. A more detailed description of client 101 is provided further below in conjunction with FIG. 2. Routers 104A-B may collectively or individually be referred to as routers 104 or router 104, respectively. Amore detailed description of router 104 implementing logical interface(s) is provided further below in conjunction with FIG. 4. It is noted that network system 100 may comprise any number of clients 101, any number of servers 102 as well as any number of routers 104 and that FIG. 1 is illustrative. It is further noted that network system 100 may comprise one or more routers (not shown) that may be coupled to server 102. These routers (not shown) may be configured to forward received packets of information to server 102. It is further noted that the connection between clients 101 and the Internet 103 may be any medium

No.	’740 Patent Claim 2	Basso
		type, e.g., wireless, wired. It is further noted that client 101 may be any type of device, e.g., wireless, Personal Digital Assistant (PDA), cell phone, personal computer system, workstation, Internet appliance, configured with the capability of connecting to the Internet 103 and consequently communicating with server 102. It is further noted that FIG. 1 is not to be limited in scope to any one particular embodiment.”) Basso at [0022] (“Referring to FIG. 1, server 102 may comprise a web page engine 105 for maintaining and providing access to an Internet web page which is enabled to forward static web pages as well as web pages to a web browser 106 of client 101 via the Internet 103. Each client lOlA-C may comprise a web browser 106A-C, respectively, which may be configured for communicating with the Internet 103 and for reading and executing web pages. Browsers 106A-C may collectively or individually be referred to as browsers 106 or browser 106, respectively. While the illustrated client engine is a web browser 106, those skilled in the art will recognize that other client engines may be used in accor-dance with the present invention.”) Basso at Figure 1

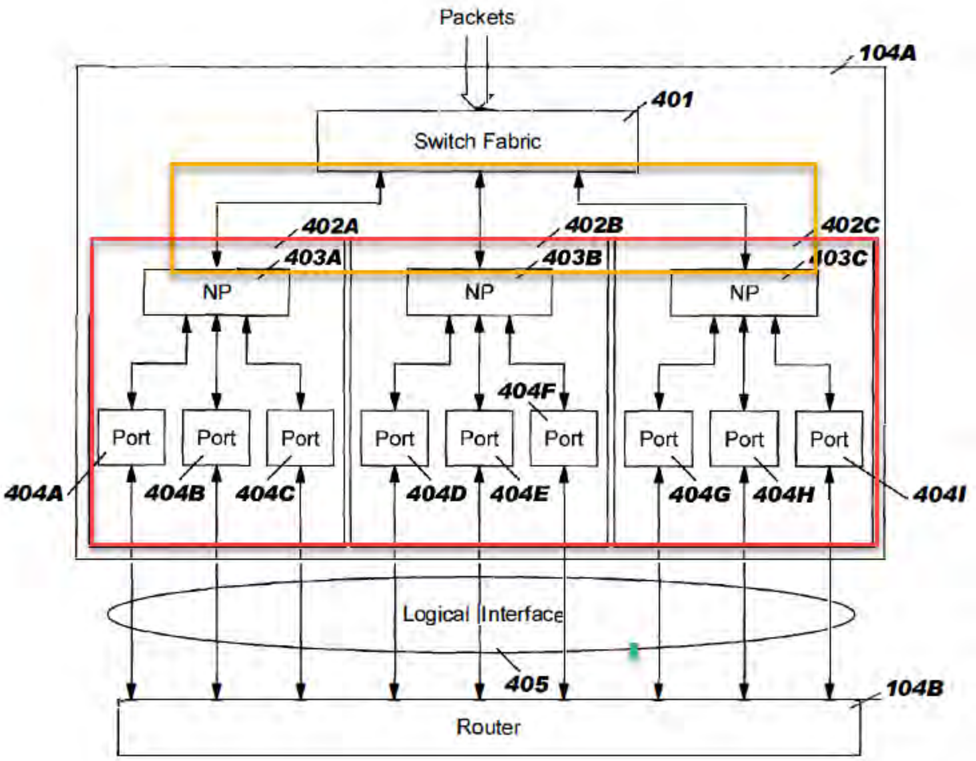
No.	’740 Patent Claim 2	Basso
		FIG. 1  100 102 Server 105 Web Page Engine 103 Internet 104B Router 104A Router 101A Client Web Browser 106A 101B Client Web Browser 106B 101C Client Web Browser 106C

No.	’740 Patent Claim 2	Basso
		Basso at [0025] (“Referring to FIG. 2, client 101 may further com-prise a communications adapter 234 coupled to bus 212. Communications adapter 234 may enable client 101 to communicate with server 102 (FIG. 1), router 104 (FIG. 1). 1/0 devices may also be connected to system bus 212 via a user interface adapter 222 and a display adapter 236. Key-board 224, mouse 226 and speaker 230 may all be inter-connected to bus 212 through user interface adapter 222. Event data may be inputted to client 101 through any of these devices. A display monitor 238 may be connected to system bus 212 by display adapter 236. In this manner, a user is capable of inputting to client 101 through keyboard 224 or mouse 226 and receiving output from client 101 via display 238.”) Basso at [0029] (“FIG. 4 illustrates an embodiment of the present invention of router 104A (FIG. 1) implementing logical interface(s) as discussed further below. It is noted that even though the following discusses an embodiment of the present invention of router 104A that the description of router 104A applies to any router 104, e.g., router 104B, of network system 100 (FIG. 1).”) Basso at [0030] (“Returning to FIG. 4, router 104A may be config-ured to receive packets of data such as from client 101 (FIG. 1), e.g., client 101A, that may be directed to another particular network device, e.g., router 104B, in network 100 (FIG. 1). Router 104A may comprise a switch fabric 401 configured to direct the incoming packets of data to particu-lar blades 402A-C coupled to switch fabric 401. Blade 402A may comprise a network processor 403A coupled with one or more ports 404A-C. Blade 402B may comprise a network processor 403B coupled with one or more ports 404D-F. Blade 402C may comprise a network processor 403C coupled with one or more ports 404G-I. Blades 402A-C may collectively or individually be referred to as blades 402 or blade 402, respectively. Network processors 403A-C may collectively or individually be referred to as network pro-cessors 403 or network processor 403, respectively. Ports 404A-I may collectively or individually be referred to as ports 404 or port 404, respectively. Each port 404 may be coupled to a particular network device, e.g., gateway, server, router such as router 104B, in network system 100. It is noted that some of ports 404 may be coupled to a separate network device and that FIG. 4 is illustrative. It is

No.	’740 Patent Claim 2	Basso
		further noted that router 104A may comprise any number of blades 402 and each blade 402 may comprise any number of network processors 403 and ports 404.”) Basso at [0034] (“Network processor 403 may be configured to trans-mit packets of data across multiple physical ports 404 to a particular network device, e.g., router 104B, by logically grouping a plurality of physical ports 404, e.g., ports 404A-I, into a logical interface 405. That is, the physical connections between router 104A and a particular network device, e.g., router 104B, may be logically grouped into a logical inter-face 405. Each set of physical connections between router 104A and a particular network device may be logically grouped into a particular logical interface 405. For example, if ports 404A-F were physically connected to network device #1, then ports 404A-F may be logically grouped into logical interface 405#1. If ports 404G-I were physically connected to network device #2, then ports 404G-I may be logically grouped into logical interface 405#2. It is noted that physical ports 404 may be logically grouped into more than one logical interface 405 and that FIG. 4 is illustrative. It is further noted that each logical interface 405 may be associated with logically grouping a plurality of physical connections to a particular network device, e.g., router, gateway, edge device, server. It is further noted that a logical interface 405 may be associated with logically grouping a plurality of ports in either one blade 402 or across multiple blades 402.”) Basso at [0039] (“Referring to FIG. 5, in conjunction with FIGS. 1 and 4, in step 501, a plurality of physical ports 404 con-nected to a particular network device, e.g., router 104B, may be mapped into a logical interface 405. For example, refer-ring to FIG. 4, ports 404A-I of router 104A connected to router 104B may be logically grouped into a logical inter-face 405 thereby enabling network processor 403, e.g., network processor 403A, to transmit packets of data across multiple physical ports 404 to a particular network device, e.g., router 104B, instead of one physical port 404 as discussed further below.”)

No.	’740 Patent Claim 3	Basso
3	The method according to claim 1, wherein the second physical links comprise backplane traces formed on a backplane to which the one or more interface modules are coupled.	Basso discloses the method according to claim 1, wherein the second physical links comprise backplane traces formed on a backplane to which the one or more interface modules are coupled. For example, Basso discloses coupling the plurality of blades to the switch fabric with physical connections. *See supra* at Claim 1. Basso at [0009] (“A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more ports. These ports may be connected to another one or more network devices. The switch fabric may be configured to direct incoming packets of data to particular blades where one or more of the network proces-sors in the recipient blade may be configured to process the received packets.”) Basso at [0030] (“Returning to FIG. 4, router 104A may be config-ured to receive packets of data such as from client 101 (FIG. 1), e.g., client 101A, that may be directed to another particular network device, e.g., router 104B, in network 100 (FIG. 1). Router 104A may comprise a switch fabric 401 configured to direct the incoming packets of data to particu-lar blades 402A-C coupled to switch fabric 401. Blade 402A may comprise a network processor 403A coupled with one or more ports 404A-C. Blade 402B may comprise a network processor 403B coupled with one or more ports 404D-F. Blade 402C may comprise a network processor 403C coupled with one or more ports 404G-I. Blades 402A-C may collectively or individually be referred to as blades 402 or blade 402, respectively. Network processors 403A-C may collectively or individually be referred to as network pro-cessors 403 or network processor 403, respectively. Ports 404A-I may collectively or individually be referred to as ports 404 or port 404, respectively. Each port 404 may be coupled to a particular network device, e.g., gateway, server, router such as router 104B, in network system 100. It is noted that some of ports 404 may be coupled to a separate network device and that FIG. 4 is illustrative. It is further noted that router 104A may comprise any number of blades 402 and each blade 402 may comprise any number of network processors 403 and ports 404.”)

No.	’740 Patent Claim 3	Basso
		Basso at [0031] (“Referring to FIG. 4, network processor 403 may be configured to receive a packet of data from switch fabric 401. Upon receiving a packet, network processor 403 may be configured to process the packet of data. Processing may include but not limited to: determining what activities to be performed on the received packet, transmitting or discarding the received packet, determining which network device, e.g., router such as router 104B, server, edge device, gateway, to transmit the received packet, etc. Network processor 403 may then transmit the processed packet to a particular network device, e.g., router such as router 104B, through a port 404 connected to that network device.”) Basso at Figure 4 (annotation added; plurality of blades, i.e., interface modules, in red; links connecting the plurality of blades to the switch fabric, i.e., a second group of second physical links arranged in parallel)

No.	’740 Patent Claim 3	Basso
		FIG. 4  Packets 104A 401 Switch Fabric 402A 403A 402B 403B 402C 403C NP NP NP 404F Port Port Port Port Port Port Port Port Port 404A 404B 404C 404D 404E 404G 404H 404I Logical Interface 405 104B Router Basso at [0035] (“By logically grouping a plurality of ports 404 coupled to a particular network device into a logical inter-face 405, network processor 403 may be configured to transmit processed packets to that particular network device via any blade 402/port 404 combination grouped in that logical interface 405. For example, referring to FIG. 4, ports 404A-404I are physically connected to router 104B. If ports 404A-404I were logically grouped into logical interface 405, then a particular network processor 403, e.g., network processor 403A, may be

No.	’740 Patent Claim 3	Basso
		configured to transmit processed packets that are determined to be transmitted to router 104B through any of ports 404A-404I in blades 402A-C, respec-tively. Network processor 403, e.g., network processor 403A, may be configured to transmit the processed packets to router 104B through ports 404, e.g., ports 404D-I, not in its blade 402, e.g., blade 402A, by forwarding the processed packets to switch fabric 401 which may then direct the processed packets to another appropriate physical blade 402/port 404 combination. Network processor 403, e.g., network processor 403A, may further be configured to transmit the processed packets to router 104B through any ports 404, e.g., ports 404A-C, in its blade 402, e.g., blade402A, instead of just one physical port 404 in its blade 402, e.g., blade 402A. A more detailed description of routing packets implementing logical interface(s) 405 is provided below in FIG. 5.”) Basso at [0040] (“In step 502, network processor 403, e.g., network processor 403A, may receive a packet of data from switch fabric 401. Upon receiving the packet of data, network processor 403, in step 503, may index into a table, com-monly referred to as a forwarding table, to determine the table associated with a particular logical interface 405 as well as the next destination address, i.e., the next hop address. The forwarding table may comprise a plurality of entries where each entry may comprise information indicat-ing a particular table associated with a particular logical interface 405 as well as the next destination address. Each logical interface 405 may be associated with a table storing a plurality of entries containing blade 402/port 404 combi-nations as discussed further below. In one embodiment, an entry may be indexed in the forwarding table using a destination address in the received packet header. It is noted that an entry may be indexed in the forwarding table using other means and that such means would be recognized by an artisan of ordinary skill in the art. It is further noted that embodiments implementing such means would fall within the scope of the present invention.”) Basso at [0046] (“In step 507, the received packet may be transmitted through the blade 402/port 404 combination identified in step 506 to the destination identified in step 503. For example, referring to FIG. 4, if network processor 403A identified blade 402B/port 404D as being the blade 402/port 404 combination to transmit the processed packet, then network processor 403A may forward the processed packet to switch fabric 401 which may then direct the processed packet to blade 402B/port 404D.”)

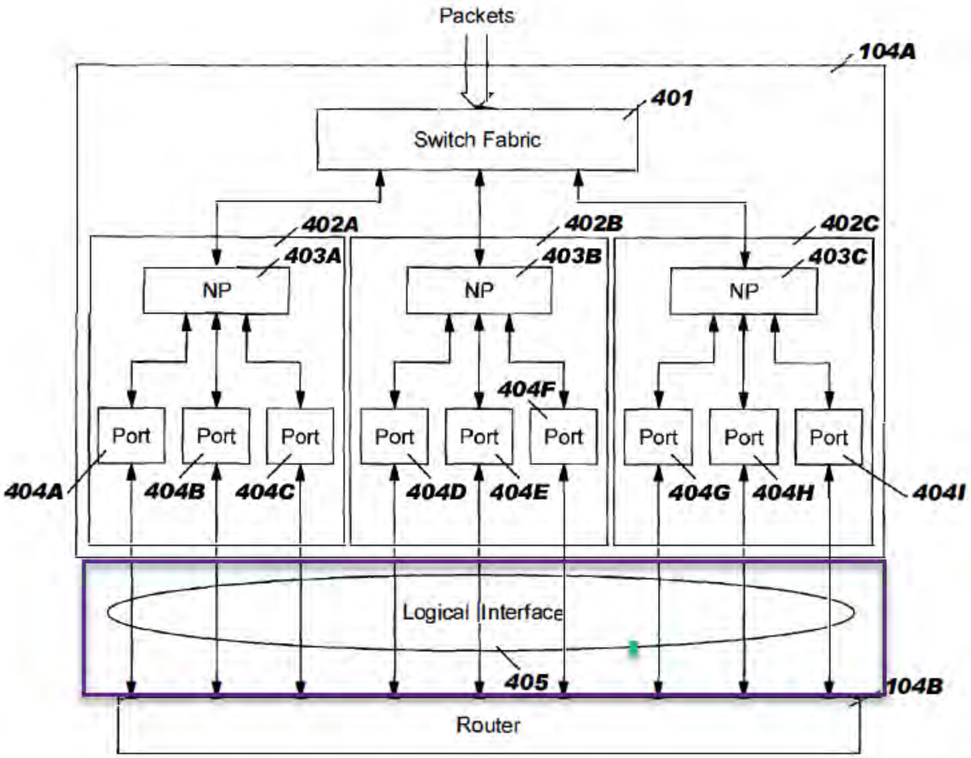
No.	’740 Patent Claim 3	Basso

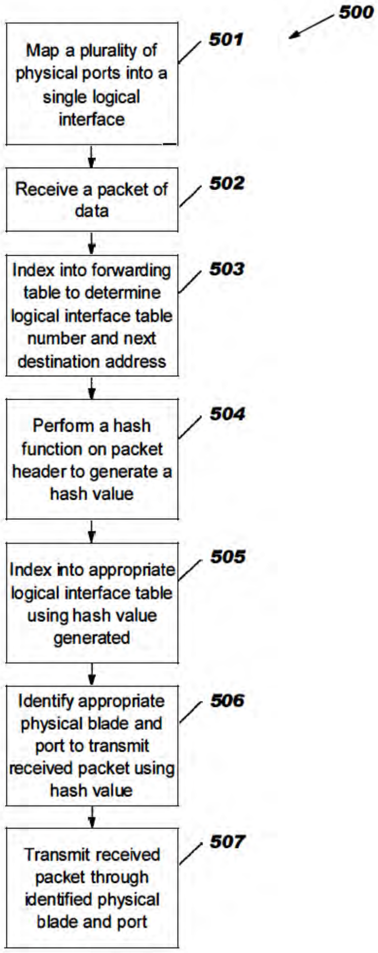
No.	’740 Patent Claim 4	Basso
4[preamble]	A method for communication, comprising:	Basso discloses a method for communication. *See supra* at 1[preamble].
4[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	Basso discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].
4[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Basso discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra* at 1[c].
4[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	Basso discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra* at 1[e].
4[d]	selecting, in a single computation based on	Basso discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group.

No.	’740 Patent Claim 4	Basso
	at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	*See supra* at 1[f].
4[e]	sending the data frame over the selected first and second physical links,	Basso discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].
4[f]	at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group.	Basso discloses at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group. For example, Basso discloses grouping the plurality of physical ports into a single logical interface. A person of ordinary skill in the art would understands other physical links could be aggregated as well. Basso at Abstract (“A method, system and computer program product for rout-ing packets. A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more physical ports. The physical ports may be connected to another one or more network devices. A plurality of physical ports across one or more blades connected to the same network device may be logically mapped into a logical interface to that network device. By logically grouping a plurality of physical ports into a logical interface to a network device, a network processor may be able to transmit packets of data to that network device across multiple ports instead of one physical port.”) Basso at [0007] (“The problems outlined above may at least in part be solved in some embodiments by mapping a plurality of physical ports connected to a network device into a logical interface to that network device, e.g., router, gateway, edge device, server. That is, a plurality of physical ports may be logically grouped into a logical interface to a network device

No.	’740 Patent Claim 4	Basso
		thereby enabling a network processor to transmit packets of data across multiple ports instead of one physical port.”) Basso at [0008] (“In one embodiment of the present invention, a method for routing packets may comprise the step of map-ping a plurality of physical ports in a network device, e.g., router, connected to another network device, e.g., router, gateway, edge device, server, into a logical interface to that network device. By logically grouping a plurality of ports into a logical interface, a network processor in the network device, e.g., router, may be able to transmit packets of data across multiple physical ports to a particular network device instead of one physical port as discussed further below.”) Basso at [0034] (“Network processor 403 may be configured to trans-mit packets of data across multiple physical ports 404 to a particular network device, e.g., router 104B, by logically grouping a plurality of physical ports 404, e.g., ports 404A-I, into a logical interface 405. That is, the physical connections between router 104A and a particular network device, e.g., router 104B, may be logically grouped into a logical inter-face 405. Each set of physical connections between router 104A and a particular network device may be logically grouped into a particular logical interface 405. For example, if ports 404A-F were physically connected to network device #1, then ports 404A-F may be logically grouped into logical interface 405#1. If ports 404G-I were physically connected to network device #2, then ports 404G-I may be logically grouped into logical interface 405#2. It is noted that physical ports 404 may be logically grouped into more than one logical interface 405 and that FIG. 4 is illustrative. It is further noted that each logical interface 405 may be associated with logically grouping a plurality of physical connections to a particular network device, e.g., router, gateway, edge device, server. It is further noted that a logical interface 405 may be associated with logically grouping a plurality of ports in either one blade 402 or across multiple blades 402.”) Basso at [0035] (“By logically grouping a plurality of ports 404 coupled to a particular network device into a logical inter-face 405, network processor 403 may be configured to transmit processed packets to that particular network device via any blade 402/port 404 combination grouped in that logical interface 405. For example, referring to FIG. 4, ports 404A-404I are physically connected to router 104B. If ports 404A-404I were logically grouped into logical

No.	’740 Patent Claim 4	Basso
		interface 405, then a particular network processor 403, e.g., network processor 403A, may be configured to transmit processed packets that are determined to be transmitted to router 104B through any of ports 404A-404I in blades 402A-C, respec-tively. Network processor 403, e.g., network processor 403A, may be configured to transmit the processed packets to router 104B through ports 404, e.g., ports 404D-I, not in its blade 402, e.g., blade 402A, by forwarding the processed packets to switch fabric 401 which may then direct the processed packets to another appropriate physical blade 402/port 404 combination. Network processor 403, e.g., network processor 403A, may further be configured to transmit the processed packets to router 104B through any ports 404, e.g., ports 404A-C, in its blade 402, e.g., blade 402A, instead of just one physical port 404 in its blade 402, e.g., blade 402A. A more detailed description of routing packets implementing logical interface(s) 405 is provided below in FIG. 5.”) Basso at Figure 4 (annotation added)

No.	’740 Patent Claim 4	Basso
		FIG. 4  Packets 104A 401 Switch Fabric 402A 403A NP 402B 403B NP 402C 403C NP 404F Port Port Port Port Port Port Port Port Port 404A 404B 404C 404D 404E 404G 404H 404I Logical Interface 405 104B Router Basso at [0039] (“Referring to FIG. 5, in conjunction with FIGS. 1 and 4, in step 501, a plurality of physical ports 404 con-nected to a particular network device, e.g., router 104B, may be mapped into a logical interface 405. For example, refer-ring to FIG. 4, ports 404A-I of router 104A connected to router 104B may be logically grouped into a logical inter-face 405 thereby enabling network processor 403, e.g., network processor 403A, to transmit

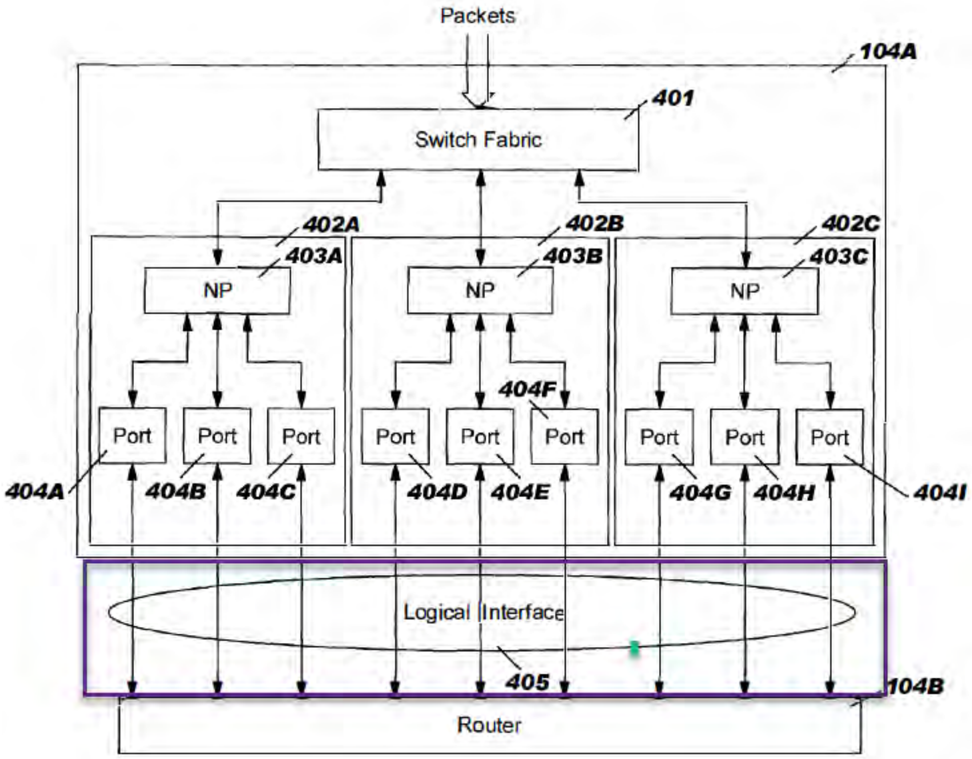
No.	’740 Patent Claim 4	Basso
		packets of data across multiple physical ports 404 to a particular network device, e.g., router 104B, instead of one physical port 404 as discussed further below.”) Basso at Figure 5 FIG. 5  500 Map a plurality of physical ports into a single logical interface 501 Receive a packet of data 502 Index into forwarding table to determine logical interface table number and next destination address 503 Perform a hash function on packet header to generate a hash value 504 Index into appropriate logical interface table using hash value generated 505 Identify appropriate physical blade and port to transmit received packet using hash value 506 Transmit received packet through identified physical blade and port 507

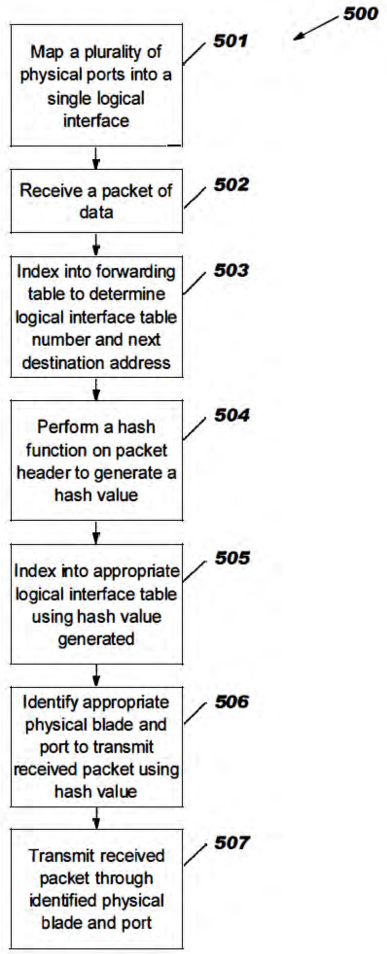
No.	’740 Patent Claim 5	Basso
5[preamble]	A method for communication, comprising:	Basso discloses a method for communication. *See supra at* 1[preamble].
5[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	Basso discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].
5[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Basso discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra at* 1[c].
5[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	Basso discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra at* 1[e].
5[d]	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link	Basso discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. *See supra at* 1[f].

No.	’740 Patent Claim 5	Basso
	out of the second group; and	
5[e]	sending the data frame over the selected first and second physical links,	Basso discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].
5[f]	coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node.	Basso discloses coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node. For example, Basso discloses grouping the plurality of physical ports which connect the network device to the plurality of blades into a single logical interface. Basso further discloses aggregating the physical ports to increase bandwidth. Basso at Abstract (“A method, system and computer program product for rout-ing packets. A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more physical ports. The physical ports may be connected to another one or more network devices. A plurality of physical ports across one or more blades connected to the same network device may be logically mapped into a logical interface to that network device. By logically grouping a plurality of physical ports into a logical interface to a network device, a network processor may be able to transmit packets of data to that network device across multiple ports instead of one physical port.”) Basso at [0007] (“The problems outlined above may at least in part be solved in some embodiments by mapping a plurality of physical ports connected to a network device into a logical interface to that network device, e.g., router, gateway, edge device, server. That is, a plurality of physical ports may be logically grouped into a logical interface to a network device thereby enabling a network processor to transmit packets of data across multiple ports instead of one physical port.”)

No.	’740 Patent Claim 5	Basso
		Basso at [0008] (“In one embodiment of the present invention, a method for routing packets may comprise the step of map-ping a plurality of physical ports in a network device, e.g., router, connected to another network device, e.g., router, gateway, edge device, server, into a logical interface to that network device. By logically grouping a plurality of ports into a logical interface, a network processor in the network device, e.g., router, may be able to transmit packets of data across multiple physical ports to a particular network device instead of one physical port as discussed further below.”) Basso at [0034] (“Network processor 403 may be configured to trans-mit packets of data across multiple physical ports 404 to a particular network device, e.g., router 104B, by logically grouping a plurality of physical ports 404, e.g., ports 404A-I, into a logical interface 405. That is, the physical connections between router 104A and a particular network device, e.g., router 104B, may be logically grouped into a logical inter-face 405. Each set of physical connections between router 104A and a particular network device may be logically grouped into a particular logical interface 405. For example, if ports 404A-F were physically connected to network device #1, then ports 404A-F may be logically grouped into logical interface 405#1. If ports 404G-I were physically connected to network device #2, then ports 404G-I may be logically grouped into logical interface 405#2. It is noted that physical ports 404 may be logically grouped into more than one logical interface 405 and that FIG. 4 is illustrative. It is further noted that each logical interface 405 may be associated with logically grouping a plurality of physical connections to a particular network device, e.g., router, gateway, edge device, server. It is further noted that a logical interface 405 may be associated with logically grouping a plurality of ports in either one blade 402 or across multiple blades 402.”) Basso at [0035] (“By logically grouping a plurality of ports 404 coupled to a particular network device into a logical inter-face 405, network processor 403 may be configured to transmit processed packets to that particular network device via any blade 402/port 404 combination grouped in that logical interface 405. For example, referring to FIG. 4, ports 404A-404I are physically connected to router 104B. If ports 404A-404I were logically grouped into logical interface 405, then a particular network processor 403, e.g., network processor 403A, may be configured to transmit processed packets that are determined to be transmitted to router 104B through any of ports 404A-404I in blades 402A-C, respec-tively. Network processor 403, e.g.,

No.	’740 Patent Claim 5	Basso
		network processor 403A, may be configured to transmit the processed packets to router 104B through ports 404, e.g., ports 404D-I, not in its blade 402, e.g., blade 402A, by forwarding the processed packets to switch fabric 401 which may then direct the processed packets to another appropriate physical blade 402/port 404 combination. Network processor 403, e.g., network processor 403A, may further be configured to transmit the processed packets to router 104B through any ports 404, e.g., ports 404A-C, in its blade 402, e.g., blade 402A, instead of just one physical port 404 in its blade 402, e.g., blade 402A. A more detailed description of routing packets implementing logical interface(s) 405 is provided below in FIG. 5.”) Basso at Figure 4 (annotation added)

No.	’740 Patent Claim 5	Basso
		FIG. 4  Packets 104A 401 Switch Fabric 402A 403A 402B 403B 402C 403C NP NP NP 404F Port Port Port Port Port Port Port Port Port 404A 404B 404C 404D 404E 404G 404H 404I Logical Interface 405 104B Router Basso at [0039] (“Referring to FIG. 5, in conjunction with FIGS. 1 and 4, in step 501, a plurality of physical ports 404 con-nected to a particular network device, e.g., router 104B, may be mapped into a logical interface 405. For example, refer-ring to FIG. 4, ports 404A-I of router 104A connected to router 104B may be logically grouped into a logical inter-face 405 thereby enabling network processor 403, e.g., network processor 403A, to transmit packets of

No.	’740 Patent Claim 5	Basso
		data across multiple physical ports 404 to a particular network device, e.g., router 104B, instead of one physical port 404 as discussed further below.”) Basso at Figure 5 FIG. 5  500 Map a plurality of physical ports into a single logical interface 501 Receive a packet of data 502 Index into forwarding table to determine logical interface table number and next destination address 503 Perform a hash function on packet header to generate a hash value 504 Index into appropriate logical interface table using hash value generated 505 Identify appropriate physical blade and port to transmit received packet using hash value 506 Transmit received packet through identified physical blade and port 507

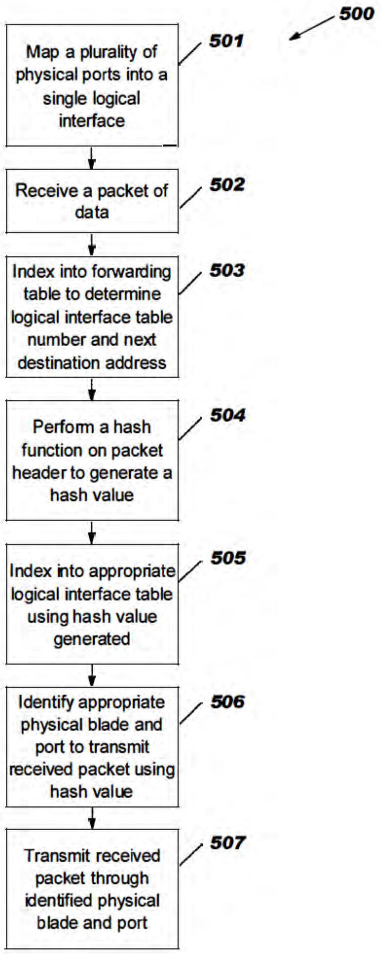
No.	’740 Patent Claim 5	Basso

No.	’740 Patent Claim 6	Basso
6	The method according to claim 1, wherein coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node.	Basso discloses the method according to claim 1, wherein coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node. For example, Basso discloses connecting the plurality of blades to the network, including grouping and distributing data packets sent in both directions between a network and network device. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Basso is found to not meet this limitation, wherein coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node would have been obvious to a person having ordinary skill in the art, as explained below. *See supra* at Claim 1. Basso at Abstract (“A method, system and computer program product for rout-ing packets. A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more physical ports. The physical ports may be connected to another one or more network devices. A plurality of physical ports across one or more blades connected to the same network device may be logically mapped into a logical interface to that network device. By logically grouping a plurality of physical ports into a logical interface to a network device, a network processor may be able to transmit packets of data to that network device across multiple ports instead of one physical port.”)

No.	’740 Patent Claim 6	Basso
		Basso at [0007] (“The problems outlined above may at least in part be solved in some embodiments by mapping a plurality of physical ports connected to a network device into a logical interface to that network device, e.g., router, gateway, edge device, server. That is, a plurality of physical ports may be logically grouped into a logical interface to a network device thereby enabling a network processor to transmit packets of data across multiple ports instead of one physical port.”) Basso at [0008] (“In one embodiment of the present invention, a method for routing packets may comprise the step of map-ping a plurality of physical ports in a network device, e.g., router, connected to another network device, e.g., router, gateway, edge device, server, into a logical interface to that network device. By logically grouping a plurality of ports into a logical interface, a network processor in the network device, e.g., router, may be able to transmit packets of data across multiple physical ports to a particular network device instead of one physical port as discussed further below.”) Basso at [0009] (“A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more ports. These ports may be connected to another one or more network devices. The switch fabric may be configured to direct incoming packets of data to particular blades where one or more of the network proces-sors in the recipient blade may be configured to process the received packets.”) Basso at [0030] (“Returning to FIG. 4, router 104A may be config-ured to receive packets of data such as from client 101 (FIG. 1), e.g., client 101A, that may be directed to another particular network device, e.g., router 104B, in network 100 (FIG. 1). Router 104A may comprise a switch fabric 401 configured to direct the incoming packets of data to particu-lar blades 402A-C coupled to switch fabric 401. Blade 402A may comprise a network processor 403A coupled with one or more ports 404A-C. Blade 402B may comprise a network processor 403B coupled with one or more ports 404D-F. Blade 402C may comprise a network processor 403C coupled with one or more ports 404G-I. Blades 402A-C may collectively or individually be referred to as blades 402 or blade 402, respectively. Network processors 403A-C may collectively or individually be referred to as network pro-cessors 403 or network processor

No.	’740 Patent Claim 6	Basso
		403, respectively. Ports 404A-I may collectively or individually be referred to as ports 404 or port 404, respectively. Each port 404 may be coupled to a particular network device, e.g., gateway, server, router such as router 104B, in network system 100. It is noted that some of ports 404 may be coupled to a separate network device and that FIG. 4 is illustrative. It is further noted that router 104A may comprise any number of blades 402 and each blade 402 may comprise any number of network processors 403 and ports 404.”) Basso at [0031] (“Referring to FIG. 4, network processor 403 may be configured to receive a packet of data from switch fabric 401. Upon receiving a packet, network processor 403 may be configured to process the packet of data. Processing may include but not limited to: determining what activities to be performed on the received packet, transmitting or discarding the received packet, determining which network device, e.g., router such as router 104B, server, edge device, gateway, to transmit the received packet, etc. Network processor 403 may then transmit the processed packet to a particular network device, e.g., router such as router 104B, through a port 404 connected to that network device.”) Basso at [0033] (“If, however, packets of data were able to be trans-mitted across multiple physical ports to a particular network device from a particular network processor, then the band-width capacity would be increased. That is, if packets of data were able to be transmitted across multiple physical ports to a particular network device by a network processor, then the rate packets of data are transmitted may be increased. It would therefore be desirable to enable a network processor to transmit packets of data across multiple physical ports to a particular network device instead of one physical port.”) Basso at [0034] (“Network processor 403 may be configured to trans-mit packets of data across multiple physical ports 404 to a particular network device, e.g., router 104B, by logically grouping a plurality of physical ports 404, e.g., ports 404A-I, into a logical interface 405. That is, the physical connections between router 104A and a particular network device, e.g., router 104B, may be logically grouped into a logical inter-face 405. Each set of physical connections between router 104A and a particular network device may be logically grouped into a particular logical interface 405. For example, if ports 404A-F were physically connected to network device #1, then ports 404A-F may be logically grouped into logical interface 405#1.

No.	’740 Patent Claim 6	Basso
		If ports 404G-I were physically connected to network device #2, then ports 404G-I may be logically grouped into logical interface 405#2. It is noted that physical ports 404 may be logically grouped into more than one logical interface 405 and that FIG. 4 is illustrative. It is further noted that each logical interface 405 may be associated with logically grouping a plurality of physical connections to a particular network device, e.g., router, gateway, edge device, server. It is further noted that a logical interface 405 may be associated with logically grouping a plurality of ports in either one blade 402 or across multiple blades 402.”) Basso at [0035] (“By logically grouping a plurality of ports 404 coupled to a particular network device into a logical inter-face 405, network processor 403 may be configured to transmit processed packets to that particular network device via any blade 402/port 404 combination grouped in that logical interface 405. For example, referring to FIG. 4, ports 404A-404I are physically connected to router 104B. If ports 404A-404I were logically grouped into logical interface 405, then a particular network processor 403, e.g., network processor 403A, may be configured to transmit processed packets that are determined to be transmitted to router 104B through any of ports 404A-404I in blades 402A-C, respec-tively. Network processor 403, e.g., network processor 403A, may be configured to transmit the processed packets to router 104B through ports 404, e.g., ports 404D-I, not in its blade 402, e.g., blade 402A, by forwarding the processed packets to switch fabric 401 which may then direct the processed packets to another appropriate physical blade 402/port 404 combination. Network processor 403, e.g., network processor 403A, may further be configured to transmit the processed packets to router 104B through any ports 404, e.g., ports 404A-C, in its blade 402, e.g., blade 402A, instead of just one physical port 404 in its blade 402, e.g., blade 402A. A more detailed description of routing packets implementing logical interface(s) 405 is provided below in FIG. 5.”) Basso at [0039] (“Referring to FIG. 5, in conjunction with FIGS. 1 and 4, in step 501, a plurality of physical ports 404 con-nected to a particular network device, e.g., router 104B, may be mapped into a logical interface 405. For example, refer-ring to FIG. 4, ports 404A-I of router 104A connected to router 104B may be logically grouped into a logical inter-face 405 thereby enabling network processor 403, e.g., network processor 403A, to transmit packets of

No.	’740 Patent Claim 6	Basso
		data across multiple physical ports 404 to a particular network device, e.g., router 104B, instead of one physical port 404 as discussed further below.”) Basso at Figure 5 FIG. 5  500 Map a plurality of physical ports into a single logical interface 501 Receive a packet of data 502 Index into forwarding table to determine logical interface table number and next destination address 503 Perform a hash function on packet header to generate a hash value 504 Index into appropriate logical interface table using hash value generated 505 Identify appropriate physical blade and port to transmit received packet using hash value 506 Transmit received packet through identified physical blade and port 507

No.	’740 Patent Claim 6	Basso
		Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Basso in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 6 of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Wiher discloses multiplexing and demultiplexing circuitry to transmit and receive ATM data cells over a data link between the ATM network and network access equipment. Wiher at 3:43-65 (“In general, in another aspect, the invention features an apparatus for communicating data cells between a data link and a backplane. The apparatus includes transceiver circuitry to transmit and receive data cells over a data link and a plurality of backplane interfaces each including at least one cell signal terminal. Each of the backplane interface is coupled to a backplane interconnection circuit. Each backplane interconnection circuit transmits and receives cells over the cell signal terminals of its associated backplane interface. The apparatus also includes de-multiplexing circuitry coupling the transceiver circuitry to each of the backplane interconnection circuits. The de-multiplexing circuitry receives a data cell from the transceiver circuitry, select a backplane interconnection circuit associated with the data cell, and provide the data cell to the selected backplane interconnection circuit for transmission over the cell signal terminals of the associated backplane interface. The apparatus also includes multiplexing circuitry coupling the plurality of backplane interconnection circuits to the transceiver circuitry. The multiplexing circuitry receives data cells from each of the backplane interconnection circuits and provide the received data cells to the transceiver circuitry.”) Wiher at 3:66-4:22 (“Implementations of the invention may include one or more of the following features. The backplane interconnection circuits may independently receive and transmit data cells over the plurality of backplane interfaces. The de-multiplexing circuitry may select a backplane interface based on data in the header field of the data cell. The apparatus may include header translation circuitry to alter header data in cells sent between the plurality of backplane interfaces and the transceiver circuitry. Each of the plurality of backplane

No.	’740 Patent Claim 6	Basso
		interfaces may include separate terminals to receive cells and separate terminals to transmit cells. The terminals to transmit cells may include a first and second control terminal and at least one outgoing cell data terminal. A backplane interface's backplane interconnection circuitry may accepts a signal on the first control terminal as indicating that a cell may be sent over the interface, asserts a 15 signal on the second control terminal to indicate that a cell is being transmitted, and transmits data bits of the cell on the outgoing cell data terminal. Each backplane interface may include a single outgoing cell data terminal and each bit of the cell may be serially transmitted over the single outgoing cell data terminal. Each backplane interface may include multiple outgoing cell data terminals and bits of the cell may be sent in parallel over the eight outgoing cell data terminals.”) For example, Lebizay discloses an optical add/drop multiplexer that multiplexers and demultiplexes data packets sent between the network boards and network. Lebizay at [0043] (“InfiniBand offers link layer Virtual Lanes (VLs) to support multiple logical channels (i.e. multiplexing) on the same physical link. Infiniband offers up to 16 virtual lanes per link. VLs provide a mechanism to avoid head-of-line blocking and the ability to support Quality of Service (QoS). The difference between a Virtual Lane and a Service Level (SL) is that a Virtual Lane is the actual logical lane (mul-tiplexed) used on a given point-to-point link. The Service Level stays constant as a packet traverses the fabric, and specifies the desired service level within a subnet. The SL (AF, EF or BE) is included in the link header, and each switch maps the SL to a VL supported by the destination link. A switch supporting a limited number of virtual lanes will map the SL field to a VL it supports. Without preserving the SL, the desired SL (AF, EF or BE) would be lost in this mapping, and later in the path, a switch supporting more VLs would be unable to recover finer granularity of SLs between two packets mapped to the same VL.”) Lebizay at [0050] (“The issue with using a ring, however, is how to map the addressing of multiple boards across these fibers. One solution is to employ Wavelength Division Multiplex-ing (WDM). A WDM optical mesh defines a meshed-topology in the wavelength space as opposed to the physical fiber space. By utilizing multiple discrete lambda-waves as optical carriers such that by meshing dedicated optical wavelengths between every two boards,

No.	’740 Patent Claim 6	Basso
		layer 2 protocols are eliminated, thereby creating a dramatic improvement in the efficiency of the transport. Today, every packet transport requires a protocol that allows the end point (and interme-diate points) to decipher the intended path (or consumer) of the packet. This protocol increases the amount of overhead required in the packet bus, allowing less room for actual data to be sent. By moving the protocol into the wavelength of the actual optical signal, the destination is implied by the wavelength and no additional bandwidth needs to be sur-rendered on the signal to provide this information. This makes the efficiency of the transport better and also speeds the routing of the packet through the network. In addition, the use of optical interconnects in a backplane environment greatly increases chassis bandwidth as well as reducing electrical radiation that often accompanies copper intercon-nects. The components involved include an optical back-plane in a physical ring topology, and the necessary trans-mitters and receivers for the size of the installation (i.e., number of slots in the chassis). In addition, optical add/drop multiplexer devices are required.”)

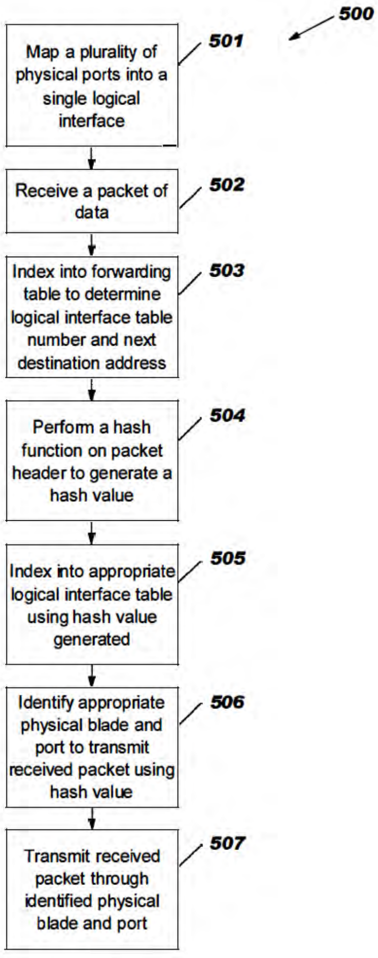
No.	’740 Patent Claim 7	Basso
7	The method according to claim 1, wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the first and second physical links.	Basso discloses the method according to claim 1, wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the first and second physical links. For example, Basso discloses selecting port and blade connections combinations to send data packets over based on a data transmission rate responsive to the bandwidth of the physical connections. *See supra* at Claim 1. Basso at [0004] (“Since the processed packets may be transmitted by the network processor across one particular address, the rate at which the processed packets are transmitted is limited to the bandwidth of the connection. That is, since the processed packets may be transmitted by the network processor across one physical port to a particular network device, the rate at which the processed packets are transmitted is limited to the bandwidth of the physical connection between the port and the network device.”)

No.	’740 Patent Claim 7	Basso
		Basso at [0005] (“If, however, the packets of data were able to be transmitted across multiple physical ports to a particular network device from a particular network processor, then the bandwidth capacity would be increased. That is, if packets of data were able to be transmitted across multiple physical ports to a particular network device by a network processor, then the rate packets of data are transmitted may be increased.”) Basso at [0006] (“It would therefore be desirable to enable a network processor to transmit packets of data across multiple physi-cal ports to a particular network device instead of one physical port.”) Basso at [0032] (“As stated in the Background Information section, each port may be physically coupled to another network device, e.g., router. Since the processed packets may be transmitted by a network processor across one particular address, the rate at which the processed packets are trans-mitted is limited to the bandwidth of the connection. That is, since the processed packets may be transmitted by the network processor across one physical port to a particular network device, the rate at which the processed packets are transmitted is limited to the bandwidth of the physical connection between the port and the network device.”) Basso at [0033] (“If, however, packets of data were able to be trans-mitted across multiple physical ports to a particular network device from a particular network processor, then the band-width capacity would be increased. That is, if packets of data were able to be transmitted across multiple physical ports to a particular network device by a network processor, then the rate packets of data are transmitted may be increased. It would therefore be desirable to enable a network processor to transmit packets of data across multiple physical ports to a particular network device instead of one physical port.”)

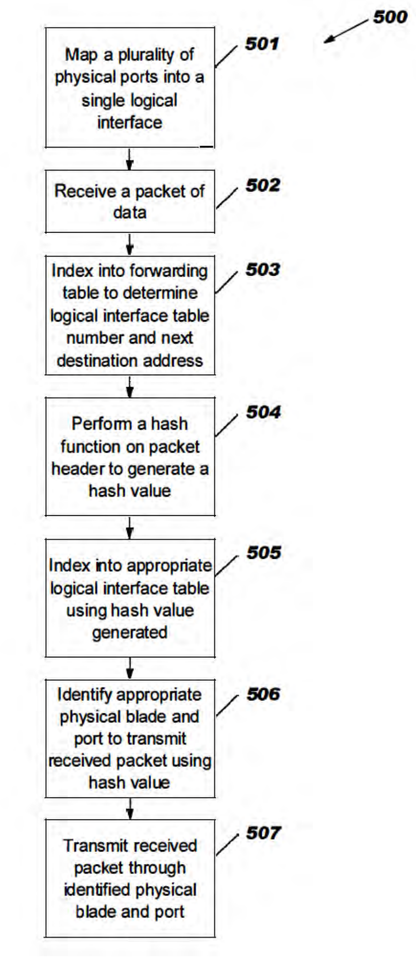
No.	’740 Patent Claim 8	Basso
8	The method according to claim 1, wherein selecting the first and	Basso discloses the method according to claim 1, wherein selecting the first and second physical links comprises applying a mapping function to the at least one of the frame attributes.

No.	’740 Patent Claim 8	Basso
	second physical links comprises applying a mapping function to the at least one of the frame attributes.	For example, Basso discloses mapping data packets to port and blade combinations (first and second physical links) based on a hash function performed on packet information. *See supra* at Claim 1. Basso at Abstract (“A method, system and computer program product for rout-ing packets. A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more physical ports. The physical ports may be connected to another one or more network devices. A plurality of physical ports across one or more blades connected to the same network device may be logically mapped into a logical interface to that network device. By logically grouping a plurality of physical ports into a logical interface to a network device, a network processor may be able to transmit packets of data to that network device across multiple ports instead of one physical port.”) Basso at [0007] (“The problems outlined above may at least in part be solved in some embodiments by mapping a plurality of physical ports connected to a network device into a logical interface to that network device, e.g., router, gateway, edge device, server. That is, a plurality of physical ports may be logically grouped into a logical interface to a network device thereby enabling a network processor to transmit packets of data across multiple ports instead of one physical port.”) Basso at [0008] (“In one embodiment of the present invention, a method for routing packets may comprise the step of map-ping a plurality of physical ports in a network device, e.g., router, connected to another network device, e.g., router, gateway, edge device, server, into a logical interface to that network device. By logically grouping a plurality of ports into a logical interface, a network processor in the network device, e.g., router, may be able to transmit packets of data across multiple physical ports to a particular network device instead of one physical port as discussed further below.”) Basso at [0009] (“A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled

No.	’740 Patent Claim 8	Basso
		to one or more ports. These ports may be connected to another one or more network devices. The switch fabric may be configured to direct incoming packets of data to particular blades where one or more of the network proces-sors in the recipient blade may be configured to process the received packets.”) Basso at [0039] (“Referring to FIG. 5, in conjunction with FIGS. 1 and 4, in step 501, a plurality of physical ports 404 con-nected to a particular network device, e.g., router 104B, may be mapped into a logical interface 405. For example, refer-ring to FIG. 4, ports 404A-I of router 104A connected to router 104B may be logically grouped into a logical inter-face 405 thereby enabling network processor 403, e.g., network processor 403A, to transmit packets of data across multiple physical ports 404 to a particular network device, e.g., router 104B, instead of one physical port 404 as discussed further below.”) Basso at Figure 5

No.	’740 Patent Claim 8	Basso
		FIG. 5  500 Map a plurality of physical ports into a single logical interface 501 Receive a packet of data 502 Index into forwarding table to determine logical interface table number and next destination address 503 Perform a hash function on packet header to generate a hash value 504 Index into appropriate logical interface table using hash value generated 505 Identify appropriate physical blade and port to transmit received packet using hash value 506 Transmit received packet through identified physical blade and port 507

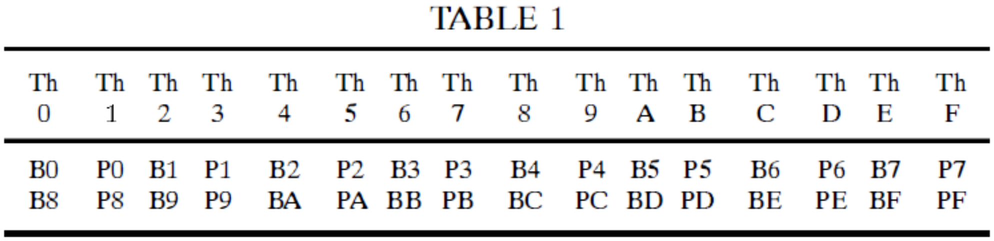
No.	’740 Patent Claim 9	Basso
9	The method according to claim 8, wherein applying the mapping function comprises applying a hashing function.	Basso discloses the method according to claim 8, wherein applying the mapping function comprises applying a hashing function. For example, Basso discloses applying a hash function to map the data packets over specific blade and port combinations. *See supra* at Claim 8. Basso at [0011] (“A hash function may then be performed on the received packet to generate a hash value. In one embodi-ment, a hash function may be performed on the source and destination address in the packet header to generate a hash value.”) Basso at [0012] (“The hash value generated may be used to index into the table associated with a particular logical interface. Upon indexing into the table associated with the logical interface, an appropriate blade/port combination may be identified to transmit the received packet of data. In one embodiment, a blade/port combination may be selected in the indexed entry of the table associated with the logical interface by using a portion of the bits of the hashed value. The received packet may then be transmitted through the identified blade/port combination to the next destination (next destination previ-ously identified by the next destination address in the forwarding table).”) Basso at [0041] (“In step 504, a hash function may be performed on the received packet to generate a hash value. In one embodi-ment, a hash function may be performed on the source and destination address in the packet header to generate a hash value. It is noted that in other embodiments a hash function may be performed on different fields, e.g., port, type of service, in the received packet to generate a hash value.”) Basso at [0042] (“In step 505, the hash value generated in step 504 may be used to index into the table associated with a particular logical interface 405 determined in step 503. Upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be identified in step 506 to transmit the received packet of data as explained below.”)

No.	’740 Patent Claim 9	Basso
		Basso at Figure 5 FIG. 5  500 501 Map a plurality of physical ports into a single logical interface 502 Receive a packet of data 503 Index into forwarding table to determine logical interface table number and next destination address 504 Perform a hash function on packet header to generate a hash value 505 Index into appropriate logical interface table using hash value generated 506 Identify appropriate physical blade and port to transmit received packet using hash value 507 Transmit received packet through identified physical blade and port Basso at [0045] (“As stated above, upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be

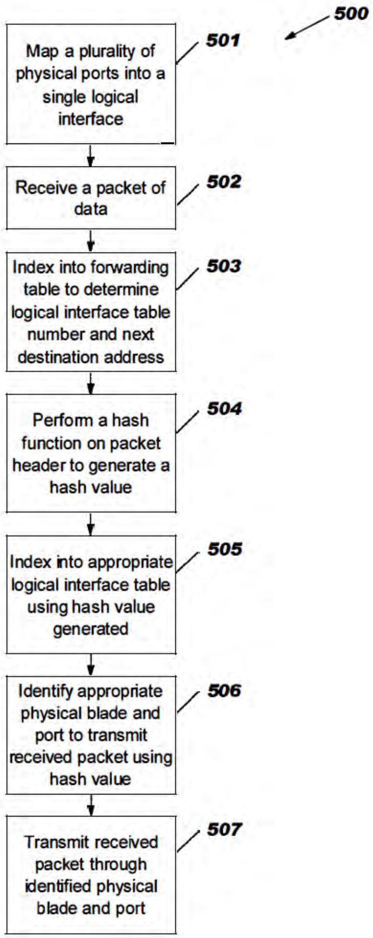
No.	’740 Patent Claim 9	Basso
		identified in step 506 to transmit the received packet of data. In one embodiment, the hash value generated in step 504 may be used to select a particular threshold value and hence a blade 402/port 404 combination associated with the selected threshold value. In one embodiment, a portion of the bits of the hash value, e.g., most significant bits, may be used to select a particular threshold value in the entry indexed in step 505. For example, referring to Table 1, since there are 16 different threshold values in each entry of the table associated with logical interface 405, only four bits of the hash value generated in step 504 may be used to select a threshold value. Upon selecting a threshold value, the blade 402/port 404 combination associated with the selected threshold value may be used to transmit the received packet.”)

No.	’740 Patent Claim 10	Basso
10[a]	The method according to claim 9, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second physical links,	Basso discloses the method according to claim 9, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second physical links. For example, Basso discloses applying a hash function that includes determining the number of blades and ports and associated threshold values. *See supra at* Claim 9. Basso at [0011] (“A hash function may then be performed on the received packet to generate a hash value. In one embodi-ment, a hash function may be performed on the source and destination address in the packet header to generate a hash value.”) Basso at [0012] (“The hash value generated may be used to index into the table associated with a particular logical interface. Upon indexing into the table associated with the logical interface, an appropriate blade/port combination may be identified to transmit the received packet of data. In one embodiment, a blade/port combination may be selected in the indexed entry of the table associated with the logical interface by using a portion of the bits of the hashed value. The received

No.	’740 Patent Claim 10	Basso
		packet may then be transmitted through the identified blade/port combination to the next destination (next destination previ-ously identified by the next destination address in the forwarding table).”) Basso at [0030] (“Returning to FIG. 4, router 104A may be config-ured to receive packets of data such as from client 101 (FIG. 1), e.g., client 101A, that may be directed to another particular network device, e.g., router 104B, in network 100 (FIG. 1). Router 104A may comprise a switch fabric 401 configured to direct the incoming packets of data to particu-lar blades 402A-C coupled to switch fabric 401. Blade 402A may comprise a network processor 403A coupled with one or more ports 404A-C. Blade 402B may comprise a network processor 403B coupled with one or more ports 404D-F. Blade 402C may comprise a network processor 403C coupled with one or more ports 404G-I. Blades 402A-C may collectively or individually be referred to as blades 402 or blade 402, respectively. Network processors 403A-C may collectively or individually be referred to as network pro-cessors 403 or network processor 403, respectively. Ports 404A-I may collectively or individually be referred to as ports 404 or port 404, respectively. Each port 404 may be coupled to a particular network device, e.g., gateway, server, router such as router 104B, in network system 100. It is noted that some of ports 404 may be coupled to a separate network device and that FIG. 4 is illustrative. It is further noted that router 104A may comprise any number of blades 402 and each blade 402 may comprise any number of network processors 403 and ports 404.”) Basso at [0041] (“In step 504, a hash function may be performed on the received packet to generate a hash value. In one embodi-ment, a hash function may be performed on the source and destination address in the packet header to generate a hash value. It is noted that in other embodiments a hash function may be performed on different fields, e.g., port, type of service, in the received packet to generate a hash value.”) Basso at [0042] (“In step 505, the hash value generated in step 504 may be used to index into the table associated with a particular logical interface 405 determined in step 503. Upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade

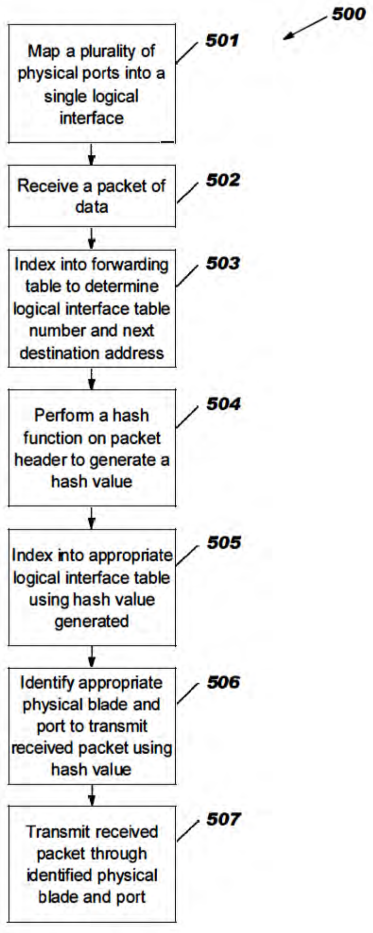
<table>
<tr><th>No.</th><th>’740 Patent Claim 10</th><th>Basso</th></tr>
<tr><td></td><td></td><td>
<p>402/port 404 combination may be identified in step 506 to transmit the received packet of data as explained below.”)</p>
<p>Basso at [0043] (“As stated above, the table associated with a par-ticular logical interface 405 may comprise a plurality of entries where each entry may comprise a threshold value associated with a particular blade 402/port 404 combination. The threshold value may represent a percentage of the total number of packets received by router 104A that may be transmitted through the blade 402/port 404 combination associated with that threshold value. In one embodiment, the threshold value may be updated periodically by a user, e.g., system administrator, in control of router 104, e.g., router 104A. For example, the threshold value, e.g., twenty percent of the number of packets received by router 104A, associ-ated with a particular blade 402/port 404 combination may be updated by lowering the threshold value by one percent during each update. An example of an entry of the table associated with a particular logical interface 405 is shown in Table 1 below:</p>

<p>TABLE 1</p>
<table>
<tr><th>Th
0</th><th>Th
1</th><th>Th
2</th><th>Th
3</th><th>Th
4</th><th>Th
5</th><th>Th
6</th><th>Th
7</th><th>Th
8</th><th>Th
9</th><th>Th
A</th><th>Th
B</th><th>Th
C</th><th>Th
D</th><th>Th
E</th><th>Th
F</th></tr>
<tr><td>B0
B8</td><td>P0
P8</td><td>B1
B9</td><td>P1
P9</td><td>B2
BA</td><td>P2
PA</td><td>B3
BB</td><td>P3
PB</td><td>B4
BC</td><td>P4
PC</td><td>B5
BD</td><td>P5
PD</td><td>B6
BE</td><td>P6
PE</td><td>B7
BF</td><td>P7
PF</td></tr>
</table>
<p>”)</p>
<p>Basso at [0044] (“Table 1 above illustrates an exemplary entry in the table associated with a particular logical interface 405. Each entry may comprise a plurality of threshold values (16 threshold values in exemplary Table 1) where each threshold value is associated with a particular blade 402/port 404 combination. For example, threshold value (Th0) is associ-ated with blade BO/port PO combination where blade BO may refer to a particular blade 402, e.g., blade 402B, and port PO may refer to a particular port 404, e.g., port 404E. Threshold value (Thl) is associated with blade Bl/port Pl combination and so forth. As stated above, each threshold value may represent a percentage of the total number of packets received by router 104A that may be</p>
</td></tr>
</table>

No.	’740 Patent Claim 10	Basso
		transmitted through the blade 402/port 404 combination associated with that threshold value. For example, threshold value (Th0) may represent a percentage of the total number of packets received by router 104A that may be transmitted throughport PO in blade BO. If port PO refers to port 404D and blade BO refers to blade 402B, then if ThO has a value of twenty percent, a maximum of twenty percent of the total packets received by router 104A may be transmitted through port 404D in blade 402B.”) Basso at Figure 5

No.	’740 Patent Claim 10	Basso
		FIG. 5  500 Map a plurality of physical ports into a single logical interface 501 Receive a packet of data 502 Index into forwarding table to determine logical interface table number and next destination address 503 Perform a hash function on packet header to generate a hash value 504 Index into appropriate logical interface table using hash value generated 505 Identify appropriate physical blade and port to transmit received packet using hash value 506 Transmit received packet through identified physical blade and port 507 Basso at [0045] (“As stated above, upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be identified in step 506 to transmit the received packet of data. In one embodiment, the hash value

No.	’740 Patent Claim 10	Basso
		generated in step 504 may be used to select a particular threshold value and hence a blade 402/port 404 combination associated with the selected threshold value. In one embodiment, a portion of the bits of the hash value, e.g., most significant bits, may be used to select a particular threshold value in the entry indexed in step 505. For example, referring to Table 1, since there are 16 different threshold values in each entry of the table associated with logical interface 405, only four bits of the hash value generated in step 504 may be used to select a threshold value. Upon selecting a threshold value, the blade 402/port 404 combination associated with the selected threshold value may be used to transmit the received packet.”)
10[b]	applying the hashing function to the at least one of the frame attributes to produce a hashing key,	Basso discloses applying the hashing function to the at least one of the frame attributes to produce a hashing key. For example, Basso discloses performing a hash function on different fields of the received packet to generate a hash value. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Basso is found to not meet this limitation, applying the hashing function to the at least one of the frame attributes to produce a hashing key would have been obvious to a person having ordinary skill in the art, as explained below. Basso at [0011] (“A hash function may then be performed on the received packet to generate a hash value. In one embodi-ment, a hash function may be performed on the source and destination address in the packet header to generate a hash value.”) Basso at [0012] (“The hash value generated may be used to index into the table associated with a particular logical interface. Upon indexing into the table associated with the logical interface, an appropriate blade/port combination may be identified to transmit the received packet of data. In one embodiment, a blade/port combination may be selected in the indexed entry of the table associated with the logical interface by using a portion of the bits of the hashed value. The received packet may then be transmitted through the identified blade/port combination to the next

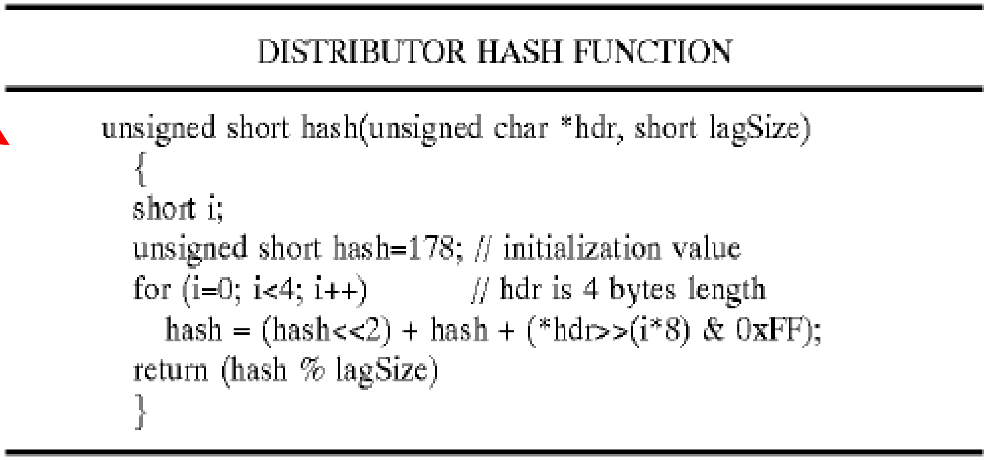
No.	’740 Patent Claim 10	Basso
		destination (next destination previ-ously identified by the next destination address in the forwarding table).”) Basso at [0030] (“Returning to FIG. 4, router 104A may be config-ured to receive packets of data such as from client 101 (FIG. 1), e.g., client 101A, that may be directed to another particular network device, e.g., router 104B, in network 100 (FIG. 1). Router 104A may comprise a switch fabric 401 configured to direct the incoming packets of data to particu-lar blades 402A-C coupled to switch fabric 401. Blade 402A may comprise a network processor 403A coupled with one or more ports 404A-C. Blade 402B may comprise a network processor 403B coupled with one or more ports 404D-F. Blade 402C may comprise a network processor 403C coupled with one or more ports 404G-I. Blades 402A-C may collectively or individually be referred to as blades 402 or blade 402, respectively. Network processors 403A-C may collectively or individually be referred to as network pro-cessors 403 or network processor 403, respectively. Ports 404A-I may collectively or individually be referred to as ports 404 or port 404, respectively. Each port 404 may be coupled to a particular network device, e.g., gateway, server, router such as router 104B, in network system 100. It is noted that some of ports 404 may be coupled to a separate network device and that FIG. 4 is illustrative. It is further noted that router 104A may comprise any number of blades 402 and each blade 402 may comprise any number of network processors 403 and ports 404.”) Basso at [0041] (“In step 504, a hash function may be performed on the received packet to generate a hash value. In one embodi-ment, a hash function may be performed on the source and destination address in the packet header to generate a hash value. It is noted that in other embodiments a hash function may be performed on different fields, e.g., port, type of service, in the received packet to generate a hash value.”) Basso at [0042] (“In step 505, the hash value generated in step 504 may be used to index into the table associated with a particular logical interface 405 determined in step 503. Upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be identified in step 506 to transmit the received packet of data as explained below.”)

No.	’740 Patent Claim 10	Basso
		Basso at Figure 5 FIG. 5  500 Map a plurality of physical ports into a single logical interface 501 Receive a packet of data 502 Index into forwarding table to determine logical interface table number and next destination address 503 Perform a hash function on packet header to generate a hash value 504 Index into appropriate logical interface table using hash value generated 505 Identify appropriate physical blade and port to transmit received packet using hash value 506 Transmit received packet through identified physical blade and port 507 Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Basso in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further

No.	’740 Patent Claim 10	Basso
		combination with (2) each (individually, as well as one or more together) of the references identified in element 10[b] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses applying a distributor hash function to the frame information which includes determining a number of the plurality of physical links. Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”)

No.	’740 Patent Claim 10	Basso
		Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”) Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:

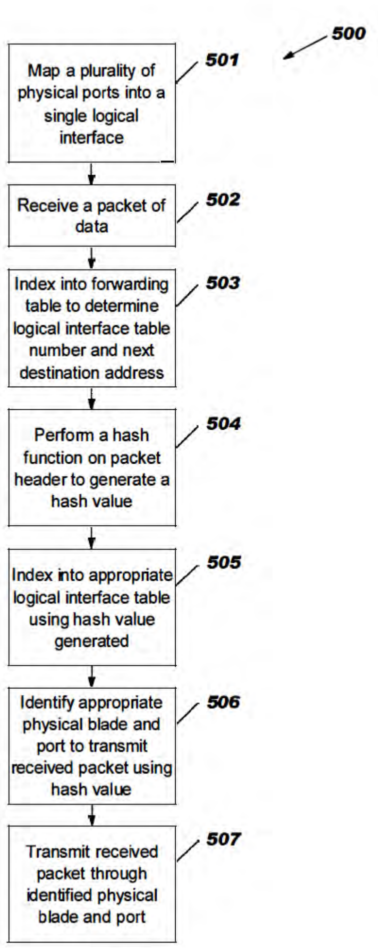
No.	’740 Patent Claim 10	Basso

		Bruckman at Table 1 (annotated)
		hashing function *“mapping function”*
		
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		For example, Alexander discloses applying a distributor hash function to the frame information which includes determining a hash key based on packet information.
		Alexander at 3:1-40 (“The hash function is preferably selected such that suc-cessive application of the hash function to all source and destination addresses expected to be seen by the Ethernet switch will produce a lowest value hash key, a highest value hash key, and a group of hash keys having intermediate values distributed evenly between the lowest and highest values.

No.	’740 Patent Claim 10	Basso
		The distribution table contains a separate port identifier look-up table for each aggregated grouping of outgoing ports. Advantageously, the hash key is an N bit hash key; and, each port identifier look-up table contains 2^N entries occupying 2^N consecutive locations, with each entry being an identifier of a particular one of the physical outgoing ports. Identifiers for particular outgoing ports are retrieved from the distribution table by extracting first and second N bit hash keys which form part of the retrieved destination and source address contexts respectively. The hash keys are combined to form an N bit connection identifier. The port identifier look-up table corresponding to the aggregated grouping represented by the retrieved destination address is selected, and the entry at the table location corresponding to the value of the N bit connection identifier is retrieved. If the address look-up table does not contain a destination address corresponding to the extracted destination address then first and second hash keys are produced by applying a hash function to the extracted source and destination addresses respectively. The hash keys are combined to form an N bit connection identifier. The incoming port on which the packet containing the extracted source address was received is identified. All of the aggregated groupings are scanned to identify all outgoing ports to which packets may be directed from the incoming port on which the packet was received. For each one of those outgoing ports, the port identifier look-up table corresponding to the aggregated grouping containing that outgoing port is selected, the entry at the table location corresponding to the value of the N bit connection identifier is retrieved, and the received packet is queued for outgoing transmission on the outgoing port corresponding to the retrieved entry.”) Alexander at 5:10-35 (“If a packet arrives bearing a source Ethernet MAC address that was not found in look-up table 12 by address resolution unit 10, learning function 16 is invoked to update look-up table 12 with the new address (i.e. processing branches along the "No" exit from FIG. 2, block 36). Learning function 16 first computes a hash function on the source Ethernet MAC address, generating an N-bit hash key ("partial connection identifier") from the 48-bit MAC address, where N is some small integer in the range of 3 to 8 (FIG. 2, block 38). The physical port on which the packet arrived is then determined. If the physical port is found to be associated with an aggregate group (i.e., it is one of a set of ports that have been bound into a single logical port),

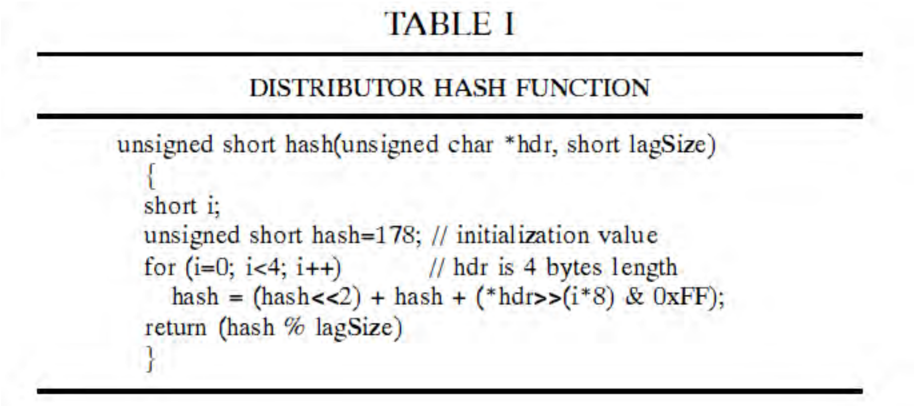
No.	’740 Patent Claim 10	Basso
		then the logical identifier assigned to the aggregate group is also determined. The hash key is then stored into address look-up table 12 in conjunction with the actual Ethernet MAC address and the port identifier (FIG. 2, block 40). The physical port identifier is used if the port is not part of an aggregate group (i.e. if processing branched along the "No" exit from block 30 and through block 32), while the logical identifier is used for ports that have been aggregated (i.e. if processing branched along the "Yes" exit from block 30 and through block 34). The hash key and port identifier are considered to form the "context" for the given MAC address.”) Alexander at 5:36-46 (“The hash function should be selected to ensure an even distribution of hash key values over the range of MAC addresses that are expected to be seen by the Ethernet switch. As a specific example, the EXACT™ Ethernet switch system employs an exclusive-OR based hash function, wherein the 48-bit MAC address is divided into 16-bit blocks, which are then exclusive-ORed together to form a single 16-bit number; the 3 least significant bits (LSBs) of this number are taken to produce a 3-bit hash key. Other schemes such as CRC-based or checksum-based hashes may also be used.”) Alexander at 6:49-65 (“If the context information for the destination address indicates, however, that the target is an aggregate group (i.e. if processing branches along the "Yes" exit from FIG. 2, block 42) then the logical identifier assigned to the aggregate group is retrieved and is used to select the proper look-up table contained within the distribution table data structure. The hash keys (partial connection identifiers) stored into the contexts for the source and destination MAC addresses are obtained from address resolution unit 10 and combined to generate a "connection identifier" with the same number of bits (FIG. 2, block 44). (In the EXACT™ Ethernet switch, a Boolean exclusive-OR operation is used to combine the hash keys without increasing the number of bits.) This connection identifier is then used to index into the selected look-up table, and finally retrieve an actual physical port index on which the packet must be transmitted (FIG. 2, block 46).”)
10[c]	calculating a modulo of a division operation	Basso discloses calculating a modulo of a division operation of the hashing key by the hashing size.

No.	’740 Patent Claim 10	Basso
	of the hashing key by the hashing size, and	For example, Basso a hash function performed on a received packet to generate a hash value, which is then used to determine the blade/port combination. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Basso is found to not meet this limitation, calculating a modulo of a division operation of the hashing key by the hashing size would have been obvious to a person having ordinary skill in the art, as explained below. Basso at [0011] (“A hash function may then be performed on the received packet to generate a hash value. In one embodi-ment, a hash function may be performed on the source and destination address in the packet header to generate a hash value.”) Basso at [0012] (“The hash value generated may be used to index into the table associated with a particular logical interface. Upon indexing into the table associated with the logical interface, an appropriate blade/port combination may be identified to transmit the received packet of data. In one embodiment, a blade/port combination may be selected in the indexed entry of the table associated with the logical interface by using a portion of the bits of the hashed value. The received packet may then be transmitted through the identified blade/port combination to the next destination (next destination previ-ously identified by the next destination address in the forwarding table).”) Basso at [0041] (“In step 504, a hash function may be performed on the received packet to generate a hash value. In one embodi-ment, a hash function may be performed on the source and destination address in the packet header to generate a hash value. It is noted that in other embodiments a hash function may be performed on different fields, e.g., port, type of service, in the received packet to generate a hash value.”) Basso at [0042] (“In step 505, the hash value generated in step 504 may be used to index into the table associated with a particular logical interface 405 determined in step 503. Upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be identified in step 506 to transmit the received packet of data as explained below.”)

No.	’740 Patent Claim 10	Basso
		Basso at Figure 5 FIG. 5  500 Map a plurality of physical ports into a single logical interface 501 Receive a packet of data 502 Index into forwarding table to determine logical interface table number and next destination address 503 Perform a hash function on packet header to generate a hash value 504 Index into appropriate logical interface table using hash value generated 505 Identify appropriate physical blade and port to transmit received packet using hash value 506 Transmit received packet through identified physical blade and port 507 Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Basso in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified

No.	’740 Patent Claim 10	Basso
		in element 10[c] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses distributing data frames over physical links and traces based on a hash function involving a division operation (%). Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0025] (“Typically, setting the protection policy includes determining a maximum number of the physical links that may fail while the logical link continues to provide at least the guaranteed bandwidth for the connection. In one embodiment, the guaranteed bandwidth is a bandwidth B, and the plurality of physical links consists of N links, and the maximum number is an integer P, and the link bandwidth allocated to each of the links is no less than B/(N-P). Conveying the data may further include managing the transmission of the data responsively to an actual number X of the physical links that have failed so that the guaranteed bandwidth on each of the links is limited to B/(N-X), X P, and an excess bandwidth on the physical links over the guaranteed bandwidth is available for other connections.”) Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”)

No.	’740 Patent Claim 10	Basso

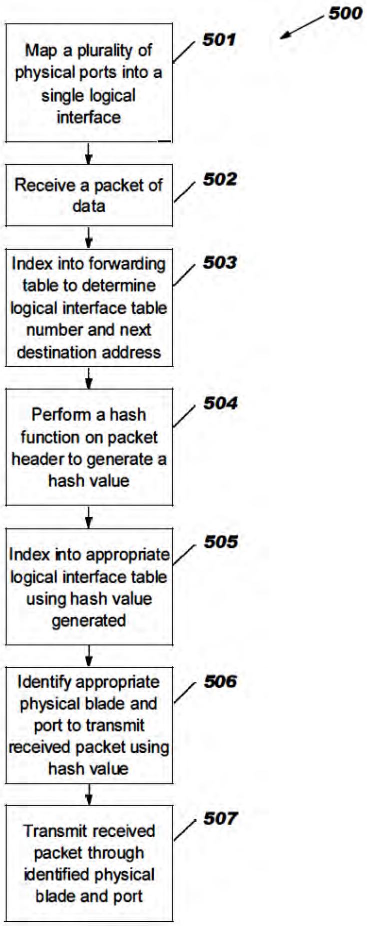
		Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:
		
		TABLE I
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		  {
		  short i;
		  unsigned short hash=178; // initialization value
		  for (i=0; i<4; i++) // hdr is 4 bytes length
		    hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		  return (hash % lagSize)
		  }
		”)
		
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		
		Bruckman at [0067] (“A similar problem may arise if there is a failure in a link in an aggregation group or in one of a number of line cards serving the aggregation group. In this case, to maintain the bandwidth allocation B made by CAC 44, each of the remaining links in the group must now carry, on average, B/(N-M) traffic, wherein M is the number of links in the group that are out of service. If only BIN has been allocated to each link, the remaining active links may not have sufficient bandwidth to continue to provide the bandwidth that has been guaranteed to the connections that they are required to carry. A similar problem arises with respect to loading of traces 52. For example, if there is a failure in LC2 or in one of links 30 in group 36 that connect to LC2, the trace connecting the multiplexer 50 in LCl will have to carry a substantially larger

No.	’740 Patent Claim 10	Basso
		share of the bandwidth, or even all of the bandwidth, that is allocated to the connection in ques-tion.”) Bruckman at [0068] (“FIG. 3 is a flow chart that schematically illustrates a method for dealing with these problems of fluctuating bandwidth requirements, in accordance with an embodiment of the present invention. In order to provide sufficient bandwidth for failure protection, CAC 44 uses a safety margin based on a protection parameter P, which is assigned at a protection setting step 60. P represents the maximum number of links in the group that can be out of service while still permitting the aggregation group to provide a given connection with the bandwidth that has been guaranteed to the connection. CAC 44 will then allocate at least B/(N-P) bandwidth to each link in the group, so that if P links fail, the group still provides total bandwidth of (N-P)*B/(N-P)= B. Setting P=l is equivalent to 1:N protection, so that the group will be unaffected by failure of a single link. In the example of group 36, shown in FIG. 2, setting P=2 will give both facility and equipment protection, i.e., the group will be unaffected not only by failure of a link, but also by failure of one of line cards 34. In the extreme case, in which P=N-1, CAC 44 will allocate the full bandwidth B on each link in the group.”) As another example, Singh discloses determining a ratio between the number of ingress and egress links and the number of links carrying data to the backplane and using a modulo to correspond to the channel’s link number. Singh at 9:30-43 (“The ratio between the number of line ingress links and the number of links carrying data to the backplane gives the backplane speedup for the system. In this example, there are 10 ingress links into the MS and 20 links (2 backplane channels) carrying that data to the backplane. This gives a backplane speedup of 2x. As another example, with 8 ingress links and 12 backplane links, there is a speedup of 1.5x. It should be noted that in addition to the backplane speedup, there is also an ingress/egress speedup. With 10 ingress links capable of carrying 2 Gbps each of raw data, this presents a 20 Gbps interface to the MS. An OC-192 only has approximately 10 Gbps worth of data. Taking into account cell overhead and cell quantization inefficiencies, there still remains excess capacity in the links.”)

No.	’740 Patent Claim 10	Basso
		Singh at 11:29-38 (“FIG. 9 is a diagram illustrating link to channel assignments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a channel, whether it is a backplane channel or an ingress/egress channel. There is no logical relationship between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.”) Singh at 15:15-39 (“The number of crossbars that are required in a system is dependent on how many links are being used to create the backplane channels. There should be an even number of crossbars and they would be divided evenly across the switch cards. The following equation, for most cases, pro-vides the correct number of crossbars: # of Crossbars=(# links per ingress channelx# of ingress channels per portx# of port cardsx speedup)/32. For the 8x8 configuration, the # of crossbars should be multiplied by (4x# of iMS)/(# backplane channels per port card). The number of port cards should be rounded up to the nearest supported configuration, i.e. 8, 16, or 32. The speedup in the case of crossbars should be the fractional speedup that is desired. Example to determine the number of arbiters and cross-bars for the following system: 4 channel port cards (40 Gbps) 8 links per channel 16 port cards Speedup=l.5 # of arbiters=(4x2x2)/2=8 # of crossbars=(8x4x16xl.5)/32=24. This would give 3crossbars per arbiter.”)

No.	’740 Patent Claim 10	Basso
		Singh at 16:28-44 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 16x16 and 32x32 is the organization of the switchplane. The port card remains the same. Backplane channels 1 and 2 are used for the backplane connectivity. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 16, correspond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”) Singh at 17:31-49 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 8x8 and 16x16 is the organi-zation of the switchplane. The port card remains the same. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Links 0-7 and 24-31 on the arbiters would not be used and would be powered off. Links 0-7 and 24-31 on the crossbars would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Backplane channels 1 and 2 are used for the backplane connectivity. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 8, corre-spond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”)
10[d]	selecting the first and second physical links responsively to the modulo.	Basso discloses selecting the first and second physical links responsively to the modulo. For example, Basso selecting the blade/port combination in response to the calculation performed in the hash function using the hash value. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Basso is found to

No.	’740 Patent Claim 10	Basso
		not meet this limitation, selecting the first and second physical links responsively to the modulo would have been obvious to a person having ordinary skill in the art, as explained below. Basso at [0011] (“A hash function may then be performed on the received packet to generate a hash value. In one embodi-ment, a hash function may be performed on the source and destination address in the packet header to generate a hash value.”) Basso at [0012] (“The hash value generated may be used to index into the table associated with a particular logical interface. Upon indexing into the table associated with the logical interface, an appropriate blade/port combination may be identified to transmit the received packet of data. In one embodiment, a blade/port combination may be selected in the indexed entry of the table associated with the logical interface by using a portion of the bits of the hashed value. The received packet may then be transmitted through the identified blade/port combination to the next destination (next destination previ-ously identified by the next destination address in the forwarding table).”) Basso at [0041] (“In step 504, a hash function may be performed on the received packet to generate a hash value. In one embodi-ment, a hash function may be performed on the source and destination address in the packet header to generate a hash value. It is noted that in other embodiments a hash function may be performed on different fields, e.g., port, type of service, in the received packet to generate a hash value.”) Basso at [0042] (“In step 505, the hash value generated in step 504 may be used to index into the table associated with a particular logical interface 405 determined in step 503. Upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be identified in step 506 to transmit the received packet of data as explained below.”) Basso at Figure 5

No.	’740 Patent Claim 10	Basso
		FIG. 5  500 Map a plurality of physical ports into a single logical interface 501 Receive a packet of data 502 Index into forwarding table to determine logical interface table number and next destination address 503 Perform a hash function on packet header to generate a hash value 504 Index into appropriate logical interface table using hash value generated 505 Identify appropriate physical blade and port to transmit received packet using hash value 506 Transmit received packet through identified physical blade and port 507 Basso at [0043] (“As stated above, the table associated with a par-ticular logical interface 405 may comprise a plurality of entries where each entry may comprise a threshold value associated with a particular blade 402/port 404 combination. The threshold value may represent a percentage of the total number of packets received by router 104A that may be transmitted through the blade

<table>
<tr><th>No.</th><th>’740 Patent Claim 10</th><th>Basso</th></tr>
<tr><td></td><td></td><td>402/port 404 combination associated with that threshold value. In one embodiment, the threshold value may be updated periodically by a user, e.g., system administrator, in control of router 104, e.g., router 104A. For example, the threshold value, e.g., twenty percent of the number of packets received by router 104A, associ-ated with a particular blade 402/port 404 combination may be updated by lowering the threshold value by one percent during each update. An example of an entry of the table associated with a particular logical interface 405 is shown in Table 1 below:

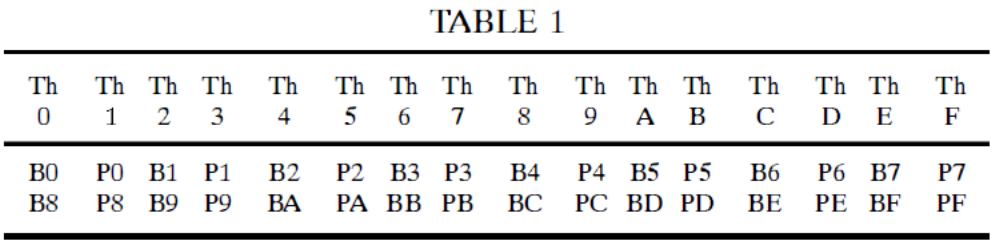


TABLE 1
<table>
<tr><th>Th 0</th><th>Th 1</th><th>Th 2</th><th>Th 3</th><th>Th 4</th><th>Th 5</th><th>Th 6</th><th>Th 7</th><th>Th 8</th><th>Th 9</th><th>Th A</th><th>Th B</th><th>Th C</th><th>Th D</th><th>Th E</th><th>Th F</th></tr>
<tr><td>B0
B8</td><td>P0
P8</td><td>B1
B9</td><td>P1
P9</td><td>B2
BA</td><td>P2
PA</td><td>B3
BB</td><td>P3
PB</td><td>B4
BC</td><td>P4
PC</td><td>B5
BD</td><td>P5
PD</td><td>B6
BE</td><td>P6
PE</td><td>B7
BF</td><td>P7
PF</td></tr>
</table>
”)

Basso at [0045] (“As stated above, upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be identified in step 506 to transmit the received packet of data. In one embodiment, the hash value generated in step 504 may be used to select a particular threshold value and hence a blade 402/port 404 combination associated with the selected threshold value. In one embodiment, a portion of the bits of the hash value, e.g., most significant bits, may be used to select a particular threshold value in the entry indexed in step 505. For example, referring to Table 1, since there are 16 different threshold values in each entry of the table associated with logical interface 405, only four bits of the hash value generated in step 504 may be used to select a threshold value. Upon selecting a threshold value, the blade 402/port 404 combination associated with the selected threshold value may be used to transmit the received packet.”)

Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Basso in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified</td></tr>
</table>

No.	’740 Patent Claim 10	Basso
		in element 10[d] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses distributing data frames over physical links and traces based on a hash function involving a division operation (%). Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0025] (“Typically, setting the protection policy includes determining a maximum number of the physical links that may fail while the logical link continues to provide at least the guaranteed bandwidth for the connection. In one embodiment, the guaranteed bandwidth is a bandwidth B, and the plurality of physical links consists of N links, and the maximum number is an integer P, and the link bandwidth allocated to each of the links is no less than B/(N-P). Conveying the data may further include managing the transmission of the data responsively to an actual number X of the physical links that have failed so that the guaranteed bandwidth on each of the links is limited to B/(N-X), X P, and an excess bandwidth on the physical links over the guaranteed bandwidth is available for other connections.”) Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”)

<table>
<tr><th>No.</th><th>’740 Patent Claim 10</th><th>Basso</th></tr>
<tr><td></td><td></td><td>

Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:

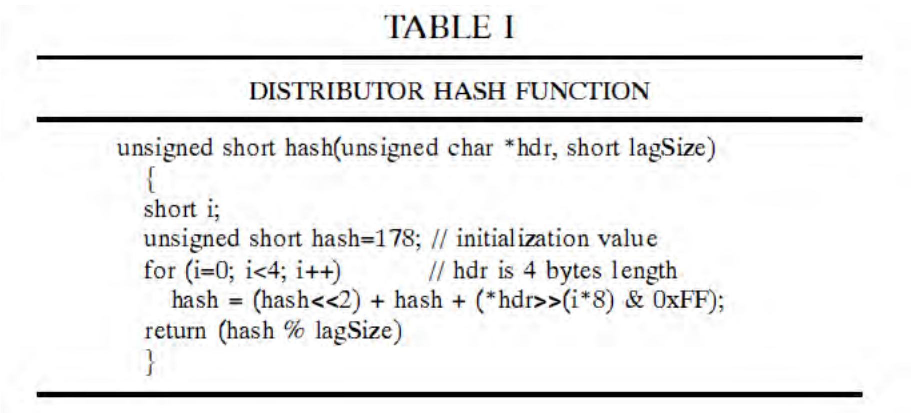


TABLE I

DISTRIBUTOR HASH FUNCTION

```
unsigned short hash(unsigned char *hdr, short lagSize)
  {
  short i;
  unsigned short hash=178; // initialization value
  for (i=0; i<4; i++)         // hdr is 4 bytes length
     hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
  return (hash % lagSize)
  }
```

”)

Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)

Bruckman at [0067] (“A similar problem may arise if there is a failure in a link in an aggregation group or in one of a number of line cards serving the aggregation group. In this case, to maintain the bandwidth allocation B made by CAC 44, each of the remaining links in the group must now carry, on average, B/(N-M) traffic, wherein M is the number of links in the group that are out of service. If only BIN has been allocated to each link, the remaining active links may not have sufficient bandwidth to continue to provide the bandwidth that has been guaranteed to the connections that they are required to carry. A similar problem arises with respect to loading of traces 52. For example, if there is a failure in LC2 or in one of links 30 in group 36 that connect to LC2, the trace connecting the multiplexer 50 in LCl will have to carry a substantially larger

</td></tr>
</table>

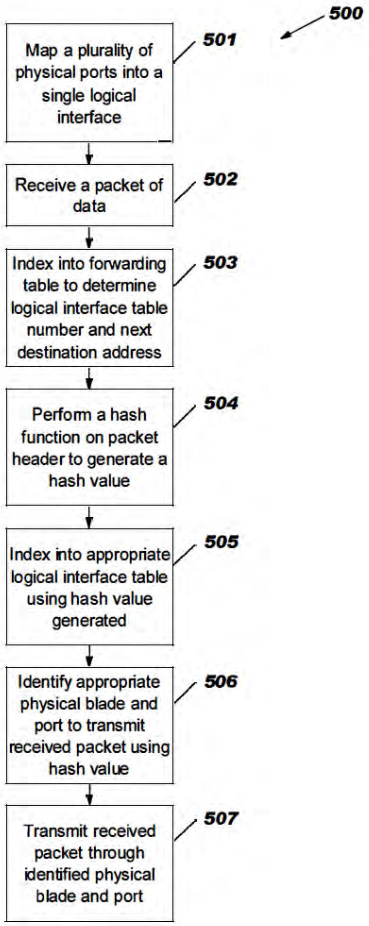
No.	’740 Patent Claim 10	Basso
		share of the bandwidth, or even all of the bandwidth, that is allocated to the connection in ques-tion.”) Bruckman at [0068] (“FIG. 3 is a flow chart that schematically illustrates a method for dealing with these problems of fluctuating bandwidth requirements, in accordance with an embodiment of the present invention. In order to provide sufficient bandwidth for failure protection, CAC 44 uses a safety margin based on a protection parameter P, which is assigned at a protection setting step 60. P represents the maximum number of links in the group that can be out of service while still permitting the aggregation group to provide a given connection with the bandwidth that has been guaranteed to the connection. CAC 44 will then allocate at least B/(N-P) bandwidth to each link in the group, so that if P links fail, the group still provides total bandwidth of (N-P)*B/(N-P)= B. Setting P=l is equivalent to 1:N protection, so that the group will be unaffected by failure of a single link. In the example of group 36, shown in FIG. 2, setting P=2 will give both facility and equipment protection, i.e., the group will be unaffected not only by failure of a link, but also by failure of one of line cards 34. In the extreme case, in which P=N-1, CAC 44 will allocate the full bandwidth B on each link in the group.”) As another example, Singh discloses determining a ratio between the number of ingress and egress links and the number of links carrying data to the backplane and using a modulo to correspond to the channel’s link number. Singh at 9:30-43 (“The ratio between the number of line ingress links and the number of links carrying data to the backplane gives the backplane speedup for the system. In this example, there are 10 ingress links into the MS and 20 links (2 backplane channels) carrying that data to the backplane. This gives a backplane speedup of 2x. As another example, with 8 ingress links and 12 backplane links, there is a speedup of 1.5x. It should be noted that in addition to the backplane speedup, there is also an ingress/egress speedup. With 10 ingress links capable of carrying 2 Gbps each of raw data, this presents a 20 Gbps interface to the MS. An OC-192 only has approximately 10 Gbps worth of data. Taking into account cell overhead and cell quantization inefficiencies, there still remains excess capacity in the links.”)

No.	’740 Patent Claim 10	Basso
		Singh at 11:29-38 (“FIG. 9 is a diagram illustrating link to channel assignments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a channel, whether it is a backplane channel or an ingress/egress channel. There is no logical relationship between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.”) Singh at 15:15-39 (“The number of crossbars that are required in a system is dependent on how many links are being used to create the backplane channels. There should be an even number of crossbars and they would be divided evenly across the switch cards. The following equation, for most cases, pro-vides the correct number of crossbars: # of Crossbars=(# links per ingress channelx# of ingress channels per portx# of port cardsx speedup)/32. For the 8x8 configuration, the # of crossbars should be multiplied by (4x# of iMS)/(# backplane channels per port card). The number of port cards should be rounded up to the nearest supported configuration, i.e. 8, 16, or 32. The speedup in the case of crossbars should be the fractional speedup that is desired. Example to determine the number of arbiters and cross-bars for the following system: 4 channel port cards (40 Gbps) 8 links per channel 16 port cards Speedup=l.5 # of arbiters=(4x2x2)/2=8 # of crossbars=(8x4x16xl.5)/32=24. This would give 3crossbars per arbiter.”)

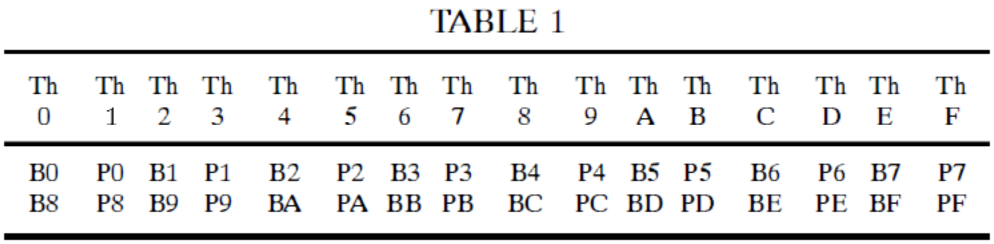
No.	’740 Patent Claim 10	Basso
		Singh at 16:28-44 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 16x16 and 32x32 is the organization of the switchplane. The port card remains the same. Backplane channels 1 and 2 are used for the backplane connectivity. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 16, correspond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”) Singh at 17:31-49 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 8x8 and 16x16 is the organi-zation of the switchplane. The port card remains the same. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Links 0-7 and 24-31 on the arbiters would not be used and would be powered off. Links 0-7 and 24-31 on the crossbars would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Backplane channels 1 and 2 are used for the backplane connectivity. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 8, corre-spond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”)

No.	’740 Patent Claim 11	Basso
11	The method according to claim 10, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo.	Basso discloses the method according to claim 10, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo. For example, Basso discloses using the hash value from the hash function to select the blade/port combination using specific bits of the product generated from the hash function. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Basso is found to not meet this limitation, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo would have been obvious to a person having ordinary skill in the art, as explained below. *See supra* at Claim 10. Basso at [0011] (“A hash function may then be performed on the received packet to generate a hash value. In one embodi-ment, a hash function may be performed on the source and destination address in the packet header to generate a hash value.”) Basso at [0012] (“The hash value generated may be used to index into the table associated with a particular logical interface. Upon indexing into the table associated with the logical interface, an appropriate blade/port combination may be identified to transmit the received packet of data. In one embodiment, a blade/port combination may be selected in the indexed entry of the table associated with the logical interface by using a portion of the bits of the hashed value. The received packet may then be transmitted through the identified blade/port combination to the next destination (next destination previ-ously identified by the next destination address in the forwarding table).”) Basso at [0041] (“In step 504, a hash function may be performed on the received packet to generate a hash value. In one embodi-ment, a hash function may be performed on the source and destination address in the packet header to generate a hash value. It is noted that in other

No.	’740 Patent Claim 11	Basso
		embodiments a hash function may be performed on different fields, e.g., port, type of service, in the received packet to generate a hash value.”) Basso at [0042] (“In step 505, the hash value generated in step 504 may be used to index into the table associated with a particular logical interface 405 determined in step 503. Upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be identified in step 506 to transmit the received packet of data as explained below.”) Basso at Figure 5

No.	’740 Patent Claim 11	Basso
		FIG. 5  500 Map a plurality of physical ports into a single logical interface 501 Receive a packet of data 502 Index into forwarding table to determine logical interface table number and next destination address 503 Perform a hash function on packet header to generate a hash value 504 Index into appropriate logical interface table using hash value generated 505 Identify appropriate physical blade and port to transmit received packet using hash value 506 Transmit received packet through identified physical blade and port 507 Basso at [0043] (“As stated above, the table associated with a par-ticular logical interface 405 may comprise a plurality of entries where each entry may comprise a threshold value associated with a particular blade 402/port 404 combination. The threshold value may represent a percentage of the total number of packets received by router 104A that may be transmitted through the blade

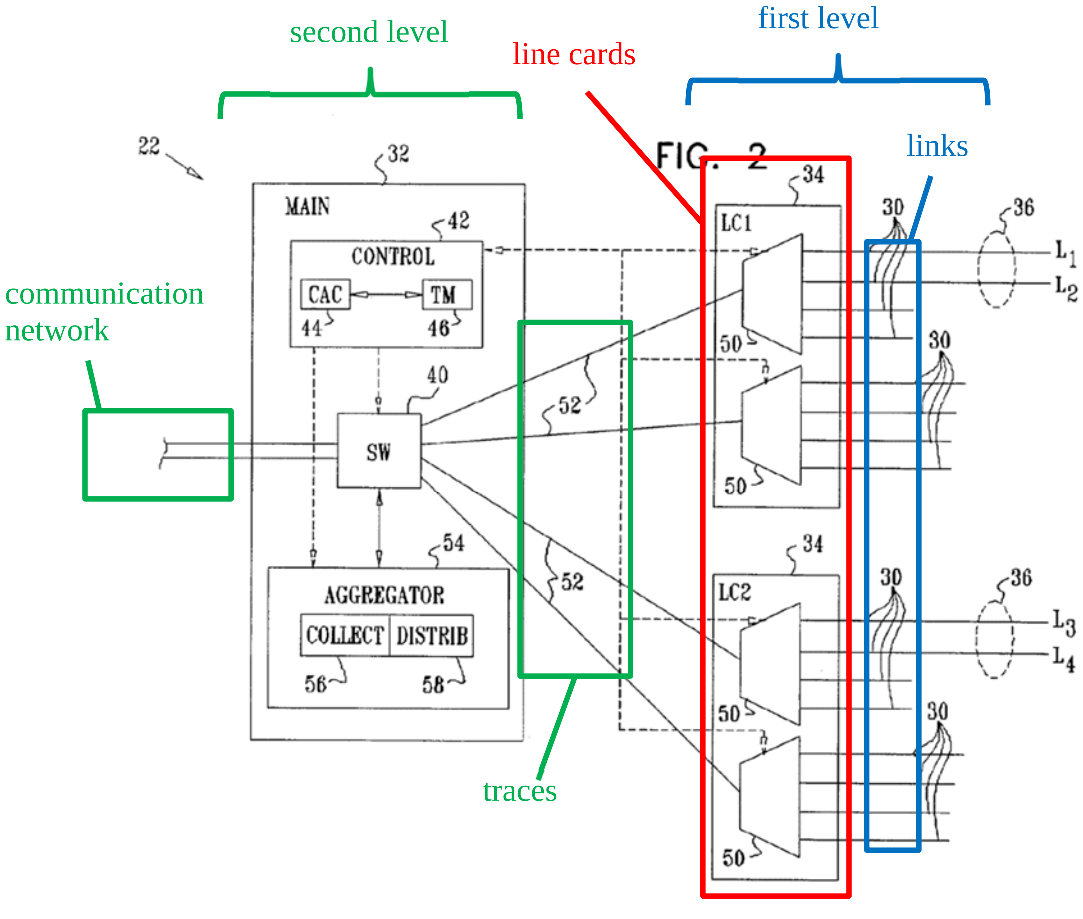
<table>
<tr><th>No.</th><th>’740 Patent Claim 11</th><th>Basso</th></tr>
<tr><td></td><td></td><td>402/port 404 combination associated with that threshold value. In one embodiment, the threshold value may be updated periodically by a user, e.g., system administrator, in control of router 104, e.g., router 104A. For example, the threshold value, e.g., twenty percent of the number of packets received by router 104A, associ-ated with a particular blade 402/port 404 combination may be updated by lowering the threshold value by one percent during each update. An example of an entry of the table associated with a particular logical interface 405 is shown in Table 1 below:

TABLE 1
<table><tr><th>Th 0</th><th>Th 1</th><th>Th 2</th><th>Th 3</th><th>Th 4</th><th>Th 5</th><th>Th 6</th><th>Th 7</th><th>Th 8</th><th>Th 9</th><th>Th A</th><th>Th B</th><th>Th C</th><th>Th D</th><th>Th E</th><th>Th F</th></tr><tr><td>B0
B8</td><td>P0
P8</td><td>B1
B9</td><td>P1
P9</td><td>B2
BA</td><td>P2
PA</td><td>B3
BB</td><td>P3
PB</td><td>B4
BC</td><td>P4
PC</td><td>B5
BD</td><td>P5
PD</td><td>B6
BE</td><td>P6
PE</td><td>B7
BF</td><td>P7
PF</td></tr></table>”)

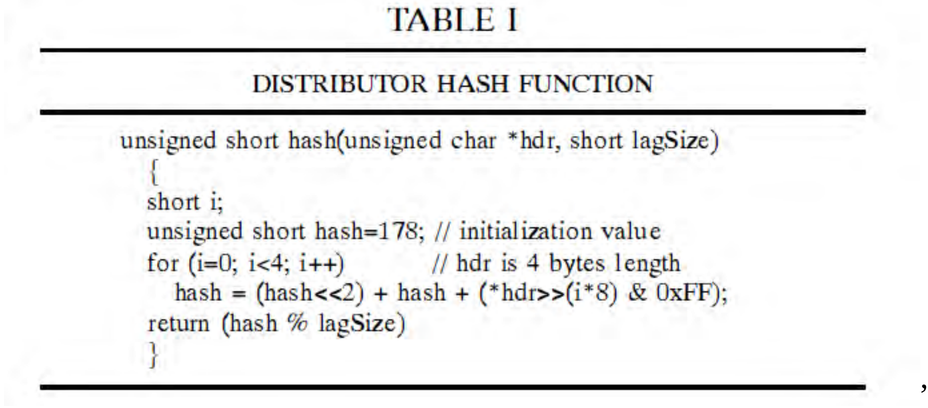
Basso at [0045] (“As stated above, upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be identified in step 506 to transmit the received packet of data. In one embodiment, the hash value generated in step 504 may be used to select a particular threshold value and hence a blade 402/port 404 combination associated with the selected threshold value. In one embodiment, a portion of the bits of the hash value, e.g., most significant bits, may be used to select a particular threshold value in the entry indexed in step 505. For example, referring to Table 1, since there are 16 different threshold values in each entry of the table associated with logical interface 405, only four bits of the hash value generated in step 504 may be used to select a threshold value. Upon selecting a threshold value, the blade 402/port 404 combination associated with the selected threshold value may be used to transmit the received packet.”)

Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Basso in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references</td></tr>
</table>

No.	’740 Patent Claim 11	Basso
		identified in element 11 of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses distributing data frames over physical links and traces based on a division operation in the hash function, i.e., selecting the first and second physical links responsively to the modulo, involving specific byte lengths of the frame information. Bruckman at Figure 2 (annotated)

No.	’740 Patent Claim 11	Basso
		 second level line cards first level links communication network traces 22 32 MAIN 42 CONTROL CAC TM 44 46 40 SW 52 52 54 AGGREGATOR COLLECT DISTRIB 56 58 34 LC1 50 50 34 LC2 50 50 30 30 30 30 36 36 L_1 L_2 L_3 L_4 FIG. 2 Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:

No.	’740 Patent Claim 11	Basso

		
		TABLE I
		DISTRIBUTOR HASH FUNCTION
		`unsigned short hash(unsigned char *hdr, short lagSize)`
		`{`
		`short i;`
		`unsigned short hash=178; // initialization value`
		`for (i=0; i<4; i++)          // hdr is 4 bytes length`
		`hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);`
		`return (hash % lagSize)`
		`}`
		”)
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		As another example, Solomon discloses using a subset of bits to encode for the selected physical port, i.e., selecting the first and second physical links responsively to the modulo, involving specific byte lengths of the frame information.
		Solomon at [0054] (“Having selected a physical port, RSVP-TE proces-sor 30 of switch A now generates a suitable MPLS label, at a label generation step 64. The preceding node upstream of switch A will subsequently attach this MPLS label to all MPLS packets transmitted through tunnel 28 to switch A. The label is assigned, in conjunction with the mapping function of mapper 34, so as to ensure that all MPLS packets carrying this label are switched through the physical port that was selected for this tunnel at step 62. For this purpose, RSVP-TE processor 30 of switch A dedicates a sub-set of the bits of MPLS label 52 to encode the serial number of the selected physical port. For example, the four least-signifi-cant bits of MPLS label 52 may be used for

No.	’740 Patent Claim 11	Basso
		encoding the selected port number. This configuration is suitable for representing LAG groups having up to 16 physical ports (N<16). The remaining bits of MPLS label 52 may be chosen at random or using any suitable method known in the art.”) Solomon at [0056] (“Mapper 34 of switch A maps the received packets belonging to tunnel 28 to the selected physical Ethernet port at a mapping step 70. For this purpose, mapper 34 extracts the MPLS label from each received packet and decodes the selected physical port number from the dedicated sub-set of bits, such as the four LSB, as described in step 64 above. The decoded value is used for mapping the packet to the selected physical port, which was allocated by the CAC processor at step 62 above. In the four-bit example described above, the mapping function may be written explicitly as: Selected port number=((MPLS label) and (0x0000F)), wherein "and" denotes the "bitwise and" operator.”)

No.	’740 Patent Claim 12	Basso
12	The method according to claim 1, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC)	Basso discloses the method according to claim 1, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port. For example, Basso discloses data packet information fields including headers, ports, and addresses, etc. *See supra* at Claim 1. Basso at [0010] (“Upon a network processor receiving a packet of data, the network processor may index into a table, com-monly referred to as a forwarding table, to determine the table associated with a particular logical interface as well as the next destination address. The forwarding table may comprise a plurality of entries where each entry may com-prise

No.	’740 Patent Claim 12	Basso
	address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port.	information indicating a particular table associated with a particular logical interface as well as the next desti-nation address. Each logical interface may be associated with a table storing a plurality of entries containing blade/ port combinations as discussed further below. In one embodiment, an entry may be indexed in the forwarding table using a destination address in the received packet header.”) Basso at [0011] (“A hash function may then be performed on the received packet to generate a hash value. In one embodi-ment, a hash function may be performed on the source and destination address in the packet header to generate a hash value.”) Basso at [0040] (“In step 502, network processor 403, e.g., network processor 403A, may receive a packet of data from switch fabric 401. Upon receiving the packet of data, network processor 403, in step 503, may index into a table, com-monly referred to as a forwarding table, to determine the table associated with a particular logical interface 405 as well as the next destination address, i.e., the next hop address. The forwarding table may comprise a plurality of entries where each entry may comprise information indicat-ing a particular table associated with a particular logical interface 405 as well as the next destination address. Each logical interface 405 may be associated with a table storing a plurality of entries containing blade 402/port 404 combi-nations as discussed further below. In one embodiment, an entry may be indexed in the forwarding table using a destination address in the received packet header. It is noted that an entry may be indexed in the forwarding table using other means and that such means would be recognized by an artisan of ordinary skill in the art. It is further noted that embodiments implementing such means would fall within the scope of the present invention.”) Basso at [0041] (“In step 504, a hash function may be performed on the received packet to generate a hash value. In one embodi-ment, a hash function may be performed on the source and destination address in the packet header to generate a hash value. It is noted that in other embodiments a hash function may be performed on different fields, e.g., port, type of service, in the received packet to generate a hash value.”)

No.	’740 Patent Claim 13	Basso
13[preamble]	A method for communication, comprising:	Basso discloses a method for communication. *See supra at* 1[preamble].
13[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	Basso discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].
13[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Basso discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra at* 1[c].
13[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	Basso discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra at* 1[e].
13[d]	selecting, in a single computation based on at least one of the frame attributes, a first physical link out	Basso discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. *See supra at* 1[f].

No.	’740 Patent Claim 13	Basso
	of the first group and a second physical link out of the second group; and	
13[e]	sending the data frame over the selected first and second physical links,	Basso discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].
13[f]	coupling the network node to the one or more interface modules and	Basso discloses coupling the network node to the one or more interface modules. *See supra at* 1[a].
13[g]	coupling each of the one or more interface modules to the communication network comprising	Basso discloses coupling each of the one or more interface modules to the communication network. *See supra at* 1[c].
13[h]	specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication	Basso discloses specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node. For example, Basso discloses connecting the blades to the network with physical connections with specific transmission rates based on bandwidth limitations. A person of ordinary skill in the art would also understand that the communication service has specified bandwidth properties. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Basso is found to not meet this limitation, coupling each of the one or more interface modules to the communication network comprising specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of

No.	’740 Patent Claim 13	Basso
	network to the network node, and	a communication service provided by the communication network to the network node would have been obvious to a person having ordinary skill in the art, as explained below. Basso at [0004] (“Since the processed packets may be transmitted by the network processor across one particular address, the rate at which the processed packets are transmitted is limited to the bandwidth of the connection. That is, since the processed packets may be transmitted by the network processor across one physical port to a particular network device, the rate at which the processed packets are transmitted is limited to the bandwidth of the physical connection between the port and the network device.”) Basso at [0005] (“If, however, the packets of data were able to be transmitted across multiple physical ports to a particular network device from a particular network processor, then the bandwidth capacity would be increased. That is, if packets of data were able to be transmitted across multiple physical ports to a particular network device by a network processor, then the rate packets of data are transmitted may be increased.”) Basso at [0032] (“As stated in the Background Information section, each port may be physically coupled to another network device, e.g., router. Since the processed packets may be transmitted by a network processor across one particular address, the rate at which the processed packets are trans-mitted is limited to the bandwidth of the connection. That is, since the processed packets may be transmitted by the network processor across one physical port to a particular network device, the rate at which the processed packets are transmitted is limited to the bandwidth of the physical connection between the port and the network device.”) Basso at [0033] (“If, however, packets of data were able to be trans-mitted across multiple physical ports to a particular network device from a particular network processor, then the band-width capacity would be increased. That is, if packets of data were able to be transmitted across multiple physical ports to a particular network device by a network processor, then the rate packets of data are transmitted may be increased. It would therefore be desirable to enable a network processor to transmit packets of data across multiple physical ports to a particular network device instead of one physical port.”)

No.	’740 Patent Claim 13	Basso
		Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Basso in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 13[h] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses specifying certain committed, excess, and guaranteed bandwidths, including CIR, EIR, and PIR, respectively. Bruckman at [0013] (“Service level agreements between network service providers and customers commonly specify a certain com-mitted bandwidth, or committed information rate (CIR), which the service provider guarantees to provide to the customer at all times, regardless of bandwidth stress on the network. Additionally or alternatively, the agreement may specify an excess bandwidth, which is available to the customer when network traffic permits. The excess band-width is typically used by customers for lower-priority services, which do not require committed bandwidth. The network service provider may guarantee the customer a certain minimum excess bandwidth, or excess information rate (EIR), in order to avoid starvation of such services in case of bandwidth stress. In general, the bandwidth guaran-teed by a service provider, referred to as the peak informa-tion rate (PIR), may include either CIR, or EIR, or both CIR and EIR (in which case PIR=CIR+EIR). The term "guaran-teed bandwidth," as used in the context of the present patent application and in the claims, includes all these types of guaranteed bandwidth.”) As another example, Solomon discloses a service property of a guaranteed bandwidth, sometimes denoted as CIR-Committed Information Rate and PIR-Peak Information Rate. Solomon at [0023] (“In another embodiment, establishing the path includes receiving an indication of a requested service property of the flow, and selecting the port includes assign-ing the port to the flow so as to comply with the requested service property. In a disclosed embodiment, the requested service property includes at least one of a guaranteed bandwidth, a peak bandwidth and a class-of-service. Addi-tionally or alternatively, assigning the port

No.	’740 Patent Claim 13	Basso
		includes selecting the port having a maximum available bandwidth out of the plurality of aggregated ports. Further additionally or alter-natively, assigning the port includes selecting the port hav-ing a minimum available bandwidth out of the plurality of aggregated ports, which is still greater than or equal to the guaranteed bandwidth.”) Solomon at [0050] (“The method of FIG. 3 begins when the preceding node asks to establish a part of tunnel 28 (comprising one or more hops) for sending MPLS packets to MPLS/LAG switch 26 A. The preceding node requests and then receives the MPLS label, which it will subsequently attach to all packets that are sent to MPLS/LAG switch 26 labeledA. The preceding node sends downstream an RSVP-TE PATH mes-sage augmented with a LABEL_REQUEST object, as defined by RSVP-TE, to MPLS/LAG switch A, at a label requesting step 60. The PATH message typically comprises information regarding service properties that are requested for tunnel 28. The service properties may comprise a guar-anteed bandwidth (sometimes denoted CIR-Committed Information Rate) and a peak bandwidth (sometimes denoted PIR-Peak Information Rate), as well as a requested CoS (Class of Service-a measure of packet priority).”)
13[i]	allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements.	Basso discloses allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements. For examples, Basso discloses allocating bandwidth across the connections between the network device, blades, and network based on the specific bandwidth limitations of the connections. Basso at [0004] (“Since the processed packets may be transmitted by the network processor across one particular address, the rate at which the processed packets are transmitted is limited to the bandwidth of the connection. That is, since the processed packets may be transmitted by the network processor across one physical port to a particular network device, the rate at which the processed packets are transmitted is limited to the bandwidth of the physical connection between the port and the network device.”)

No.	’740 Patent Claim 13	Basso
		Basso at [0005] (“If, however, the packets of data were able to be transmitted across multiple physical ports to a particular network device from a particular network processor, then the bandwidth capacity would be increased. That is, if packets of data were able to be transmitted across multiple physical ports to a particular network device by a network processor, then the rate packets of data are transmitted may be increased.”) Basso at [0032] (“As stated in the Background Information section, each port may be physically coupled to another network device, e.g., router. Since the processed packets may be transmitted by a network processor across one particular address, the rate at which the processed packets are trans-mitted is limited to the bandwidth of the connection. That is, since the processed packets may be transmitted by the network processor across one physical port to a particular network device, the rate at which the processed packets are transmitted is limited to the bandwidth of the physical connection between the port and the network device.”) Basso at [0033] (“If, however, packets of data were able to be trans-mitted across multiple physical ports to a particular network device from a particular network processor, then the band-width capacity would be increased. That is, if packets of data were able to be transmitted across multiple physical ports to a particular network device by a network processor, then the rate packets of data are transmitted may be increased. It would therefore be desirable to enable a network processor to transmit packets of data across multiple physical ports to a particular network device instead of one physical port.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Basso in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 13[i] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For examples, Bruckman discloses allocating bandwidth over the physical links and traces based on bandwidth requirements, margins, and fluctuations.

No.	’740 Patent Claim 13	Basso
		Bruckman at [0013] (“Service level agreements between network service providers and customers commonly specify a certain com-mitted bandwidth, or committed information rate (CIR), which the service provider guarantees to provide to the customer at all times, regardless of bandwidth stress on the network. Additionally or alternatively, the agreement may specify an excess bandwidth, which is available to the customer when network traffic permits. The excess band-width is typically used by customers for lower-priority services, which do not require committed bandwidth. The network service provider may guarantee the customer a certain minimum excess bandwidth, or excess information rate (EIR), in order to avoid starvation of such services in case of bandwidth stress. In general, the bandwidth guaran-teed by a service provider, referred to as the peak informa-tion rate (PIR), may include either CIR, or EIR, or both CIR and EIR (in which case PIR=CIR+EIR). The term "guaran-teed bandwidth," as used in the context of the present patent application and in the claims, includes all these types of guaranteed bandwidth.”) As another example, Solomon discloses selecting the path to transmit the flow to comply with the service properties of the flow, i.e., allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements. Solomon at [0023] (“In another embodiment, establishing the path includes receiving an indication of a requested service property of the flow, and selecting the port includes assign-ing the port to the flow so as to comply with the requested service property. In a disclosed embodiment, the requested service property includes at least one of a guaranteed bandwidth, a peak bandwidth and a class-of-service. Addi-tionally or alternatively, assigning the port includes selecting the port having a maximum available bandwidth out of the plurality of aggregated ports. Further additionally or alter-natively, assigning the port includes selecting the port hav-ing a minimum available bandwidth out of the plurality of aggregated ports, which is still greater than or equal to the guaranteed bandwidth.”) Solomon at [0050] (“The method of FIG. 3 begins when the preceding node asks to establish a part of tunnel 28 (comprising one or more hops) for sending MPLS packets to MPLS/LAG switch 26 A. The preceding node requests and then receives the MPLS label, which it will

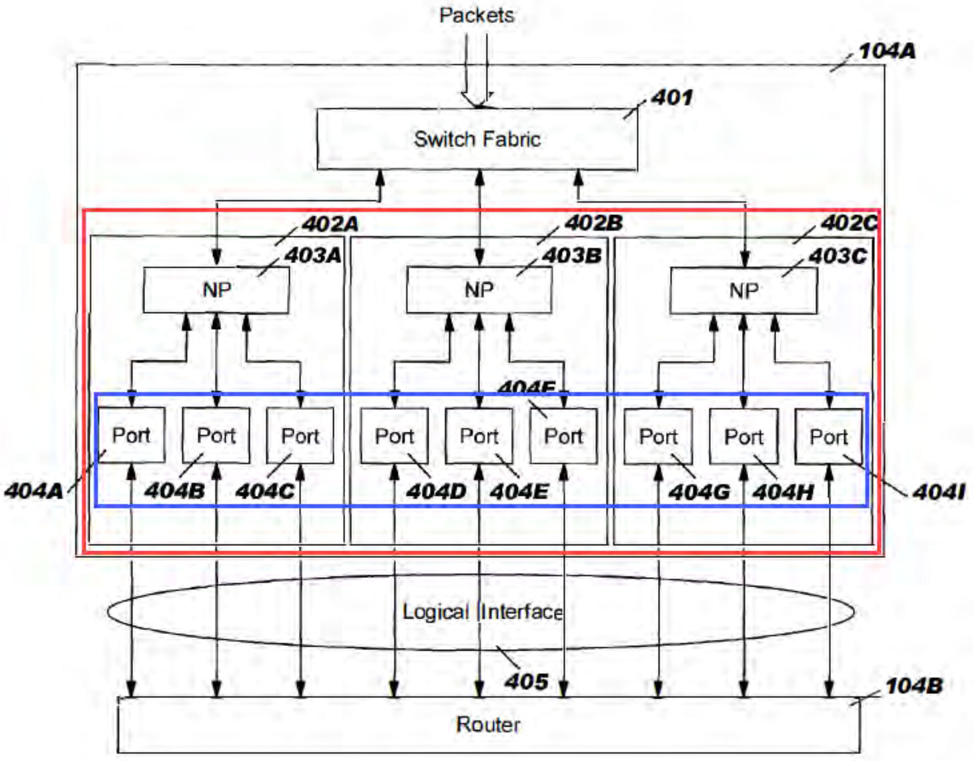
No.	’740 Patent Claim 13	Basso
		subsequently attach to all packets that are sent to MPLS/LAG switch 26 labeledA. The preceding node sends downstream an RSVP-TE PATH mes-sage augmented with a LABEL_REQUEST object, as defined by RSVP-TE, to MPLS/LAG switch A, at a label requesting step 60. The PATH message typically comprises information regarding service properties that are requested for tunnel 28. The service properties may comprise a guar-anteed bandwidth (sometimes denoted CIR-Committed Information Rate) and a peak bandwidth (sometimes denoted PIR-Peak Information Rate), as well as a requested CoS (Class of Service-a measure of packet priority).”)

No.	’740 Patent Claim 14	Basso
14[preamble]	A method for connecting user ports to a communication network, comprising:	Basso discloses a method for connecting user ports to a communication network. For example, Basso discloses a method for coupling physical ports on a plurality of blades to a network. Basso at Abstract (“A method, system and computer program product for rout-ing packets. A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more physical ports. The physical ports may be connected to another one or more network devices. A plurality of physical ports across one or more blades connected to the same network device may be logically mapped into a logical interface to that network device. By logically grouping a plurality of physical ports into a logical interface to a network device, a network processor may be able to transmit packets of data to that network device across multiple ports instead of one physical port.”) Basso at [0007] (“The problems outlined above may at least in part be solved in some embodiments by mapping a plurality of physical ports connected to a network device into a logical interface to that network device, e.g., router, gateway, edge device, server. That is, a

No.	’740 Patent Claim 14	Basso
		plurality of physical ports may be logically grouped into a logical interface to a network device thereby enabling a network processor to transmit packets of data across multiple ports instead of one physical port.”) Basso at [0008] (“In one embodiment of the present invention, a method for routing packets may comprise the step of map-ping a plurality of physical ports in a network device, e.g., router, connected to another network device, e.g., router, gateway, edge device, server, into a logical interface to that network device. By logically grouping a plurality of ports into a logical interface, a network processor in the network device, e.g., router, may be able to transmit packets of data across multiple physical ports to a particular network device instead of one physical port as discussed further below.”) Basso at [0009] (“A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more ports. These ports may be connected to another one or more network devices. The switch fabric may be configured to direct incoming packets of data to particular blades where one or more of the network proces-sors in the recipient blade may be configured to process the received packets.”) Basso at [0030] (“Returning to FIG. 4, router 104A may be config-ured to receive packets of data such as from client 101 (FIG. 1), e.g., client 101A, that may be directed to another particular network device, e.g., router 104B, in network 100 (FIG. 1). Router 104A may comprise a switch fabric 401 configured to direct the incoming packets of data to particu-lar blades 402A-C coupled to switch fabric 401. Blade 402A may comprise a network processor 403A coupled with one or more ports 404A-C. Blade 402B may comprise a network processor 403B coupled with one or more ports 404D-F. Blade 402C may comprise a network processor 403C coupled with one or more ports 404G-I. Blades 402A-C may collectively or individually be referred to as blades 402 or blade 402, respectively. Network processors 403A-C may collectively or individually be referred to as network pro-cessors 403 or network processor 403, respectively. Ports 404A-I may collectively or individually be referred to as ports 404 or port 404, respectively. Each port 404 may be coupled to a particular network device, e.g.,

No.	’740 Patent Claim 14	Basso
		gateway, server, router such as router 104B, in network system 100. It is noted that some of ports 404 may be coupled to a separate network device and that FIG. 4 is illustrative. It is further noted that router 104A may comprise any number of blades 402 and each blade 402 may comprise any number of network processors 403 and ports 404.”) Basso ay [0031] (“Referring to FIG. 4, network processor 403 may be configured to receive a packet of data from switch fabric 401. Upon receiving a packet, network processor 403 may be configured to process the packet of data. Processing may include but not limited to: determining what activities to be performed on the received packet, transmitting or discarding the received packet, determining which network device, e.g., router such as router 104B, server, edge device, gateway, to transmit the received packet, etc. Network processor 403 may then transmit the processed packet to a particular network device, e.g., router such as router 104B, through a port 404 connected to that network device.”) Basso at [0033] (“If, however, packets of data were able to be trans-mitted across multiple physical ports to a particular network device from a particular network processor, then the band-width capacity would be increased. That is, if packets of data were able to be transmitted across multiple physical ports to a particular network device by a network processor, then the rate packets of data are transmitted may be increased. It would therefore be desirable to enable a network processor to transmit packets of data across multiple physical ports to a particular network device instead of one physical port.”) Basso at [0034] (“Network processor 403 may be configured to trans-mit packets of data across multiple physical ports 404 to a particular network device, e.g., router 104B, by logically grouping a plurality of physical ports 404, e.g., ports 404A-I, into a logical interface 405. That is, the physical connections between router 104A and a particular network device, e.g., router 104B, may be logically grouped into a logical inter-face 405. Each set of physical connections between router 104A and a particular network device may be logically grouped into a particular logical interface 405. For example, if ports 404A-F were physically connected to network device #1, then ports 404A-F may be logically grouped into logical interface 405#1. If ports 404G-I were physically connected to network device #2, then ports 404G-I may be logically

No.	’740 Patent Claim 14	Basso
		grouped into logical interface 405#2. It is noted that physical ports 404 may be logically grouped into more than one logical interface 405 and that FIG. 4 is illustrative. It is further noted that each logical interface 405 may be associated with logically grouping a plurality of physical connections to a particular network device, e.g., router, gateway, edge device, server. It is further noted that a logical interface 405 may be associated with logically grouping a plurality of ports in either one blade 402 or across multiple blades 402.”) Basso at [0035] (“By logically grouping a plurality of ports 404 coupled to a particular network device into a logical inter-face 405, network processor 403 may be configured to transmit processed packets to that particular network device via any blade 402/port 404 combination grouped in that logical interface 405. For example, referring to FIG. 4, ports 404A-404I are physically connected to router 104B. If ports 404A-404I were logically grouped into logical interface 405, then a particular network processor 403, e.g., network processor 403A, may be configured to transmit processed packets that are determined to be transmitted to router 104B through any of ports 404A-404I in blades 402A-C, respec-tively. Network processor 403, e.g., network processor 403A, may be configured to transmit the processed packets to router 104B through ports 404, e.g., ports 404D-I, not in its blade 402, e.g., blade 402A, by forwarding the processed packets to switch fabric 401 which may then direct the processed packets to another appropriate physical blade 402/port 404 combination. Network processor 403, e.g., network processor 403A, may further be configured to transmit the processed packets to router 104B through any ports 404, e.g., ports 404A-C, in its blade 402, e.g., blade 402A, instead of just one physical port 404 in its blade 402, e.g., blade 402A. A more detailed description of routing packets implementing logical interface(s) 405 is provided below in FIG. 5.”) Basso at Figure 4

No.	’740 Patent Claim 14	Basso
		FIG. 4  Packets 104A 401 Switch Fabric 402A 402B 402C 403A 403B 403C NP NP NP 404F Port Port Port Port Port Port Port Port Port 404A 404B 404C 404D 404E 404G 404H 404I Logical Interface 405 104B Router
14[a]	coupling the user ports to one or more user interface modules;	Basso discloses coupling the user ports to one or more user interface modules. *See supra at* 1[a].

No.	’740 Patent Claim 14	Basso
14[b]	coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel,	Basso discloses coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel. *See supra at* 1[c], 3.
14[c]	at least one of said backplane traces being bi-directional and operative to communicate in both an upstream direction and a downstream direction;	Basso discloses at least one of said backplane traces being bi-directional and operative to communicate in both an upstream direction and a downstream direction. *See supra at* 14[b], 1[d].
14[d]	receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes;	Basso discloses receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes. *See supra at* 14[a], 1[e].
14[e]	for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or	Basso discloses for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces. *See supra at* 14[b], 1[f].

No.	’740 Patent Claim 14	Basso
	more backplane traces; and	
14[f]	sending the data frame over the selected backplane trace;	Basso discloses sending the data frame over the selected backplane trace. *See supra at* 14[e], 1[g].
14[g]	said sending comprising communicating along said at least one of said backplane traces.	Basso discloses said sending comprising communicating along said at least one of said backplane traces. *See supra at* 14[f], 1[h].

No.	’740 Patent Claim 15	Basso
15[preamble]	A method for connecting user ports to a communication network, comprising:	Basso discloses a method for connecting user ports to a communication network. *See supra at* 14[preamble].
15[a]	coupling the user ports to one or more user interface modules;	Basso discloses coupling the user ports to one or more user interface modules. *See supra at* 14[a].
15[b]	coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel;	Basso discloses coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel. *See supra at* 14[b].

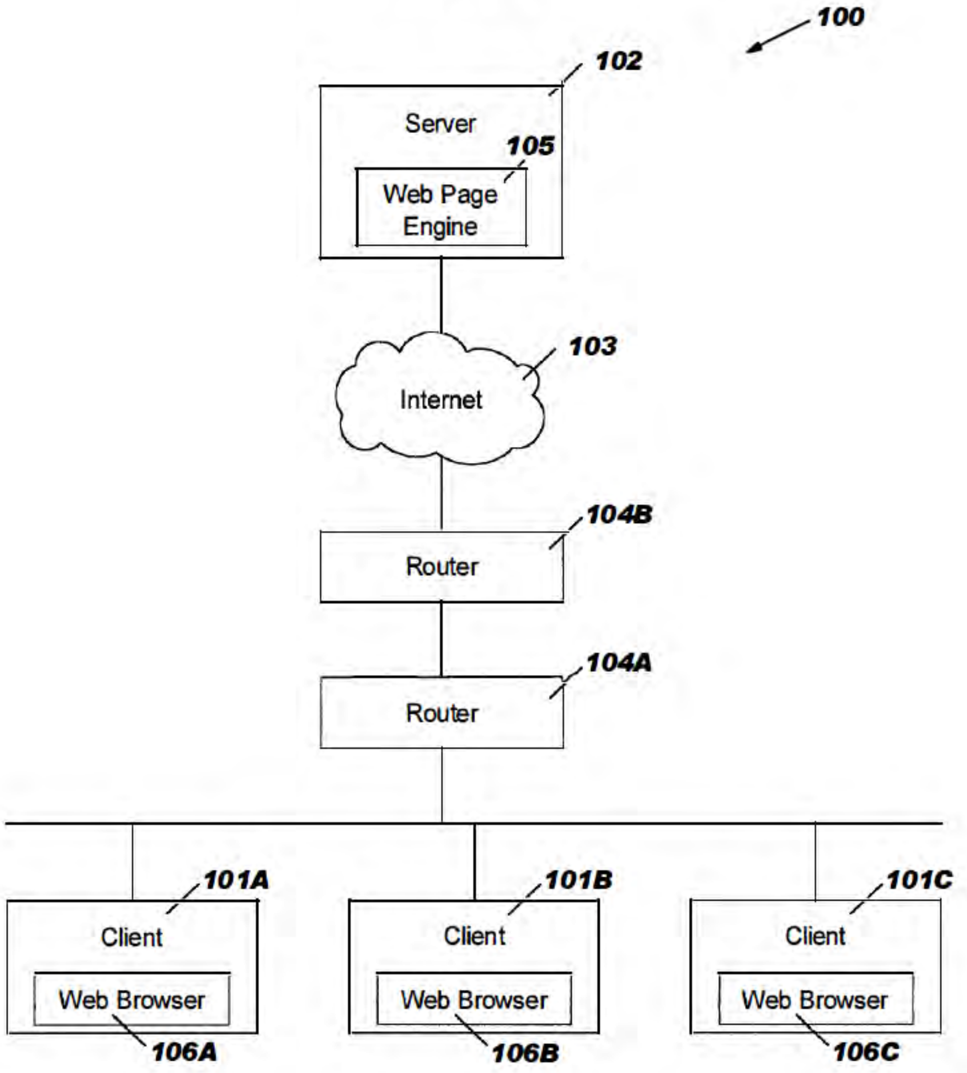
No.	’740 Patent Claim 15	Basso
15[c]	receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes;	Basso discloses receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes. *See supra at* 14[d].
15[d]	for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces; and	Basso discloses for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces. *See supra at* 14[e].
15[e]	sending the data frame over the selected backplane trace,	Basso discloses sending the data frame over the selected backplane trace. *See supra at* 14[f].
15[f]	at least some of the backplane traces being aggregated into an Ethernet link aggregation (LAG) group.	Basso discloses at least some of the backplane traces being aggregated into an Ethernet link aggregation (LAG) group. *See supra at* 15[e], 4[f], 3.

No.	’740 Patent Claim 16	Basso
16	The method according to claim 14, wherein selecting the backplane trace comprises applying a hashing function to the at least one of the frame attributes.	Basso discloses the method according to claim 14, wherein selecting the backplane trace comprises applying a hashing function to the at least one of the frame attributes. *See supra* at 14, 9, 8.

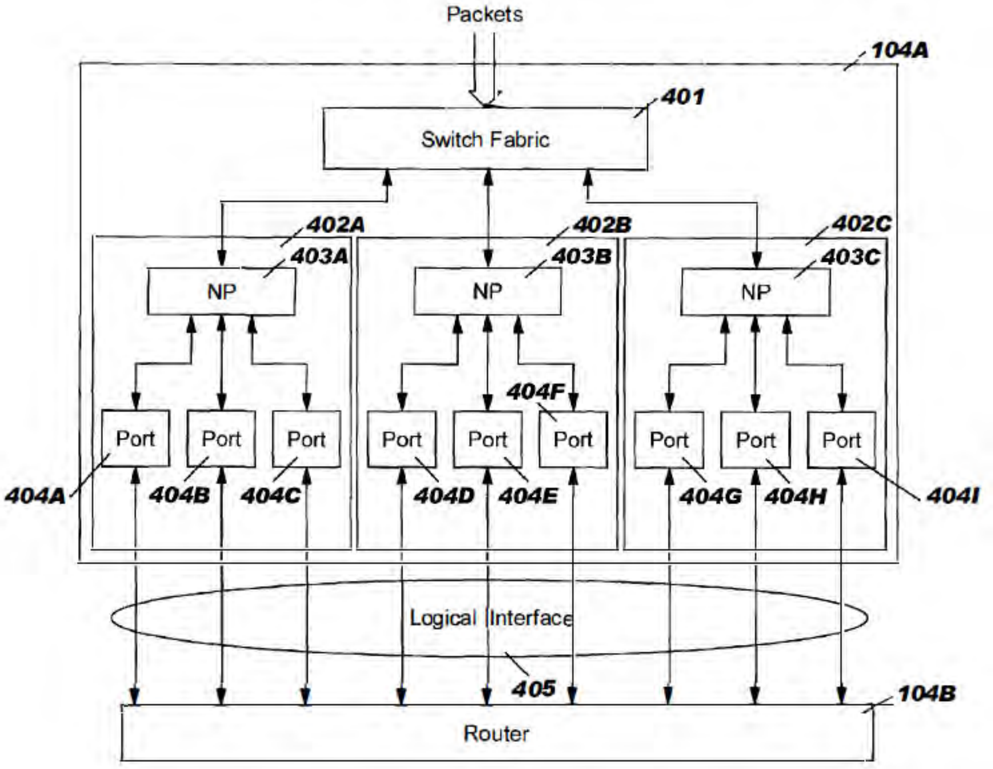
No.	’740 Patent Claim 17	Basso
17[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Basso discloses apparatus for connecting a network node with a communication network. For example, Basso discloses a system for coupling a network device with a network. Basso at Abstract (“A method, system and computer program product for rout-ing packets. A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more physical ports. The physical ports may be connected to another one or more network devices. A plurality of physical ports across one or more blades connected to the same network device may be logically mapped into a logical interface to that network device. By logically grouping a plurality of physical ports into a logical interface to a network device, a network processor may be able to transmit packets of data to that network device across multiple ports instead of one physical port.”) Basso at [0007] (“The problems outlined above may at least in part be solved in some embodiments by mapping a plurality of physical ports connected to a network device into a logical interface to that network device, e.g., router, gateway, edge device, server. That is, a plurality of physical ports may be logically grouped into a logical interface to a network device thereby enabling a network processor to transmit packets of data across multiple ports instead of one physical port.”)

No.	’740 Patent Claim 17	Basso
		Basso at [0009] (“A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more ports. These ports may be connected to another one or more network devices. The switch fabric may be configured to direct incoming packets of data to particular blades where one or more of the network proces-sors in the recipient blade may be configured to process the received packets.”) Basso at [0021] (“FIG. 1 illustrates one embodiment of the present invention of a network system 100. Network system 100 may comprise one or more clients 101A-C connected to a server 102 via the Internet 103. A more detailed description of server 102 is provided further below in conjunction with FIG. 3. The Internet 103 may refer to a network of com-puters. Network system 100 may further comprise one or more routers, e.g., 104A-B, that may be coupled to one or more clients lOlA-C. Routers 104A-B may be configured to forward packets of information from the one or more clients lOlA-C to the Internet 103. Clients lOlA-C may collec-tively or individually be referred to as clients 101 or client 101, respectively. A more detailed description of client 101 is provided further below in conjunction with FIG. 2. Routers 104A-B may collectively or individually be referred to as routers 104 or router 104, respectively. Amore detailed description of router 104 implementing logical interface(s) is provided further below in conjunction with FIG. 4. It is noted that network system 100 may comprise any number of clients 101, any number of servers 102 as well as any number of routers 104 and that FIG. 1 is illustrative. It is further noted that network system 100 may comprise one or more routers (not shown) that may be coupled to server 102. These routers (not shown) may be configured to forward received packets of information to server 102. It is further noted that the connection between clients 101 and the Internet 103 may be any medium type, e.g., wireless, wired. It is further noted that client 101 may be any type of device, e.g., wireless, Personal Digital Assistant (PDA), cell phone, personal computer system, workstation, Internet appliance, configured with the capability of connecting to the Internet 103 and consequently communicating with server 102. It is further noted that FIG. 1 is not to be limited in scope to any one particular embodiment.”)

No.	’740 Patent Claim 17	Basso
		Basso at Figure 1

		FIG. 1  100 102 Server 105 Web Page Engine 103 Internet 104B Router 104A Router 101A Client Web Browser 106A 101B Client Web Browser 106B 101C Client Web Browser 106C

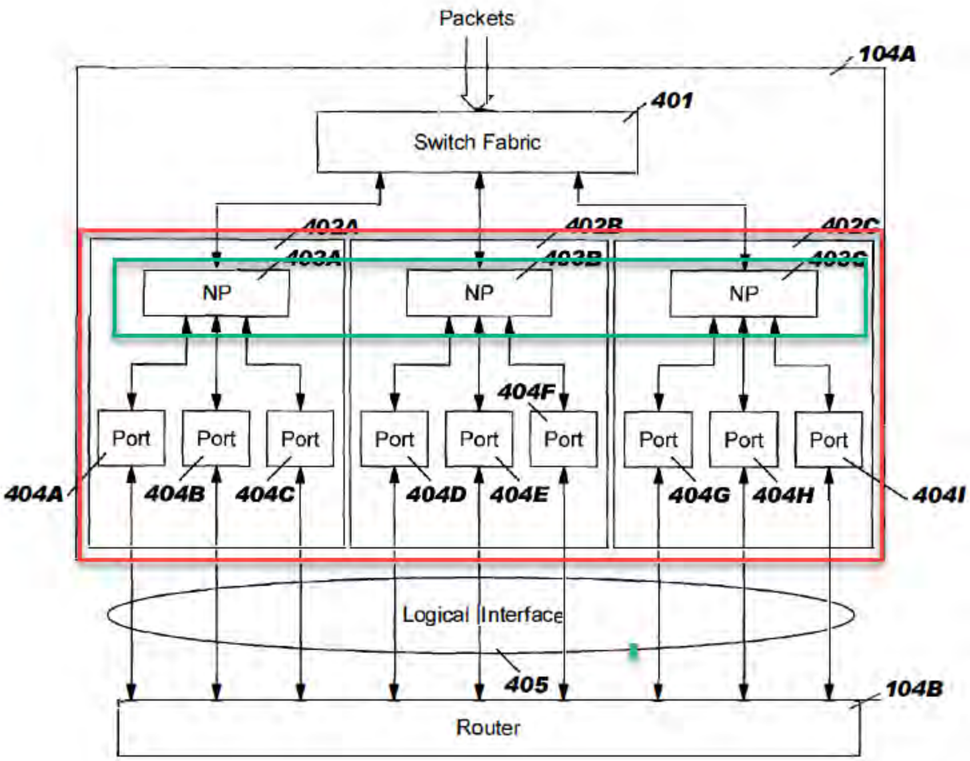
No.	’740 Patent Claim 17	Basso
		Basso at [0030] (“Returning to FIG. 4, router 104A may be config-ured to receive packets of data such as from client 101 (FIG. 1), e.g., client 101A, that may be directed to another particular network device, e.g., router 104B, in network 100 (FIG. 1). Router 104A may comprise a switch fabric 401 configured to direct the incoming packets of data to particu-lar blades 402A-C coupled to switch fabric 401. Blade 402A may comprise a network processor 403A coupled with one or more ports 404A-C. Blade 402B may comprise a network processor 403B coupled with one or more ports 404D-F. Blade 402C may comprise a network processor 403C coupled with one or more ports 404G-I. Blades 402A-C may collectively or individually be referred to as blades 402 or blade 402, respectively. Network processors 403A-C may collectively or individually be referred to as network pro-cessors 403 or network processor 403, respectively. Ports 404A-I may collectively or individually be referred to as ports 404 or port 404, respectively. Each port 404 may be coupled to a particular network device, e.g., gateway, server, router such as router 104B, in network system 100. It is noted that some of ports 404 may be coupled to a separate network device and that FIG. 4 is illustrative. It is further noted that router 104A may comprise any number of blades 402 and each blade 402 may comprise any number of network processors 403 and ports 404.”) Basso at Figure 4

No.	’740 Patent Claim 17	Basso
		FIG. 4  Packets 104A 401 Switch Fabric 402A 402B 402C 403A 403B 403C NP NP NP 404F Port Port Port Port Port Port Port Port Port 404A 404B 404C 404D 404E 404G 404H 404I Logical Interface 405 104B Router
17[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the	Basso discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. For example, Basso discloses a plurality of blades with network processors that can be configured to process data packets with specific data fields that are sent between network devices and the network.

No.	’740 Patent Claim 17	Basso
	communication network,	Basso at Abstract (“A method, system and computer program product for rout-ing packets. A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more physical ports. The physical ports may be connected to another one or more network devices. A plurality of physical ports across one or more blades connected to the same network device may be logically mapped into a logical interface to that network device. By logically grouping a plurality of physical ports into a logical interface to a network device, a network processor may be able to transmit packets of data to that network device across multiple ports instead of one physical port.”) Basso at [0009] (“A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more ports. These ports may be connected to another one or more network devices. The switch fabric may be configured to direct incoming packets of data to particular blades where one or more of the network proces-sors in the recipient blade may be configured to process the received packets.) Basso at [0030] (“Returning to FIG. 4, router 104A may be config-ured to receive packets of data such as from client 101 (FIG. 1), e.g., client 101A, that may be directed to another particular network device, e.g., router 104B, in network 100 (FIG. 1). Router 104A may comprise a switch fabric 401 configured to direct the incoming packets of data to particu-lar blades 402A-C coupled to switch fabric 401. Blade 402A may comprise a network processor 403A coupled with one or more ports 404A-C. Blade 402B may comprise a network processor 403B coupled with one or more ports 404D-F. Blade 402C may comprise a network processor 403C coupled with one or more ports 404G-I. Blades 402A-C may collectively or individually be referred to as blades 402 or blade 402, respectively. Network processors 403A-C may collectively or individually be referred to as network pro-cessors 403 or network processor 403, respectively. Ports 404A-I may collectively or individually be referred to as ports 404 or port 404, respectively. Each port 404 may be coupled to a particular network device, e.g., gateway, server, router such as router 104B, in network system 100. It is noted that some of ports 404 may be coupled to a separate network device and that FIG. 4 is illustrative. It is

No.	’740 Patent Claim 17	Basso
		further noted that router 104A may comprise any number of blades 402 and each blade 402 may comprise any number of network processors 403 and ports 404.”) Basso ay [0031] (“Referring to FIG. 4, network processor 403 may be configured to receive a packet of data from switch fabric 401. Upon receiving a packet, network processor 403 may be configured to process the packet of data. Processing may include but not limited to: determining what activities to be performed on the received packet, transmitting or discarding the received packet, determining which network device, e.g., router such as router 104B, server, edge device, gateway, to transmit the received packet, etc. Network processor 403 may then transmit the processed packet to a particular network device, e.g., router such as router 104B, through a port 404 connected to that network device.”) Basso at [0032] (“As stated in the Background Information section, each port may be physically coupled to another network device, e.g., router. Since the processed packets may be transmitted by a network processor across one particular address, the rate at which the processed packets are trans-mitted is limited to the bandwidth of the connection. That is, since the processed packets may be transmitted by the network processor across one physical port to a particular network device, the rate at which the processed packets are transmitted is limited to the bandwidth of the physical connection between the port and the network device.”) Basso at [0034] (“Network processor 403 may be configured to trans-mit packets of data across multiple physical ports 404 to a particular network device, e.g., router 104B, by logically grouping a plurality of physical ports 404, e.g., ports 404A-I, into a logical interface 405. That is, the physical connections between router 104A and a particular network device, e.g., router 104B, may be logically grouped into a logical inter-face 405. Each set of physical connections between router 104A and a particular network device may be logically grouped into a particular logical interface 405. For example, if ports 404A-F were physically connected to network device #1, then ports 404A-F may be logically grouped into logical interface 405#1. If ports 404G-I were physically connected to network device #2, then ports 404G-I may be logically grouped into logical interface 405#2. It is noted that physical ports 404 may be logically grouped into more than one logical interface 405 and that FIG. 4 is illustrative. It is further

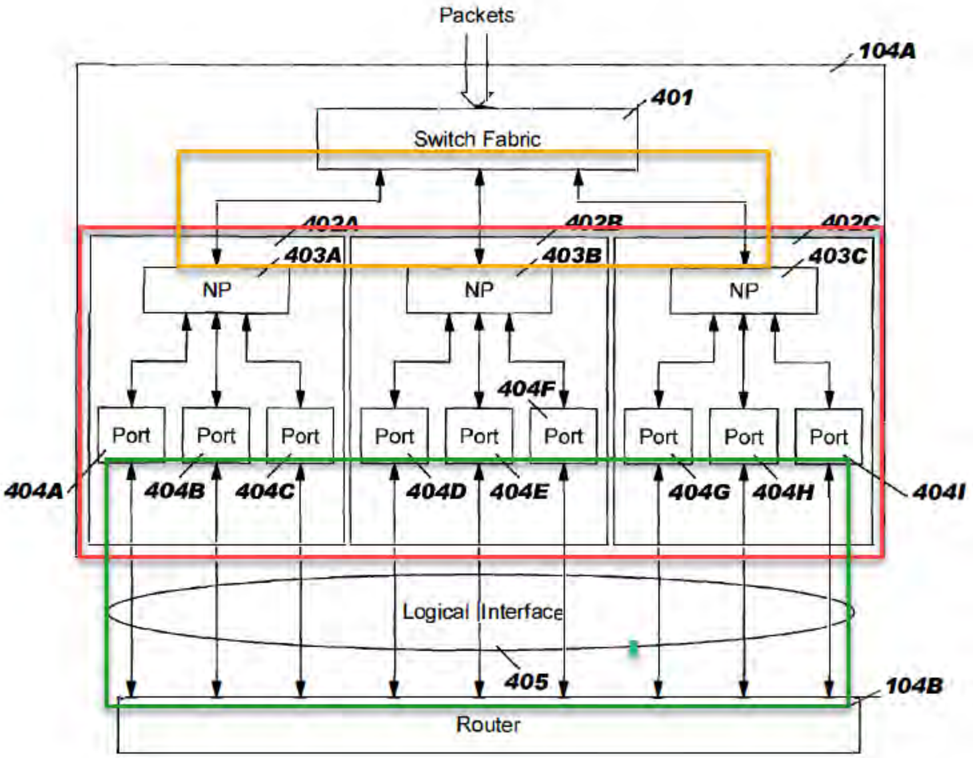
No.	’740 Patent Claim 17	Basso
		noted that each logical interface 405 may be associated with logically grouping a plurality of physical connections to a particular network device, e.g., router, gateway, edge device, server. It is further noted that a logical interface 405 may be associated with logically grouping a plurality of ports in either one blade 402 or across multiple blades 402.”) Basso at [0035] (“By logically grouping a plurality of ports 404 coupled to a particular network device into a logical inter-face 405, network processor 403 may be configured to transmit processed packets to that particular network device via any blade 402/port 404 combination grouped in that logical interface 405. For example, referring to FIG. 4, ports 404A-404I are physically connected to router 104B. If ports 404A-404I were logically grouped into logical interface 405, then a particular network processor 403, e.g., network processor 403A, may be configured to transmit processed packets that are determined to be transmitted to router 104B through any of ports 404A-404I in blades 402A-C, respec-tively. Network processor 403, e.g., network processor 403A, may be configured to transmit the processed packets to router 104B through ports 404, e.g., ports 404D-I, not in its blade 402, e.g., blade 402A, by forwarding the processed packets to switch fabric 401 which may then direct the processed packets to another appropriate physical blade 402/port 404 combination. Network processor 403, e.g., network processor 403A, may further be configured to transmit the processed packets to router 104B through any ports 404, e.g., ports 404A-C, in its blade 402, e.g., blade 402A, instead of just one physical port 404 in its blade 402, e.g., blade 402A. A more detailed description of routing packets implementing logical interface(s) 405 is provided below in FIG. 5.”) Basso at Figure 4 (annotation added)

No.	’740 Patent Claim 17	Basso
		FIG. 4  Packets 104A 401 Switch Fabric 402A 403A 402B 403B 402C 403C NP NP NP 404F Port Port Port Port Port Port Port Port Port 404A 404B 404C 404D 404E 404G 404H 404I Logical Interface 405 104B Router
17[b]	at least one of said interface modules being operative to communicate in both an upstream direction	Basso discloses at least one of said interface modules being operative to communicate in both an upstream direction and a downstream direction. For example, Basso discloses a plurality of blades that receive and transmit data packets in both incoming and outgoing directions.

No.	’740 Patent Claim 17	Basso
	and a downstream direction;	Basso at Abstract (“A method, system and computer program product for rout-ing packets. A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more physical ports. The physical ports may be connected to another one or more network devices. A plurality of physical ports across one or more blades connected to the same network device may be logically mapped into a logical interface to that network device. By logically grouping a plurality of physical ports into a logical interface to a network device, a network processor may be able to transmit packets of data to that network device across multiple ports instead of one physical port.”) Basso at [0009] (“A network device, e.g., router, may comprise a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more ports. These ports may be connected to another one or more network devices. The switch fabric may be configured to direct incoming packets of data to particular blades where one or more of the network proces-sors in the recipient blade may be configured to process the received packets.) Basso at [0030] (“Returning to FIG. 4, router 104A may be config-ured to receive packets of data such as from client 101 (FIG. 1), e.g., client 101A, that may be directed to another particular network device, e.g., router 104B, in network 100 (FIG. 1). Router 104A may comprise a switch fabric 401 configured to direct the incoming packets of data to particu-lar blades 402A-C coupled to switch fabric 401. Blade 402A may comprise a network processor 403A coupled with one or more ports 404A-C. Blade 402B may comprise a network processor 403B coupled with one or more ports 404D-F. Blade 402C may comprise a network processor 403C coupled with one or more ports 404G-I. Blades 402A-C may collectively or individually be referred to as blades 402 or blade 402, respectively. Network processors 403A-C may collectively or individually be referred to as network pro-cessors 403 or network processor 403, respectively. Ports 404A-I may collectively or individually be referred to as ports 404 or port 404, respectively. Each port 404 may be coupled to a particular network device, e.g., gateway, server, router such as router 104B, in network system 100. It is noted that some of

No.	’740 Patent Claim 17	Basso
		ports 404 may be coupled to a separate network device and that FIG. 4 is illustrative. It is further noted that router 104A may comprise any number of blades 402 and each blade 402 may comprise any number of network processors 403 and ports 404.”) Basso ay [0031] (“Referring to FIG. 4, network processor 403 may be configured to receive a packet of data from switch fabric 401. Upon receiving a packet, network processor 403 may be configured to process the packet of data. Processing may include but not limited to: determining what activities to be performed on the received packet, transmitting or discarding the received packet, determining which network device, e.g., router such as router 104B, server, edge device, gateway, to transmit the received packet, etc. Network processor 403 may then transmit the processed packet to a particular network device, e.g., router such as router 104B, through a port 404 connected to that network device.”) Basso at [0032] (“As stated in the Background Information section, each port may be physically coupled to another network device, e.g., router. Since the processed packets may be transmitted by a network processor across one particular address, the rate at which the processed packets are trans-mitted is limited to the bandwidth of the connection. That is, since the processed packets may be transmitted by the network processor across one physical port to a particular network device, the rate at which the processed packets are transmitted is limited to the bandwidth of the physical connection between the port and the network device.”) Basso at [0034] (“Network processor 403 may be configured to trans-mit packets of data across multiple physical ports 404 to a particular network device, e.g., router 104B, by logically grouping a plurality of physical ports 404, e.g., ports 404A-I, into a logical interface 405. That is, the physical connections between router 104A and a particular network device, e.g., router 104B, may be logically grouped into a logical inter-face 405. Each set of physical connections between router 104A and a particular network device may be logically grouped into a particular logical interface 405. For example, if ports 404A-F were physically connected to network device #1, then ports 404A-F may be logically grouped into logical interface 405#1. If ports 404G-I were physically connected to network device #2, then ports 404G-I may be logically grouped into logical interface 405#2. It is noted that physical ports 404 may be logically

No.	’740 Patent Claim 17	Basso
		grouped into more than one logical interface 405 and that FIG. 4 is illustrative. It is further noted that each logical interface 405 may be associated with logically grouping a plurality of physical connections to a particular network device, e.g., router, gateway, edge device, server. It is further noted that a logical interface 405 may be associated with logically grouping a plurality of ports in either one blade 402 or across multiple blades 402.”) Basso at [0035] (“By logically grouping a plurality of ports 404 coupled to a particular network device into a logical inter-face 405, network processor 403 may be configured to transmit processed packets to that particular network device via any blade 402/port 404 combination grouped in that logical interface 405. For example, referring to FIG. 4, ports 404A-404I are physically connected to router 104B. If ports 404A-404I were logically grouped into logical interface 405, then a particular network processor 403, e.g., network processor 403A, may be configured to transmit processed packets that are determined to be transmitted to router 104B through any of ports 404A-404I in blades 402A-C, respec-tively. Network processor 403, e.g., network processor 403A, may be configured to transmit the processed packets to router 104B through ports 404, e.g., ports 404D-I, not in its blade 402, e.g., blade 402A, by forwarding the processed packets to switch fabric 401 which may then direct the processed packets to another appropriate physical blade 402/port 404 combination. Network processor 403, e.g., network processor 403A, may further be configured to transmit the processed packets to router 104B through any ports 404, e.g., ports 404A-C, in its blade 402, e.g., blade 402A, instead of just one physical port 404 in its blade 402, e.g., blade 402A. A more detailed description of routing packets implementing logical interface(s) 405 is provided below in FIG. 5.”) Basso at Figure 4 (annotation added)

No.	’740 Patent Claim 17	Basso
		FIG. 4  Packets 104A 401 Switch Fabric 402A 402B 402C 403A 403B 403C NP NP NP 404F Port Port Port Port Port Port Port Port Port 404A 404B 404C 404D 404E 404G 404H 404I Logical Interface 405 104B Router
17[c]	a first group of first physical links arranged in parallel so as to couple the network node to the	Basso discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 1[a].

No.	’740 Patent Claim 17	Basso
	one or more interface modules;	
17[d]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Basso discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 1[c].
17[e]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame;	Basso discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 1[f].
17[f]	at least one of said first physical links and at least one of said second links	Basso discloses at least one of said first physical links and at least one of said second links being bi-directional links operative to communicate in both said upstream direction and said downstream direction.

No.	’740 Patent Claim 17	Basso
	being bi-directional links operative to communicate in both said upstream direction and said downstream direction.	*See supra at* 1[b], 1[d].

No.	’740 Patent Claim 18	Basso
18[a]	The apparatus according to claim 17, and comprising a backplane to which the one or more interface modules are coupled,	Basso discloses the apparatus according to claim 17, and comprising a backplane to which the one or more interface modules are coupled. *See supra at* 3, 17.
18[b]	wherein the second physical links comprise backplane traces formed on the backplane.	Basso discloses wherein the second physical links comprise backplane traces formed on the backplane. *See supra at* 3, 17.

No.	’740 Patent Claim 19	Basso
19[preamble]	Apparatus for connecting a network node with a	Basso discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].

No.	’740 Patent Claim 19	Basso
	communication network, comprising:	
19[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	Basso discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].
19[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Basso discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].
19[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Basso discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].
19[d]	a control module, which is arranged to select for each data frame sent between the communication	Basso discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame.

No.	’740 Patent Claim 19	Basso
	network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,	*See supra at* 17[e].
19[e]	at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group.	Basso discloses at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group. *See supra at* 4[f].

No.	’740 Patent Claim 20	Basso
20[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Basso discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
20[a]	one or more interface modules, which are arranged to process data frames having	Basso discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].

No.	’740 Patent Claim 20	Basso
	frame attributes sent between the network node and the communication network;	
20[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Basso discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].
20[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Basso discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].
20[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first	Basso discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].

No.	’740 Patent Claim 20	Basso
	physical link out of the first group and a second physical link out of the second group over which to send the data frame,	
20[e]	two or more of the first physical links being aggregated into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node.	Basso discloses two or more of the first physical links being aggregated into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node. *See supra at* 19[e], 5[f].

No.	’740 Patent Claim 21	Basso
21	The apparatus according to claim 17, and comprising a multiplexer, which is arranged to perform at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing	Basso discloses the apparatus according to claim 17, and comprising a multiplexer, which is arranged to perform at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node. *See supra at* 6, 17.

No.	’740 Patent Claim 21	Basso
	downstream data frames sent from the communication network to the network node.	

No.	’740 Patent Claim 22	Basso
22	The apparatus according to claim 17, wherein the control module is arranged to balance a frame data rate among at least some of the first and second physical links.	Basso discloses the apparatus according to claim 17, wherein the control module is arranged to balance a frame data rate among at least some of the first and second physical links. *See supra at* 7, 17.

No.	’740 Patent Claim 23	Basso
23	The apparatus according to claim 17, wherein the control module is arranged to apply a mapping function to the at least one of the frame attributes so as to select the first and	Basso discloses the apparatus according to claim 17, wherein the control module is arranged to apply a mapping function to the at least one of the frame attributes so as to select the first and second physical links. *See supra at* 8, 17.

No.	’740 Patent Claim 23	Basso
	second physical links.	

No.	’740 Patent Claim 24	Basso
24	The apparatus according to claim 23, wherein the mapping function comprises a hashing function.	Basso discloses the apparatus according to claim 23, wherein the mapping function comprises a hashing function. *See supra at* 9, 23.

No.	’740 Patent Claim 25	Basso
25[a]	The apparatus according to claim 24, wherein the control module is arranged to determine a hashing size responsively to a number of at least some of the first and second physical links,	Basso discloses the apparatus according to claim 24, wherein the control module is arranged to determine a hashing size responsively to a number of at least some of the first and second physical links. *See supra at* 10[a], 24.
25[b]	to apply the hashing function to the at least one of the frame attributes to produce a hashing key,	Basso discloses to apply the hashing function to the at least one of the frame attributes to produce a hashing key. *See supra at* 10[b].

No.	’740 Patent Claim 25	Basso
25[c]	to calculate a modulo of a division operation of the hashing key by the hashing size, and	Basso discloses to calculate a modulo of a division operation of the hashing key by the hashing size. *See supra at* 10[c].
25[d]	to select the first and second physical links responsively to the modulo.	Basso discloses to select the first and second physical links responsively to the modulo. *See supra at* 10[d].

No.	’740 Patent Claim 26	Basso
26	The apparatus according to claim 25, wherein the control module is arranged to select the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo.	Basso discloses the apparatus according to claim 25, wherein the control module is arranged to select the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo. *See supra at* 11, 25.

No.	’740 Patent Claim 27	Basso
27	The apparatus according to claim	Basso discloses the apparatus according to claim 17, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header

No.	’740 Patent Claim 27	Basso
	17, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port.	field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port. *See supra at* 12, 17.

No.	’740 Patent Claim 28	Basso
28[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Basso discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
28[a]	one or more interface modules, which are arranged to process data frames having	Basso discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].

No.	’740 Patent Claim 28	Basso
	frame attributes sent between the network node and the communication network;	
28[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Basso discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].
28[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Basso discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].
28[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first	Basso discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].

No.	’740 Patent Claim 28	Basso
	physical link out of the first group and a second physical link out of the second group over which to send the data frame,	
28[e]	the communication network being arranged to provide a communication service to the network node,	Basso discloses the communication network being arranged to provide a communication service to the network node. *See supra at* 2[b].
28[f]	the service having specified bandwidth requirements comprising at least one of a committed information rate (CR), a peak information rate (PIR) and an excess information rate (EIR), and	Basso discloses the service having specified bandwidth requirements comprising at least one of a committed information rate (CR), a peak information rate (PIR) and an excess information rate (EIR). *See supra at* 13[i].
28[g]	the first and second groups of physical links being dimensioned to provide an allocated bandwidth for the communication service responsively	Basso discloses the first and second groups of physical links being dimensioned to provide an allocated bandwidth for the communication service responsively to the bandwidth requirements. *See supra at* 13[j].

No.	’740 Patent Claim 28	Basso
	to the bandwidth requirements.	

No.	’740 Patent Claim 29	Basso
29[preamble]	Apparatus for connecting user ports to a communication network, comprising:	Basso discloses apparatus for connecting user ports to a communication network. *See supra at* 17[preamble], 14[preamble].
29[a]	one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network,	Basso discloses one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network. *See supra at* 17[a], 14[a].
29[b]	at least one of said user interface modules being bi-directional and operative to communicate in both an upstream direction and a downstream direction;	Basso discloses at least one of said user interface modules being bi-directional and operative to communicate in both an upstream direction and a downstream direction. *See supra at* 17[b], 14[c].

No.	’740 Patent Claim 29	Basso
29[c]	a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network,	Basso discloses a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network. *See supra at* 14[b]-[e].
29[d]	at least one of said backplane traces being bi-directional and operative to communicate in both said upstream direction and said downstream direction; and	Basso discloses at least one of said backplane traces being bi-directional and operative to communicate in both said upstream direction and said downstream direction. *See supra at* 14[c], 17[b].
29[e]	a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from	Basso discloses a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame. *See supra at* 14[e], 17[e].

No.	’740 Patent Claim 29	Basso
	the plurality of backplane traces over which to send the data frame.	

No.	’740 Patent Claim 30	Basso
30[preamble]	Apparatus for connecting user ports to a communication network, comprising:	Basso discloses apparatus for connecting user ports to a communication network. *See supra at* 29[preamble].
30[a]	one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network;	Basso discloses one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network. *See supra at* 29[a].
30[b]	a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data	Basso discloses a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network. *See supra at* 29[c].

No.	’740 Patent Claim 30	Basso
	frames between the one or more user interface modules and the communication network;	
30[c]	a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame;	Basso discloses a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame. *See supra at* 29[e].
30[d]	at least some of the backplane traces are aggregated into an Ethernet link aggregation (LAG) group.	Basso discloses at least some of the backplane traces are aggregated into an Ethernet link aggregation (LAG) group. *See supra at* 4[f], 15[f].

No.	’740 Patent Claim 31	Basso
31	The apparatus according to claim 29, wherein the control module is arranged to apply a	Basso discloses the apparatus according to claim 29, wherein the control module is arranged to apply a hashing function to the at least one of the frame attributes so as to select the backplane trace. *See supra at* 16, 29, 30[c].

No.	’740 Patent Claim 31	Basso
	hashing function to the at least one of the frame attributes so as to select the backplane trace.	

EXHIBIT C-8
Defendant's Preliminary Invalidity Contentions
Orckit Corporation v. Cisco Systems, Inc., 2:22-cv-00276-JRG-RSP

Chart for U.S. Patent 7,545,740 ("the '740 Patent")
U.S. Patent Publication No. 2006/0039366 to Ghosh et al. ("Ghosh")

As shown in the chart below, all Asserted Claims of the '740 Patent are invalid under (1) 35 U.S.C. § 102 (a), (e), and (g) because Ghosh meets each element of those claims, and/or (2) 35 U.S.C. § 103 because Ghosh renders those claims obvious either alone, or in combination with the knowledge of a person having ordinary skill in the art, and in further combination with the references specifically identified below and in the following claim chart and/or one or more references identified in Defendant's Preliminary Invalidity Contentions. The following quotations and diagrams come from Ghosh titled "Port Aggregation For Fibre Channel Interfaces", which was filed on August 20, 2004, and published on February 23, 2006.

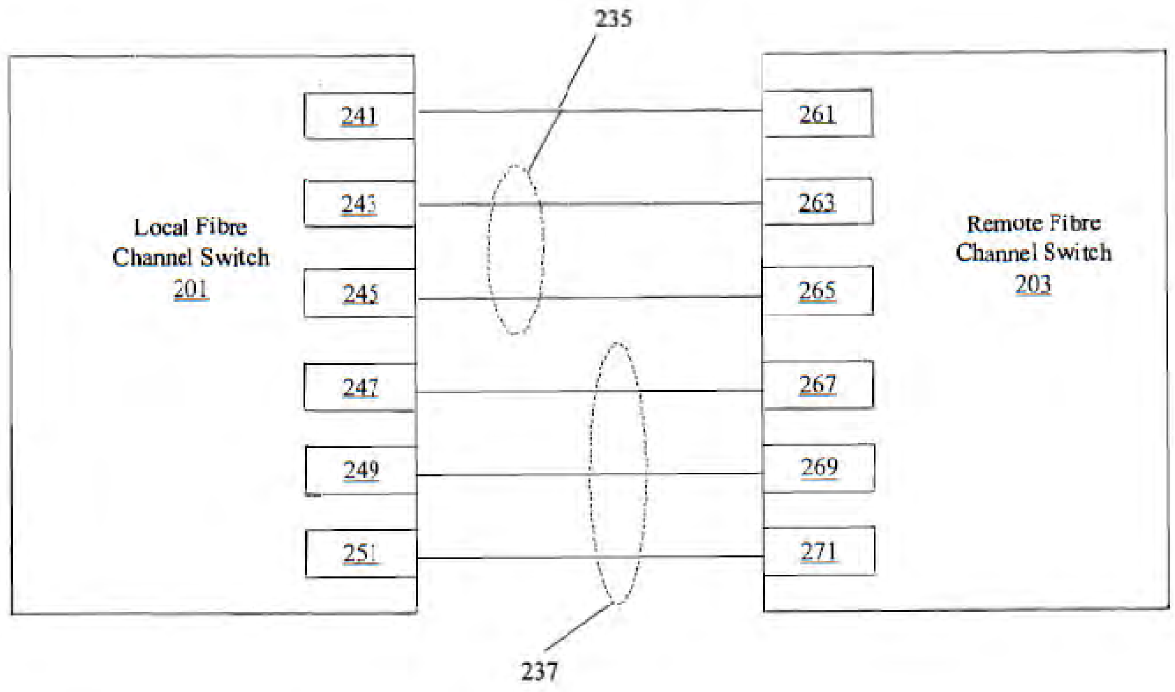
Motivations to combine the disclosures in Ghosh with disclosures in other publications known in the art, as explained in this chart, include at least the similarity in subject matter between the references to the extent they concern methods of data communication systems, and specifically to methods and systems for link aggregation in a data communication network. Insofar as the references cite other patents or publications, or suggest additional changes, one of ordinary skill in the art would look beyond a single reference to other references in the field.

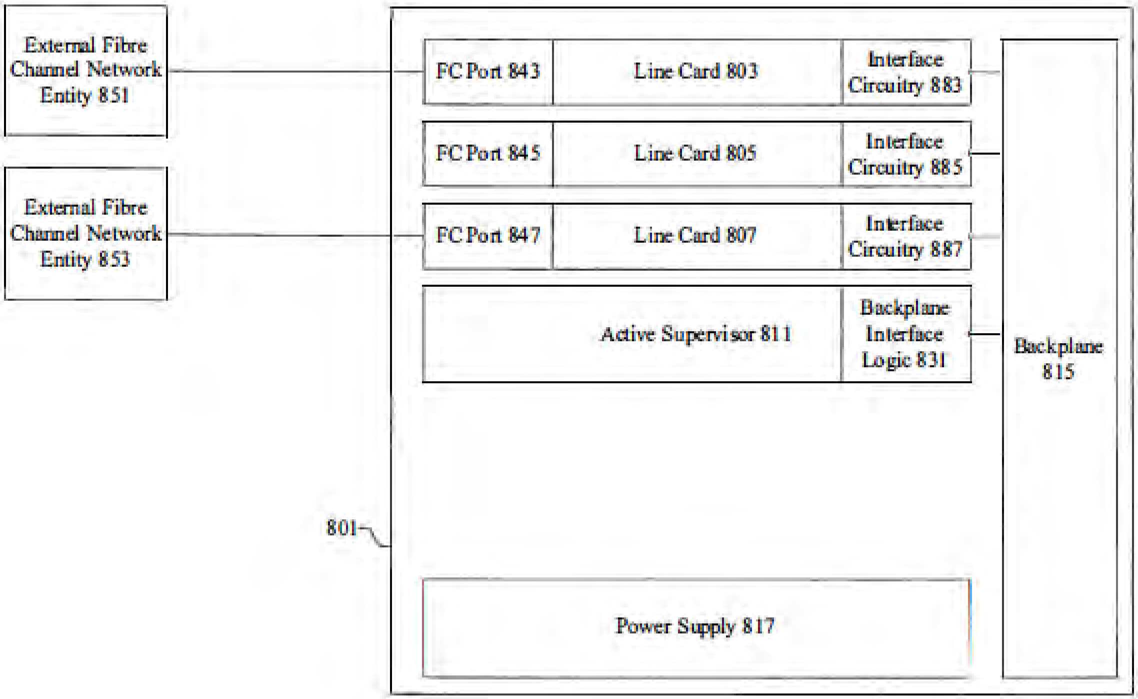
These invalidity contentions are based on Defendant's present understanding of the asserted claims, and Orckit's apparent construction of the claims in its November 3, 2022 Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1, and Orckit's January 19, 2023 First Amended Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1 (Orckit's "Infringement Disclosures"), which is deficient at least insofar as it fails to cite any documents or identify accused structures, acts, or materials in the Accused Products with particularity. Defendant does not agree with Orckit's application of the claims, or that the claims satisfy the requirements of 35 U.S.C. § 112. Defendant's contentions herein are not, and should in no way be seen as, admissions or adoptions as to any particular claim scope or construction, or as any admission that any particular element is met by any accused product in any particular way. Defendant objects to any attempt to imply claim construction from this chart. Defendant's prior art invalidity contentions are made in a variety of alternatives and do not represent Defendant's agreement or view as to the meaning, definiteness, written description support for, or enablement of any claim contained therein.

The following contentions are subject to revision and amendment pursuant to Federal Rule of Civil Procedure 26(e), the Local Rules, and the Orders of record in this matter subject to further investigation and discovery regarding the prior art and the Court's construction of the claims at issue.

No.	'740 Patent Claim 1	Ghosh
1[preamble]	A method for communication, comprising:	Ghosh discloses a method for communication. For example, Ghosh discloses a method for communicating between network entities using a port channel. Ghosh at Abstract ("According to the present invention, methods and apparatus are provided to allow efficient and effective aggregation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent gen-eration of port channel map tables.") Ghosh at [0007] ("According to the present invention, methods and apparatus are provided to allow efficient and effective aggre-gation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent generation of port channel map tables.") Ghosh at [0008] ("In one embodiment, a method for aggregating ports in a fibre channel fabric is provided. It is determined that a plurality of local ports at a local fibre channel switch are compatible. Identifiers for the plurality of local ports are sent to a remote fibre channel switch. The remote fibre channel switch determines if a plurality of remote ports are compat-ible, the plurality of remote ports corresponding to the plurality of local ports. An indication that one or more of the remote physical ports are compatible is received. A port channel including one or more of the local ports correspond-ing to the compatible remote ports is created.") Ghosh at [0009] ("In another embodiment, a fibre channel switch is provided. The fibre channel switch includes memory, a plurality of local ports, and a processor. The plurality of local ports

No.	’740 Patent Claim 1	Ghosh
		are coupled to a remote fibre channel switch through a plurality of remote ports. The processor is con-figured to determine that a subset of the plurality of local ports at a local fibre channel switch are compatible and send identifiers for the subset of the plurality of local ports to a remote fibre channel switch. The remote fibre channel switch determines if a subset of the plurality of remote ports are compatible. The subset of the plurality of remote ports corresponds to the subset of the plurality of local ports.”) Ghosh at [0010] (“In another embodiment, a fibre channel network is described. The fibre channel network includes a local fibre channel switch and a remote fibre channel switch. The local fibre channel switch aggregates a compatible subset of the plurality of local ports and sends identifiers for the compat-ible subset of the plurality of local ports to the remote fibre channel switch. The remote fibre channel switch determines if a subset of the plurality of remote ports are compatible. The subset of the plurality of remote ports corresponds to the compatible subset of the plurality of local ports.”) Ghosh at [0059] (“Line cards 803, 805, and 807 can communicate with an active supervisor 811 through interface circuitry 883, 885, and 887 and the backplane 815. According to various embodiments, each line card includes a plurality of ports that can act as either input ports or output ports for communication with external fibre channel network entities 851 and 853. The backplane 815 can provide a communi-cations channel for all traffic between line cards and super- visors. Individual line cards 803 and 807 can also be coupled to external fibre channel network entities 851 and 853 through fibre channel ports 843 and 847.”) Ghosh at [0060] (“External fibre channel network entities 851 and 853 can be nodes such as other fibre channel switches, disks, RAIDS, tape libraries, or servers. It should be noted that the switch can support any number of line cards and supervisors. In the embodiment shown, only a single supervisor is connected to the backplane 815 and the single supervisor communicates with many different line cards. The active supervisor 811 may be configured or designed to run a plurality of applications such as routing, domain manager, system manager, and utility applications.”)

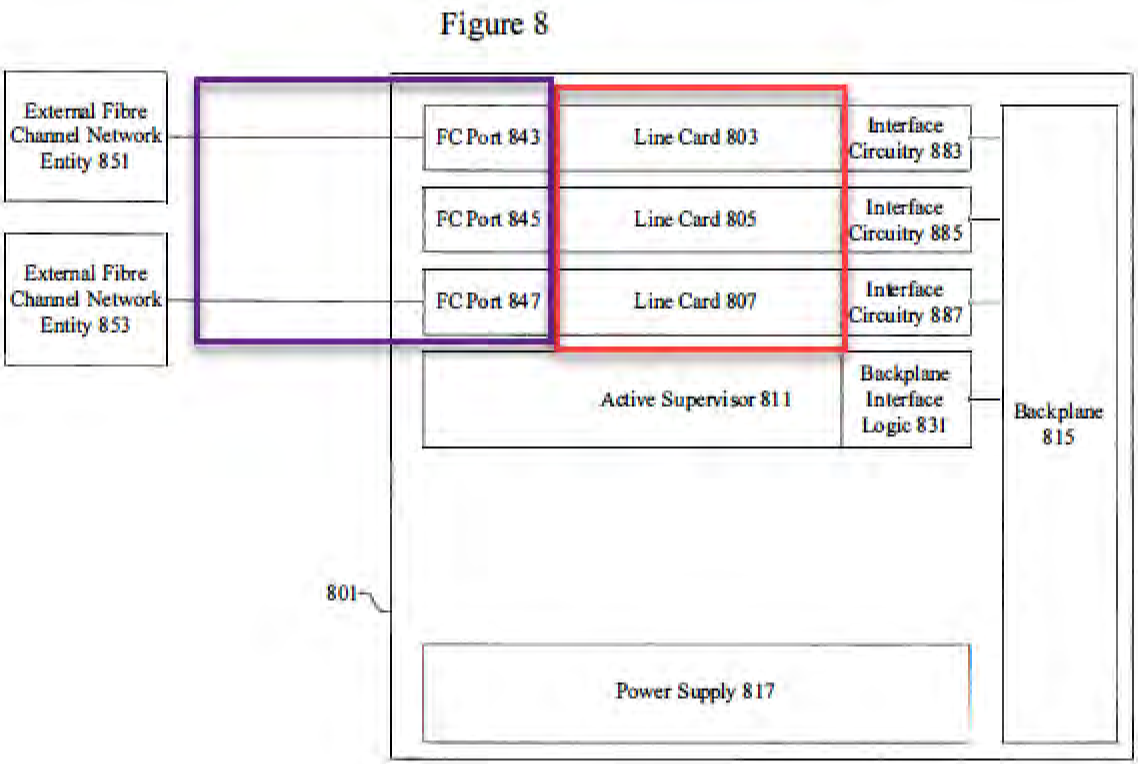
No.	’740 Patent Claim 1	Ghosh
		Ghosh at Figure 2  235 Local Fibre Channel Switch 201 241 243 245 247 249 251 261 263 265 267 269 271 Remote Fibre Channel Switch 203 237 Figure 2 Ghosh at Figure 8

No.	’740 Patent Claim 1	Ghosh
		Figure 8  External Fibre Channel Network Entity 851 External Fibre Channel Network Entity 853 FC Port 843 Line Card 803 Interface Circuitry 883 FC Port 845 Line Card 805 Interface Circuitry 885 FC Port 847 Line Card 807 Interface Circuitry 887 Active Supervisor 811 Backplane Interface Logic 831 Backplane 815 801 Power Supply 817
1[a]	coupling a network node to one or more interface modules using a	Ghosh discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel.

No.	’740 Patent Claim 1	Ghosh
	first group of first physical links arranged in parallel,	For example, Ghosh discloses line cards that use a plurality of ports that can act as either parallel input ports or output ports for communication with external fibre channel network entities. Ghosh at [0022] (“Switches in a fibre channel network are typically interconnected using multiple physical links. The physical links connecting a pair of switches allows transmission of data and control signals. In some instances, it is useful to aggregate multiple physical links into a logical link. Physi-cal links are also referred to herein as physical interfaces and channels while logical links are also referred to herein as logical interfaces and port channels. For example, a local switch may be connected to a remote switch through four physical links. Instead of having to transmit data through a particular physical link, the physical links can be aggregated to form one or more logical links. In one example, all four physical links are aggregated into a single logical link. Instead of having data transmitted through a particular physical link, the data can merely be transmitted over a particular logical link without regard to the particular physi-cal interface used. Aggregating physical links into a logical link allows for higher aggregated bandwidth, load balancing, and link redundancy. For example, if a particular physical link fails or is overloaded, data can still be transmitted over the logical link.”) Ghosh at [0027] (“FIG. 1 shows one example of a storage area network implemented using fibre channel that can use efficient port channel configuration mechanisms. A switch 101 is coupled to switches 103 and 105 as well as to a host 111 and storage 121. Switch 101 may be connected to other entities through multiple physical links or channels config-ured as logical links or port channels. In one embodiment, host 111 may be a server or client system while storage 121 may be single disk or a redundant array of independent disks (RAID). Switches 103 and 105 are both coupled to switch 107. Switch 107 is connected to host 113 and switch 103 is connected to storage 123. Switch 109 is connected to host 115, switch 107, disk array 153, and an external network 151 that may or may not use fibre channel.”) Ghosh at [0029] (“FIG. 2 is a diagrammatic representation showing links between two switches, such as two fibre channel switches shown in FIG. 1. A local fibre channel switch 201 includes local ports 241, 243, 245, 247, 249, and 251. A remote fibre channel switch 203 includes remote ports 261, 263, 265, 267, 269, and 271. Local port 241 is coupled to remote port 261 through an

No.	’740 Patent Claim 1	Ghosh
		individual physical link or channel. Connected ports are also referred to herein as peer ports. Local port 243 is coupled to remote port 263 and local port 245 is coupled to remote port 265. The two resulting physical links are aggregated to form port channel 235. Local ports 247, 249, and 251 are coupled to remote ports 267, 269, and 271 respectively. The three resulting physical links are aggregated to form port channel 237.”) Ghosh at [0039] (“At 431, remote switch 403 uses the information received from the local switch 401 to update a port channel database. In one example, the remote switch 403 can check if the port Bl is already assigned to a different port channel. If the port is not already assigned to a different port channel, the remote switch 403 can proceed and send a sync accept message 413 in response to the sync message 411. The sync accept message 413 includes a remote switch 403 assigned world wide name for the remote port channel identifier. The sync accept message indicates that a port channel can now be formed. At 423, local switch 401 uses the information to update its own port channel database. However, the port channel may not yet be operational until the hardware configuration is completed. The local switch 401 continues hardware configuration such as line card configuration to make the port Al part of the port channel. An acknowledg-ment 427 is sent and received by remote switch 403 at 429. In some examples, the local switch 401 sends a commit signal 415 when hardware configuration is complete.”) Ghosh at [0044] (“At 531, remote switch 503 uses the information received from the local switch 501 to verify port B2 is compatible with other port in port channel C2. In one example, configuration parameters associated with B2 are checked against configuration parameters associated with Bl. The remote switch 503 can also check if the port B2 is already assigned to a different port channel. If the port B2 is compatible with port Bl, the remote switch 503 can proceed and send a sync accept message 513 in response to the sync message 511 to indicate that the port B2 can be aggregated into the port channel. The sync accept message indicates that a port channel can now be modified. At 523, local switch 501 uses the information to update its own port channel database. However, the port channel may not yet be fully operational until the hardware configuration is completed. The local switch 501 continues hardware configuration such as line card configuration to make the port A2 part of the port channel Cl. An

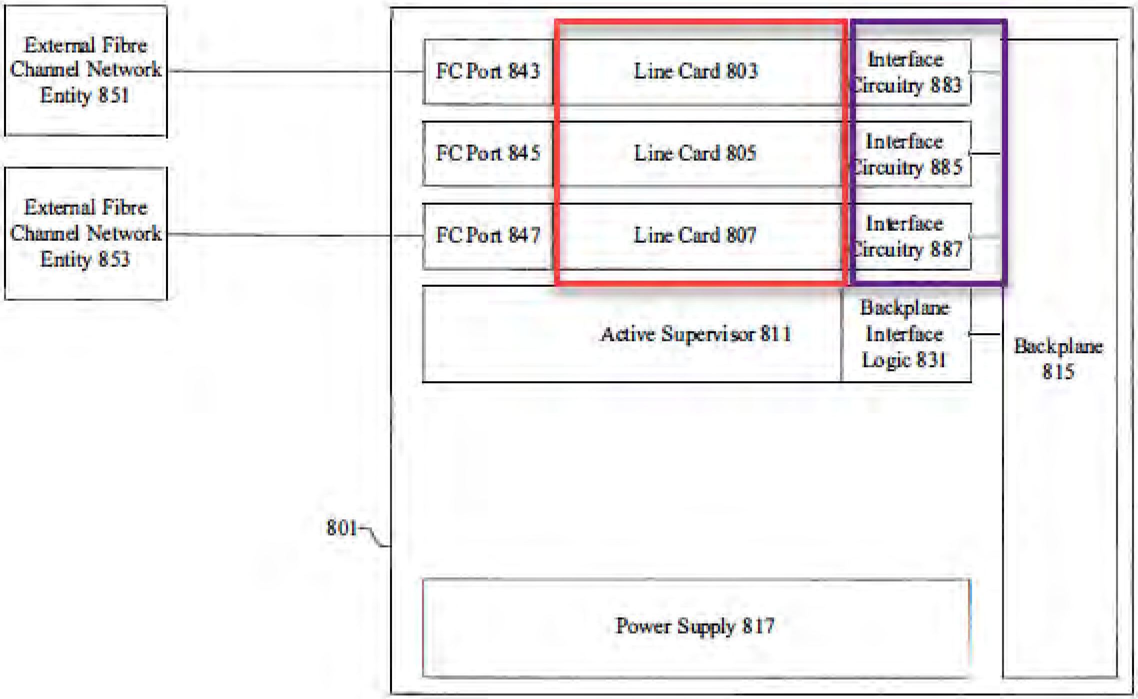
No.	’740 Patent Claim 1	Ghosh
		acknowledgment 527 is sent and received by remote switch 503 at 529. In some examples, the local switch 501 sends a commit signal 515 when hardware configuration is complete.”) Ghosh at [0057] (“As described above, techniques for aggregating ports may be performed in a variety of network devices or switches. According to various embodiments, a switch includes a processor, network interfaces, and memory. A variety of ports, Media Access Control (MAC) blocks, and buffers can also be provided as will be appreciated by one of skill in the art.”) Ghosh at [0058] (“FIG. 8 is a diagrammatic representation of one example of a fibre channel switch that can be used to implement techniques of the present invention. Although one particular configuration will be described, it should be noted that a wide variety of switch and router configurations are available. The fibre channel switch 801 may include one or more supervisors 811. According to various embodi-ments, the supervisor 811 has its own processor, memory, and storage resources.”) Ghosh at [0059] (“Line cards 803, 805, and 807 can communicate with an active supervisor 811 through interface circuitry 883, 885, and 887 and the backplane 815. According to various embodiments, each line card includes a plurality of ports that can act as either input ports or output ports for communication with external fibre channel network entities 851 and 853. The backplane 815 can provide a communi-cations channel for all traffic between line cards and super- visors. Individual line cards 803 and 807 can also be coupled to external fibre channel network entities 851 and 853 through fibre channel ports 843 and 847.”) Ghosh at [0060] (“External fibre channel network entities 851 and 853 can be nodes such as other fibre channel switches, disks, RAIDS, tape libraries, or servers. It should be noted that the switch can support any number of line cards and supervisors. In the embodiment shown, only a single supervisor is connected to the backplane 815 and the single supervisor communicates with many different line cards. The active supervisor 811 may be configured or designed to run a plurality of applications such as routing, domain manager, system manager, and utility applications.”)

No.	’740 Patent Claim 1	Ghosh
		Ghosh at Figure 8  Figure 8 External Fibre Channel Network Entity 851 External Fibre Channel Network Entity 853 FC Port 843 FC Port 845 FC Port 847 Line Card 803 Line Card 805 Line Card 807 Interface Circuitry 883 Interface Circuitry 885 Interface Circuitry 887 Active Supervisor 811 Backplane Interface Logic 831 Backplane 815 801 Power Supply 817
1[b]	at least one of said first physical links being a bi-directional link operative to communicate in	Ghosh discloses at least one of said first physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction. For example, Ghosh discloses input and output ports that exchange data frames for a given flow over the same physical link in both directions.

No.	’740 Patent Claim 1	Ghosh
	both an upstream direction and a downstream direction	Ghosh at [0026] (“Consequently, the techniques and mechanisms of the present invention allow automatic detection of compat-ible ports to enable automatic creation of port channels. Port channels can be effectively brought up at either a local switch or a remote switch after either automatic creation of port channels or manual configuration of port channels. Robust error detection capabilities allow the correction of improper configurations and connections. Member physical ports of a port channel can operate as individual links if they cannot be configured to be part of port channel. Further-more, synchronization is supported so that requests and responses belonging to the same flow can be carried over the same physical link in a port channel in both directions. In many conventional implementations, such as Ethernet for example, a flow belonging to a particular port channel could be carried over different physical links during send and receive phases.”) Ghosh at [0032] (“Each entity can also have additional parameters to aid in the set up of port channels. According to various embodiments, parameters such as a channeling model, a channeling intent and a channeling status are included. A channeling model indicates to a peer port the channel group is automatically created or user configured. A channeling intent parameter indicates the peer port if this port intends to participate in a port channel. Otherwise, the port intends to operate as an individual port. The channeling status param-eter tells the peer port about its current channeling status. This parameter is exchanged by the attached peer ports to agree upon the channeling status of the link and to ensure that both ends are synchronized.”) Ghosh at [0055] (“Based on this scheme, frames for a given flow are transmitted through the same physical port of the port channel. However, unless there is proper synchronization of the port channel map tables at the two ends of a port channel, it is possible that requests and responses for the same flow are carried over two different physical links. For example, requests may be carried over link 721 while responses are carried over 723. This is undesirable for port channels as it affects applications like write acceleration that assume the traffic for a given flow is carried over the same physical link in both directions.”) Ghosh at [0056] (“According to various embodiments of the present invention, links are brought up in order. A first link is selected for bring up. No effort is made to bring up other links is

No.	’740 Patent Claim 1	Ghosh
		attempted until the exchange associated with the first link is completed. Consequently, port channel map tables at the local switch 701 and at the remote switch 703 are consistent. Port channel map table entries are synchronized at both ends so that all frames for a given flow are carried over the same physical link in both directions after the exchanges are completed.”) Ghosh at [0059] (“Line cards 803, 805, and 807 can communicate with an active supervisor 811 through interface circuitry 883, 885, and 887 and the backplane 815. According to various embodiments, each line card includes a plurality of ports that can act as either input ports or output ports for communication with external fibre channel network entities 851 and 853. The backplane 815 can provide a communi-cations channel for all traffic between line cards and super- visors. Individual line cards 803 and 807 can also be coupled to external fibre channel network entities 851 and 853 through fibre channel ports 843 and 847.”)
1[c]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel,	Ghosh discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. For example, Ghosh discloses connecting the line cards to the active supervisor via the backplane using parallel interface circuitry. Ghosh at [0059] (“Line cards 803, 805, and 807 can communicate with an active supervisor 811 through interface circuitry 883, 885, and 887 and the backplane 815. According to various embodiments, each line card includes a plurality of ports that can act as either input ports or output ports for communication with external fibre channel network entities 851 and 853. The backplane 815 can provide a communi-cations channel for all traffic between line cards and super- visors. Individual line cards 803 and 807 can also be coupled to external fibre channel network entities 851 and 853 through fibre channel ports 843 and 847.”) Ghosh at [0060] (“External fibre channel network entities 851 and 853 can be nodes such as other fibre channel switches, disks, RAIDS, tape libraries, or servers. It should be noted that the switch can support any number of line cards and supervisors. In the embodiment shown, only a

No.	’740 Patent Claim 1	Ghosh
		single supervisor is connected to the backplane 815 and the single supervisor communicates with many different line cards. The active supervisor 811 may be configured or designed to run a plurality of applications such as routing, domain manager, system manager, and utility applications.”) Ghosh at [0061] (“According to one embodiment, the routing appli-cation is configured to provide credits to a sender upon recognizing that a frame has been forwarded to a next hop. A utility application can be configured to track the number of buffers and the number of credits used. A domain manager application can be used to assign domains in the fibre channel storage area network. Various supervisor applica-tions may also be configured to provide functionality such as flow control, credit management, and quality of service (QoS) functionality for various fibre channel protocol layers.”) Ghosh at Figure 8 (annotation added)

No.	’740 Patent Claim 1	Ghosh
		Figure 8  External Fibre Channel Network Entity 851 External Fibre Channel Network Entity 853 FC Port 843 FC Port 845 FC Port 847 Line Card 803 Line Card 805 Line Card 807 Interface Circuitry 883 Interface Circuitry 885 Interface Circuitry 887 Active Supervisor 811 Backplane Interface Logic 831 Backplane 815 801 Power Supply 817
1[d]	at least one of said second physical links being a bi-directional link operative to communicate in both an upstream	Ghosh discloses at least one of said second physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction. For example, Ghosh discloses interface circuitry connecting line cards and the backplane that exchange data frames for a given flow over the same physical link in both directions.

No.	’740 Patent Claim 1	Ghosh
	direction and a downstream direction;	Ghosh at [0026] (“Consequently, the techniques and mechanisms of the present invention allow automatic detection of compat-ible ports to enable automatic creation of port channels. Port channels can be effectively brought up at either a local switch or a remote switch after either automatic creation of port channels or manual configuration of port channels. Robust error detection capabilities allow the correction of improper configurations and connections. Member physical ports of a port channel can operate as individual links if they cannot be configured to be part of port channel. Further-more, synchronization is supported so that requests and responses belonging to the same flow can be carried over the same physical link in a port channel in both directions. In many conventional implementations, such as Ethernet for example, a flow belonging to a particular port channel could be carried over different physical links during send and receive phases.”) Ghosh at [0032] (“Each entity can also have additional parameters to aid in the set up of port channels. According to various embodiments, parameters such as a channeling model, a channeling intent and a channeling status are included. A channeling model indicates to a peer port the channel group is automatically created or user configured. A channeling intent parameter indicates the peer port if this port intends to participate in a port channel. Otherwise, the port intends to operate as an individual port. The channeling status param-eter tells the peer port about its current channeling status. This parameter is exchanged by the attached peer ports to agree upon the channeling status of the link and to ensure that both ends are synchronized.”) Ghosh at [0055] (“Based on this scheme, frames for a given flow are transmitted through the same physical port of the port channel. However, unless there is proper synchronization of the port channel map tables at the two ends of a port channel, it is possible that requests and responses for the same flow are carried over two different physical links. For example, requests may be carried over link 721 while responses are carried over 723. This is undesirable for port channels as it affects applications like write acceleration that assume the traffic for a given flow is carried over the same physical link in both directions.”) Ghosh at [0056] (“According to various embodiments of the present invention, links are brought up in order. A first link is selected for bring up. No effort is made to bring up other links is

No.	’740 Patent Claim 1	Ghosh
		attempted until the exchange associated with the first link is completed. Consequently, port channel map tables at the local switch 701 and at the remote switch 703 are consistent. Port channel map table entries are synchronized at both ends so that all frames for a given flow are carried over the same physical link in both directions after the exchanges are completed.”) Ghosh at [0059] (“Line cards 803, 805, and 807 can communicate with an active supervisor 811 through interface circuitry 883, 885, and 887 and the backplane 815. According to various embodiments, each line card includes a plurality of ports that can act as either input ports or output ports for communication with external fibre channel network entities 851 and 853. The backplane 815 can provide a communi-cations channel for all traffic between line cards and super- visors. Individual line cards 803 and 807 can also be coupled to external fibre channel network entities 851 and 853 through fibre channel ports 843 and 847.”)
1[e]	receiving a data frame having frame attributes sent between the communication network and the network node:	Ghosh discloses receiving a data frame having frame attributes sent between the communication network and the network node. For example, Ghosh discloses data frames with identifying information in a flow that are transmitted through particular links between network entities and the network. Ghosh at [0004] (“Neighboring nodes in a fibre channel network are typically interconnected through multiple physical links. For example, a local fibre channel switch may be connected to a remote fibre channel switch through four physical links. In many instances, it may be beneficial to aggregate some of the physical links into logical links. That is, multiple physi-cal links can be combined to form a logical interface to provide higher aggregate bandwidth, load balancing, and link redundancy. When a frame is being transmitted over a logical link, it does not matter what particular physical link is being used as long as all the frames of a given flow are transmitted through the same link. If a constituent physical link goes down, the logical link can still remain operational.”) Ghosh at [0054] (“A database such as a port channel map table is used store the links included in a port channel. The links and/or port information is entered into the table based on the order in

No.	’740 Patent Claim 1	Ghosh
		which they were brought up. The port channel map table is used to select the physical link through which a frame is transmitted. In conventional implementations, the order of the peer ports listed in the port channel map table can be different for both the local switch 701 and the remote switch 703.”) Ghosh at [0055] (“Based on this scheme, frames for a given flow are transmitted through the same physical port of the port channel. However, unless there is proper synchronization of the port channel map tables at the two ends of a port channel, it is possible that requests and responses for the same flow are carried over two different physical links. For example, requests may be carried over link 721 while responses are carried over 723. This is undesirable for port channels as it affects applications like write acceleration that assume the traffic for a given flow is carried over the same physical link in both directions.”) Ghosh at [0056] (“According to various embodiments of the present invention, links are brought up in order. A first link is selected for bring up. No effort is made to bring up other links is attempted until the exchange associated with the first link is completed. Consequently, port channel map tables at the local switch 701 and at the remote switch 703 are consistent. Port channel map table entries are synchronized at both ends so that all frames for a given flow are carried over the same physical link in both directions after the exchanges are completed.”) Ghosh at [0061] (“According to one embodiment, the routing appli-cation is configured to provide credits to a sender upon recognizing that a frame has been forwarded to a next hop. A utility application can be configured to track the number of buffers and the number of credits used. A domain manager application can be used to assign domains in the fibre channel storage area network. Various supervisor applica-tions may also be configured to provide functionality such as flow control, credit management, and quality of service (QoS) functionality for various fibre channel protocol layers.”)
1[f]:	selecting, in a single computation based on at least	Ghosh discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group.

No.	’740 Patent Claim 1	Ghosh
	one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	For example, Ghosh discloses using a port channel map table to select the physical links through which a frame is transmitted. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Ghosh is found to not meet this limitation, selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group would have been obvious to a person having ordinary skill in the art, as explained below. Ghosh at [0004] (“Neighboring nodes in a fibre channel network are typically interconnected through multiple physical links. For example, a local fibre channel switch may be connected to a remote fibre channel switch through four physical links. In many instances, it may be beneficial to aggregate some of the physical links into logical links. That is, multiple physi-cal links can be combined to form a logical interface to provide higher aggregate bandwidth, load balancing, and link redundancy. When a frame is being transmitted over a logical link, it does not matter what particular physical link is being used as long as all the frames of a given flow are transmitted through the same link. If a constituent physical link goes down, the logical link can still remain operational.”) Ghosh at [0054] (“A database such as a port channel map table is used store the links included in a port channel. The links and/or port information is entered into the table based on the order in which they were brought up. The port channel map table is used to select the physical link through which a frame is transmitted. In conventional implementations, the order of the peer ports listed in the port channel map table can be different for both the local switch 701 and the remote switch 703.”) Ghosh at [0055] (“Based on this scheme, frames for a given flow are transmitted through the same physical port of the port channel. However, unless there is proper synchronization of the port channel map tables at the two ends of a port channel, it is possible that requests and responses for the same flow are carried over two different physical links. For example, requests may be carried over link 721 while responses are carried over 723. This is undesirable for port channels as it affects applications like write acceleration that assume the traffic for a given flow is carried over the same physical link in both directions.”)

No.	’740 Patent Claim 1	Ghosh
		Ghosh at [0056] (“According to various embodiments of the present invention, links are brought up in order. A first link is selected for bring up. No effort is made to bring up other links is attempted until the exchange associated with the first link is completed. Consequently, port channel map tables at the local switch 701 and at the remote switch 703 are consistent. Port channel map table entries are synchronized at both ends so that all frames for a given flow are carried over the same physical link in both directions after the exchanges are completed.”) Ghosh at [0061] (“According to one embodiment, the routing appli-cation is configured to provide credits to a sender upon recognizing that a frame has been forwarded to a next hop. A utility application can be configured to track the number of buffers and the number of credits used. A domain manager application can be used to assign domains in the fibre channel storage area network. Various supervisor applica-tions may also be configured to provide functionality such as flow control, credit management, and quality of service (QoS) functionality for various fibre channel protocol layers.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Ghosh in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 1[f] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Basso discloses using a hash function and index table to select for blade/port combinations over which to send the packet over a user port and a switch fabric link. Basso further discloses that this selection is performed based upon packet information. Basso at [0010] (“Upon a network processor receiving a packet of data, the network processor may index into a table, commonly referred to as a forwarding table, to determine the table associated with a particular logical interface as well as the next destination address. The forwarding table may comprise a plurality of entries where each entry may comprise information indicating a particular table associated with a particular logical interface as well as the next destination address. Each

No.	’740 Patent Claim 1	Ghosh
		logical interface may be associated with a table storing a plurality of entries containing blade/ port combinations as discussed further below. In one embodiment, an entry may be indexed in the forwarding table using a destination address in the received packet header.”) Basso at [0011] (“A hash function may then be performed on the received packet to generate a hash value. In one embodiment, a hash function may be performed on the source and destination address in the packet header to generate a hash value.”) Basso at [0012] (“The hash value generated may be used to index into the table associated with a particular logical interface. Upon indexing into the table associated with the logical interface, an appropriate blade/port combination may be identified to transmit the received packet of data. In one embodiment, a blade/port combination may be selected in the indexed entry of the table associated with the logical interface by using a portion of the bits of the hashed value. The received packet may then be transmitted through the identified blade/port combination to the next destination (next destination previously identified by the next destination address in the forwarding table).”) Basso at [0035] (“By logically grouping a plurality of ports 404 coupled to a particular network device into a logical interface 405, network processor 403 may be configured to transmit processed packets to that particular network device via any blade 402/port 404 combination grouped in that logical interface 405. For example, referring to FIG. 4, ports 404A-404I are physically connected to router 104B. If ports 404A-404I were logically grouped into logical interface 405, then a particular network processor 403, e.g., network processor 403A, may be configured to transmit processed packets that are determined to be transmitted to router 104B through any of ports 404A-404I in blades 402A-C, respectively. Network processor 403, e.g., network processor 403A, may be configured to transmit the processed packets to router 104B through ports 404, e.g., ports 404D-I, not in its blade 402, e.g., blade 402A, by forwarding the processed packets to switch fabric 401 which may then direct the processed packets to another appropriate physical blade 402/port 404 combination. Network processor 403, e.g., network processor 403A, may further be configured to transmit the processed packets to router 104B through any ports 404, e.g., ports 404A-C, in its blade 402, e.g., blade402A, instead of just one physical port 404 in its blade 402, e.g., blade 402A.

No.	’740 Patent Claim 1	Ghosh
		A more detailed description of routing packets implementing logical interface(s) 405 is provided below in FIG. 5.”) Basso at [0040] (“In step 502, network processor 403, e.g., network processor 403A, may receive a packet of data from switch fabric 401. Upon receiving the packet of data, network processor 403, in step 503, may index into a table, commonly referred to as a forwarding table, to determine the table associated with a particular logical interface 405 as well as the next destination address, i.e., the next hop address. The forwarding table may comprise a plurality of entries where each entry may comprise information indicating a particular table associated with a particular logical interface 405 as well as the next destination address. Each logical interface 405 may be associated with a table storing a plurality of entries containing blade 402/port 404 combinations as discussed further below. In one embodiment, an entry may be indexed in the forwarding table using a destination address in the received packet header. It is noted that an entry may be indexed in the forwarding table usingother means and that such means would be recognized by an artisan of ordinary skill in the art. It is further noted that embodiments implementing such means would fall within the scope of the present invention.”) Basso at [0041] (“In step 504, a hash function may be performed on the received packet to generate a hash value. In one embodiment, a hash function may be performed on the source and destination address in the packet header to generate a hash value. It is noted that in other embodiments a hash function may be performed on different fields, e.g., port, type of service, in the received packet to generate a hash value.”) Basso at [0042] (“In step 505, the hash value generated in step 504 may be used to index into the table associated with a particular logical interface 405 determined in step 503. Upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be identified in step 506 to transmit the received packet of data as explained below.”) Basso at [0043] (“As stated above, the table associated with a particular logical interface 405 may comprise a plurality of entries where each entry may comprise a threshold value associated with a

<table>
<tr><th>No.</th><th>’740 Patent Claim 1</th><th>Ghosh</th></tr>
<tr><td></td><td></td><td>

particular blade 402/port 404 combination. The threshold value may represent a percentage of the total number of packets received by router 104A that may be transmitted through the blade 402/port 404 combination associated with that threshold value. In one embodiment, the threshold value may be updated periodically by a user, e.g., system administrator, in control of router 104, e.g., router 104A. For example, the threshold value, e.g., twenty percent of the number of packets received by router 104A, associated with a particular blade 402/port 404 combination may be updated by lowering the threshold value by one percent during each update. An example of an entry of the table associated with a particular logical interface 405 is shown in Table 1 below:

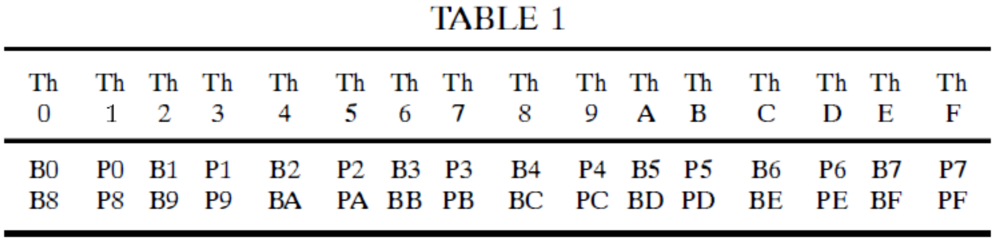


TABLE 1

<table>
<tr><th>Th 0</th><th>Th 1</th><th>Th 2</th><th>Th 3</th><th>Th 4</th><th>Th 5</th><th>Th 6</th><th>Th 7</th><th>Th 8</th><th>Th 9</th><th>Th A</th><th>Th B</th><th>Th C</th><th>Th D</th><th>Th E</th><th>Th F</th></tr>
<tr><td>B0</td><td>P0</td><td>B1</td><td>P1</td><td>B2</td><td>P2</td><td>B3</td><td>P3</td><td>B4</td><td>P4</td><td>B5</td><td>P5</td><td>B6</td><td>P6</td><td>B7</td><td>P7</td></tr>
<tr><td>B8</td><td>P8</td><td>B9</td><td>P9</td><td>BA</td><td>PA</td><td>BB</td><td>PB</td><td>BC</td><td>PC</td><td>BD</td><td>PD</td><td>BE</td><td>PE</td><td>BF</td><td>PF</td></tr>
</table>

Basso at [0044] (“Table 1 above illustrates an exemplary entry in the table associated with a particular logical interface 405. Each entry may comprise a plurality of threshold values (16 threshold values in exemplary Table 1) where each threshold value is associated with a particular blade 402/port 404 combination. For example, threshold value (Th0) is associated with blade B0/port P0 combination where blade BO may refer to a particular blade 402, e.g., blade 402B, and port PO may refer to a particular port 404, e.g., port 404E. Threshold value (Thl) is associated with blade Bl/port Pl combination and so forth. As stated above, each threshold value may represent a percentage of the total number of packets received by router 104A that may be transmitted through the blade 402/port 404 combination associated with that threshold value. For example, threshold value (Th0) may represent a percentage of the total number of packets received by router 104A that may be transmitted through port PO in blade BO. If port PO refers to port 404D and blade BO refers to blade 402B, then if Th0 has a value of twenty percent, a maximum of twenty percent of the total packets received by router 104A may be transmitted through port 404D in blade 402B.”)

</td></tr>
</table>

No.	’740 Patent Claim 1	Ghosh
		Basso at [0045] (“As stated above, upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be identified in step 506 to transmit the received packet of data. In one embodiment, the hash value generated in step 504 may be used to select a particular threshold value and hence a blade 402/port 404 combination associated with the selected threshold value. In one embodiment, a portion of the bits of the hash value, e.g., most significant bits, may be used to select a particular threshold value in the entry indexed in step 505. For example, referring to Table 1, since there are 16 different threshold values in each entry of the table associated with logical interface 405, only four bits of the hash value generated in step 504 may be used to select a threshold value. Upon selecting a threshold value, the blade 402/port 404 combination associated with the selected threshold value may be used to transmit the received packet.”) As another example, Wiher discloses using cell header information to at each node to select and route the ATM data cell over a data link and selected backplane. Wiher at 3:43-65 (“In general, in another aspect, the invention features an apparatus for communicating data cells between a data link and a backplane. The apparatus includes transceiver circuitry to transmit and receive data cells over a data link and a plurality of backplane interfaces each including at least one cell signal terminal. Each of the backplane interface is coupled to a backplane interconnection circuit. Each backplane interconnection circuit transmits and receives cells over the cell signal terminals of its associated backplane interface. The apparatus also includes de-multiplexing circuitry coupling the transceiver circuitry to each of the backplane interconnection circuits. The de-multiplexing circuitry receives a data cell from the transceiver circuitry, select a backplane interconnection circuit associated with the data cell, and provide the data cell to the selected backplane interconnection circuit for transmission over the cell signal terminals of the associated backplane interface. The apparatus also includes multiplexing circuitry coupling the plurality of backplane interconnection circuits to the transceiver circuitry. The multiplexing circuitry receives data cells from each of the

No.	’740 Patent Claim 1	Ghosh
		backplane interconnection circuits and provide the received data cells to the transceiver circuitry.”) Wiher at 3:66-4:22 (“Implementations of the invention may include one or more of the following features. The backplane interconnection circuits may independently receive and transmit data cells over the plurality of backplane interfaces. The de-multiplexing circuitry may select a backplane interface based on data in the header field of the data cell. The apparatus may include header translation circuitry to alter header data in cells sent between the plurality of backplane interfaces and the transceiver circuitry. Each of the plurality of backplane interfaces may include separate terminals to receive cells and separate terminals to transmit cells. The terminals to transmit cells may include a first and second control terminal and at least one outgoing cell data terminal. A backplane interface's backplane interconnection circuitry may accepts a signal on the first control terminal as indicating that a cell may be sent over the interface, asserts a 15 signal on the second control terminal to indicate that a cell is being transmitted, and transmits data bits of the cell on the outgoing cell data terminal. Each backplane interface may include a single outgoing cell data terminal and each bit of the cell may be serially transmitted over the single outgoing cell data terminal. Each backplane interface may include multiple outgoing cell data terminals and bits of the cell may be sent in parallel over the eight outgoing cell data terminals.”)
1[g]	sending the data frame over the selected first and second physical links,	Ghosh discloses sending the data frame over the selected first and second physical links. For example, Ghosh discloses transmitting frames over ports and interface circuitry. Ghosh at [0004] (“Neighboring nodes in a fibre channel network are typically interconnected through multiple physical links. For example, a local fibre channel switch may be connected to a remote fibre channel switch through four physical links. In many instances, it may be beneficial to aggregate some of the physical links into logical links. That is, multiple physi-cal links can be combined to form a logical interface to provide higher aggregate bandwidth, load balancing, and link redundancy. When a frame is being transmitted over a logical link, it does not matter what particular physical link is being used as long as all the frames of a given flow are transmitted

No.	’740 Patent Claim 1	Ghosh
		through the same link. If a constituent physical link goes down, the logical link can still remain operational.”) Ghosh at [0054] (“A database such as a port channel map table is used store the links included in a port channel. The links and/or port information is entered into the table based on the order in which they were brought up. The port channel map table is used to select the physical link through which a frame is transmitted. In conventional implementations, the order of the peer ports listed in the port channel map table can be different for both the local switch 701 and the remote switch 703.”) Ghosh at [0055] (“Based on this scheme, frames for a given flow are transmitted through the same physical port of the port channel. However, unless there is proper synchronization of the port channel map tables at the two ends of a port channel, it is possible that requests and responses for the same flow are carried over two different physical links. For example, requests may be carried over link 721 while responses are carried over 723. This is undesirable for port channels as it affects applications like write acceleration that assume the traffic for a given flow is carried over the same physical link in both directions.”) Ghosh at [0056] (“According to various embodiments of the present invention, links are brought up in order. A first link is selected for bring up. No effort is made to bring up other links is attempted until the exchange associated with the first link is completed. Consequently, port channel map tables at the local switch 701 and at the remote switch 703 are consistent. Port channel map table entries are synchronized at both ends so that all frames for a given flow are carried over the same physical link in both directions after the exchanges are completed.”) Ghosh at [0061] (“According to one embodiment, the routing appli-cation is configured to provide credits to a sender upon recognizing that a frame has been forwarded to a next hop. A utility application can be configured to track the number of buffers and the number of credits used. A domain manager application can be used to assign domains in the fibre channel storage area network. Various supervisor applica-tions may also be configured to provide functionality

No.	’740 Patent Claim 1	Ghosh
		such as flow control, credit management, and quality of service (QoS) functionality for various fibre channel protocol layers.”)
1[h]	said sending comprising communicating along at least one of said bi-directional links.	Ghosh discloses said sending comprising communicating along at least one of said bi-directional links. *See supra at* 1[b], 1[d], 1[g].

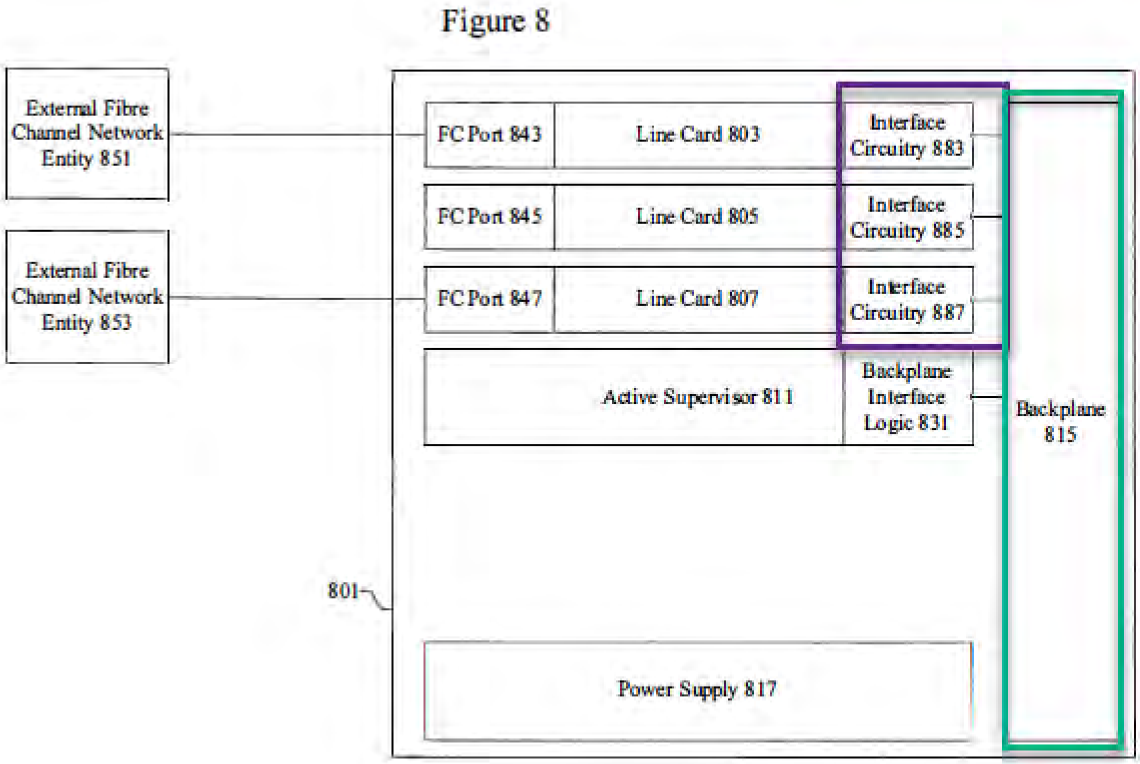
No.	’740 Patent Claim 2	Ghosh
2[a]	The method according to claim 1, wherein the network node comprises a user node, and	Ghosh discloses the method according to claim 1, wherein the network node comprises a user node. For example, Ghosh discloses network entities may include client system operated by a user. Ghosh at Abstract (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggregation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent gen-eration of port channel map tables.”) Ghosh at [0027] (“FIG. 1 shows one example of a storage area network implemented using fibre channel that can use efficient port channel configuration mechanisms. A switch 101 is coupled to switches 103 and 105 as well as to a host 111 and storage 121. Switch 101 may be connected to other entities through multiple physical links or channels config-ured as logical links or port channels. In one embodiment, host 111 may be a server or client system while storage 121 may be single disk or a redundant array of independent disks (RAID). Switches 103 and 105 are both coupled to switch 107. Switch 107 is connected to host 113

No.	’740 Patent Claim 2	Ghosh
		and switch 103 is connected to storage 123. Switch 109 is connected to host 115, switch 107, disk array 153, and an external network 151 that may or may not use fibre channel.”) Ghosh at [0032] (“Each entity can also have additional parameters to aid in the set up of port channels. According to various embodiments, parameters such as a channeling model, a channeling intent and a channeling status are included. A channeling model indicates to a peer port the channel group is automatically created or user configured. A channeling intent parameter indicates the peer port if this port intends to participate in a port channel. Otherwise, the port intends to operate as an individual port. The channeling status param-eter tells the peer port about its current channeling status. This parameter is exchanged by the attached peer ports to agree upon the channeling status of the link and to ensure that both ends are synchronized.”) Ghosh at [0059] (“Line cards 803, 805, and 807 can communicate with an active supervisor 811 through interface circuitry 883, 885, and 887 and the backplane 815. According to various embodiments, each line card includes a plurality of ports that can act as either input ports or output ports for communication with external fibre channel network entities 851 and 853. The backplane 815 can provide a communi-cations channel for all traffic between line cards and super- visors. Individual line cards 803 and 807 can also be coupled to external fibre channel network entities 851 and 853 through fibre channel ports 843 and 847.”) Ghosh at [0060] (“External fibre channel network entities 851 and 853 can be nodes such as other fibre channel switches, disks, RAIDS, tape libraries, or servers. It should be noted that the switch can support any number of line cards and supervisors. In the embodiment shown, only a single supervisor is connected to the backplane 815 and the single supervisor communicates with many different line cards. The active supervisor 811 may be configured or designed to run a plurality of applications such as routing, domain manager, system manager, and utility applications.”) Ghosh at [0061] (“According to one embodiment, the routing appli-cation is configured to provide credits to a sender upon recognizing that a frame has been forwarded to a next hop. A utility application can be configured to track the number of buffers and the number of

No.	’740 Patent Claim 2	Ghosh
		credits used. A domain manager application can be used to assign domains in the fibre channel storage area network. Various supervisor applica-tions may also be configured to provide functionality such as flow control, credit management, and quality of service (QoS) functionality for various fibre channel protocol layers.”)
2[b]	wherein sending the data frame comprises establishing a communication service between the user node and the communication network.	Ghosh discloses wherein sending the data frame comprises establishing a communication service between the user node and the communication network. For example, Ghosh discloses establishing a fibre channel network among network entities, including client systems, for transmitting data frames. Ghosh at Abstract (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggregation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent gen-eration of port channel map tables.”) Ghosh at [0027] (“FIG. 1 shows one example of a storage area network implemented using fibre channel that can use efficient port channel configuration mechanisms. A switch 101 is coupled to switches 103 and 105 as well as to a host 111 and storage 121. Switch 101 may be connected to other entities through multiple physical links or channels config-ured as logical links or port channels. In one embodiment, host 111 may be a server or client system while storage 121 may be single disk or a redundant array of independent disks (RAID). Switches 103 and 105 are both coupled to switch 107. Switch 107 is connected to host 113 and switch 103 is connected to storage 123. Switch 109 is connected to host 115, switch 107, disk array 153, and an external network 151 that may or may not use fibre channel.”) Ghosh at [0032] (“Each entity can also have additional parameters to aid in the set up of port channels. According to various embodiments, parameters such as a channeling model, a channeling intent and a channeling status are included. A channeling model indicates to a peer port the channel group is automatically created or user configured. A channeling intent

No.	’740 Patent Claim 2	Ghosh
		parameter indicates the peer port if this port intends to participate in a port channel. Otherwise, the port intends to operate as an individual port. The channeling status param-eter tells the peer port about its current channeling status. This parameter is exchanged by the attached peer ports to agree upon the channeling status of the link and to ensure that both ends are synchronized.”) Ghosh at [0059] (“Line cards 803, 805, and 807 can communicate with an active supervisor 811 through interface circuitry 883, 885, and 887 and the backplane 815. According to various embodiments, each line card includes a plurality of ports that can act as either input ports or output ports for communication with external fibre channel network entities 851 and 853. The backplane 815 can provide a communi-cations channel for all traffic between line cards and super- visors. Individual line cards 803 and 807 can also be coupled to external fibre channel network entities 851 and 853 through fibre channel ports 843 and 847.”) Ghosh at [0060] (“External fibre channel network entities 851 and 853 can be nodes such as other fibre channel switches, disks, RAIDS, tape libraries, or servers. It should be noted that the switch can support any number of line cards and supervisors. In the embodiment shown, only a single supervisor is connected to the backplane 815 and the single supervisor communicates with many different line cards. The active supervisor 811 may be configured or designed to run a plurality of applications such as routing, domain manager, system manager, and utility applications.”) Ghosh at [0061] (“According to one embodiment, the routing appli-cation is configured to provide credits to a sender upon recognizing that a frame has been forwarded to a next hop. A utility application can be configured to track the number of buffers and the number of credits used. A domain manager application can be used to assign domains in the fibre channel storage area network. Various supervisor applica-tions may also be configured to provide functionality such as flow control, credit management, and quality of service (QoS) functionality for various fibre channel protocol layers.”)

No.	’740 Patent Claim 3	Ghosh
3	The method according to claim 1, wherein the second physical links comprise backplane traces formed on a backplane to which the one or more interface modules are coupled.	Ghosh discloses the method according to claim 1, wherein the second physical links comprise backplane traces formed on a backplane to which the one or more interface modules are coupled. For example, Ghosh discloses connecting the line cards to the backplane using parallel interface circuitry. Ghosh at [0059] (“Line cards 803, 805, and 807 can communicate with an active supervisor 811 through interface circuitry 883, 885, and 887 and the backplane 815. According to various embodiments, each line card includes a plurality of ports that can act as either input ports or output ports for communication with external fibre channel network entities 851 and 853. The backplane 815 can provide a communi-cations channel for all traffic between line cards and super- visors. Individual line cards 803 and 807 can also be coupled to external fibre channel network entities 851 and 853 through fibre channel ports 843 and 847.”) Ghosh at [0060] (“External fibre channel network entities 851 and 853 can be nodes such as other fibre channel switches, disks, RAIDS, tape libraries, or servers. It should be noted that the switch can support any number of line cards and supervisors. In the embodiment shown, only a single supervisor is connected to the backplane 815 and the single supervisor communicates with many different line cards. The active supervisor 811 may be configured or designed to run a plurality of applications such as routing, domain manager, system manager, and utility applications.”) Ghosh at [0061] (“According to one embodiment, the routing appli-cation is configured to provide credits to a sender upon recognizing that a frame has been forwarded to a next hop. A utility application can be configured to track the number of buffers and the number of credits used. A domain manager application can be used to assign domains in the fibre channel storage area network. Various supervisor applica-tions may also be configured to provide functionality such as flow control, credit management, and quality of service (QoS) functionality for various fibre channel protocol layers.”)

No.	’740 Patent Claim 3	Ghosh
		Ghosh at Figure 8 (annotation added)  Figure 8 External Fibre Channel Network Entity 851 External Fibre Channel Network Entity 853 FC Port 843 Line Card 803 Interface Circuitry 883 FC Port 845 Line Card 805 Interface Circuitry 885 FC Port 847 Line Card 807 Interface Circuitry 887 Active Supervisor 811 Backplane Interface Logic 831 Backplane 815 801 Power Supply 817

No.	’740 Patent Claim 4	Ghosh
4[preamble]	A method for communication, comprising:	Ghosh discloses a method for communication. *See supra* at 1[preamble].
4[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	Ghosh discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].
4[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Ghosh discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra* at 1[c].
4[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	Ghosh discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra* at 1[e].
4[d]	selecting, in a single computation	Ghosh discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group.

No.	’740 Patent Claim 4	Ghosh
	based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	*See supra* at 1[f].
4[e]	sending the data frame over the selected first and second physical links,	Ghosh discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].
4[f]	at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group.	Ghosh discloses at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group. For example, Ghosh discloses aggregating physical links, including ports, into aggregate port channels that form a single logical link to increase bandwidth. Ghosh at Abstract (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggregation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent gen-eration of port channel map tables.”) Ghosh at [0004] (“Neighboring nodes in a fibre channel network are typically interconnected through multiple physical links. For example, a local fibre channel switch may be connected to a remote fibre channel switch through four physical links. In many instances, it may be beneficial to aggregate some of the physical links into logical links. That is, multiple physi-cal links can be combined to form a logical interface to provide higher aggregate bandwidth, load balancing, and

No.	’740 Patent Claim 4	Ghosh
		link redundancy. When a frame is being transmitted over a logical link, it does not matter what particular physical link is being used as long as all the frames of a given flow are transmitted through the same link. If a constituent physical link goes down, the logical link can still remain operational.”) Ghosh at [0007] (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggre-gation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent generation of port channel map tables.”) Ghosh at [0008] (“In one embodiment, a method for aggregating ports in a fibre channel fabric is provided. It is determined that a plurality of local ports at a local fibre channel switch are compatible. Identifiers for the plurality of local ports are sent to a remote fibre channel switch. The remote fibre channel switch determines if a plurality of remote ports are compat-ible, the plurality of remote ports corresponding to the plurality of local ports. An indication that one or more of the remote physical ports are compatible is received. A port channel including one or more of the local ports correspond-ing to the compatible remote ports is created.”) Ghosh at [0010] (“In another embodiment, a fibre channel network is described. The fibre channel network includes a local fibre channel switch and a remote fibre channel switch. The local fibre channel switch aggregates a compatible subset of the plurality of local ports and sends identifiers for the compat-ible subset of the plurality of local ports to the remote fibre channel switch. The remote fibre channel switch determines if a subset of the plurality of remote ports are compatible. The subset of the plurality of remote ports corresponds to the compatible subset of the plurality of local ports.”) Ghosh at [0022] (“Switches in a fibre channel network are typically interconnected using multiple physical links. The physical links connecting a pair of switches allows transmission of data and control signals. In some instances, it is useful to aggregate multiple physical links into a logical link. Physi-cal links are also referred to herein as physical interfaces and channels while

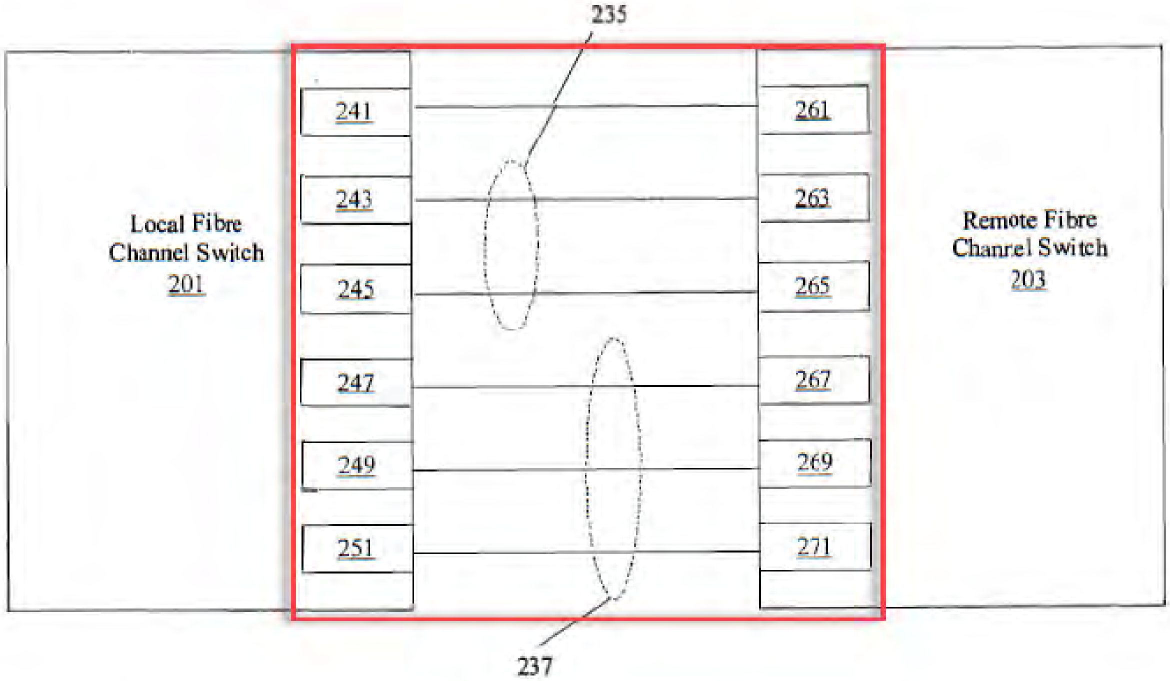
No.	’740 Patent Claim 4	Ghosh
		logical links are also referred to herein as logical interfaces and port channels. For example, a local switch may be connected to a remote switch through four physical links. Instead of having to transmit data through a particular physical link, the physical links can be aggregated to form one or more logical links. In one example, all four physical links are aggregated into a single logical link. Instead of having data transmitted through a particular physical link, the data can merely be transmitted over a particular logical link without regard to the particular physi-cal interface used. Aggregating physical links into a logical link allows for higher aggregated bandwidth, load balancing, and link redundancy. For example, if a particular physical link fails or is overloaded, data can still be transmitted over the logical link.”) Ghosh at [0029] (“FIG. 2 is a diagrammatic representation showing links between two switches, such as two fibre channel switches shown in FIG. 1. A local fibre channel switch 201 includes local ports 241, 243, 245, 247, 249, and 251. A remote fibre channel switch 203 includes remote ports 261, 263, 265, 267, 269, and 271. Local port 241 is coupled to remote port 261 through an individual physical link or channel. Connected ports are also referred to herein as peer ports. Local port 243 is coupled to remote port 263 and local port 245 is coupled to remote port 265. The two resulting physical links are aggregated to form port channel 235. Local ports 247, 249, and 251 are coupled to remote ports 267, 269, and 271 respectively. The three resulting physical links are aggregated to form port channel 237.”) Ghosh at [0030] (“According to various embodiments, local fibre channel switch 201 and remote fibre channel switch both have associated identifiers. In some examples, the identifiers are globally unique identifiers such as a global switch world wide names (WWNs). Each local port 241, 243, 245, 247, 249, and 251 and each remote port 261,263,265,267, 269, and 271 can also be associated with identifiers. In some examples, the identifiers are port WWNs. The port WWNs are typically used for debugging or identifying the peer port in alert or warning messages. However, according to various embodiments, the techniques of the present invention use WWNs as globally unique identifiers to aggregate ports instead of using compatibility keys which are only locally unique. Compatibility keys are mechanisms typically used by other protocols such as Ethernet for aggregation.”)

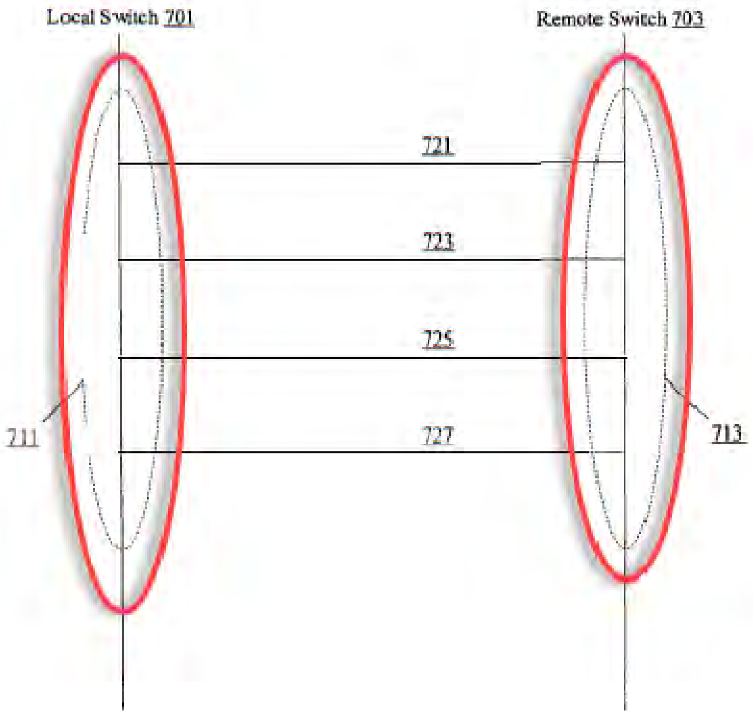
No.	’740 Patent Claim 4	Ghosh
		Ghosh at [0033] (“A variety of parameters can be used to aggregate physical ports. FIG. 3 is a flow process diagram showing one technique for aggregating physical ports into a logical port. And 301, it is determined if auto create functionality is enabled. According to various embodiments, auto create functionality allows automatic configuration and detection of compatible physical ports as well as aggregation into one or more logical ports. Auto creation does not require user intervention. In other examples, administrators can manu-ally arrange ports for aggregation.”) Ghosh at [0037] (“FIG. 4 is an exchange diagram showing one example of a bring up procedure used for a port creating a new port channel. A local switch 401 is coupled to a remote switch 403. The local switch 401 includes a physical port Al coupled to physical port Bl included in remote switch 403. When two peer ports Al and Bl are being aggregated into a port channel, the peer switches 401 and 403 typically already know the world wide names of the individual physical peer ports. However, the peer switches only know the world wide name of their own logical port or port channel. That is, both switches have the individual physical link configured, but the link is not yet part of a port channel. At 421, a local switch 401 sends a synchronize (sync) message 411 to the remote switch 403 to begin the process of creating a port channel including ports Al and Bl.”) Ghosh at [0038] (“In some examples, the sync message 411 includes a local port channel identifier and a remote port channel identifier. In one particular example, the local port channel identifier is set to the world wide name of the local port channel assigned by the local switch 401. The remote port channel identifier is left blank to indicate that the port Al is being aggregated as part of a new port channel. The sync message 411 can also include other parameters such as channel status, channel model, or channel intent.”) Ghosh at [0042] (“When two peer ports A2 and B2 are aggregated into a port channel Cl, the peer switches 501 and 503 typically already know the world wide names of the indi-vidual physical peer ports A2 and B2 as well as the world wide name information of the port channel Cl. Conse-quently, the port channel is already successfully established. According to various embodiments, local switch 501 and remote switch 503 perform parameter checking to ensure that

No.	’740 Patent Claim 4	Ghosh
		the new physical port A2 and B2 can be safely added to the existing port channel Cl. At 521, a local switch can check configuration parameters to ensure that physical ports Al and A2 at the local switch 501 are compatible. The compatibility checking can be performed anytime. In some examples, compatibility checking is checked before a local switch 501 sends a synchronize (sync) message 511 to the remote switch 503 to begin the process of aggregating ports A2 and B2 into the port channel.) Ghosh at [0043] (“In some examples, the sync message 511 includes local port channel identifier and a remote port channel identifier. In one particular example, the local port channel identifier is set to the world wide name of the local port channel assigned by the local switch 501. The remote port channel identifier is filled with the existing port channel identifier to indicate that the port A2 is being aggregated into existing port channel C2. The sync message 511 can also include other parameters such as channel status, channel model, or channel intent.”) Ghosh at [0044] (“At 531, remote switch 503 uses the information received from the local switch 501 to verify port B2 is compatible with other port in port channel C2. In one example, configuration parameters associated with B2 are checked against configuration parameters associated with Bl. The remote switch 503 can also check if the port B2 is already assigned to a different port channel. If the port B2 is compatible with port Bl, the remote switch 503 can proceed and send a sync accept message 513 in response to the sync message 511 to indicate that the port B2 can be aggregated into the port channel. The sync accept message indicates that a port channel can now be modified. At 523, local switch 501 uses the information to update its own port channel database. However, the port channel may not yet be fully operational until the hardware configuration is completed. The local switch 501 continues hardware configuration such as line card configuration to make the port A2 part of the port channel Cl. An acknowledgment 527 is sent and received by remote switch 503 at 529. In some examples, the local switch 501 sends a commit signal 515 when hardware configuration is complete.”) Ghosh at [0045] (“The remote switch 503 receives the commit signal at 533 and begins its own hardware configuration. On completion of its hardware configuration, remote switch 503 sends out a commit accept signal 517 to indicate to local switch 501 that hardware configuration is

No.	’740 Patent Claim 4	Ghosh
		completed. According to various embodiments, local switch 501 receives the commit accept signal 517 and notifies relevant applications that the port channel is now fully operational at 525 and that port A2 has been aggregated into port channel Cl. The local switch 501 can also send out an acknowledge message 519. When the remote switch 503 receives the acknowledge, it notifies relevant applications that the port channel is operational at 535 and that port B2 has been aggregated into port channel Cl. In one embodiments, the techniques of the present invention contemplate using a two phase SYNC and COMMIT mechanism similar to the mechanism used in EPP.”) Ghosh at [0046] (“FIGS. 4 and 5 show examples of ports being aggregated into a port channel. At a particular switch, ports can be selected for aggregation into a port channel in a variety of manners. FIG. 6 is an exchange diagram showing automatic selection of ports at a switch for aggregation into a port channel. A local switch 601 is coupled to a remote switch 603. In one example, the local switch 601 includes physical ports Al, A2, A3, and A4 while remote switch 603 includes physical ports Bl, B2, B3, and B4. No port chan-nels have been formed.”) Ghosh at [0049] (“At 631, remote switch 603 uses the information received from the local switch 601 to verify that the peer ports of Al, A2, and A4 are compatible. That is, ports Bl, B2, and B4 are checked for compatibility. In one example, only ports Bl and B2 may be compatible, and consequently only ports Al, A2, Bl, and B2 can be included in the port channel. In another example, ports Bl, B2, and B4 are compatible, so ports Al, A2, A4, Bl, B2, and B4 can be aggregated into port channel Cl. According to various embodiments, if the port B2 is compatible with port Bl, the remote switch 603 can proceed and send a sync accept message 613 in response to the sync message 611 to indicate that the port B2 can be aggregated into the port channel. It should be noted that remote switch 603 can send a list indicating that ports B2 and B4 are compatible with Bl. However, the remote switch 603 sends only one compatible port B2 back for several reasons, and in the process of selection compatible port channels get priority over com-patible individual ports.”) Ghosh at [0050] (“One reason is that aggregation mechanisms and techniques can be implemented more elegantly by handling ports on an individual basis. Any individual port will

No.	’740 Patent Claim 4	Ghosh
		either start a new port channel, be added to an existing port channel, or operate stand alone. There is no need to keep track of groups of ports to be aggregated. Another reason is that fewer ports need to be locked if only a single port is being aggregated at any one time. The sync accept message indicates that a port channel can now be modified. At 623, local switch 601 receives the information and recognizes that Al and A2 can now be aggregated into port channel Cl. However, the port channel may not yet be fully operational until the hardware configuration is completed. An acknowl-edgment 627 is sent and received by remote switch 603 at 629. In some examples, the local switch 601 sends a commit signal 615 when hardware configuration is complete.”) Ghosh at [0051] (“The remote switch 603 receives the commit signal at 633 to create port channel Cl including ports Bl and B2. Hardware configuration can now be performed. On comple-tion of its hardware configuration, remote switch 603 sends out a commit accept signal 617 to indicate to local switch 601 that hardware configuration is completed. According to various embodiments, local switch 601 receives the commit accept signal 617 and notifies relevant applications that the port channel is now fully operational at 625 and that ports Al and A2 have been aggregated into port channel Cl. The local switch 601 can also send out an acknowledge message 619. When the remote switch 603 receives the acknowledge, it notifies relevant applications that the port channel is fully operational at 635 and that ports Bl and B2 have been aggregated into port channel Cl.”) Ghosh at [0053] (“FIG. 7 is a diagrammatic representation showing synchronous aggregation of ports into a port channel. A local switch 701 is coupled to a remote switch 703 through links 721, 723, 725, and 727. According to various embodiments, the links are being aggregated into port channel 711 at the local switch 701 and port channel 713 at the remote switch 703 in a synchronous manner. That is the peer ports corre-sponding to each link are brought up in the same order at both the local switch 701 and the remote switch 703.”) Ghosh at Figure 2 (annotation added)

No.	’740 Patent Claim 4	Ghosh
		 235 241 243 245 247 249 251 261 263 265 267 269 271 Local Fibre Channel Switch 201 Remote Fibre Chanrel Switch 203 237 Figure 2 Ghosh at Figure 7 (annotation added)

No.	’740 Patent Claim 4	Ghosh
		 Local Switch 701 Remote Switch 703 721 723 725 727 711 713 Figure 7

No.	’740 Patent Claim 5	Ghosh
5[preamble]	A method for communication, comprising:	Ghosh discloses a method for communication. *See supra at* 1[preamble].
5[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	Ghosh discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].
5[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Ghosh discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra at* 1[c].
5[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	Ghosh discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra at* 1[e].
5[d]	selecting, in a single computation based on at least	Ghosh discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group.

No.	’740 Patent Claim 5	Ghosh
	one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	*See supra at* 1[f].
5[e]	sending the data frame over the selected first and second physical links,	Ghosh discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].
5[f]	coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node.	Ghosh discloses coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node. For example, Ghosh discloses aggregating physical links, including ports, into aggregate port channels that form a single logical link to increase bandwidth, load balancing, and link redundancy. Ghosh at Abstract (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggregation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent gen-eration of port channel map tables.”) Ghosh at [0004] (“Neighboring nodes in a fibre channel network are typically interconnected through multiple physical links. For example, a local fibre channel switch may be connected to a remote fibre channel switch through four physical links. In many instances, it may be beneficial to aggregate some of the physical links into logical links. That is, multiple physi-cal links can be

No.	’740 Patent Claim 5	Ghosh
		combined to form a logical interface to provide higher aggregate bandwidth, load balancing, and link redundancy. When a frame is being transmitted over a logical link, it does not matter what particular physical link is being used as long as all the frames of a given flow are transmitted through the same link. If a constituent physical link goes down, the logical link can still remain operational.”) Ghosh at [0007] (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggre-gation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent generation of port channel map tables.”) Ghosh at [0008] (“In one embodiment, a method for aggregating ports in a fibre channel fabric is provided. It is determined that a plurality of local ports at a local fibre channel switch are compatible. Identifiers for the plurality of local ports are sent to a remote fibre channel switch. The remote fibre channel switch determines if a plurality of remote ports are compat-ible, the plurality of remote ports corresponding to the plurality of local ports. An indication that one or more of the remote physical ports are compatible is received. A port channel including one or more of the local ports correspond-ing to the compatible remote ports is created.”) Ghosh at [0010] (“In another embodiment, a fibre channel network is described. The fibre channel network includes a local fibre channel switch and a remote fibre channel switch. The local fibre channel switch aggregates a compatible subset of the plurality of local ports and sends identifiers for the compat-ible subset of the plurality of local ports to the remote fibre channel switch. The remote fibre channel switch determines if a subset of the plurality of remote ports are compatible. The subset of the plurality of remote ports corresponds to the compatible subset of the plurality of local ports.”) Ghosh at [0022] (“Switches in a fibre channel network are typically interconnected using multiple physical links. The physical links connecting a pair of switches allows transmission of data and control signals. In some instances, it is useful to aggregate multiple physical links into a

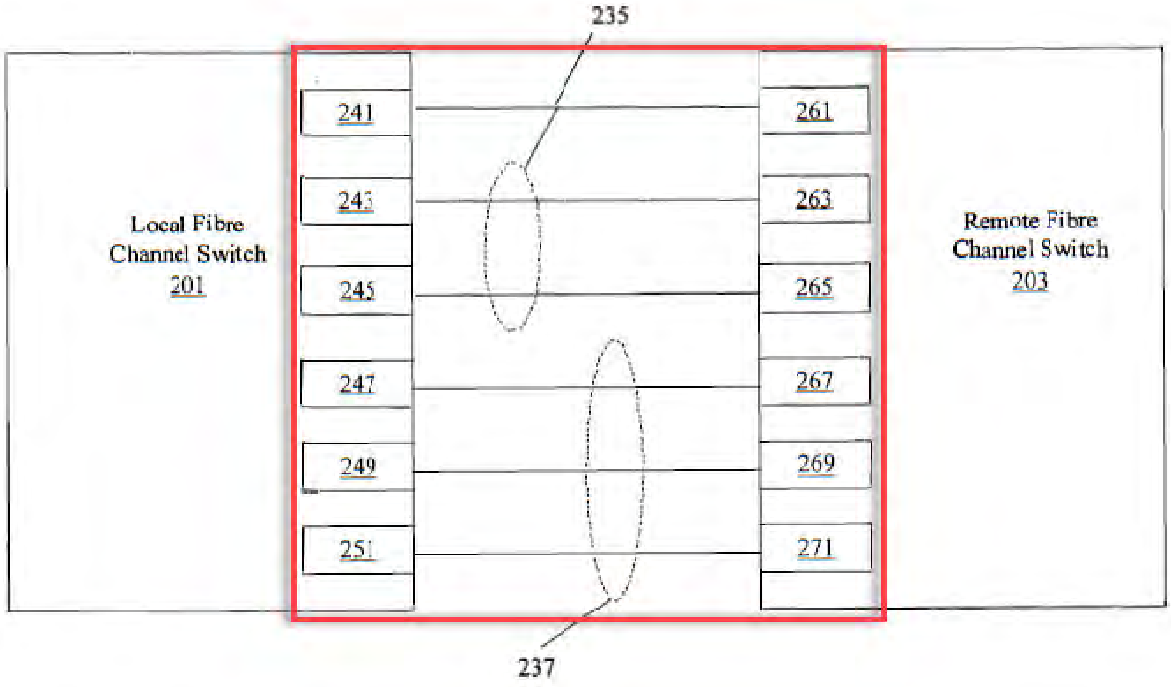
No.	’740 Patent Claim 5	Ghosh
		logical link. Physi-cal links are also referred to herein as physical interfaces and channels while logical links are also referred to herein as logical interfaces and port channels. For example, a local switch may be connected to a remote switch through four physical links. Instead of having to transmit data through a particular physical link, the physical links can be aggregated to form one or more logical links. In one example, all four physical links are aggregated into a single logical link. Instead of having data transmitted through a particular physical link, the data can merely be transmitted over a particular logical link without regard to the particular physi-cal interface used. Aggregating physical links into a logical link allows for higher aggregated bandwidth, load balancing, and link redundancy. For example, if a particular physical link fails or is overloaded, data can still be transmitted over the logical link.”) Ghosh at [0029] (“FIG. 2 is a diagrammatic representation showing links between two switches, such as two fibre channel switches shown in FIG. 1. A local fibre channel switch 201 includes local ports 241, 243, 245, 247, 249, and 251. A remote fibre channel switch 203 includes remote ports 261, 263, 265, 267, 269, and 271. Local port 241 is coupled to remote port 261 through an individual physical link or channel. Connected ports are also referred to herein as peer ports. Local port 243 is coupled to remote port 263 and local port 245 is coupled to remote port 265. The two resulting physical links are aggregated to form port channel 235. Local ports 247, 249, and 251 are coupled to remote ports 267, 269, and 271 respectively. The three resulting physical links are aggregated to form port channel 237.”) Ghosh at [0030] (“According to various embodiments, local fibre channel switch 201 and remote fibre channel switch both have associated identifiers. In some examples, the identifiers are globally unique identifiers such as a global switch world wide names (WWNs). Each local port 241, 243, 245, 247, 249, and 251 and each remote port 261,263,265,267, 269, and 271 can also be associated with identifiers. In some examples, the identifiers are port WWNs. The port WWNs are typically used for debugging or identifying the peer port in alert or warning messages. However, according to various embodiments, the techniques of the present invention use WWNs as globally unique identifiers to aggregate ports instead of using compatibility keys which are only locally unique. Compatibility keys are mechanisms typically used by other protocols such as Ethernet for aggregation.”)

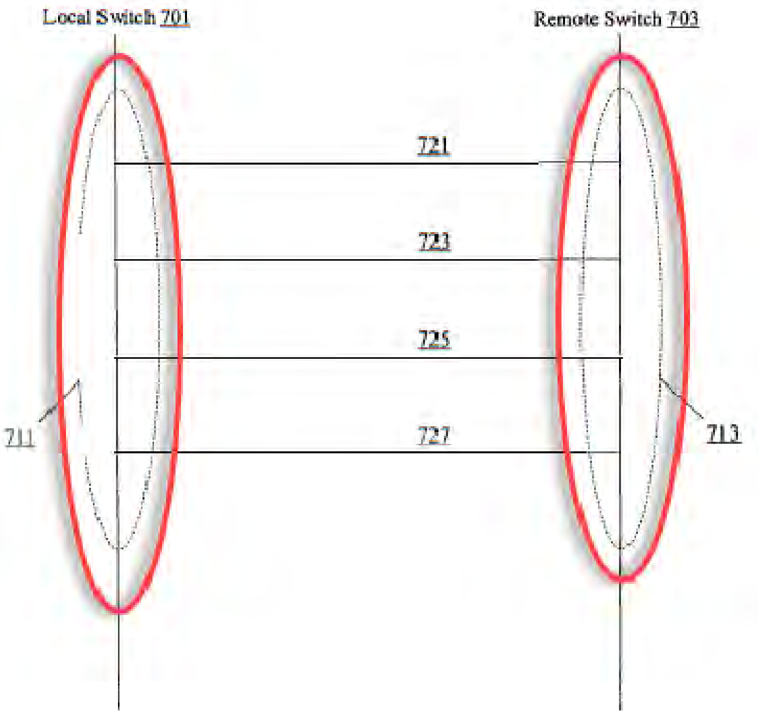
No.	’740 Patent Claim 5	Ghosh
		Ghosh at [0033] (“A variety of parameters can be used to aggregate physical ports. FIG. 3 is a flow process diagram showing one technique for aggregating physical ports into a logical port. And 301, it is determined if auto create functionality is enabled. According to various embodiments, auto create functionality allows automatic configuration and detection of compatible physical ports as well as aggregation into one or more logical ports. Auto creation does not require user intervention. In other examples, administrators can manu-ally arrange ports for aggregation.”) Ghosh at [0037] (“FIG. 4 is an exchange diagram showing one example of a bring up procedure used for a port creating a new port channel. A local switch 401 is coupled to a remote switch 403. The local switch 401 includes a physical port Al coupled to physical port Bl included in remote switch 403. When two peer ports Al and Bl are being aggregated into a port channel, the peer switches 401 and 403 typically already know the world wide names of the individual physical peer ports. However, the peer switches only know the world wide name of their own logical port or port channel. That is, both switches have the individual physical link configured, but the link is not yet part of a port channel. At 421, a local switch 401 sends a synchronize (sync) message 411 to the remote switch 403 to begin the process of creating a port channel including ports Al and Bl.”) Ghosh at [0038] (“In some examples, the sync message 411 includes a local port channel identifier and a remote port channel identifier. In one particular example, the local port channel identifier is set to the world wide name of the local port channel assigned by the local switch 401. The remote port channel identifier is left blank to indicate that the port Al is being aggregated as part of a new port channel. The sync message 411 can also include other parameters such as channel status, channel model, or channel intent.”) Ghosh at [0042] (“When two peer ports A2 and B2 are aggregated into a port channel Cl, the peer switches 501 and 503 typically already know the world wide names of the indi-vidual physical peer ports A2 and B2 as well as the world wide name information of the port channel Cl. Conse-quently, the port channel is already successfully established. According to various

No.	’740 Patent Claim 5	Ghosh
		embodiments, local switch 501 and remote switch 503 perform parameter checking to ensure that the new physical port A2 and B2 can be safely added to the existing port channel Cl. At 521, a local switch can check configuration parameters to ensure that physical ports Al and A2 at the local switch 501 are compatible. The compatibility checking can be performed anytime. In some examples, compatibility checking is checked before a local switch 501 sends a synchronize (sync) message 511 to the remote switch 503 to begin the process of aggregating ports A2 and B2 into the port channel.) Ghosh at [0043] (“In some examples, the sync message 511 includes local port channel identifier and a remote port channel identifier. In one particular example, the local port channel identifier is set to the world wide name of the local port channel assigned by the local switch 501. The remote port channel identifier is filled with the existing port channel identifier to indicate that the port A2 is being aggregated into existing port channel C2. The sync message 511 can also include other parameters such as channel status, channel model, or channel intent.”) Ghosh at [0044] (“At 531, remote switch 503 uses the information received from the local switch 501 to verify port B2 is compatible with other port in port channel C2. In one example, configuration parameters associated with B2 are checked against configuration parameters associated with Bl. The remote switch 503 can also check if the port B2 is already assigned to a different port channel. If the port B2 is compatible with port Bl, the remote switch 503 can proceed and send a sync accept message 513 in response to the sync message 511 to indicate that the port B2 can be aggregated into the port channel. The sync accept message indicates that a port channel can now be modified. At 523, local switch 501 uses the information to update its own port channel database. However, the port channel may not yet be fully operational until the hardware configuration is completed. The local switch 501 continues hardware configuration such as line card configuration to make the port A2 part of the port channel Cl. An acknowledgment 527 is sent and received by remote switch 503 at 529. In some examples, the local switch 501 sends a commit signal 515 when hardware configuration is complete.”) Ghosh at [0045] (“The remote switch 503 receives the commit signal at 533 and begins its own hardware configuration. On completion of its hardware configuration, remote switch 503 sends

No.	’740 Patent Claim 5	Ghosh
		out a commit accept signal 517 to indicate to local switch 501 that hardware configuration is completed. According to various embodiments, local switch 501 receives the commit accept signal 517 and notifies relevant applications that the port channel is now fully operational at 525 and that port A2 has been aggregated into port channel Cl. The local switch 501 can also send out an acknowledge message 519. When the remote switch 503 receives the acknowledge, it notifies relevant applications that the port channel is operational at 535 and that port B2 has been aggregated into port channel Cl. In one embodiments, the techniques of the present invention contemplate using a two phase SYNC and COMMIT mechanism similar to the mechanism used in EPP.”) Ghosh at [0046] (“FIGS. 4 and 5 show examples of ports being aggregated into a port channel. At a particular switch, ports can be selected for aggregation into a port channel in a variety of manners. FIG. 6 is an exchange diagram showing automatic selection of ports at a switch for aggregation into a port channel. A local switch 601 is coupled to a remote switch 603. In one example, the local switch 601 includes physical ports Al, A2, A3, and A4 while remote switch 603 includes physical ports Bl, B2, B3, and B4. No port chan-nels have been formed.”) Ghosh at [0049] (“At 631, remote switch 603 uses the information received from the local switch 601 to verify that the peer ports of Al, A2, and A4 are compatible. That is, ports Bl, B2, and B4 are checked for compatibility. In one example, only ports Bl and B2 may be compatible, and consequently only ports Al, A2, Bl, and B2 can be included in the port channel. In another example, ports Bl, B2, and B4 are compatible, so ports Al, A2, A4, Bl, B2, and B4 can be aggregated into port channel Cl. According to various embodiments, if the port B2 is compatible with port Bl, the remote switch 603 can proceed and send a sync accept message 613 in response to the sync message 611 to indicate that the port B2 can be aggregated into the port channel. It should be noted that remote switch 603 can send a list indicating that ports B2 and B4 are compatible with Bl. However, the remote switch 603 sends only one compatible port B2 back for several reasons, and in the process of selection compatible port channels get priority over com-patible individual ports.”)

No.	’740 Patent Claim 5	Ghosh
		Ghosh at [0050] (“One reason is that aggregation mechanisms and techniques can be implemented more elegantly by handling ports on an individual basis. Any individual port will either start a new port channel, be added to an existing port channel, or operate stand alone. There is no need to keep track of groups of ports to be aggregated. Another reason is that fewer ports need to be locked if only a single port is being aggregated at any one time. The sync accept message indicates that a port channel can now be modified. At 623, local switch 601 receives the information and recognizes that Al and A2 can now be aggregated into port channel Cl. However, the port channel may not yet be fully operational until the hardware configuration is completed. An acknowl-edgment 627 is sent and received by remote switch 603 at 629. In some examples, the local switch 601 sends a commit signal 615 when hardware configuration is complete.”) Ghosh at [0051] (“The remote switch 603 receives the commit signal at 633 to create port channel Cl including ports Bl and B2. Hardware configuration can now be performed. On comple-tion of its hardware configuration, remote switch 603 sends out a commit accept signal 617 to indicate to local switch 601 that hardware configuration is completed. According to various embodiments, local switch 601 receives the commit accept signal 617 and notifies relevant applications that the port channel is now fully operational at 625 and that ports Al and A2 have been aggregated into port channel Cl. The local switch 601 can also send out an acknowledge message 619. When the remote switch 603 receives the acknowledge, it notifies relevant applications that the port channel is fully operational at 635 and that ports Bl and B2 have been aggregated into port channel Cl.”) Ghosh at [0053] (“FIG. 7 is a diagrammatic representation showing synchronous aggregation of ports into a port channel. A local switch 701 is coupled to a remote switch 703 through links 721, 723, 725, and 727. According to various embodiments, the links are being aggregated into port channel 711 at the local switch 701 and port channel 713 at the remote switch 703 in a synchronous manner. That is the peer ports corre-sponding to each link are brought up in the same order at both the local switch 701 and the remote switch 703.”) Ghosh at Figure 2 (annotation added)

No.	’740 Patent Claim 5	Ghosh
		 235 241 243 245 247 249 251 261 263 265 267 269 271 Local Fibre Channel Switch 201 Remote Fibre Channel Switch 203 237 Figure 2 Ghosh at Figure 7 (annotation added)

No.	’740 Patent Claim 5	Ghosh
		 Local Switch 701 Remote Switch 703 721 723 725 727 711 713 Figure 7

No.	’740 Patent Claim 6	Ghosh
6	The method according to claim 1, wherein coupling each of the one or more interface	Ghosh discloses the method according to claim 1, wherein coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and

No.	’740 Patent Claim 6	Ghosh
	modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node.	demultiplexing downstream data frames sent from the communication network to the network node. For example, Ghosh discloses exchanging data frames for a given flow in both directions. A person of ordinary skill in the art would understand that networking systems and methods employ multiplexers and demultiplexers as standard components to efficiently transmit upstream and downstream traffic. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Ghosh is found to not meet this limitation, wherein coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node would have been obvious to a person having ordinary skill in the art, as explained below. *See supra* Claim 1. Ghosh at [0026] (“Consequently, the techniques and mechanisms of the present invention allow automatic detection of compat-ible ports to enable automatic creation of port channels. Port channels can be effectively brought up at either a local switch or a remote switch after either automatic creation of port channels or manual configuration of port channels. Robust error detection capabilities allow the correction of improper configurations and connections. Member physical ports of a port channel can operate as individual links if they cannot be configured to be part of port channel. Further-more, synchronization is supported so that requests and responses belonging to the same flow can be carried over the same physical link in a port channel in both directions. In many conventional implementations, such as Ethernet for example, a flow belonging to a particular port channel could be carried over different physical links during send and receive phases.”) Ghosh at [0032] (“Each entity can also have additional parameters to aid in the set up of port channels. According to various embodiments, parameters such as a channeling model, a

No.	’740 Patent Claim 6	Ghosh
		channeling intent and a channeling status are included. A channeling model indicates to a peer port the channel group is automatically created or user configured. A channeling intent parameter indicates the peer port if this port intends to participate in a port channel. Otherwise, the port intends to operate as an individual port. The channeling status param-eter tells the peer port about its current channeling status. This parameter is exchanged by the attached peer ports to agree upon the channeling status of the link and to ensure that both ends are synchronized.”) Ghosh at [0055] (“Based on this scheme, frames for a given flow are transmitted through the same physical port of the port channel. However, unless there is proper synchronization of the port channel map tables at the two ends of a port channel, it is possible that requests and responses for the same flow are carried over two different physical links. For example, requests may be carried over link 721 while responses are carried over 723. This is undesirable for port channels as it affects applications like write acceleration that assume the traffic for a given flow is carried over the same physical link in both directions.”) Ghosh at [0056] (“According to various embodiments of the present invention, links are brought up in order. A first link is selected for bring up. No effort is made to bring up other links is attempted until the exchange associated with the first link is completed. Consequently, port channel map tables at the local switch 701 and at the remote switch 703 are consistent. Port channel map table entries are synchronized at both ends so that all frames for a given flow are carried over the same physical link in both directions after the exchanges are completed.”) Ghosh at [0059] (“Line cards 803, 805, and 807 can communicate with an active supervisor 811 through interface circuitry 883, 885, and 887 and the backplane 815. According to various embodiments, each line card includes a plurality of ports that can act as either input ports or output ports for communication with external fibre channel network entities 851 and 853. The backplane 815 can provide a communi-cations channel for all traffic between line cards and super- visors. Individual line cards 803 and 807 can also be coupled to external fibre channel network entities 851 and 853 through fibre channel ports 843 and 847.”)

No.	’740 Patent Claim 6	Ghosh
		Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Ghosh in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 6 of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Wiher discloses multiplexing and demultiplexing circuitry to transmit and receive ATM data cells over a data link between the ATM network and network access equipment. Wiher at 3:43-65 (“In general, in another aspect, the invention features an apparatus for communicating data cells between a data link and a backplane. The apparatus includes transceiver circuitry to transmit and receive data cells over a data link and a plurality of backplane interfaces each including at least one cell signal terminal. Each of the backplane interface is coupled to a backplane interconnection circuit. Each backplane interconnection circuit transmits and receives cells over the cell signal terminals of its associated backplane interface. The apparatus also includes de-multiplexing circuitry coupling the transceiver circuitry to each of the backplane interconnection circuits. The de-multiplexing circuitry receives a data cell from the transceiver circuitry, select a backplane interconnection circuit associated with the data cell, and provide the data cell to the selected backplane interconnection circuit for transmission over the cell signal terminals of the associated backplane interface. The apparatus also includes multiplexing circuitry coupling the plurality of backplane interconnection circuits to the transceiver circuitry. The multiplexing circuitry receives data cells from each of the backplane interconnection circuits and provide the received data cells to the transceiver circuitry.”) Wiher at 3:66-4:22 (“Implementations of the invention may include one or more of the following features. The backplane interconnection circuits may independently receive and transmit data cells over the plurality of backplane interfaces. The de-multiplexing circuitry may select a backplane interface based on data in the header field of the data cell. The apparatus may include header translation circuitry to alter header data in cells sent between the plurality of backplane interfaces and the transceiver circuitry. Each of the plurality of

No.	’740 Patent Claim 6	Ghosh
		backplane interfaces may include separate terminals to receive cells and separate terminals to transmit cells. The terminals to transmit cells may include a first and second control terminal and at least one outgoing cell data terminal. A backplane interface's backplane interconnection circuitry may accepts a signal on the first control terminal as indicating that a cell may be sent over the interface, asserts a 15 signal on the second control terminal to indicate that a cell is being transmitted, and transmits data bits of the cell on the outgoing cell data terminal. Each backplane interface may include a single outgoing cell data terminal and each bit of the cell may be serially transmitted over the single outgoing cell data terminal. Each backplane interface may include multiple outgoing cell data terminals and bits of the cell may be sent in parallel over the eight outgoing cell data terminals.”) For example, Lebizay discloses an optical add/drop multiplexer that multiplexers and demultiplexes data packets sent between the network boards and network. Lebizay at [0043] (“InfiniBand offers link layer Virtual Lanes (VLs) to support multiple logical channels (i.e. multiplexing) on the same physical link. Infiniband offers up to 16 virtual lanes per link. VLs provide a mechanism to avoid head-of-line blocking and the ability to support Quality of Service (QoS). The difference between a Virtual Lane and a Service Level (SL) is that a Virtual Lane is the actual logical lane (mul-tiplexed) used on a given point-to-point link. The Service Level stays constant as a packet traverses the fabric, and specifies the desired service level within a subnet. The SL (AF, EF or BE) is included in the link header, and each switch maps the SL to a VL supported by the destination link. A switch supporting a limited number of virtual lanes will map the SL field to a VL it supports. Without preserving the SL, the desired SL (AF, EF or BE) would be lost in this mapping, and later in the path, a switch supporting more VLs would be unable to recover finer granularity of SLs between two packets mapped to the same VL.”) Lebizay at [0050] (“The issue with using a ring, however, is how to map the addressing of multiple boards across these fibers. One solution is to employ Wavelength Division Multiplex-ing (WDM). A WDM optical mesh defines a meshed-topology in the wavelength space as opposed to the physical fiber space. By utilizing multiple discrete lambda-waves as optical carriers such that by meshing dedicated optical wavelengths between every two

No.	’740 Patent Claim 6	Ghosh
		boards, layer 2 protocols are eliminated, thereby creating a dramatic improvement in the efficiency of the transport. Today, every packet transport requires a protocol that allows the end point (and interme-diate points) to decipher the intended path (or consumer) of the packet. This protocol increases the amount of overhead required in the packet bus, allowing less room for actual data to be sent. By moving the protocol into the wavelength of the actual optical signal, the destination is implied by the wavelength and no additional bandwidth needs to be sur-rendered on the signal to provide this information. This makes the efficiency of the transport better and also speeds the routing of the packet through the network. In addition, the use of optical interconnects in a backplane environment greatly increases chassis bandwidth as well as reducing electrical radiation that often accompanies copper intercon-nects. The components involved include an optical back-plane in a physical ring topology, and the necessary trans-mitters and receivers for the size of the installation (i.e., number of slots in the chassis). In addition, optical add/drop multiplexer devices are required.”)

No.	’740 Patent Claim 7	Ghosh
7	The method according to claim 1, wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the first and second physical links.	Ghosh discloses the method according to claim 1, wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the first and second physical links. For example, Ghosh discloses aggregating physical links, including ports, into aggregate port channels that form a single logical link to increase bandwidth, load balancing, and link redundancy. *See supra* Claim 1 Ghosh at Abstract (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggregation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port

No.	’740 Patent Claim 7	Ghosh
		channels. Ports can be aggregated synchronously to allow consistent gen-eration of port channel map tables.”) Ghosh at [0004] (“Neighboring nodes in a fibre channel network are typically interconnected through multiple physical links. For example, a local fibre channel switch may be connected to a remote fibre channel switch through four physical links. In many instances, it may be beneficial to aggregate some of the physical links into logical links. That is, multiple physi-cal links can be combined to form a logical interface to provide higher aggregate bandwidth, load balancing, and link redundancy. When a frame is being transmitted over a logical link, it does not matter what particular physical link is being used as long as all the frames of a given flow are transmitted through the same link. If a constituent physical link goes down, the logical link can still remain operational.”) Ghosh at [0007] (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggre-gation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent generation of port channel map tables.”) Ghosh at [0008] (“In one embodiment, a method for aggregating ports in a fibre channel fabric is provided. It is determined that a plurality of local ports at a local fibre channel switch are compatible. Identifiers for the plurality of local ports are sent to a remote fibre channel switch. The remote fibre channel switch determines if a plurality of remote ports are compat-ible, the plurality of remote ports corresponding to the plurality of local ports. An indication that one or more of the remote physical ports are compatible is received. A port channel including one or more of the local ports correspond-ing to the compatible remote ports is created.”) Ghosh at [0010] (“In another embodiment, a fibre channel network is described. The fibre channel network includes a local fibre channel switch and a remote fibre channel switch. The local fibre channel switch aggregates a compatible subset of the plurality of local ports and

No.	’740 Patent Claim 7	Ghosh
		sends identifiers for the compat-ible subset of the plurality of local ports to the remote fibre channel switch. The remote fibre channel switch determines if a subset of the plurality of remote ports are compatible. The subset of the plurality of remote ports corresponds to the compatible subset of the plurality of local ports.”) Ghosh at [0022] (“Switches in a fibre channel network are typically interconnected using multiple physical links. The physical links connecting a pair of switches allows transmission of data and control signals. In some instances, it is useful to aggregate multiple physical links into a logical link. Physi-cal links are also referred to herein as physical interfaces and channels while logical links are also referred to herein as logical interfaces and port channels. For example, a local switch may be connected to a remote switch through four physical links. Instead of having to transmit data through a particular physical link, the physical links can be aggregated to form one or more logical links. In one example, all four physical links are aggregated into a single logical link. Instead of having data transmitted through a particular physical link, the data can merely be transmitted over a particular logical link without regard to the particular physi-cal interface used. Aggregating physical links into a logical link allows for higher aggregated bandwidth, load balancing, and link redundancy. For example, if a particular physical link fails or is overloaded, data can still be transmitted over the logical link.”) Ghosh at [0029] (“FIG. 2 is a diagrammatic representation showing links between two switches, such as two fibre channel switches shown in FIG. 1. A local fibre channel switch 201 includes local ports 241, 243, 245, 247, 249, and 251. A remote fibre channel switch 203 includes remote ports 261, 263, 265, 267, 269, and 271. Local port 241 is coupled to remote port 261 through an individual physical link or channel. Connected ports are also referred to herein as peer ports. Local port 243 is coupled to remote port 263 and local port 245 is coupled to remote port 265. The two resulting physical links are aggregated to form port channel 235. Local ports 247, 249, and 251 are coupled to remote ports 267, 269, and 271 respectively. The three resulting physical links are aggregated to form port channel 237.”) Ghosh at [0030] (“According to various embodiments, local fibre channel switch 201 and remote fibre channel switch both have associated identifiers. In some examples, the

No.	’740 Patent Claim 7	Ghosh
		identifiers are globally unique identifiers such as a global switch world wide names (WWNs). Each local port 241, 243, 245, 247, 249, and 251 and each remote port 261,263,265,267, 269, and 271 can also be associated with identifiers. In some examples, the identifiers are port WWNs. The port WWNs are typically used for debugging or identifying the peer port in alert or warning messages. However, according to various embodiments, the techniques of the present invention use WWNs as globally unique identifiers to aggregate ports instead of using compatibility keys which are only locally unique. Compatibility keys are mechanisms typically used by other protocols such as Ethernet for aggregation.”) Ghosh at [0033] (“A variety of parameters can be used to aggregate physical ports. FIG. 3 is a flow process diagram showing one technique for aggregating physical ports into a logical port. And 301, it is determined if auto create functionality is enabled. According to various embodiments, auto create functionality allows automatic configuration and detection of compatible physical ports as well as aggregation into one or more logical ports. Auto creation does not require user intervention. In other examples, administrators can manu-ally arrange ports for aggregation.”) Ghosh at [0037] (“FIG. 4 is an exchange diagram showing one example of a bring up procedure used for a port creating a new port channel. A local switch 401 is coupled to a remote switch 403. The local switch 401 includes a physical port Al coupled to physical port Bl included in remote switch 403. When two peer ports Al and Bl are being aggregated into a port channel, the peer switches 401 and 403 typically already know the world wide names of the individual physical peer ports. However, the peer switches only know the world wide name of their own logical port or port channel. That is, both switches have the individual physical link configured, but the link is not yet part of a port channel. At 421, a local switch 401 sends a synchronize (sync) message 411 to the remote switch 403 to begin the process of creating a port channel including ports Al and Bl.”) Ghosh at [0038] (“In some examples, the sync message 411 includes a local port channel identifier and a remote port channel identifier. In one particular example, the local port channel identifier is set to the world wide name of the local port channel assigned by the local switch 401. The remote port channel identifier is left blank to indicate that the port Al

No.	’740 Patent Claim 7	Ghosh
		is being aggregated as part of a new port channel. The sync message 411 can also include other parameters such as channel status, channel model, or channel intent.”) Ghosh at [0042] (“When two peer ports A2 and B2 are aggregated into a port channel Cl, the peer switches 501 and 503 typically already know the world wide names of the indi-vidual physical peer ports A2 and B2 as well as the world wide name information of the port channel Cl. Conse-quently, the port channel is already successfully established. According to various embodiments, local switch 501 and remote switch 503 perform parameter checking to ensure that the new physical port A2 and B2 can be safely added to the existing port channel Cl. At 521, a local switch can check configuration parameters to ensure that physical ports Al and A2 at the local switch 501 are compatible. The compatibility checking can be performed anytime. In some examples, compatibility checking is checked before a local switch 501 sends a synchronize (sync) message 511 to the remote switch 503 to begin the process of aggregating ports A2 and B2 into the port channel.) Ghosh at [0043] (“In some examples, the sync message 511 includes local port channel identifier and a remote port channel identifier. In one particular example, the local port channel identifier is set to the world wide name of the local port channel assigned by the local switch 501. The remote port channel identifier is filled with the existing port channel identifier to indicate that the port A2 is being aggregated into existing port channel C2. The sync message 511 can also include other parameters such as channel status, channel model, or channel intent.”) Ghosh at [0044] (“At 531, remote switch 503 uses the information received from the local switch 501 to verify port B2 is compatible with other port in port channel C2. In one example, configuration parameters associated with B2 are checked against configuration parameters associated with Bl. The remote switch 503 can also check if the port B2 is already assigned to a different port channel. If the port B2 is compatible with port Bl, the remote switch 503 can proceed and send a sync accept message 513 in response to the sync message 511 to indicate that the port B2 can be aggregated into the port channel. The sync accept message indicates that a port channel can now be modified. At 523, local switch 501 uses the information to update its own port channel database. However, the port channel may

No.	’740 Patent Claim 7	Ghosh
		not yet be fully operational until the hardware configuration is completed. The local switch 501 continues hardware configuration such as line card configuration to make the port A2 part of the port channel Cl. An acknowledgment 527 is sent and received by remote switch 503 at 529. In some examples, the local switch 501 sends a commit signal 515 when hardware configuration is complete.”) Ghosh at [0045] (“The remote switch 503 receives the commit signal at 533 and begins its own hardware configuration. On completion of its hardware configuration, remote switch 503 sends out a commit accept signal 517 to indicate to local switch 501 that hardware configuration is completed. According to various embodiments, local switch 501 receives the commit accept signal 517 and notifies relevant applications that the port channel is now fully operational at 525 and that port A2 has been aggregated into port channel Cl. The local switch 501 can also send out an acknowledge message 519. When the remote switch 503 receives the acknowledge, it notifies relevant applications that the port channel is operational at 535 and that port B2 has been aggregated into port channel Cl. In one embodiments, the techniques of the present invention contemplate using a two phase SYNC and COMMIT mechanism similar to the mechanism used in EPP.”) Ghosh at [0046] (“FIGS. 4 and 5 show examples of ports being aggregated into a port channel. At a particular switch, ports can be selected for aggregation into a port channel in a variety of manners. FIG. 6 is an exchange diagram showing automatic selection of ports at a switch for aggregation into a port channel. A local switch 601 is coupled to a remote switch 603. In one example, the local switch 601 includes physical ports Al, A2, A3, and A4 while remote switch 603 includes physical ports Bl, B2, B3, and B4. No port chan-nels have been formed.”) Ghosh at [0049] (“At 631, remote switch 603 uses the information received from the local switch 601 to verify that the peer ports of Al, A2, and A4 are compatible. That is, ports Bl, B2, and B4 are checked for compatibility. In one example, only ports Bl and B2 may be compatible, and consequently only ports Al, A2, Bl, and B2 can be included in the port channel. In another example, ports Bl, B2, and B4 are compatible, so ports Al, A2, A4, Bl, B2, and B4 can be aggregated into port channel Cl. According to various embodiments, if

No.	’740 Patent Claim 7	Ghosh
		the port B2 is compatible with port Bl, the remote switch 603 can proceed and send a sync accept message 613 in response to the sync message 611 to indicate that the port B2 can be aggregated into the port channel. It should be noted that remote switch 603 can send a list indicating that ports B2 and B4 are compatible with Bl. However, the remote switch 603 sends only one compatible port B2 back for several reasons, and in the process of selection compatible port channels get priority over com-patible individual ports.”) Ghosh at [0050] (“One reason is that aggregation mechanisms and techniques can be implemented more elegantly by handling ports on an individual basis. Any individual port will either start a new port channel, be added to an existing port channel, or operate stand alone. There is no need to keep track of groups of ports to be aggregated. Another reason is that fewer ports need to be locked if only a single port is being aggregated at any one time. The sync accept message indicates that a port channel can now be modified. At 623, local switch 601 receives the information and recognizes that Al and A2 can now be aggregated into port channel Cl. However, the port channel may not yet be fully operational until the hardware configuration is completed. An acknowl-edgment 627 is sent and received by remote switch 603 at 629. In some examples, the local switch 601 sends a commit signal 615 when hardware configuration is complete.”) Ghosh at [0051] (“The remote switch 603 receives the commit signal at 633 to create port channel Cl including ports Bl and B2. Hardware configuration can now be performed. On comple-tion of its hardware configuration, remote switch 603 sends out a commit accept signal 617 to indicate to local switch 601 that hardware configuration is completed. According to various embodiments, local switch 601 receives the commit accept signal 617 and notifies relevant applications that the port channel is now fully operational at 625 and that ports Al and A2 have been aggregated into port channel Cl. The local switch 601 can also send out an acknowledge message 619. When the remote switch 603 receives the acknowledge, it notifies relevant applications that the port channel is fully operational at 635 and that ports Bl and B2 have been aggregated into port channel Cl.”) Ghosh at [0053] (“FIG. 7 is a diagrammatic representation showing synchronous aggregation of ports into a port channel. A local switch 701 is coupled to a remote switch 703 through

No.	’740 Patent Claim 7	Ghosh
		links 721, 723, 725, and 727. According to various embodiments, the links are being aggregated into port channel 711 at the local switch 701 and port channel 713 at the remote switch 703 in a synchronous manner. That is the peer ports corre-sponding to each link are brought up in the same order at both the local switch 701 and the remote switch 703.”)

No.	’740 Patent Claim 8	Ghosh
8	The method according to claim 1, wherein selecting the first and second physical links comprises applying a mapping function to the at least one of the frame attributes.	Ghosh discloses the method according to claim 1, wherein selecting the first and second physical links comprises applying a mapping function to the at least one of the frame attributes. For example, Ghosh discloses using a port channel map table to select the physical links through which a frame is transmitted. A person of ordinary skill in the art would understand a port channel map table is a variety of a mapping function which applies to frame specific information. *See supra* Claim 1 Ghosh at [0004] (“Neighboring nodes in a fibre channel network are typically interconnected through multiple physical links. For example, a local fibre channel switch may be connected to a remote fibre channel switch through four physical links. In many instances, it may be beneficial to aggregate some of the physical links into logical links. That is, multiple physi-cal links can be combined to form a logical interface to provide higher aggregate bandwidth, load balancing, and link redundancy. When a frame is being transmitted over a logical link, it does not matter what particular physical link is being used as long as all the frames of a given flow are transmitted through the same link. If a constituent physical link goes down, the logical link can still remain operational.”) Ghosh at [0054] (“A database such as a port channel map table is used store the links included in a port channel. The links and/or port information is entered into the table based on the order in which they were brought up. The port channel map table is used to select the physical link through which a frame is transmitted. In conventional implementations, the

No.	’740 Patent Claim 8	Ghosh
		order of the peer ports listed in the port channel map table can be different for both the local switch 701 and the remote switch 703.”) Ghosh at [0055] (“Based on this scheme, frames for a given flow are transmitted through the same physical port of the port channel. However, unless there is proper synchronization of the port channel map tables at the two ends of a port channel, it is possible that requests and responses for the same flow are carried over two different physical links. For example, requests may be carried over link 721 while responses are carried over 723. This is undesirable for port channels as it affects applications like write acceleration that assume the traffic for a given flow is carried over the same physical link in both directions.”) Ghosh at [0056] (“According to various embodiments of the present invention, links are brought up in order. A first link is selected for bring up. No effort is made to bring up other links is attempted until the exchange associated with the first link is completed. Consequently, port channel map tables at the local switch 701 and at the remote switch 703 are consistent. Port channel map table entries are synchronized at both ends so that all frames for a given flow are carried over the same physical link in both directions after the exchanges are completed.”) Ghosh at [0061] (“According to one embodiment, the routing appli-cation is configured to provide credits to a sender upon recognizing that a frame has been forwarded to a next hop. A utility application can be configured to track the number of buffers and the number of credits used. A domain manager application can be used to assign domains in the fibre channel storage area network. Various supervisor applica-tions may also be configured to provide functionality such as flow control, credit management, and quality of service (QoS) functionality for various fibre channel protocol layers.”)

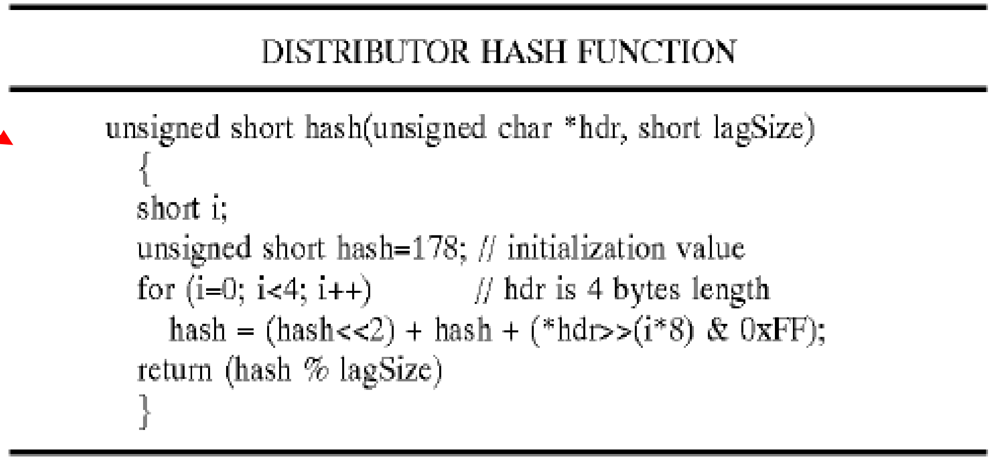
No.	’740 Patent Claim 9	Ghosh
9	The method according to claim 8, wherein applying the mapping function comprises applying a hashing function.	Ghosh discloses the method according to claim 8, wherein applying the mapping function comprises applying a hashing function. For example, Ghosh discloses using a port channel map table to select the physical links through which a frame is transmitted. A person of ordinary skill in the art would understand a port channel map table is a variety of a mapping function which applies to frame specific information. A person of ordinary skill in the art would further understand that a hashing function is a standard algorithm used to route data traffic. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Ghosh is found to not meet this limitation, wherein applying the mapping function comprises applying a hashing function would have been obvious to a person having ordinary skill in the art, as explained below. *See supra* Claim 8 Ghosh at [0004] (“Neighboring nodes in a fibre channel network are typically interconnected through multiple physical links. For example, a local fibre channel switch may be connected to a remote fibre channel switch through four physical links. In many instances, it may be beneficial to aggregate some of the physical links into logical links. That is, multiple physi-cal links can be combined to form a logical interface to provide higher aggregate bandwidth, load balancing, and link redundancy. When a frame is being transmitted over a logical link, it does not matter what particular physical link is being used as long as all the frames of a given flow are transmitted through the same link. If a constituent physical link goes down, the logical link can still remain operational.”) Ghosh at [0054] (“A database such as a port channel map table is used store the links included in a port channel. The links and/or port information is entered into the table based on the order in which they were brought up. The port channel map table is used to select the physical link through which a frame is transmitted. In conventional implementations, the order of the peer

No.	’740 Patent Claim 9	Ghosh
		ports listed in the port channel map table can be different for both the local switch 701 and the remote switch 703.”) Ghosh at [0055] (“Based on this scheme, frames for a given flow are transmitted through the same physical port of the port channel. However, unless there is proper synchronization of the port channel map tables at the two ends of a port channel, it is possible that requests and responses for the same flow are carried over two different physical links. For example, requests may be carried over link 721 while responses are carried over 723. This is undesirable for port channels as it affects applications like write acceleration that assume the traffic for a given flow is carried over the same physical link in both directions.”) Ghosh at [0056] (“According to various embodiments of the present invention, links are brought up in order. A first link is selected for bring up. No effort is made to bring up other links is attempted until the exchange associated with the first link is completed. Consequently, port channel map tables at the local switch 701 and at the remote switch 703 are consistent. Port channel map table entries are synchronized at both ends so that all frames for a given flow are carried over the same physical link in both directions after the exchanges are completed.”) Ghosh at [0061] (“According to one embodiment, the routing appli-cation is configured to provide credits to a sender upon recognizing that a frame has been forwarded to a next hop. A utility application can be configured to track the number of buffers and the number of credits used. A domain manager application can be used to assign domains in the fibre channel storage area network. Various supervisor applica-tions may also be configured to provide functionality such as flow control, credit management, and quality of service (QoS) functionality for various fibre channel protocol layers.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Ghosh in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 9 of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references.

No.	’740 Patent Claim 9	Ghosh
		For example, Bruckman discloses applying a distributor hash function. Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be

No.	’740 Patent Claim 9	Ghosh
		carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”) Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I: Bruckman at Table 1 (annotated)

No.	’740 Patent Claim 9	Ghosh

		hashing function *“mapping function”*
		
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		As another example, Solomon discloses applying a mapping function which comprises a hashing function performed by the mapper.
		Solomon at [0024] (“In another embodiment, switching the data packets includes mapping the data packets to the selected port responsively to the label. Additionally or alternatively, map-ping the data packets includes applying a hashing function to the label so as to determine a number of the selected port, and choosing the label includes applying an inverse of the hashing function to the number of the selected port.”)

No.	’740 Patent Claim 9	Ghosh
		Solomon at [0048] (“The mapping function typically uses MPLS label 52 for mapping, since the MPLS label uniquely identifies MPLS tunnel 28, and it is required that all MPLS packets belonging to the same tunnel be switched through the same physical port 24. Additionally or alternatively, the mapping function uses a "PW" label (pseudo wire label, formerly known as a virtual connection, or VC label), which is optionally added to MPLS header 50. The PW label com-prises information that the egress node requires for deliver-ing the packet to its destination, and is optionally added during the encapsulation of MPLS packets. Additional details regarding the VC label can be found in an IETF draft by Martini et al. entitled "Encapsulation Methods for Trans-port of Ethernet Frames Over IP/MPLS Networks" (IETF draft-ietf-pwe3-ethernet-encap-07.txt, May, 2004), which is incorporated herein by reference. In some embodiments, mapper 34 applies a hashing function to the MPLS and/or PW label, as will be described below.”) Solomon at [0059] (“In this method, the mapping function used by mapper 34 of switch A is a hashing function. Various hashing functions are known in the art, and any suitable hashing function may be used in mapper 34. Since the hashing operation is performed for each packet, it is desirable to have a hashing function that is computationally simple.”) Solomon at [0060] (“As mentioned above, the hashing function typically hashes the value of MPLS label 52 to determine the selected physical port, as the MPLS label uniquely identifies tunnel 28. For example, the following hashing function may be used by mapper 34: Selected port number=l+((MPLS label) mod N), wherein N denotes the number of physical Ethernet ports in LAG group 25, and "mod" denotes the modulus operator. Assuming the values of MPLS labels are distrib-uted uniformly over a certain range, this function achieves a uniform distribution of port allocations for the different MPLS labels. It can also be seen that all packets carrying the same MPLS label (in other words-belonging to the same MPLS tunnel) will be mapped to the same physical port.”) Solomon at [0065] (“Mapper 34 of switch A maps each received packet to the selected physical port of LAG group 25 using the hashing function, at a hashing step 90. Mapper 34 extracts the MPLS label from each received packet and uses the hashing function to calculate the serial number of the selected physical port, which was selected by the CAC processor at step 82.

No.	’740 Patent Claim 9	Ghosh
		Following the numerical example given above, the mapper extracts MPLS label=65647 from the packet. Substituting this value and N=3 into the hashing function gives: Selected port number=1+(65647 mod 3)=2, which is indeed the port number selected in the example above.”)

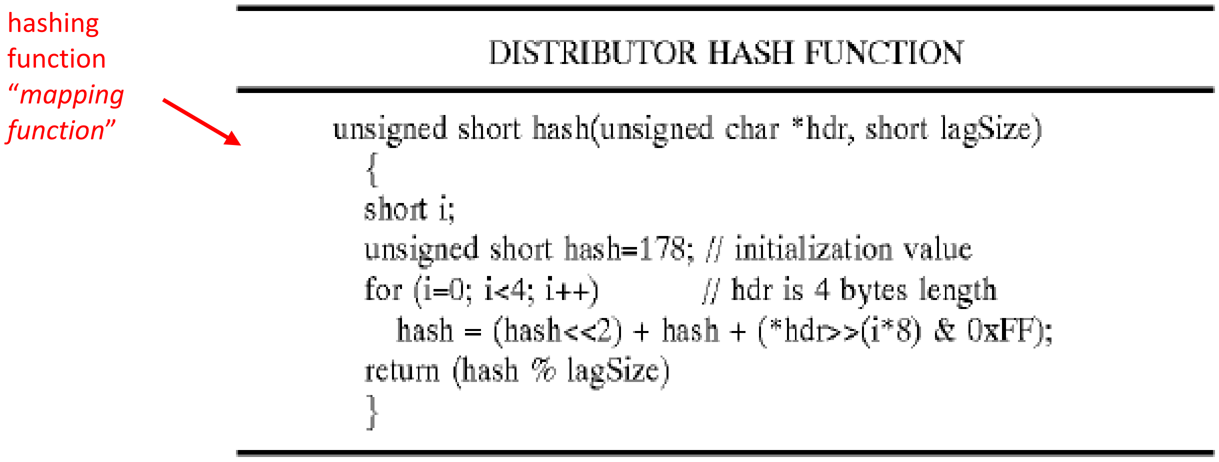
No.	’740 Patent Claim 10	Ghosh
10[a]	The method according to claim 9, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second physical links,	Ghosh discloses the method according to claim 9, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second physical links. For example, Ghosh discloses using a port channel map table to select the physical links through which a frame is transmitted. A person of ordinary skill in the art would further understand that a hashing function is a standard algorithm used to route data traffic that involves information specific to the system architecture, including involving the number of physical links. *See supra at* Claim 9. Ghosh at [0004] (“Neighboring nodes in a fibre channel network are typically interconnected through multiple physical links. For example, a local fibre channel switch may be connected to a remote fibre channel switch through four physical links. In many instances, it may be beneficial to aggregate some of the physical links into logical links. That is, multiple physi-cal links can be combined to form a logical interface to provide higher aggregate bandwidth, load balancing, and link redundancy. When a frame is being transmitted over a logical link, it does not matter what particular physical link is being used as long as all the frames of a given flow are transmitted through the same link. If a constituent physical link goes down, the logical link can still remain operational.”) Ghosh at [0054] (“A database such as a port channel map table is used store the links included in a port channel. The links and/or port information is entered into the table based on the order in

No.	’740 Patent Claim 10	Ghosh
		which they were brought up. The port channel map table is used to select the physical link through which a frame is transmitted. In conventional implementations, the order of the peer ports listed in the port channel map table can be different for both the local switch 701 and the remote switch 703.”) Ghosh at [0055] (“Based on this scheme, frames for a given flow are transmitted through the same physical port of the port channel. However, unless there is proper synchronization of the port channel map tables at the two ends of a port channel, it is possible that requests and responses for the same flow are carried over two different physical links. For example, requests may be carried over link 721 while responses are carried over 723. This is undesirable for port channels as it affects applications like write acceleration that assume the traffic for a given flow is carried over the same physical link in both directions.”) Ghosh at [0056] (“According to various embodiments of the present invention, links are brought up in order. A first link is selected for bring up. No effort is made to bring up other links is attempted until the exchange associated with the first link is completed. Consequently, port channel map tables at the local switch 701 and at the remote switch 703 are consistent. Port channel map table entries are synchronized at both ends so that all frames for a given flow are carried over the same physical link in both directions after the exchanges are completed.”) Ghosh at [0061] (“According to one embodiment, the routing appli-cation is configured to provide credits to a sender upon recognizing that a frame has been forwarded to a next hop. A utility application can be configured to track the number of buffers and the number of credits used. A domain manager application can be used to assign domains in the fibre channel storage area network. Various supervisor applica-tions may also be configured to provide functionality such as flow control, credit management, and quality of service (QoS) functionality for various fibre channel protocol layers.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Ghosh in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified

No.	’740 Patent Claim 10	Ghosh
		in element 10[a] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses applying a distributor hash function to the frame information which includes determining a number of the plurality of physical link. Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”)

No.	’740 Patent Claim 10	Ghosh
		Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”) Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:

No.	’740 Patent Claim 10	Ghosh

		Bruckman at Table 1 (annotated)
		
		hashing function *“mapping function”*
		DISTRIBUTOR HASH FUNCTION
		`unsigned short hash(unsigned char *hdr, short lagSize)`
		`{`
		`short i;`
		`unsigned short hash=178; // initialization value`
		`for (i=0; i<4; i++)          // hdr is 4 bytes length`
		`    hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);`
		`return (hash % lagSize)`
		`}`
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		For example, Solomon discloses applying a distributor hash function to the frame information which includes determining a number of the plurality of physical links.
		Solomon at [0024] (“In another embodiment, switching the data packets includes mapping the data packets to the selected port responsively to the label. Additionally or alternatively, map-ping the data packets includes applying a hashing function to the label so as to determine a number of the selected port, and choosing the label includes applying an inverse of the hashing function to the number of the selected port.”)

No.	’740 Patent Claim 10	Ghosh
		Solomon at [0048] (“The mapping function typically uses MPLS label 52 for mapping, since the MPLS label uniquely identifies MPLS tunnel 28, and it is required that all MPLS packets belonging to the same tunnel be switched through the same physical port 24. Additionally or alternatively, the mapping function uses a "PW" label (pseudo wire label, formerly known as a virtual connection, or VC label), which is optionally added to MPLS header 50. The PW label com-prises information that the egress node requires for deliver-ing the packet to its destination, and is optionally added during the encapsulation of MPLS packets. Additional details regarding the VC label can be found in an IETF draft by Martini et al. entitled "Encapsulation Methods for Trans-port of Ethernet Frames Over IP/MPLS Networks" (IETF draft-ietf-pwe3-ethernet-encap-07.txt, May, 2004), which is incorporated herein by reference. In some embodiments, mapper 34 applies a hashing function to the MPLS and/or PW label, as will be described below.”) Solomon at [0059] (“In this method, the mapping function used by mapper 34 of switch A is a hashing function. Various hashing functions are known in the art, and any suitable hashing function may be used in mapper 34. Since the hashing operation is performed for each packet, it is desirable to have a hashing function that is computationally simple.”) Solomon at [0060] (“As mentioned above, the hashing function typically hashes the value of MPLS label 52 to determine the selected physical port, as the MPLS label uniquely identifies tunnel 28. For example, the following hashing function may be used by mapper 34: Selected port number=l+((MPLS label) mod N), wherein N denotes the number of physical Ethernet ports in LAG group 25, and "mod" denotes the modulus operator. Assuming the values of MPLS labels are distrib-uted uniformly over a certain range, this function achieves a uniform distribution of port allocations for the different MPLS labels. It can also be seen that all packets carrying the same MPLS label (in other words-belonging to the same MPLS tunnel) will be mapped to the same physical port.”) Solomon at [0065] (“Mapper 34 of switch A maps each received packet to the selected physical port of LAG group 25 using the hashing function, at a hashing step 90. Mapper 34 extracts the MPLS label from each received packet and uses the hashing function to calculate the serial number of the selected physical port, which was selected by the CAC processor at step 82.

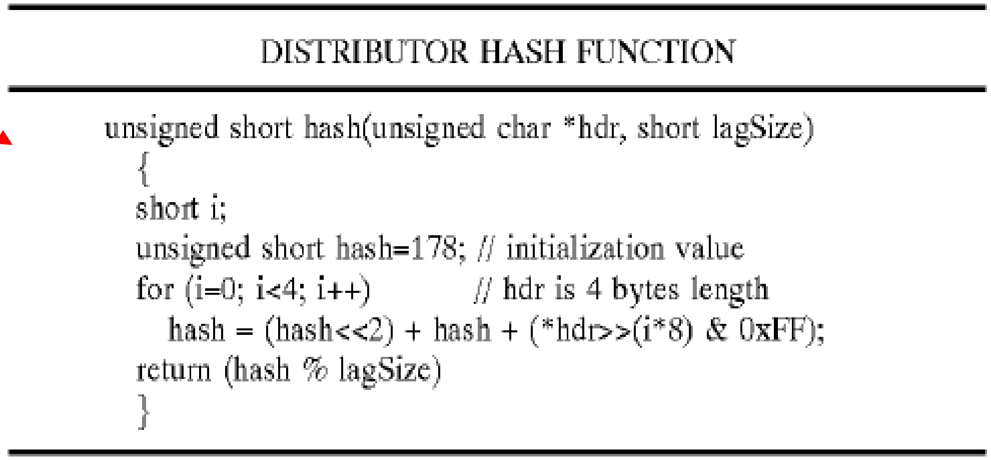
No.	’740 Patent Claim 10	Ghosh
		Following the numerical example given above, the mapper extracts MPLS label=65647 from the packet. Substituting this value and N=3 into the hashing function gives: Selected port number=1+(65647 mod 3)=2, which is indeed the port number selected in the example above.”)
10[b]	applying the hashing function to the at least one of the frame attributes to produce a hashing key,	Ghosh discloses applying the hashing function to the at least one of the frame attributes to produce a hashing key. For example, Ghosh discloses using a port channel map table to select the physical links through which a frame is transmitted. A person of ordinary skill in the art would further understand that a hashing function is a standard algorithm used to route data traffic that involves information specific to the system architecture, including involving the specific information of the data frame. *See supra* Claim 1 Ghosh at [0004] (“Neighboring nodes in a fibre channel network are typically interconnected through multiple physical links. For example, a local fibre channel switch may be connected to a remote fibre channel switch through four physical links. In many instances, it may be beneficial to aggregate some of the physical links into logical links. That is, multiple physi-cal links can be combined to form a logical interface to provide higher aggregate bandwidth, load balancing, and link redundancy. When a frame is being transmitted over a logical link, it does not matter what particular physical link is being used as long as all the frames of a given flow are transmitted through the same link. If a constituent physical link goes down, the logical link can still remain operational.”) Ghosh at [0054] (“A database such as a port channel map table is used store the links included in a port channel. The links and/or port information is entered into the table based on the order in which they were brought up. The port channel map table is used to select the physical link through which a frame is transmitted. In conventional implementations, the order of the peer

No.	’740 Patent Claim 10	Ghosh
		ports listed in the port channel map table can be different for both the local switch 701 and the remote switch 703.”) Ghosh at [0055] (“Based on this scheme, frames for a given flow are transmitted through the same physical port of the port channel. However, unless there is proper synchronization of the port channel map tables at the two ends of a port channel, it is possible that requests and responses for the same flow are carried over two different physical links. For example, requests may be carried over link 721 while responses are carried over 723. This is undesirable for port channels as it affects applications like write acceleration that assume the traffic for a given flow is carried over the same physical link in both directions.”) Ghosh at [0056] (“According to various embodiments of the present invention, links are brought up in order. A first link is selected for bring up. No effort is made to bring up other links is attempted until the exchange associated with the first link is completed. Consequently, port channel map tables at the local switch 701 and at the remote switch 703 are consistent. Port channel map table entries are synchronized at both ends so that all frames for a given flow are carried over the same physical link in both directions after the exchanges are completed.”) Ghosh at [0061] (“According to one embodiment, the routing appli-cation is configured to provide credits to a sender upon recognizing that a frame has been forwarded to a next hop. A utility application can be configured to track the number of buffers and the number of credits used. A domain manager application can be used to assign domains in the fibre channel storage area network. Various supervisor applica-tions may also be configured to provide functionality such as flow control, credit management, and quality of service (QoS) functionality for various fibre channel protocol layers.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Ghosh in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[b] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references.

No.	’740 Patent Claim 10	Ghosh
		For example, Bruckman discloses applying a distributor hash function to the frame information which includes determining a number of the plurality of physical links. Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor.

No.	’740 Patent Claim 10	Ghosh
		Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”) Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I: Bruckman at Table 1 (annotated)

No.	’740 Patent Claim 10	Ghosh

		hashing function *“mapping function”*
		
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		  hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		For example, Alexander discloses applying a distributor hash function to the frame information which includes determining a hash key based on packet information.
		Alexander at 3:1-40 (“The hash function is preferably selected such that suc-cessive application of the hash function to all source and destination addresses expected to be seen by the Ethernet switch will produce a lowest value hash key, a highest value hash key, and a group of hash keys having intermediate values distributed evenly between the lowest and highest values.

No.	’740 Patent Claim 10	Ghosh
		The distribution table contains a separate port identifier look-up table for each aggregated grouping of outgoing ports. Advantageously, the hash key is an N bit hash key; and, each port identifier look-up table contains 2^N entries occupying 2^N consecutive locations, with each entry being an identifier of a particular one of the physical outgoing ports. Identifiers for particular outgoing ports are retrieved from the distribution table by extracting first and second N bit hash keys which form part of the retrieved destination and source address contexts respectively. The hash keys are combined to form an N bit connection identifier. The port identifier look-up table corresponding to the aggregated grouping represented by the retrieved destination address is selected, and the entry at the table location corresponding to the value of the N bit connection identifier is retrieved. If the address look-up table does not contain a destination address corresponding to the extracted destination address then first and second hash keys are produced by applying a hash function to the extracted source and destination addresses respectively. The hash keys are combined to form an N bit connection identifier. The incoming port on which the packet containing the extracted source address was received is identified. All of the aggregated groupings are scanned to identify all outgoing ports to which packets may be directed from the incoming port on which the packet was received. For each one of those outgoing ports, the port identifier look-up table corresponding to the aggregated grouping containing that outgoing port is selected, the entry at the table location corresponding to the value of the N bit connection identifier is retrieved, and the received packet is queued for outgoing transmission on the outgoing port corresponding to the retrieved entry.”) Alexander at 5:10-35 (“If a packet arrives bearing a source Ethernet MAC address that was not found in look-up table 12 by address resolution unit 10, learning function 16 is invoked to update look-up table 12 with the new address (i.e. processing branches along the "No" exit from FIG. 2, block 36). Learning function 16 first computes a hash function on the source Ethernet MAC address, generating an N-bit hash key ("partial connection identifier") from the 48-bit MAC address, where N is some small integer in the range of 3 to 8 (FIG. 2, block 38). The physical port on which the packet arrived is then determined. If the physical port is found to be associated with an aggregate group (i.e., it is one of a set of ports that have been bound into a single logical port), then the logical identifier assigned to the aggregate group is also

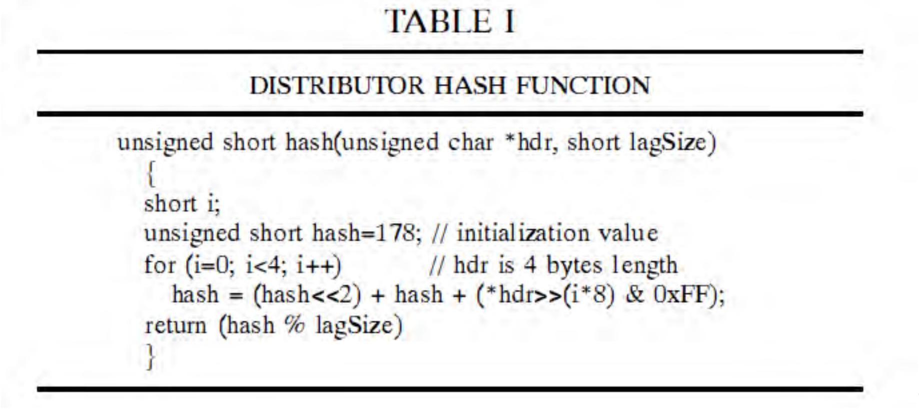
No.	’740 Patent Claim 10	Ghosh
		determined. The hash key is then stored into address look-up table 12 in conjunction with the actual Ethernet MAC address and the port identifier (FIG. 2, block 40). The physical port identifier is used if the port is not part of an aggregate group (i.e. if processing branched along the "No" exit from block 30 and through block 32), while the logical identifier is used for ports that have been aggregated (i.e. if processing branched along the "Yes" exit from block 30 and through block 34). The hash key and port identifier are considered to form the "context" for the given MAC address.”) Alexander at 5:36-46 (“The hash function should be selected to ensure an even distribution of hash key values over the range of MAC addresses that are expected to be seen by the Ethernet switch. As a specific example, the EXACT™ Ethernet switch system employs an exclusive-OR based hash function, wherein the 48-bit MAC address is divided into 16-bit blocks, which are then exclusive-ORed together to form a single 16-bit number; the 3 least significant bits (LSBs) of this number are taken to produce a 3-bit hash key. Other schemes such as CRC-based or checksum-based hashes may also be used.”) Alexander at 6:49-65 (“If the context information for the destination address indicates, however, that the target is an aggregate group (i.e. if processing branches along the "Yes" exit from FIG. 2, block 42) then the logical identifier assigned to the aggregate group is retrieved and is used to select the proper look-up table contained within the distribution table data structure. The hash keys (partial connection identifiers) stored into the contexts for the source and destination MAC addresses are obtained from address resolution unit 10 and combined to generate a "connection identifier" with the same number of bits (FIG. 2, block 44). (In the EXACT™ Ethernet switch, a Boolean exclusive-OR operation is used to combine the hash keys without increasing the number of bits.) This connection identifier is then used to index into the selected look-up table, and finally retrieve an actual physical port index on which the packet must be transmitted (FIG. 2, block 46).”)
10[c]	calculating a modulo of a division operation	Ghosh discloses calculating a modulo of a division operation of the hashing key by the hashing size.

No.	’740 Patent Claim 10	Ghosh
	of the hashing key by the hashing size, and	For example, Ghosh discloses using a port channel map table to select the physical links through which a frame is transmitted. A person of ordinary skill in the art would understand that a hashing function is a standard algorithm used to route data traffic that involves information specific to the system architecture and includes calculations of the generated values. *See supra* Claim 1 Ghosh at [0004] (“Neighboring nodes in a fibre channel network are typically interconnected through multiple physical links. For example, a local fibre channel switch may be connected to a remote fibre channel switch through four physical links. In many instances, it may be beneficial to aggregate some of the physical links into logical links. That is, multiple physi-cal links can be combined to form a logical interface to provide higher aggregate bandwidth, load balancing, and link redundancy. When a frame is being transmitted over a logical link, it does not matter what particular physical link is being used as long as all the frames of a given flow are transmitted through the same link. If a constituent physical link goes down, the logical link can still remain operational.”) Ghosh at [0054] (“A database such as a port channel map table is used store the links included in a port channel. The links and/or port information is entered into the table based on the order in which they were brought up. The port channel map table is used to select the physical link through which a frame is transmitted. In conventional implementations, the order of the peer ports listed in the port channel map table can be different for both the local switch 701 and the remote switch 703.”) Ghosh at [0055] (“Based on this scheme, frames for a given flow are transmitted through the same physical port of the port channel. However, unless there is proper synchronization of the port channel map tables at the two ends of a port channel, it is possible that requests and responses for the same flow are carried over two different physical links. For example, requests may be carried over link 721 while responses are carried over 723. This is undesirable for port channels as it affects applications like write acceleration that assume the traffic for a given flow is carried over the same physical link in both directions.”)

No.	’740 Patent Claim 10	Ghosh
		Ghosh at [0056] (“According to various embodiments of the present invention, links are brought up in order. A first link is selected for bring up. No effort is made to bring up other links is attempted until the exchange associated with the first link is completed. Consequently, port channel map tables at the local switch 701 and at the remote switch 703 are consistent. Port channel map table entries are synchronized at both ends so that all frames for a given flow are carried over the same physical link in both directions after the exchanges are completed.”) Ghosh at [0061] (“According to one embodiment, the routing appli-cation is configured to provide credits to a sender upon recognizing that a frame has been forwarded to a next hop. A utility application can be configured to track the number of buffers and the number of credits used. A domain manager application can be used to assign domains in the fibre channel storage area network. Various supervisor applica-tions may also be configured to provide functionality such as flow control, credit management, and quality of service (QoS) functionality for various fibre channel protocol layers.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Ghosh in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[c] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses distributing data frames over physical links and traces based on a hash function involving a division operation (%). Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0025] (“Typically, setting the protection policy includes determining a maximum number of the physical links that may fail while the logical link continues to provide at least the

No.	’740 Patent Claim 10	Ghosh
		guaranteed bandwidth for the connection. In one embodiment, the guaranteed bandwidth is a bandwidth B, and the plurality of physical links consists of N links, and the maximum number is an integer P, and the link bandwidth allocated to each of the links is no less than B/(N-P). Conveying the data may further include managing the transmission of the data responsively to an actual number X of the physical links that have failed so that the guaranteed bandwidth on each of the links is limited to B/(N-X), X P, and an excess bandwidth on the physical links over the guaranteed bandwidth is available for other connections.”) Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”) Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:

No.	’740 Patent Claim 10	Ghosh

		
		TABLE I
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		”)
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		Bruckman at [0067] (“A similar problem may arise if there is a failure in a link in an aggregation group or in one of a number of line cards serving the aggregation group. In this case, to maintain the bandwidth allocation B made by CAC 44, each of the remaining links in the group must now carry, on average, B/(N-M) traffic, wherein M is the number of links in the group that are out of service. If only BIN has been allocated to each link, the remaining active links may not have sufficient bandwidth to continue to provide the bandwidth that has been guaranteed to the connections that they are required to carry. A similar problem arises with respect to loading of traces 52. For example, if there is a failure in LC2 or in one of links 30 in group 36 that connect to LC2, the trace connecting the multiplexer 50 in LCl will have to carry a substantially larger share of the bandwidth, or even all of the bandwidth, that is allocated to the connection in ques-tion.”)

No.	’740 Patent Claim 10	Ghosh
		Bruckman at [0068] (“FIG. 3 is a flow chart that schematically illustrates a method for dealing with these problems of fluctuating bandwidth requirements, in accordance with an embodiment of the present invention. In order to provide sufficient bandwidth for failure protection, CAC 44 uses a safety margin based on a protection parameter P, which is assigned at a protection setting step 60. P represents the maximum number of links in the group that can be out of service while still permitting the aggregation group to provide a given connection with the bandwidth that has been guaranteed to the connection. CAC 44 will then allocate at least B/(N-P) bandwidth to each link in the group, so that if P links fail, the group still provides total bandwidth of (N-P)*B/(N-P)= B. Setting P=l is equivalent to 1:N protection, so that the group will be unaffected by failure of a single link. In the example of group 36, shown in FIG. 2, setting P=2 will give both facility and equipment protection, i.e., the group will be unaffected not only by failure of a link, but also by failure of one of line cards 34. In the extreme case, in which P=N-1, CAC 44 will allocate the full bandwidth B on each link in the group.”) As another example, Singh discloses determining a ratio between the number of ingress and egress links and the number of links carrying data to the backplane and using a modulo to correspond to the channel’s link number. Singh at 9:30-43 (“The ratio between the number of line ingress links and the number of links carrying data to the backplane gives the backplane speedup for the system. In this example, there are 10 ingress links into the MS and 20 links (2 backplane channels) carrying that data to the backplane. This gives a backplane speedup of 2x. As another example, with 8 ingress links and 12 backplane links, there is a speedup of 1.5x. It should be noted that in addition to the backplane speedup, there is also an ingress/egress speedup. With 10 ingress links capable of carrying 2 Gbps each of raw data, this presents a 20 Gbps interface to the MS. An OC-192 only has approximately 10 Gbps worth of data. Taking into account cell overhead and cell quantization inefficiencies, there still remains excess capacity in the links.”) Singh at 11:29-38 (“FIG. 9 is a diagram illustrating link to channel assignments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the

No.	’740 Patent Claim 10	Ghosh
		speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a channel, whether it is a backplane channel or an ingress/egress channel. There is no logical relationship between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.”) Singh at 15:15-39 (“The number of crossbars that are required in a system is dependent on how many links are being used to create the backplane channels. There should be an even number of crossbars and they would be divided evenly across the switch cards. The following equation, for most cases, pro-vides the correct number of crossbars: # of Crossbars=(# links per ingress channelx# of ingress channels per portx# of port cardsx speedup)/32. For the 8x8 configuration, the # of crossbars should be multiplied by (4x# of iMS)/(# backplane channels per port card). The number of port cards should be rounded up to the nearest supported configuration, i.e. 8, 16, or 32. The speedup in the case of crossbars should be the fractional speedup that is desired. Example to determine the number of arbiters and cross-bars for the following system: 4 channel port cards (40 Gbps) 8 links per channel 16 port cards Speedup=l.5 # of arbiters=(4x2x2)/2=8 # of crossbars=(8x4x16xl.5)/32=24. This would give 3crossbars per arbiter.”) Singh at 16:28-44 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 16x16 and 32x32 is the organization of the switchplane. The port card remains the same. Backplane channels 1 and 2 are used for the backplane connectivity. Ingress and egress links 30-39 on the MS would not be used

No.	’740 Patent Claim 10	Ghosh
		and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 16, correspond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”) Singh at 17:31-49 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 8x8 and 16x16 is the organi-zation of the switchplane. The port card remains the same. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Links 0-7 and 24-31 on the arbiters would not be used and would be powered off. Links 0-7 and 24-31 on the crossbars would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Backplane channels 1 and 2 are used for the backplane connectivity. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 8, corre-spond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”)
10[d]	selecting the first and second physical links responsively to the modulo.	Ghosh discloses selecting the first and second physical links responsively to the modulo. For example, Ghosh discloses using a port channel map table to select the physical links through which a frame is transmitted. A person of ordinary skill in the art would understand that a hashing function is a standard algorithm used to route data traffic that involves information specific to the system architecture and includes calculations of the generated values. A person of ordinary skill in the art would further understand that the physical links over which the frame is sent are determined based on the calculations of the hashing function.

No.	’740 Patent Claim 10	Ghosh
		Ghosh at [0004] (“Neighboring nodes in a fibre channel network are typically interconnected through multiple physical links. For example, a local fibre channel switch may be connected to a remote fibre channel switch through four physical links. In many instances, it may be beneficial to aggregate some of the physical links into logical links. That is, multiple physi-cal links can be combined to form a logical interface to provide higher aggregate bandwidth, load balancing, and link redundancy. When a frame is being transmitted over a logical link, it does not matter what particular physical link is being used as long as all the frames of a given flow are transmitted through the same link. If a constituent physical link goes down, the logical link can still remain operational.”) Ghosh at [0054] (“A database such as a port channel map table is used store the links included in a port channel. The links and/or port information is entered into the table based on the order in which they were brought up. The port channel map table is used to select the physical link through which a frame is transmitted. In conventional implementations, the order of the peer ports listed in the port channel map table can be different for both the local switch 701 and the remote switch 703.”) Ghosh at [0055] (“Based on this scheme, frames for a given flow are transmitted through the same physical port of the port channel. However, unless there is proper synchronization of the port channel map tables at the two ends of a port channel, it is possible that requests and responses for the same flow are carried over two different physical links. For example, requests may be carried over link 721 while responses are carried over 723. This is undesirable for port channels as it affects applications like write acceleration that assume the traffic for a given flow is carried over the same physical link in both directions.”) Ghosh at [0056] (“According to various embodiments of the present invention, links are brought up in order. A first link is selected for bring up. No effort is made to bring up other links is attempted until the exchange associated with the first link is completed. Consequently, port channel map tables at the local switch 701 and at the remote switch 703 are consistent. Port

No.	’740 Patent Claim 10	Ghosh
		channel map table entries are synchronized at both ends so that all frames for a given flow are carried over the same physical link in both directions after the exchanges are completed.”) Ghosh at [0061] (“According to one embodiment, the routing appli-cation is configured to provide credits to a sender upon recognizing that a frame has been forwarded to a next hop. A utility application can be configured to track the number of buffers and the number of credits used. A domain manager application can be used to assign domains in the fibre channel storage area network. Various supervisor applica-tions may also be configured to provide functionality such as flow control, credit management, and quality of service (QoS) functionality for various fibre channel protocol layers.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Ghosh in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[d] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses distributing data frames over physical links and traces based on a hash function involving a division operation (%). Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0025] (“Typically, setting the protection policy includes determining a maximum number of the physical links that may fail while the logical link continues to provide at least the guaranteed bandwidth for the connection. In one embodiment, the guaranteed bandwidth is a bandwidth B, and the plurality of physical links consists of N links, and the maximum number is an integer P, and the link bandwidth allocated to each of the links is no less than B/(N-P). Conveying the data may further include managing the transmission of the data responsively to an actual number X of the physical links that have failed so that the guaranteed bandwidth on

<table>
<tr><th>No.</th><th>’740 Patent Claim 10</th><th>Ghosh</th></tr>
<tr><td></td><td></td><td>

each of the links is limited to B/(N-X), X P, and an excess bandwidth on the physical links over the guaranteed bandwidth is available for other connections.”)

Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”)

Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:

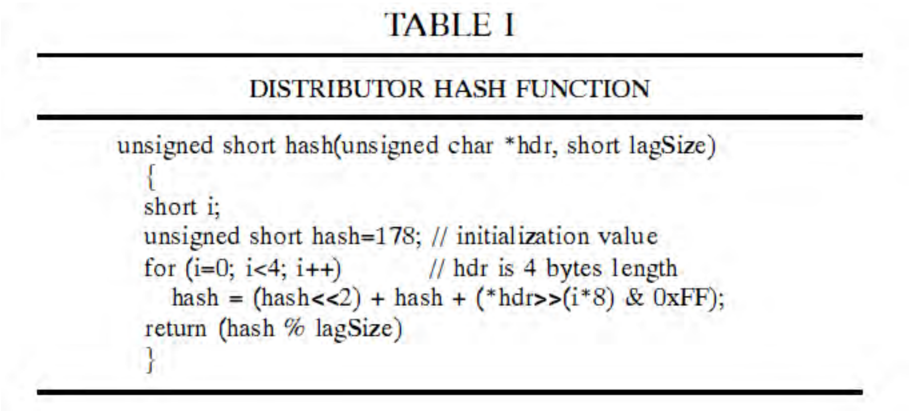


TABLE I

DISTRIBUTOR HASH FUNCTION

```
unsigned short hash(unsigned char *hdr, short lagSize)
  {
  short i;
  unsigned short hash=178; // initialization value
  for (i=0; i<4; i++)          // hdr is 4 bytes length
     hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
  return (hash % lagSize)
  }
```

”)

</td></tr>
</table>

No.	’740 Patent Claim 10	Ghosh
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”) Bruckman at [0067] (“A similar problem may arise if there is a failure in a link in an aggregation group or in one of a number of line cards serving the aggregation group. In this case, to maintain the bandwidth allocation B made by CAC 44, each of the remaining links in the group must now carry, on average, B/(N-M) traffic, wherein M is the number of links in the group that are out of service. If only BIN has been allocated to each link, the remaining active links may not have sufficient bandwidth to continue to provide the bandwidth that has been guaranteed to the connections that they are required to carry. A similar problem arises with respect to loading of traces 52. For example, if there is a failure in LC2 or in one of links 30 in group 36 that connect to LC2, the trace connecting the multiplexer 50 in LCl will have to carry a substantially larger share of the bandwidth, or even all of the bandwidth, that is allocated to the connection in ques-tion.”) Bruckman at [0068] (“FIG. 3 is a flow chart that schematically illustrates a method for dealing with these problems of fluctuating bandwidth requirements, in accordance with an embodiment of the present invention. In order to provide sufficient bandwidth for failure protection, CAC 44 uses a safety margin based on a protection parameter P, which is assigned at a protection setting step 60. P represents the maximum number of links in the group that can be out of service while still permitting the aggregation group to provide a given connection with the bandwidth that has been guaranteed to the connection. CAC 44 will then allocate at least B/(N-P) bandwidth to each link in the group, so that if P links fail, the group still provides total bandwidth of (N-P)*B/(N-P)= B. Setting P=l is equivalent to 1:N protection, so that the group will be unaffected by failure of a single link. In the example of group 36, shown in FIG. 2, setting P=2 will give both facility and equipment protection, i.e., the group will be unaffected not only by failure of a link, but also by failure of one of line cards 34. In the extreme case, in which P=N-1, CAC 44 will allocate the full bandwidth B on each link in the group.”)

No.	’740 Patent Claim 10	Ghosh
		As another example, Singh discloses determining a ratio between the number of ingress and egress links and the number of links carrying data to the backplane and using a modulo to correspond to the channel’s link number. Singh at 9:30-43 (“The ratio between the number of line ingress links and the number of links carrying data to the backplane gives the backplane speedup for the system. In this example, there are 10 ingress links into the MS and 20 links (2 backplane channels) carrying that data to the backplane. This gives a backplane speedup of 2x. As another example, with 8 ingress links and 12 backplane links, there is a speedup of 1.5x. It should be noted that in addition to the backplane speedup, there is also an ingress/egress speedup. With 10 ingress links capable of carrying 2 Gbps each of raw data, this presents a 20 Gbps interface to the MS. An OC-192 only has approximately 10 Gbps worth of data. Taking into account cell overhead and cell quantization inefficiencies, there still remains excess capacity in the links.”) Singh at 11:29-38 (“FIG. 9 is a diagram illustrating link to channel assignments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a channel, whether it is a backplane channel or an ingress/egress channel. There is no logical relationship between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.”) Singh at 15:15-39 (“The number of crossbars that are required in a system is dependent on how many links are being used to create the backplane channels. There should be an even number of crossbars and they would be divided evenly across the switch cards. The following equation, for most cases, pro-vides the correct number of crossbars: # of Crossbars=(# links per ingress channelx# of ingress channels per portx# of port cardsx speedup)/32.

No.	’740 Patent Claim 10	Ghosh
		For the 8x8 configuration, the # of crossbars should be multiplied by (4x# of iMS)/(# backplane channels per port card). The number of port cards should be rounded up to the nearest supported configuration, i.e. 8, 16, or 32. The speedup in the case of crossbars should be the fractional speedup that is desired. Example to determine the number of arbiters and cross-bars for the following system: 4 channel port cards (40 Gbps) 8 links per channel 16 port cards Speedup=l.5 # of arbiters=(4x2x2)/2=8 # of crossbars=(8x4x16xl.5)/32=24. This would give 3crossbars per arbiter.”) Singh at 16:28-44 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 16x16 and 32x32 is the organization of the switchplane. The port card remains the same. Backplane channels 1 and 2 are used for the backplane connectivity. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 16, correspond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”) Singh at 17:31-49 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 8x8 and 16x16 is the organi-zation of the switchplane. The port card remains the same. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Links 0-7 and 24-31 on the arbiters would not be used and would be powered off. Links 0-7 and 24-31 on the crossbars would not

No.	’740 Patent Claim 10	Ghosh
		be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Backplane channels 1 and 2 are used for the backplane connectivity. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 8, corre-spond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”)

No.	’740 Patent Claim 11	Ghosh
11	The method according to claim 10, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo.	Ghosh discloses the method according to claim 10, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo. For example, Ghosh discloses using a port channel map table to select the physical links through which a frame is transmitted. A person of ordinary skill in the art would understand that a hashing function is a standard algorithm used to route data traffic that involves information specific to the system architecture and includes calculations of the generated values. A person of ordinary skill in the art would further understand that the physical links over which the frame is sent are determined based on the calculations of the hashing function, including the subsets of bits in a binary representation of the modulo. *See supra* Claim 10 Ghosh at [0004] (“Neighboring nodes in a fibre channel network are typically interconnected through multiple physical links. For example, a local fibre channel switch may be connected to a remote fibre channel switch through four physical links. In many instances, it may be

No.	’740 Patent Claim 11	Ghosh
		beneficial to aggregate some of the physical links into logical links. That is, multiple physi-cal links can be combined to form a logical interface to provide higher aggregate bandwidth, load balancing, and link redundancy. When a frame is being transmitted over a logical link, it does not matter what particular physical link is being used as long as all the frames of a given flow are transmitted through the same link. If a constituent physical link goes down, the logical link can still remain operational.”) Ghosh at [0054] (“A database such as a port channel map table is used store the links included in a port channel. The links and/or port information is entered into the table based on the order in which they were brought up. The port channel map table is used to select the physical link through which a frame is transmitted. In conventional implementations, the order of the peer ports listed in the port channel map table can be different for both the local switch 701 and the remote switch 703.”) Ghosh at [0055] (“Based on this scheme, frames for a given flow are transmitted through the same physical port of the port channel. However, unless there is proper synchronization of the port channel map tables at the two ends of a port channel, it is possible that requests and responses for the same flow are carried over two different physical links. For example, requests may be carried over link 721 while responses are carried over 723. This is undesirable for port channels as it affects applications like write acceleration that assume the traffic for a given flow is carried over the same physical link in both directions.”) Ghosh at [0056] (“According to various embodiments of the present invention, links are brought up in order. A first link is selected for bring up. No effort is made to bring up other links is attempted until the exchange associated with the first link is completed. Consequently, port channel map tables at the local switch 701 and at the remote switch 703 are consistent. Port channel map table entries are synchronized at both ends so that all frames for a given flow are carried over the same physical link in both directions after the exchanges are completed.”) Ghosh at [0061] (“According to one embodiment, the routing appli-cation is configured to provide credits to a sender upon recognizing that a frame has been forwarded to a next hop. A

No.	’740 Patent Claim 11	Ghosh
		utility application can be configured to track the number of buffers and the number of credits used. A domain manager application can be used to assign domains in the fibre channel storage area network. Various supervisor applica-tions may also be configured to provide functionality such as flow control, credit management, and quality of service (QoS) functionality for various fibre channel protocol layers.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Ghosh in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 11 of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses distributing data frames over physical links and traces based on a division operation in the hash function, involving specific byte lengths of the frame information. Bruckman at Figure 2 (annotated)

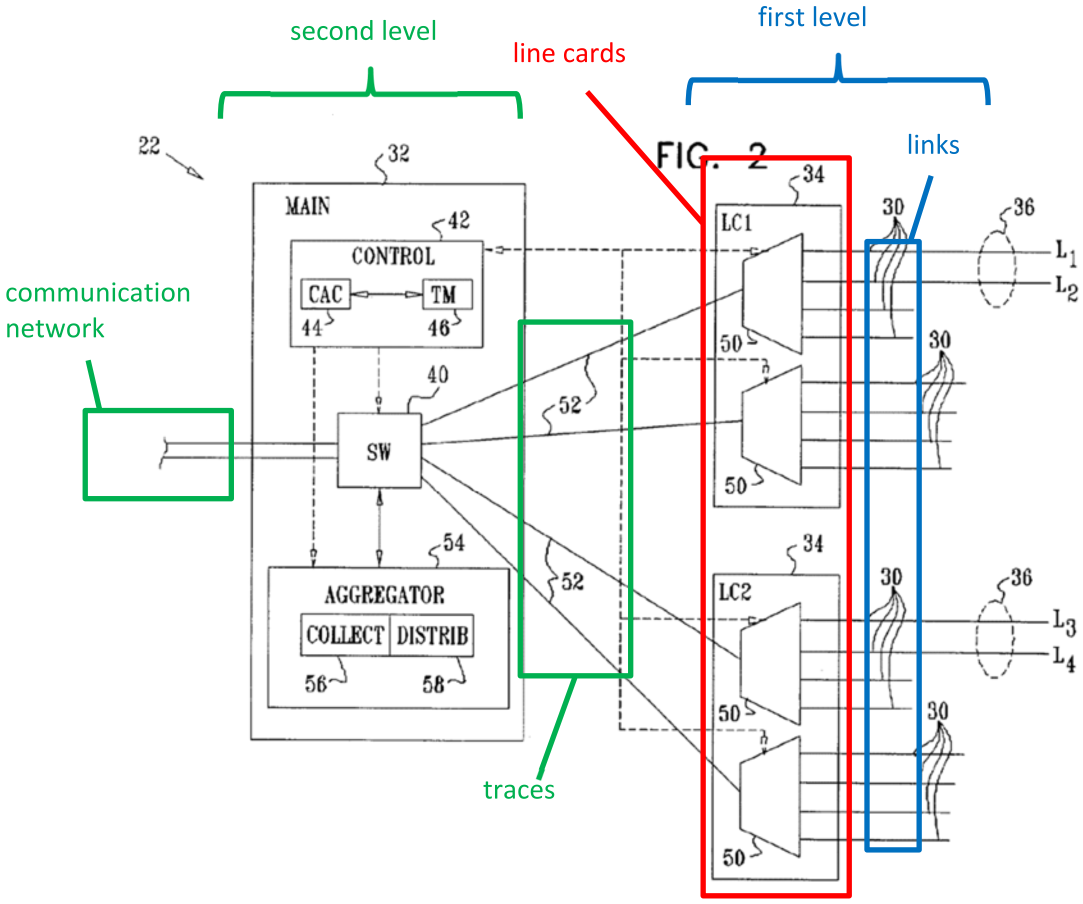
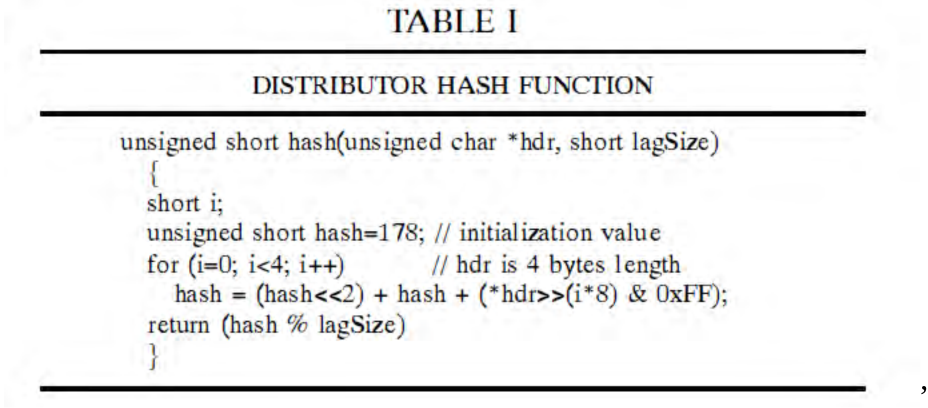
<table>
<tr><th>No.</th><th>’740 Patent Claim 11</th><th>Ghosh</th></tr>
<tr><td></td><td></td><td>


FIG. 2

Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:
</td></tr>
</table>

No.	’740 Patent Claim 11	Ghosh

		
		TABLE I
		DISTRIBUTOR HASH FUNCTION
		`unsigned short hash(unsigned char *hdr, short lagSize)`
		`  {`
		`  short i;`
		`  unsigned short hash=178; // initialization value`
		`  for (i=0; i<4; i++)          // hdr is 4 bytes length`
		`    hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);`
		`  return (hash % lagSize)`
		`  }`
		”)
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		As another example, Solomon discloses using a subset of bits to encode for the selected physical port, involving specific byte lengths of the frame information.
		Solomon at [0054] (“Having selected a physical port, RSVP-TE proces-sor 30 of switch A now generates a suitable MPLS label, at a label generation step 64. The preceding node upstream of switch A will subsequently attach this MPLS label to all MPLS packets transmitted through tunnel 28 to switch A. The label is assigned, in conjunction with the mapping function of mapper 34, so as to ensure that all MPLS packets carrying this label are switched through the physical port that was selected for this tunnel at step 62. For this purpose, RSVP-TE processor 30 of switch A dedicates a sub-set of the bits of MPLS label 52 to encode the serial number of the selected physical port. For example, the four least-signifi-cant bits of MPLS label 52 may be used for encoding the selected port number. This configuration is suitable for representing LAG

No.	’740 Patent Claim 11	Ghosh
		groups having up to 16 physical ports (N<16). The remaining bits of MPLS label 52 may be chosen at random or using any suitable method known in the art.”) Solomon at [0056] (“Mapper 34 of switch A maps the received packets belonging to tunnel 28 to the selected physical Ethernet port at a mapping step 70. For this purpose, mapper 34 extracts the MPLS label from each received packet and decodes the selected physical port number from the dedicated sub-set of bits, such as the four LSB, as described in step 64 above. The decoded value is used for mapping the packet to the selected physical port, which was allocated by the CAC processor at step 62 above. In the four-bit example described above, the mapping function may be written explicitly as: Selected port number=((MPLS label) and (0x0000F)), wherein "and" denotes the "bitwise and" operator.”)

No.	’740 Patent Claim 12	Ghosh
12	The method according to claim 1, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC)	Ghosh discloses the method according to claim 1, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port. For example, Ghosh discloses data frames with identifying information in a flow that are transmitted. A person of ordinary skill in the art would understand that such frame information could include at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port, which are used to route the frame according to the port channel map table. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Ghosh is found to not meet this limitation, wherein the at least one of the frame attributes comprises at

No.	’740 Patent Claim 12	Ghosh
	address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port.	least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port would have been obvious to a person having ordinary skill in the art, as explained below. *See supra* Claim 1 Ghosh at [0004] (“Neighboring nodes in a fibre channel network are typically interconnected through multiple physical links. For example, a local fibre channel switch may be connected to a remote fibre channel switch through four physical links. In many instances, it may be beneficial to aggregate some of the physical links into logical links. That is, multiple physi-cal links can be combined to form a logical interface to provide higher aggregate bandwidth, load balancing, and link redundancy. When a frame is being transmitted over a logical link, it does not matter what particular physical link is being used as long as all the frames of a given flow are transmitted through the same link. If a constituent physical link goes down, the logical link can still remain operational.”) Ghosh at [0054] (“A database such as a port channel map table is used store the links included in a port channel. The links and/or port information is entered into the table based on the order in which they were brought up. The port channel map table is used to select the physical link through which a frame is transmitted. In conventional implementations, the order of the peer ports listed in the port channel map table can be different for both the local switch 701 and the remote switch 703.”) Ghosh at [0055] (“Based on this scheme, frames for a given flow are transmitted through the same physical port of the port channel. However, unless there is proper synchronization of the port channel map tables at the two ends of a port channel, it is possible that requests and responses for the same flow are carried over two different physical links. For example,

No.	’740 Patent Claim 12	Ghosh
		requests may be carried over link 721 while responses are carried over 723. This is undesirable for port channels as it affects applications like write acceleration that assume the traffic for a given flow is carried over the same physical link in both directions.”) Ghosh at [0056] (“According to various embodiments of the present invention, links are brought up in order. A first link is selected for bring up. No effort is made to bring up other links is attempted until the exchange associated with the first link is completed. Consequently, port channel map tables at the local switch 701 and at the remote switch 703 are consistent. Port channel map table entries are synchronized at both ends so that all frames for a given flow are carried over the same physical link in both directions after the exchanges are completed.”) Ghosh at [0061] (“According to one embodiment, the routing appli-cation is configured to provide credits to a sender upon recognizing that a frame has been forwarded to a next hop. A utility application can be configured to track the number of buffers and the number of credits used. A domain manager application can be used to assign domains in the fibre channel storage area network. Various supervisor applica-tions may also be configured to provide functionality such as flow control, credit management, and quality of service (QoS) functionality for various fibre channel protocol layers.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Ghosh in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 12 of Exhibit E-3 renders the claim, including the present limitation, obvious. Below is one such example. For example, Bruckman discloses frame information including header fields, source addresses, destination addresses, ports, etc. Bruckman specifically discloses Ethernet frame information including source MAC address, destination MAC address, reception port, type of destination address, Ethernet Length/Type value, and higher layer protocol information. Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of

No.	’740 Patent Claim 12	Ghosh
		the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria:

<table>
<tr><th>No.</th><th>’740 Patent Claim 12</th><th>Ghosh</th></tr>
<tr><td></td><td></td><td>

[0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link.
[0060] The hash function uses all the specified input data from the frame headers.
[0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values
[0062] The hash function generates very different hash values for similar data.”)

Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:

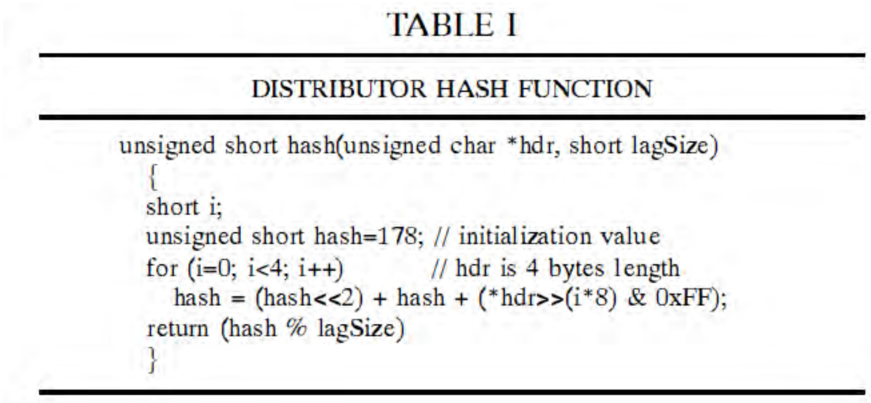


TABLE I

DISTRIBUTOR HASH FUNCTION

```
unsigned short hash(unsigned char *hdr, short lagSize)
  {
  short i;
  unsigned short hash=178; // initialization value
  for (i=0; i<4; i++)          // hdr is 4 bytes length
     hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
  return (hash % lagSize)
  }
```

”)

Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)

</td></tr>
</table>

No.	’740 Patent Claim 13	Ghosh
13[preamble]	A method for communication, comprising:	Ghosh discloses a method for communication. *See supra at* 1[preamble].
13[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	Ghosh discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].
13[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Ghosh discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra at* 1[c].
13[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	Ghosh discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra at* 1[e].
13[d]	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and	Ghosh discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. *See supra at* 1[f].

No.	’740 Patent Claim 13	Ghosh
	a second physical link out of the second group; and	
13[e]	sending the data frame over the selected first and second physical links,	Ghosh discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].
13[f]	coupling the network node to the one or more interface modules and	Ghosh discloses coupling the network node to the one or more interface modules. *See supra at* 1[a].
13[g]	coupling each of the one or more interface modules to the communication network comprising	Ghosh discloses coupling each of the one or more interface modules to the communication network. *See supra at* 1[c].
13[h]	specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication	Ghosh discloses specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node. For example, Ghosh discloses bandwidth considerations prescribed by the physical links coupling the line cards to the network that may be increased by aggregation. A person of ordinary skill in the art would understand that such bandwidth consideration could include a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Ghosh is found to not meet this limitation, coupling each of the one or more interface modules to the communication network comprising specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of

No.	’740 Patent Claim 13	Ghosh
	network to the network node, and	a communication service provided by the communication network to the network node would have been obvious to a person having ordinary skill in the art, as explained below. Ghosh at [0004] (“Neighboring nodes in a fibre channel network are typically interconnected through multiple physical links. For example, a local fibre channel switch may be connected to a remote fibre channel switch through four physical links. In many instances, it may be beneficial to aggregate some of the physical links into logical links. That is, multiple physi-cal links can be combined to form a logical interface to provide higher aggregate bandwidth, load balancing, and link redundancy. When a frame is being transmitted over a logical link, it does not matter what particular physical link is being used as long as all the frames of a given flow are transmitted through the same link. If a constituent physical link goes down, the logical link can still remain operational.”) Ghosh at [0022] (“Switches in a fibre channel network are typically interconnected using multiple physical links. The physical links connecting a pair of switches allows transmission of data and control signals. In some instances, it is useful to aggregate multiple physical links into a logical link. Physi-cal links are also referred to herein as physical interfaces and channels while logical links are also referred to herein as logical interfaces and port channels. For example, a local switch may be connected to a remote switch through four physical links. Instead of having to transmit data through a particular physical link, the physical links can be aggregated to form one or more logical links. In one example, all four physical links are aggregated into a single logical link. Instead of having data transmitted through a particular physical link, the data can merely be transmitted over a particular logical link without regard to the particular physi-cal interface used. Aggregating physical links into a logical link allows for higher aggregated bandwidth, load balancing, and link redundancy. For example, if a particular physical link fails or is overloaded, data can still be transmitted over the logical link.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Ghosh in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references

No.	’740 Patent Claim 13	Ghosh
		identified in element 13[h] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses specifying certain committed, excess, and guaranteed bandwidths, including CIR, EIR, and PIR, respectively. Bruckman at [0013] (“Service level agreements between network service providers and customers commonly specify a certain com-mitted bandwidth, or committed information rate (CIR), which the service provider guarantees to provide to the customer at all times, regardless of bandwidth stress on the network. Additionally or alternatively, the agreement may specify an excess bandwidth, which is available to the customer when network traffic permits. The excess band-width is typically used by customers for lower-priority services, which do not require committed bandwidth. The network service provider may guarantee the customer a certain minimum excess bandwidth, or excess information rate (EIR), in order to avoid starvation of such services in case of bandwidth stress. In general, the bandwidth guaran-teed by a service provider, referred to as the peak informa-tion rate (PIR), may include either CIR, or EIR, or both CIR and EIR (in which case PIR=CIR+EIR). The term "guaran-teed bandwidth," as used in the context of the present patent application and in the claims, includes all these types of guaranteed bandwidth.”) As another example, Solomon discloses a service property of a guaranteed bandwidth, sometimes denoted as CIR-Committed Information Rate and PIR-Peak Information Rate. Solomon at [0023] (“In another embodiment, establishing the path includes receiving an indication of a requested service property of the flow, and selecting the port includes assign-ing the port to the flow so as to comply with the requested service property. In a disclosed embodiment, the requested service property includes at least one of a guaranteed bandwidth, a peak bandwidth and a class-of-service. Addi-tionally or alternatively, assigning the port includes selecting the port having a maximum available bandwidth out of the plurality of aggregated ports. Further additionally or alter-natively, assigning the port

No.	’740 Patent Claim 13	Ghosh
		includes selecting the port hav-ing a minimum available bandwidth out of the plurality of aggregated ports, which is still greater than or equal to the guaranteed bandwidth.”) Solomon at [0050] (“The method of FIG. 3 begins when the preceding node asks to establish a part of tunnel 28 (comprising one or more hops) for sending MPLS packets to MPLS/LAG switch 26 A. The preceding node requests and then receives the MPLS label, which it will subsequently attach to all packets that are sent to MPLS/LAG switch 26 labeledA. The preceding node sends downstream an RSVP-TE PATH mes-sage augmented with a LABEL_REQUEST object, as defined by RSVP-TE, to MPLS/LAG switch A, at a label requesting step 60. The PATH message typically comprises information regarding service properties that are requested for tunnel 28. The service properties may comprise a guar-anteed bandwidth (sometimes denoted CIR-Committed Information Rate) and a peak bandwidth (sometimes denoted PIR-Peak Information Rate), as well as a requested CoS (Class of Service-a measure of packet priority).”)
13[i]	allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements.	Ghosh discloses allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements. For example, Ghosh discloses distributing and load balancing frames over ports and interface circuitry in accordance with the bandwidth considerations. Ghosh at [0004] (“Neighboring nodes in a fibre channel network are typically interconnected through multiple physical links. For example, a local fibre channel switch may be connected to a remote fibre channel switch through four physical links. In many instances, it may be beneficial to aggregate some of the physical links into logical links. That is, multiple physi-cal links can be combined to form a logical interface to provide higher aggregate bandwidth, load balancing, and link redundancy. When a frame is being transmitted over a logical link, it does not matter what particular physical link is being used as long as all the frames of a given flow are transmitted through the same link. If a constituent physical link goes down, the logical link can still remain operational.”)

No.	’740 Patent Claim 13	Ghosh
		Ghosh at [0022] (“Switches in a fibre channel network are typically interconnected using multiple physical links. The physical links connecting a pair of switches allows transmission of data and control signals. In some instances, it is useful to aggregate multiple physical links into a logical link. Physi-cal links are also referred to herein as physical interfaces and channels while logical links are also referred to herein as logical interfaces and port channels. For example, a local switch may be connected to a remote switch through four physical links. Instead of having to transmit data through a particular physical link, the physical links can be aggregated to form one or more logical links. In one example, all four physical links are aggregated into a single logical link. Instead of having data transmitted through a particular physical link, the data can merely be transmitted over a particular logical link without regard to the particular physi-cal interface used. Aggregating physical links into a logical link allows for higher aggregated bandwidth, load balancing, and link redundancy. For example, if a particular physical link fails or is overloaded, data can still be transmitted over the logical link.”)

No.	’740 Patent Claim 14	Ghosh
14[preamble]	A method for connecting user ports to a communication network, comprising:	Ghosh discloses a method for connecting user ports to a communication network. For example, Ghosh discloses methods for connecting ports in a port channel to a fibre channel network. Ghosh at Abstract (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggregation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent gen-eration of port channel map tables.”) Ghosh at [0007] (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggre-gation of ports into port channels in a fibre channel

No.	’740 Patent Claim 14	Ghosh
		network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent generation of port channel map tables.”) Ghosh at [0008] (“In one embodiment, a method for aggregating ports in a fibre channel fabric is provided. It is determined that a plurality of local ports at a local fibre channel switch are compatible. Identifiers for the plurality of local ports are sent to a remote fibre channel switch. The remote fibre channel switch determines if a plurality of remote ports are compat-ible, the plurality of remote ports corresponding to the plurality of local ports. An indication that one or more of the remote physical ports are compatible is received. A port channel including one or more of the local ports correspond-ing to the compatible remote ports is created.”) Ghosh at [0009] (“In another embodiment, a fibre channel switch is provided. The fibre channel switch includes memory, a plurality of local ports, and a processor. The plurality of local ports are coupled to a remote fibre channel switch through a plurality of remote ports. The processor is con-figured to determine that a subset of the plurality of local ports at a local fibre channel switch are compatible and send identifiers for the subset of the plurality of local ports to a remote fibre channel switch. The remote fibre channel switch determines if a subset of the plurality of remote ports are compatible. The subset of the plurality of remote ports corresponds to the subset of the plurality of local ports.”) Ghosh at [0010] (“In another embodiment, a fibre channel network is described. The fibre channel network includes a local fibre channel switch and a remote fibre channel switch. The local fibre channel switch aggregates a compatible subset of the plurality of local ports and sends identifiers for the compat-ible subset of the plurality of local ports to the remote fibre channel switch. The remote fibre channel switch determines if a subset of the plurality of remote ports are compatible. The subset of the plurality of remote ports corresponds to the compatible subset of the plurality of local ports.”)

No.	’740 Patent Claim 14	Ghosh
		Ghosh at [0059] (“Line cards 803, 805, and 807 can communicate with an active supervisor 811 through interface circuitry 883, 885, and 887 and the backplane 815. According to various embodiments, each line card includes a plurality of ports that can act as either input ports or output ports for communication with external fibre channel network entities 851 and 853. The backplane 815 can provide a communi-cations channel for all traffic between line cards and super- visors. Individual line cards 803 and 807 can also be coupled to external fibre channel network entities 851 and 853 through fibre channel ports 843 and 847.”) Ghosh at [0060] (“External fibre channel network entities 851 and 853 can be nodes such as other fibre channel switches, disks, RAIDS, tape libraries, or servers. It should be noted that the switch can support any number of line cards and supervisors. In the embodiment shown, only a single supervisor is connected to the backplane 815 and the single supervisor communicates with many different line cards. The active supervisor 811 may be configured or designed to run a plurality of applications such as routing, domain manager, system manager, and utility applications.”)
14[a]	coupling the user ports to one or more user interface modules;	Ghosh discloses coupling the user ports to one or more user interface modules. *See supra at* 1[a].
14[b]	coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel,	Ghosh discloses coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel. *See supra at* 1[c], 3.
14[c]	at least one of said backplane traces being bi-directional and operative to communicate in both an	Ghosh discloses at least one of said backplane traces being bi-directional and operative to communicate in both an upstream direction and a downstream direction. *See supra at* 14[b], 1[d].

No.	’740 Patent Claim 14	Ghosh
	upstream direction and a downstream direction;	
14[d]	receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes;	Ghosh discloses receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes. *See supra at* 14[a], 1[e].
14[e]	for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces; and	Ghosh discloses for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces. *See supra at* 14[b], 1[f].
14[f]	sending the data frame over the selected backplane trace;	Ghosh discloses sending the data frame over the selected backplane trace. *See supra at* 14[e], 1[g].
14[g]	said sending comprising communicating along said at least one of said backplane traces.	Ghosh discloses said sending comprising communicating along said at least one of said backplane traces. *See supra at* 14[f], 1[h].

No.	’740 Patent Claim 15	Ghosh
15[preamble]	A method for connecting user ports to a communication network, comprising:	Ghosh discloses a method for connecting user ports to a communication network. *See supra at* 14[preamble].

No.	’740 Patent Claim 15	Ghosh
15[a]	coupling the user ports to one or more user interface modules;	Ghosh discloses coupling the user ports to one or more user interface modules. *See supra at* 14[a].
15[b]	coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel;	Ghosh discloses coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel. *See supra at* 14[b].
15[c]	receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes;	Ghosh discloses receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes. *See supra at* 14[d].
15[d]	for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces; and	Ghosh discloses for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces. *See supra at* 14[e].
15[e]	sending the data frame over the	Ghosh discloses sending the data frame over the selected backplane trace.

No.	’740 Patent Claim 15	Ghosh
	selected backplane trace,	*See supra at* 14[f].
15[f]	at least some of the backplane traces being aggregated into an Ethernet link aggregation (LAG) group.	Ghosh discloses at least some of the backplane traces being aggregated into an Ethernet link aggregation (LAG) group. *See supra at* 15[e], 4[f], 3.

No.	’740 Patent Claim 16	Ghosh
16	The method according to claim 14, wherein selecting the backplane trace comprises applying a hashing function to the at least one of the frame attributes.	Ghosh discloses the method according to claim 14, wherein selecting the backplane trace comprises applying a hashing function to the at least one of the frame attributes. *See supra* at 14, 9, 8.

No.	’740 Patent Claim 17	Ghosh
17[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Ghosh discloses apparatus for connecting a network node with a communication network. For example, Ghosh discloses an apparatus for connecting network entities with a network for data traffic transmission. Ghosh at Abstract (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggregation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels.

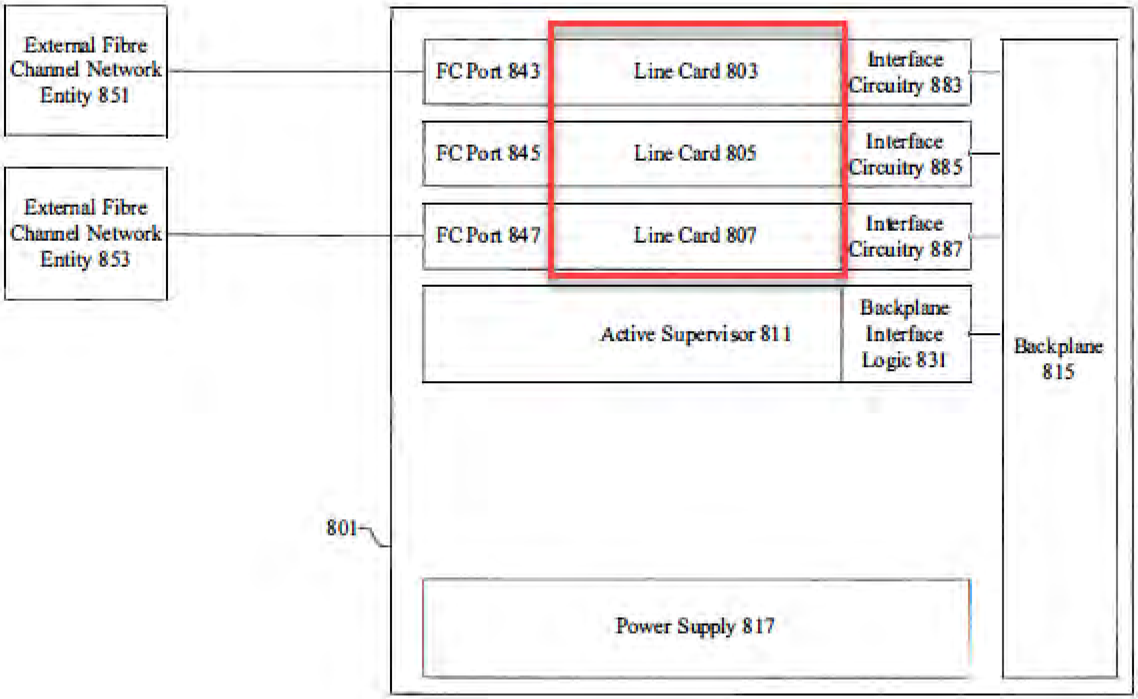
No.	’740 Patent Claim 17	Ghosh
		Ports can be aggregated synchronously to allow consistent gen-eration of port channel map tables.”) Ghosh at [0007] (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggre-gation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent generation of port channel map tables.”) Ghosh at [0008] (“In one embodiment, a method for aggregating ports in a fibre channel fabric is provided. It is determined that a plurality of local ports at a local fibre channel switch are compatible. Identifiers for the plurality of local ports are sent to a remote fibre channel switch. The remote fibre channel switch determines if a plurality of remote ports are compat-ible, the plurality of remote ports corresponding to the plurality of local ports. An indication that one or more of the remote physical ports are compatible is received. A port channel including one or more of the local ports correspond-ing to the compatible remote ports is created.”) Ghosh at [0009] (“In another embodiment, a fibre channel switch is provided. The fibre channel switch includes memory, a plurality of local ports, and a processor. The plurality of local ports are coupled to a remote fibre channel switch through a plurality of remote ports. The processor is con-figured to determine that a subset of the plurality of local ports at a local fibre channel switch are compatible and send identifiers for the subset of the plurality of local ports to a remote fibre channel switch. The remote fibre channel switch determines if a subset of the plurality of remote ports are compatible. The subset of the plurality of remote ports corresponds to the subset of the plurality of local ports.”) Ghosh at [0010] (“In another embodiment, a fibre channel network is described. The fibre channel network includes a local fibre channel switch and a remote fibre channel switch. The local fibre channel switch aggregates a compatible subset of the plurality of local ports and

No.	’740 Patent Claim 17	Ghosh
		sends identifiers for the compat-ible subset of the plurality of local ports to the remote fibre channel switch. The remote fibre channel switch determines if a subset of the plurality of remote ports are compatible. The subset of the plurality of remote ports corresponds to the compatible subset of the plurality of local ports.”) Ghosh at [0059] (“Line cards 803, 805, and 807 can communicate with an active supervisor 811 through interface circuitry 883, 885, and 887 and the backplane 815. According to various embodiments, each line card includes a plurality of ports that can act as either input ports or output ports for communication with external fibre channel network entities 851 and 853. The backplane 815 can provide a communi-cations channel for all traffic between line cards and super- visors. Individual line cards 803 and 807 can also be coupled to external fibre channel network entities 851 and 853 through fibre channel ports 843 and 847.”) Ghosh at [0060] (“External fibre channel network entities 851 and 853 can be nodes such as other fibre channel switches, disks, RAIDS, tape libraries, or servers. It should be noted that the switch can support any number of line cards and supervisors. In the embodiment shown, only a single supervisor is connected to the backplane 815 and the single supervisor communicates with many different line cards. The active supervisor 811 may be configured or designed to run a plurality of applications such as routing, domain manager, system manager, and utility applications.”)
17[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network,	Ghosh discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. For example, Ghosh discloses line cards that process frames with specific frame information transmitted between network entities in a fibre channel network using a port channel map table. Ghosh at [0022] (“Switches in a fibre channel network are typically interconnected using multiple physical links. The physical links connecting a pair of switches allows transmission of data and control signals. In some instances, it is useful to aggregate multiple physical links

No.	’740 Patent Claim 17	Ghosh
		into a logical link. Physi-cal links are also referred to herein as physical interfaces and channels while logical links are also referred to herein as logical interfaces and port channels. For example, a local switch may be connected to a remote switch through four physical links. Instead of having to transmit data through a particular physical link, the physical links can be aggregated to form one or more logical links. In one example, all four physical links are aggregated into a single logical link. Instead of having data transmitted through a particular physical link, the data can merely be transmitted over a particular logical link without regard to the particular physi-cal interface used. Aggregating physical links into a logical link allows for higher aggregated bandwidth, load balancing, and link redundancy. For example, if a particular physical link fails or is overloaded, data can still be transmitted over the logical link.”) Ghosh at [0027] (“FIG. 1 shows one example of a storage area network implemented using fibre channel that can use efficient port channel configuration mechanisms. A switch 101 is coupled to switches 103 and 105 as well as to a host 111 and storage 121. Switch 101 may be connected to other entities through multiple physical links or channels config-ured as logical links or port channels. In one embodiment, host 111 may be a server or client system while storage 121 may be single disk or a redundant array of independent disks (RAID). Switches 103 and 105 are both coupled to switch 107. Switch 107 is connected to host 113 and switch 103 is connected to storage 123. Switch 109 is connected to host 115, switch 107, disk array 153, and an external network 151 that may or may not use fibre channel.”) Ghosh at [0029] (“FIG. 2 is a diagrammatic representation showing links between two switches, such as two fibre channel switches shown in FIG. 1. A local fibre channel switch 201 includes local ports 241, 243, 245, 247, 249, and 251. A remote fibre channel switch 203 includes remote ports 261, 263, 265, 267, 269, and 271. Local port 241 is coupled to remote port 261 through an individual physical link or channel. Connected ports are also referred to herein as peer ports. Local port 243 is coupled to remote port 263 and local port 245 is coupled to remote port 265. The two resulting physical links are aggregated to form port channel 235. Local ports 247, 249, and 251 are coupled to remote ports 267, 269, and 271 respectively. The three resulting physical links are aggregated to form port channel 237.”)

No.	’740 Patent Claim 17	Ghosh
		Ghosh at [0039] (“At 431, remote switch 403 uses the information received from the local switch 401 to update a port channel database. In one example, the remote switch 403 can check if the port Bl is already assigned to a different port channel. If the port is not already assigned to a different port channel, the remote switch 403 can proceed and send a sync accept message 413 in response to the sync message 411. The sync accept message 413 includes a remote switch 403 assigned world wide name for the remote port channel identifier. The sync accept message indicates that a port channel can now be formed. At 423, local switch 401 uses the information to update its own port channel database. However, the port channel may not yet be operational until the hardware configuration is completed. The local switch 401 continues hardware configuration such as line card configuration to make the port Al part of the port channel. An acknowledg-ment 427 is sent and received by remote switch 403 at 429. In some examples, the local switch 401 sends a commit signal 415 when hardware configuration is complete.”) Ghosh at [0044] (“At 531, remote switch 503 uses the information received from the local switch 501 to verify port B2 is compatible with other port in port channel C2. In one example, configuration parameters associated with B2 are checked against configuration parameters associated with Bl. The remote switch 503 can also check if the port B2 is already assigned to a different port channel. If the port B2 is compatible with port Bl, the remote switch 503 can proceed and send a sync accept message 513 in response to the sync message 511 to indicate that the port B2 can be aggregated into the port channel. The sync accept message indicates that a port channel can now be modified. At 523, local switch 501 uses the information to update its own port channel database. However, the port channel may not yet be fully operational until the hardware configuration is completed. The local switch 501 continues hardware configuration such as line card configuration to make the port A2 part of the port channel Cl. An acknowledgment 527 is sent and received by remote switch 503 at 529. In some examples, the local switch 501 sends a commit signal 515 when hardware configuration is complete.”)

No.	’740 Patent Claim 17	Ghosh
		Ghosh at [0057] (“As described above, techniques for aggregating ports may be performed in a variety of network devices or switches. According to various embodiments, a switch includes a processor, network interfaces, and memory. A variety of ports, Media Access Control (MAC) blocks, and buffers can also be provided as will be appreciated by one of skill in the art.”) Ghosh at [0058] (“FIG. 8 is a diagrammatic representation of one example of a fibre channel switch that can be used to implement techniques of the present invention. Although one particular configuration will be described, it should be noted that a wide variety of switch and router configurations are available. The fibre channel switch 801 may include one or more supervisors 811. According to various embodi-ments, the supervisor 811 has its own processor, memory, and storage resources.”) Ghosh at [0059] (“Line cards 803, 805, and 807 can communicate with an active supervisor 811 through interface circuitry 883, 885, and 887 and the backplane 815. According to various embodiments, each line card includes a plurality of ports that can act as either input ports or output ports for communication with external fibre channel network entities 851 and 853. The backplane 815 can provide a communi-cations channel for all traffic between line cards and super- visors. Individual line cards 803 and 807 can also be coupled to external fibre channel network entities 851 and 853 through fibre channel ports 843 and 847.”) Ghosh at [0060] (“External fibre channel network entities 851 and 853 can be nodes such as other fibre channel switches, disks, RAIDS, tape libraries, or servers. It should be noted that the switch can support any number of line cards and supervisors. In the embodiment shown, only a single supervisor is connected to the backplane 815 and the single supervisor communicates with many different line cards. The active supervisor 811 may be configured or designed to run a plurality of applications such as routing, domain manager, system manager, and utility applications.”) Ghosh at Figure 8 (annotation added)

No.	’740 Patent Claim 17	Ghosh
		Figure 8  External Fibre Channel Network Entity 851 External Fibre Channel Network Entity 853 FC Port 843 Line Card 803 Interface Circuitry 883 FC Port 845 Line Card 805 Interface Circuitry 885 FC Port 847 Line Card 807 Interface Circuitry 887 Active Supervisor 811 Backplane Interface Logic 831 Backplane 815 801 Power Supply 817
17[b]	at least one of said interface modules being operative to communicate in both an upstream direction and a downstream direction;	Ghosh discloses at least one of said interface modules being operative to communicate in both an upstream direction and a downstream direction. For example, Ghosh discloses line cards that can transmit frames in both directions. Ghosh at [0026] (“Consequently, the techniques and mechanisms of the present invention allow automatic detection of compat-ible ports to enable automatic creation of port channels.

No.	’740 Patent Claim 17	Ghosh
		Port channels can be effectively brought up at either a local switch or a remote switch after either automatic creation of port channels or manual configuration of port channels. Robust error detection capabilities allow the correction of improper configurations and connections. Member physical ports of a port channel can operate as individual links if they cannot be configured to be part of port channel. Further-more, synchronization is supported so that requests and responses belonging to the same flow can be carried over the same physical link in a port channel in both directions. In many conventional implementations, such as Ethernet for example, a flow belonging to a particular port channel could be carried over different physical links during send and receive phases.”) Ghosh at [0032] (“Each entity can also have additional parameters to aid in the set up of port channels. According to various embodiments, parameters such as a channeling model, a channeling intent and a channeling status are included. A channeling model indicates to a peer port the channel group is automatically created or user configured. A channeling intent parameter indicates the peer port if this port intends to participate in a port channel. Otherwise, the port intends to operate as an individual port. The channeling status param-eter tells the peer port about its current channeling status. This parameter is exchanged by the attached peer ports to agree upon the channeling status of the link and to ensure that both ends are synchronized.”) Ghosh at [0055] (“Based on this scheme, frames for a given flow are transmitted through the same physical port of the port channel. However, unless there is proper synchronization of the port channel map tables at the two ends of a port channel, it is possible that requests and responses for the same flow are carried over two different physical links. For example, requests may be carried over link 721 while responses are carried over 723. This is undesirable for port channels as it affects applications like write acceleration that assume the traffic for a given flow is carried over the same physical link in both directions.”) Ghosh at [0056] (“According to various embodiments of the present invention, links are brought up in order. A first link is selected for bring up. No effort is made to bring up other links is attempted until the exchange associated with the first link is completed.

No.	’740 Patent Claim 17	Ghosh
		Consequently, port channel map tables at the local switch 701 and at the remote switch 703 are consistent. Port channel map table entries are synchronized at both ends so that all frames for a given flow are carried over the same physical link in both directions after the exchanges are completed.”) Ghosh at [0059] (“Line cards 803, 805, and 807 can communicate with an active supervisor 811 through interface circuitry 883, 885, and 887 and the backplane 815. According to various embodiments, each line card includes a plurality of ports that can act as either input ports or output ports for communication with external fibre channel network entities 851 and 853. The backplane 815 can provide a communi-cations channel for all traffic between line cards and super- visors. Individual line cards 803 and 807 can also be coupled to external fibre channel network entities 851 and 853 through fibre channel ports 843 and 847.”)
17[c]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Ghosh discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 1[a].
17[d]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Ghosh discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 1[c].
17[e]	a control module, which is arranged to select for each data	Ghosh discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on

No.	’740 Patent Claim 17	Ghosh
	frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame;	at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 1[f].
17[f]	at least one of said first physical links and at least one of said second links being bi-directional links operative to communicate in both said upstream direction and said downstream direction.	Ghosh discloses at least one of said first physical links and at least one of said second links being bi-directional links operative to communicate in both said upstream direction and said downstream direction. *See supra at* 1[b], 1[d].

No.	’740 Patent Claim 18	Ghosh
18[a]	The apparatus according to claim 17, and comprising a	Ghosh discloses the apparatus according to claim 17, and comprising a backplane to which the one or more interface modules are coupled.

No.	’740 Patent Claim 18	Ghosh
	backplane to which the one or more interface modules are coupled,	*See supra at* 3, 17.
18[b]	wherein the second physical links comprise backplane traces formed on the backplane.	Ghosh discloses wherein the second physical links comprise backplane traces formed on the backplane. *See supra at* 3, 17.

No.	’740 Patent Claim 19	Ghosh
19[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Ghosh discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
19[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	Ghosh discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].
19[b]	a first group of first physical links arranged in parallel	Ghosh discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules.

No.	’740 Patent Claim 19	Ghosh
	so as to couple the network node to the one or more interface modules;	*See supra at* 17[c].
19[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Ghosh discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].
19[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,	Ghosh discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].
19[e]	at least one of the first and second	Ghosh discloses at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group.

No.	’740 Patent Claim 19	Ghosh
	groups of physical links comprising an Ethernet link aggregation (LAG) group.	*See supra at* 4[f].

No.	’740 Patent Claim 20	Ghosh
20[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Ghosh discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
20[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	Ghosh discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].
20[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Ghosh discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].
20[c]	a second group of second physical links	Ghosh discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network.

No.	’740 Patent Claim 20	Ghosh
	arranged in parallel so as to couple the one or more interface modules to the communication network; and	*See supra at* 17[d].
20[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,	Ghosh discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].
20[e]	two or more of the first physical links being aggregated into an external Ethernet link aggregation (LAG) group so as to increase a data	Ghosh discloses two or more of the first physical links being aggregated into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node. *See supra at* 19[e], 5[f].

No.	’740 Patent Claim 20	Ghosh
	bandwidth provided to the network node.	

No.	’740 Patent Claim 21	Ghosh
21	The apparatus according to claim 17, and comprising a multiplexer, which is arranged to perform at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node.	Ghosh discloses the apparatus according to claim 17, and comprising a multiplexer, which is arranged to perform at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node. *See supra at* 6, 17.

No.	’740 Patent Claim 22	Ghosh
22	The apparatus according to claim 17, wherein the control module is arranged to balance a	Ghosh discloses the apparatus according to claim 17, wherein the control module is arranged to balance a frame data rate among at least some of the first and second physical links. *See supra at* 7, 17.

No.	’740 Patent Claim 22	Ghosh
	frame data rate among at least some of the first and second physical links.	

No.	’740 Patent Claim 23	Ghosh
23	The apparatus according to claim 17, wherein the control module is arranged to apply a mapping function to the at least one of the frame attributes so as to select the first and second physical links.	Ghosh discloses the apparatus according to claim 17, wherein the control module is arranged to apply a mapping function to the at least one of the frame attributes so as to select the first and second physical links. *See supra at* 8, 17.

No.	’740 Patent Claim 24	Ghosh
24	The apparatus according to claim 23, wherein the mapping function comprises a hashing function.	Ghosh discloses the apparatus according to claim 23, wherein the mapping function comprises a hashing function. *See supra at* 9, 23.

No.	’740 Patent Claim 25	Ghosh
25[a]	The apparatus according to claim 24, wherein the control module is arranged to determine a hashing size responsively to a number of at least some of the first and second physical links,	Ghosh discloses the apparatus according to claim 24, wherein the control module is arranged to determine a hashing size responsively to a number of at least some of the first and second physical links. *See supra at* 10[a], 24.
25[b]	to apply the hashing function to the at least one of the frame attributes to produce a hashing key,	Ghosh discloses to apply the hashing function to the at least one of the frame attributes to produce a hashing key. *See supra at* 10[b].
25[c]	to calculate a modulo of a division operation of the hashing key by the hashing size, and	Ghosh discloses to calculate a modulo of a division operation of the hashing key by the hashing size. *See supra at* 10[c].
25[d]	to select the first and second physical links responsively to the modulo.	Ghosh discloses to select the first and second physical links responsively to the modulo. *See supra at* 10[d].

No.	’740 Patent Claim 26	Ghosh
26	The apparatus according to claim 25, wherein the control module is arranged to select the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo.	Ghosh discloses the apparatus according to claim 25, wherein the control module is arranged to select the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo. *See supra at* 11, 25.

No.	’740 Patent Claim 27	Ghosh
27	The apparatus according to claim 17, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC)	Ghosh discloses the apparatus according to claim 17, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port. *See supra at* 12, 17.

No.	’740 Patent Claim 27	Ghosh
	address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port.	

No.	’740 Patent Claim 28	Ghosh
28[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Ghosh discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
28[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	Ghosh discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].
28[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Ghosh discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].

No.	’740 Patent Claim 28	Ghosh
28[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Ghosh discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].
28[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,	Ghosh discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].
28[e]	the communication network being arranged to provide a communication service to the network node,	Ghosh discloses the communication network being arranged to provide a communication service to the network node. *See supra at* 2[b].

No.	’740 Patent Claim 28	Ghosh
28[f]	the service having specified bandwidth requirements comprising at least one of a committed information rate (CR), a peak information rate (PIR) and an excess information rate (EIR), and	Ghosh discloses the service having specified bandwidth requirements comprising at least one of a committed information rate (CR), a peak information rate (PIR) and an excess information rate (EIR). *See supra at* 13[i].
28[g]	the first and second groups of physical links being dimensioned to provide an allocated bandwidth for the communication service responsively to the bandwidth requirements.	Ghosh discloses the first and second groups of physical links being dimensioned to provide an allocated bandwidth for the communication service responsively to the bandwidth requirements. *See supra at* 13[j].

No.	’740 Patent Claim 29	Ghosh
29[preamble]	Apparatus for connecting user ports to a communication network, comprising:	Ghosh discloses apparatus for connecting user ports to a communication network. *See supra at* 17[preamble], 14[preamble].
29[a]	one or more user interface modules coupled to the user	Ghosh discloses one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network.

No.	’740 Patent Claim 29	Ghosh
	ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network,	*See supra at* 17[a], 14[a].
29[b]	at least one of said user interface modules being bi-directional and operative to communicate in both an upstream direction and a downstream direction;	Ghosh discloses at least one of said user interface modules being bi-directional and operative to communicate in both an upstream direction and a downstream direction. *See supra at* 17[b], 14[c].
29[c]	a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the	Ghosh discloses a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network. *See supra at* 14[b]-[e].

No.	’740 Patent Claim 29	Ghosh
	communication network,	
29[d]	at least one of said backplane traces being bi-directional and operative to communicate in both said upstream direction and said downstream direction; and	Ghosh discloses at least one of said backplane traces being bi-directional and operative to communicate in both said upstream direction and said downstream direction. *See supra at* 14[c], 17[b].
29[e]	a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame.	Ghosh discloses a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame. *See supra at* 14[e], 17[e].

No.	’740 Patent Claim 30	Ghosh
30[preamble]	Apparatus for connecting user ports to a communication network, comprising:	Ghosh discloses apparatus for connecting user ports to a communication network. *See supra at* 29[preamble].

No.	’740 Patent Claim 30	Ghosh
30[a]	one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network;	Ghosh discloses one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network. *See supra at* 29[a].
30[b]	a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network;	Ghosh discloses a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network. *See supra at* 29[c].
30[c]	a control module, which is arranged to select, for each data frame, responsively to at least one of the	Ghosh discloses a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame. *See supra at* 29[e].

No.	’740 Patent Claim 30	Ghosh
	frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame;	
30[d]	at least some of the backplane traces are aggregated into an Ethernet link aggregation (LAG) group.	Ghosh discloses at least some of the backplane traces are aggregated into an Ethernet link aggregation (LAG) group. *See supra at* 4[f], 15[f].

No.	’740 Patent Claim 31	Ghosh
31	The apparatus according to claim 29, wherein the control module is arranged to apply a hashing function to the at least one of the frame attributes so as to select the backplane trace.	Ghosh discloses the apparatus according to claim 29, wherein the control module is arranged to apply a hashing function to the at least one of the frame attributes so as to select the backplane trace. *See supra at* 16, 29, 30[c].

EXHIBIT C-9
Defendant's Preliminary Invalidity Contentions
Orckit Corporation v. Cisco Systems, Inc., 2:22-cv-00276-JRG-RSP

Chart for U.S. Patent 7,545,740 ("the '740 Patent")
U.S. Patent Publication No. 2004/0042448 to Lebizay et al. ("Lebizay")

As shown in the chart below, all Asserted Claims of the '740 Patent are invalid under (1) 35 U.S.C. §§ 102 (a), (b), (e), and (g) because Lebizay meets each element of those claims, and/or (2) 35 U.S.C. § 103 because Lebizay renders those claims obvious either alone, or in combination with the knowledge of a person having ordinary skill in the art, and in further combination with the references specifically identified below and in the following claim chart and/or one or more references identified in Defendant's Preliminary Invalidity Contentions. The following quotations and diagrams come from Lebizay titled "Multi-Port High-Speed Serial Fabric Interconnect Chip In A Meshed Configuration", which was filed on August 30, 2002, and published on March 4, 2004.

Motivations to combine the disclosures in Lebizay with disclosures in other publications known in the art, as explained in this chart, include at least the similarity in subject matter between the references to the extent they concern methods of data communication systems, and specifically to methods and systems for link aggregation in a data communication network. Insofar as the references cite other patents or publications, or suggest additional changes, one of ordinary skill in the art would look beyond a single reference to other references in the field.

These invalidity contentions are based on Defendant's present understanding of the asserted claims, and Orckit's apparent construction of the claims in its November 3, 2022 Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1, and Orckit's January 19, 2023 First Amended Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1 (Orckit's "Infringement Disclosures"), which is deficient at least insofar as it fails to cite any documents or identify accused structures, acts, or materials in the Accused Products with particularity. Defendant does not agree with Orckit's application of the claims, or that the claims satisfy the requirements of 35 U.S.C. § 112. Defendant's contentions herein are not, and should in no way be seen as, admissions or adoptions as to any particular claim scope or construction, or as any admission that any particular element is met by any accused product in any particular way. Defendant objects to any attempt to imply claim construction from this chart. Defendant's prior art invalidity contentions are made in a variety of alternatives and do not represent Defendant's agreement or view as to the meaning, definiteness, written description support for, or enablement of any claim contained therein.

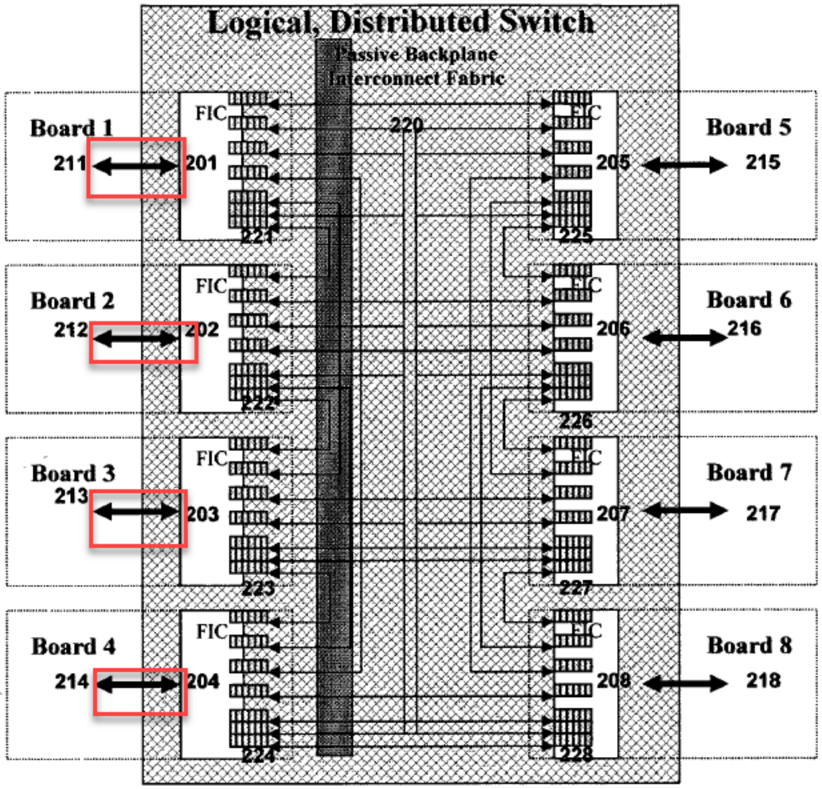
The following contentions are subject to revision and amendment pursuant to Federal Rule of Civil Procedure 26(e), the Local Rules, and the Orders of record in this matter subject to further investigation and discovery regarding the prior art and the Court's construction of the claims at issue.

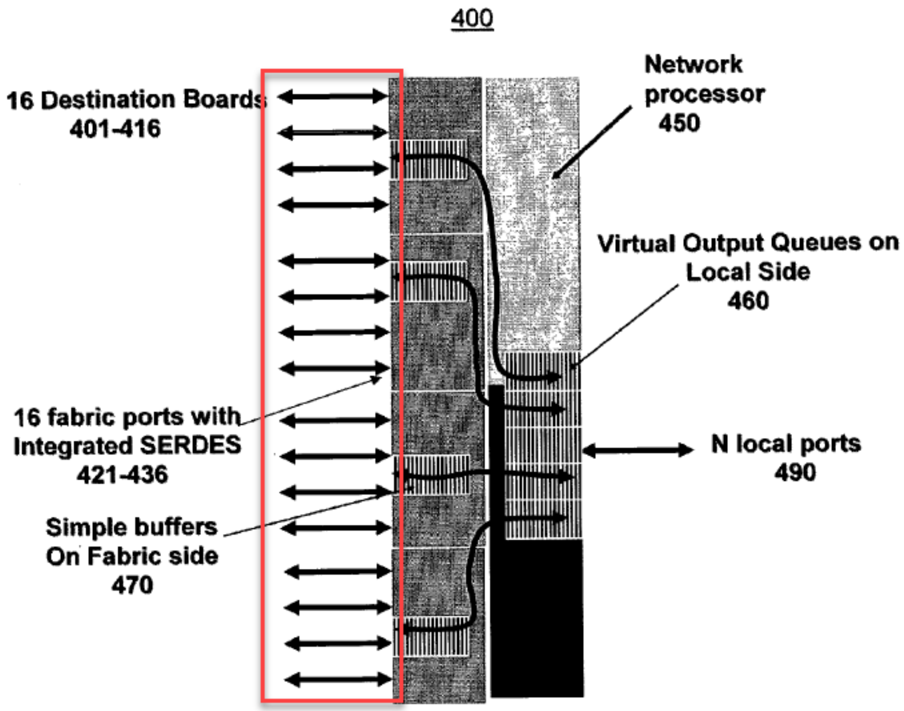
No.	'740 Patent Claim 1	Lebizay
1[preamble]	A method for communication, comprising:	Lebizay discloses a method for communication. For example, Lebizay discloses a communication system that bridges multiple boards for data transfer. Lebizay at Abstract ("Multiple boards are connected within a chassis, using a multi-port Target Channel Adapter (TCA). Data is transported from a TCA on a board directly to a TCA on another board through a meshed backplane. The meshed backplane is equipped to mount boards via connectors and may consist of a fabric of copper conductors or optical fibers. Commu-nication from TCA to TCA requires placing ports on each individual TCA along with the appropriate input and output buffering. A multi-port TCA capable of performing multiple bridging functions simultaneously i.e., bridging from a high speed serial meshed backplane to multiple local busses, i.e., Gigabit Ethernet, Fibre Channel, and TCP/IP devices, is referred to as a Fabric Interconnect Chip (FIC).") Lebizay at [0022] ("The InfiniBand Architecture uses basically the VIA primitives for its operation at the transport layer. In order for an application to communicate with another application over the InfiniBand it must first create a work queue that consists of a queue pair (QP). In order for the application to execute an operation, it must place a work queue element (WQE) in the work queue. From there, the operation is picked-up for execution by the channel adapter. Therefore, the Work Queue forms the communications medium between appli-cations and the channel adapter, relieving the operating system from having to deal with this responsibility.") Lebizay at [0030] ("The FIC provides data queuing first in first out (FIFOs) on both its inbound and outbound sides for each of its multiple ports. By placing FICs on every board within a chassis that is connected to the meshed backplane, the boards can inter-communicate with each other in a "meshed" topology. Therefore, every board has a point to point connection with

No.	’740 Patent Claim 1	Lebizay
		every other board in the backplane, making a complete cross-connect without the need for a separate switch device.”)
1[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel,	Lebizay discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. For example, Lebizay discloses connecting Target Channel Adapters (TCA), sometimes known as Fabric Interconnect Chips (FIC) to network boards using multiple parallel ports. Lebizay at [0024] (“High speed serial fabrics such as the InfiniBand architecture may also be used for connecting multiple com-ponents (boards) communicating to one another within a single chassis over a common fabric or switched fabric. For instance, a new standard, AdvancedTCATM (Advanced Tele-communications Compute Architecture), is currently being developed out of the industry forum PCI Industrial Com-puter Manufacturers Group, which previously formulated the CompactPCI™ standard. CompactPCI™ is a variation of the Eurocard form factor that uses PCI buses in a back plane. The Eurocard form factor is a physical form factor of line cards that have a front panel. A line card has a solid face plate with handles, slides into the front of a chassis by the use of guides, and makes contact with a backplane or midplane within the chassis. This allows direct access to the cards unlike a PC chassis or server where a lid is opened to expose the PCI cards sitting inside the chassis. AdvancedTCA TM defines star and meshed topologies for a packet switched fabric in the back plane of a chassis. The proposed meshed fabric solution utilizes Infiniband technol-ogy and a meshed fabric solution. However, the design methodology being proposed potentially puts a switch on every single board in the chassis. For example, suppose there are 16 boards or slots in a chassis, if a 16 port switch exists on every single board, 16 switches are required in the chassis. Each board communicates with all other boards to form a meshed or switched fabric.”) Lebizay at [0028] (“Embodiments of the present invention exists within the context of connecting multiple entities within a system (specifically multiple boards within a chassis), using mul-tiple TCAs. The components consist of multi-port TCAs and a meshed backplane that is equipped to mount boards via connectors on the backplane. A multi-port TCA capable of performing multiple bridging functions simultaneously i.e., bridging from an Infiniband

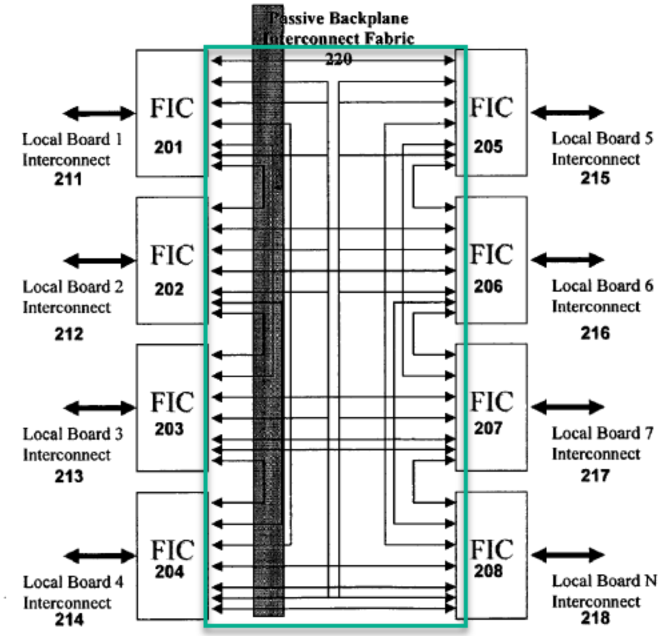
No.	’740 Patent Claim 1	Lebizay
		meshed backplane to multiple local busses, i.e., Gigabit Ethernet, Fibre Channel, and TCP/IP devices, is referred to as a Fabric Interconnect Chip (FIC). FIG. 1 depicts the connection of a FIC 100 that interconnects between local components on a board (local busses) 120 and a backplane mesh fabric 110.”) Lebizay at [0029] (“FIG. 2 depicts connection of multiple intercon-nected FICs. Each board 211-218 on the mesh 220 contains one FIC 201-208 . Each FIC may provide an interconnection to every other board in the shelf (chassis). FIG. 2 shows an 8-way (8 board) mesh, but any size mesh is realizable, hence the 8,h board 218 is labeled "N". The lines drawn show how the traces from each FIC 'port' travels across the passive backplane interconnect fabric 220 to its corresponding 'port' on another FIC, (located on another board).”) Lebizay at [0030] (“The FIC provides data queuing first in first out (FIFOs) on both its inbound and outbound sides for each of its multiple ports. By placing FICs on every board within a chassis that is connected to the meshed backplane, the boards can inter-communicate with each other in a "meshed" topology. Therefore, every board has a point to point connection with every other board in the backplane, making a complete cross-connect without the need for a separate switch device.”)
1[b]	at least one of said first physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction	Lebizay discloses at least one of said first physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction. For example, Lebizay discloses that its ports comprise queues capable of communicating in both inbound and outbound directions. Lebizay at [0026] (“All switch devices have a given amount of ineffi-ciency based upon their queuing logic. Most often, this leads to head-of-line blocking in the switch because multiple contenders are vying for a single bottle neck. However, by placing multiple queues both inbound and outbound, for each port on the switch, these head-of-line blocking laten-cies can

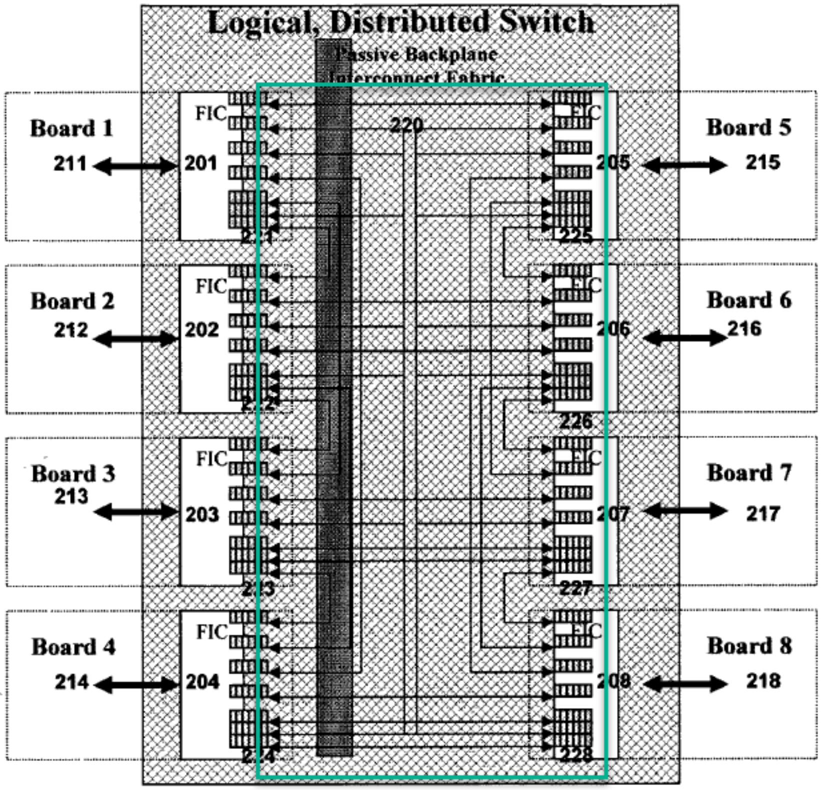
No.	’740 Patent Claim 1	Lebizay
		be largely removed. A single device that incorpo-rates such a queuing model is both costly and large, almost too large and expensive to put into a single piece of usable silicon. However, by distributing this functionality across multiple components, using the inherent cross-connect of the meshed backplane, an efficient logical distributed switch may be built at much lower cost and higher efficiency.”) Lebizay at [0030] (“The FIC provides data queuing first in first out (FIFOs) on both its inbound and outbound sides for each of its multiple ports. By placing FICs on every board within a chassis that is connected to the meshed backplane, the boards can inter-communicate with each other in a "meshed" topology. Therefore, every board has a point to point connection with every other board in the backplane, making a complete cross-connect without the need for a separate switch device.”) Lebizay at [0031] (“In addition, by incorporating the queuing logic on both the inbound and outbound sides of the FIC, coupled with the inherent cross-connect of the mesh, a logical, distributed switch is created, without really having a switch device. FIG. 3 illustrates the same 8-way mesh depicted in FIG. 2, highlighting the logic switch which is created by distributing the switch function across FICs 201-208 on all the boards 211-218. Each multi-port FIC 201-208 is shown having inbound and outbound queues 221-228 for each port. In total, what is depicted is a 8-way logical distributed switch, where each local bus (board 211-218) represents an access point on the logical switch.”) Lebizay at Figure 3 (annotation added)

No.	’740 Patent Claim 1	Lebizay
		 Logical, Distributed Switch Passive Backplane Interconnect Fabric 220 Board 1 211 201 FIC 221 Board 2 212 202 FIC 222 Board 3 213 203 FIC 223 Board 4 214 204 FIC 224 Board 5 215 205 FIC 225 Board 6 216 206 FIC 226 Board 7 217 207 FIC 227 Board 8 218 208 FIC 228 FIG. 3 Lebizay at [0045] (“FIG. 4 also illustrates simple buffers (receive queues) 470 on the egress fabric side. The 16 fabric ports 421-436 have integrated serializer/de-serializers (SERDES). By allocating buffers (queues) to each port on both the receive and transmit sides, traffic can

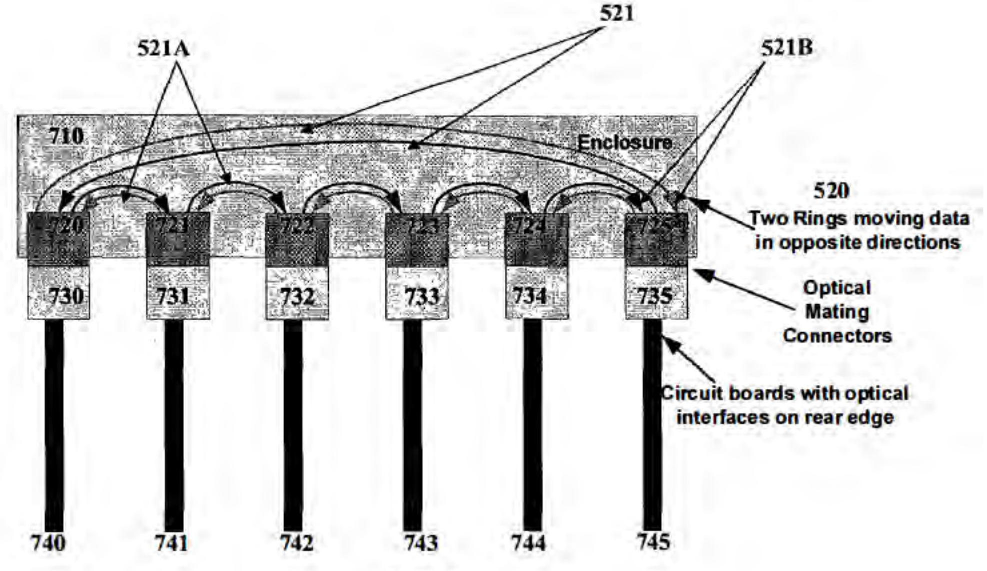
No.	’740 Patent Claim 1	Lebizay
		be managed without head of line blocking or congestion. This enables each board to have a private communications channel with every other board in the system.”) Lebizay at Figure 4  400 16 Destination Boards 401-416 Network processor 450 Virtual Output Queues on Local Side 460 16 fabric ports with Integrated SERDES 421-436 N local ports 490 Simple buffers On Fabric side 470 FIG. 4
1[c]	coupling each of the one or more interface modules to a	Lebizay discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel.

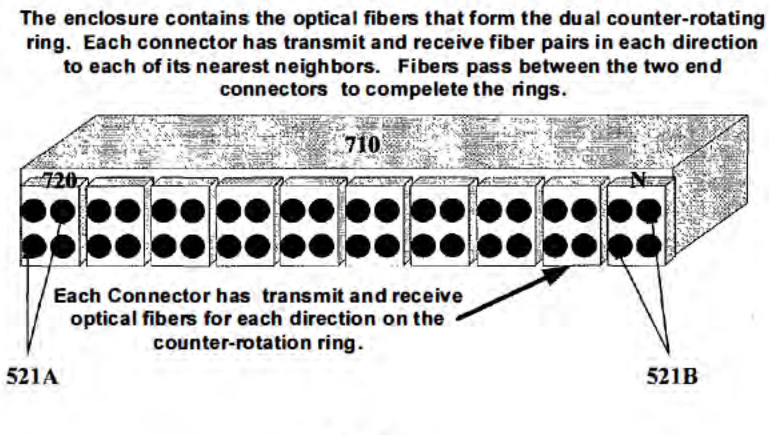
No.	’740 Patent Claim 1	Lebizay
	communication network using a second group of second physical links arranged in parallel,	For example, Lebizay discloses parallel connectors or traces on a backplane, that couple FICs. Lebizay at [0028] (“Embodiments of the present invention exists within the context of connecting multiple entities within a system (specifically multiple boards within a chassis), using mul-tiple TCAs. The components consist of multi-port TCAs and a meshed backplane that is equipped to mount boards via connectors on the backplane. A multi-port TCA capable of performing multiple bridging functions simultaneously i.e., bridging from an Infiniband meshed backplane to multiple local busses, i.e., Gigabit Ethernet, Fibre Channel, and TCP/IP devices, is referred to as a Fabric Interconnect Chip (FIC). FIG. 1 depicts the connection of a FIC 100 that interconnects between local components on a board (local busses) 120 and a backplane mesh fabric 110.”) Lebizay at [0029] (“FIG. 2 depicts connection of multiple interconnected FICs. Each board 211-218 on the mesh 220 contains one FIC 201-208 . Each FIC may provide an interconnection to every other board in the shelf (chassis). FIG. 2 shows an 8-way (8 board) mesh, but any size mesh is realizable, hence the 8,h board 218 is labeled "N". The lines drawn show how the traces from each FIC 'port' travels across the passive backplane interconnect fabric 220 to its corresponding 'port' on another FIC, (located on another board).”) Lebizay at Figure 2 (annotation added)

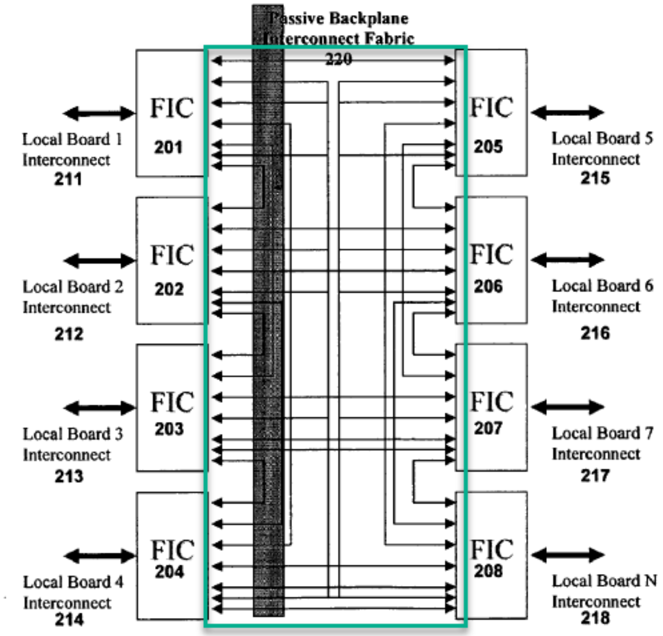
No.	’740 Patent Claim 1	Lebizay
		 Passive Backplane Interconnect Fabric 220 FIC 201 Local Board 1 Interconnect 211 FIC 202 Local Board 2 Interconnect 212 FIC 203 Local Board 3 Interconnect 213 FIC 204 Local Board 4 Interconnect 214 FIC 205 Local Board 5 Interconnect 215 FIC 206 Local Board 6 Interconnect 216 FIC 207 Local Board 7 Interconnect 217 FIC 208 Local Board N Interconnect 218 FIG. 2 Lebizay at Figure 3 (annotation added)

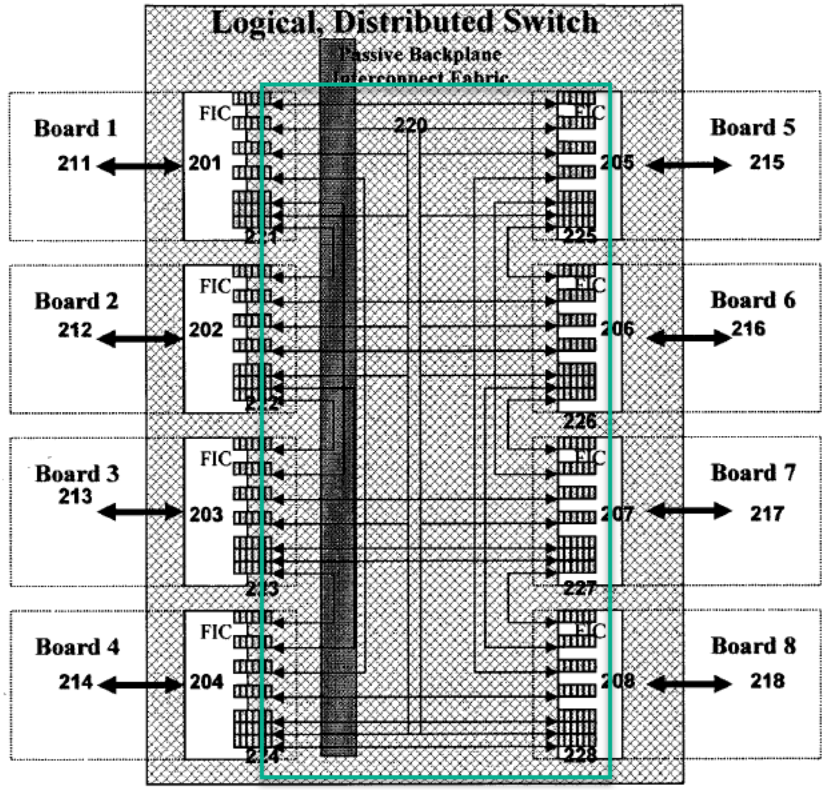
No.	’740 Patent Claim 1	Lebizay
		 Logical, Distributed Switch Passive Backplane Interconnect Fabric 220 Board 1 211 FIC 201 221 Board 2 212 FIC 202 222 Board 3 213 FIC 203 223 Board 4 214 FIC 204 224 Board 5 215 FIC 205 225 Board 6 216 FIC 206 226 Board 7 217 FIC 207 227 Board 8 218 FIC 208 228 **FIG. 3** Lebizay at [0032] (“Since each FIC (201-208) has queuing (221-228) on both the inbound and outbound sides, coupled with the inherent cross-connect in the backplane 220, an extremely efficient logically distributed switch can be built at much lower cost, greater reliability, higher efficiency, lower power and lower latency than a typical, centrally switched topol-ogy, or any architecture that places a switch on every board of a meshed topology.”)

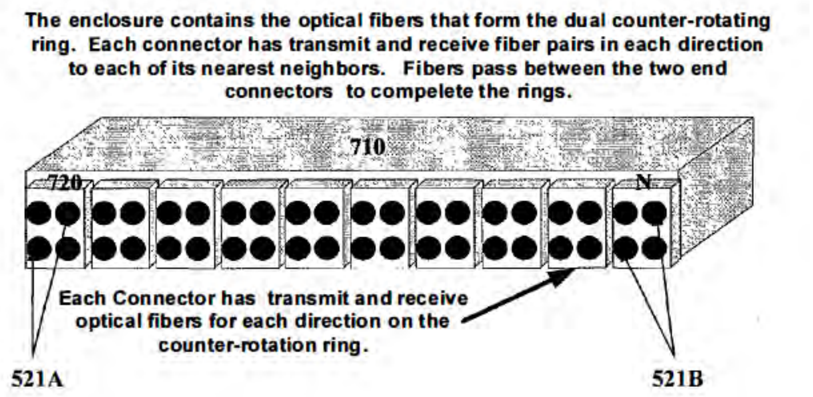
No.	’740 Patent Claim 1	Lebizay
		Lebizay at [0046] (“Today, boards in a common chassis interconnect to each other over copper traces embedded in the chassis backplane. Existing backplanes use copper traces within a PCB substrate to establish the communications paths between boards. Topologies such as common bus PCI and switched star (i.e. some Ethernet approaches) are popular today. The copper traces and the electrical drivers that communicate over them are limited in terms of their capac-ity to carry data bandwidth with respect to the Local Area Network (LAN) and Wide Area Network (WAN) optical signals that these boards are intended to terminate.”) Lebizay at [0058] (“Referring to FIG. 7, the basic components of the dual counter-rotating ring optical backplane include a set of fibers 521, arranged in a dual-counter-rotating ring 520, housed in a overlay module 710 that allows fiber cavity mechanical connectors 720-725 to mount to the overlay module 710 front surface. Each mechanical connector 720-725 mates up to a mated connector 730-735 that mounts to the individual boards 740-745 in a chassis. Mechanical connectors 720-725 may also contain an OADM 541 ring adaptor previously depicted in FIG. 5 and FIG. 6.”) Lebizay at Figure 7

No.	’740 Patent Claim 1	Lebizay
		 521 521A 521B 710 Enclosure 520 Two Rings moving data in opposite directions 720 721 722 723 724 725 730 731 732 733 734 735 Optical Mating Connectors Circuit boards with optical interfaces on rear edge 740 741 742 743 744 745 FIG. 7 Lebizay at [0059] (“FIG. 8 illustrates an overlay module according to an embodiment of the present invention. The overlay mod-ule 710 includes a series of connectors 720-N mounted to the front of the enclosure. The overlay module 710 contains the optical fibers 521 that form the dual counter-rotating ring 520. Referring to FIG. 7 and FIG. 8, each connector 720-725 has transmit 521A and receive 521B optical fibers for each direction on the counterrotation ring 520. Each connector 720-725 (720-N) has transmit 521A and receive fiber 521B in each direction to each of its nearest neighbors. Fibers 521 pass between the two end connectors 720, 735 (720, N) to complete the rings 520.”) Lebizay at Figure 8

No.	’740 Patent Claim 1	Lebizay
		 The enclosure contains the optical fibers that form the dual counter-rotating ring. Each connector has transmit and receive fiber pairs in each direction to each of its nearest neighbors. Fibers pass between the two end connectors to compelete the rings. 710 720 N Each Connector has transmit and receive optical fibers for each direction on the counter-rotation ring. 521A 521B FIG. 8
1[d]	at least one of said second physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction;	Lebizay discloses at least one of said second physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction. For example, Lebizay discloses connectors on the backplane capable of receiving and transmitting data in both upstream and downstream directions. Lebizay at [0032] (“Since each FIC (201-208) has queuing (221-228) on both the inbound and outbound sides, coupled with the inherent cross-connect in the backplane 220, an extremely efficient logically distributed switch can be built at much lower cost, greater reliability, higher efficiency, lower power and lower latency than a typical, centrally switched topol-ogy, or any architecture that places a switch on every board of a meshed topology.”) Lebizay at Figure 2 (annotation added)

No.	’740 Patent Claim 1	Lebizay
		 Passive Backplane Interconnect Fabric 220 FIC 201 FIC 202 FIC 203 FIC 204 FIC 205 FIC 206 FIC 207 FIC 208 Local Board 1 Interconnect 211 Local Board 2 Interconnect 212 Local Board 3 Interconnect 213 Local Board 4 Interconnect 214 Local Board 5 Interconnect 215 Local Board 6 Interconnect 216 Local Board 7 Interconnect 217 Local Board N Interconnect 218 FIG. 2 Lebizay at Figure 3 (annotation added)

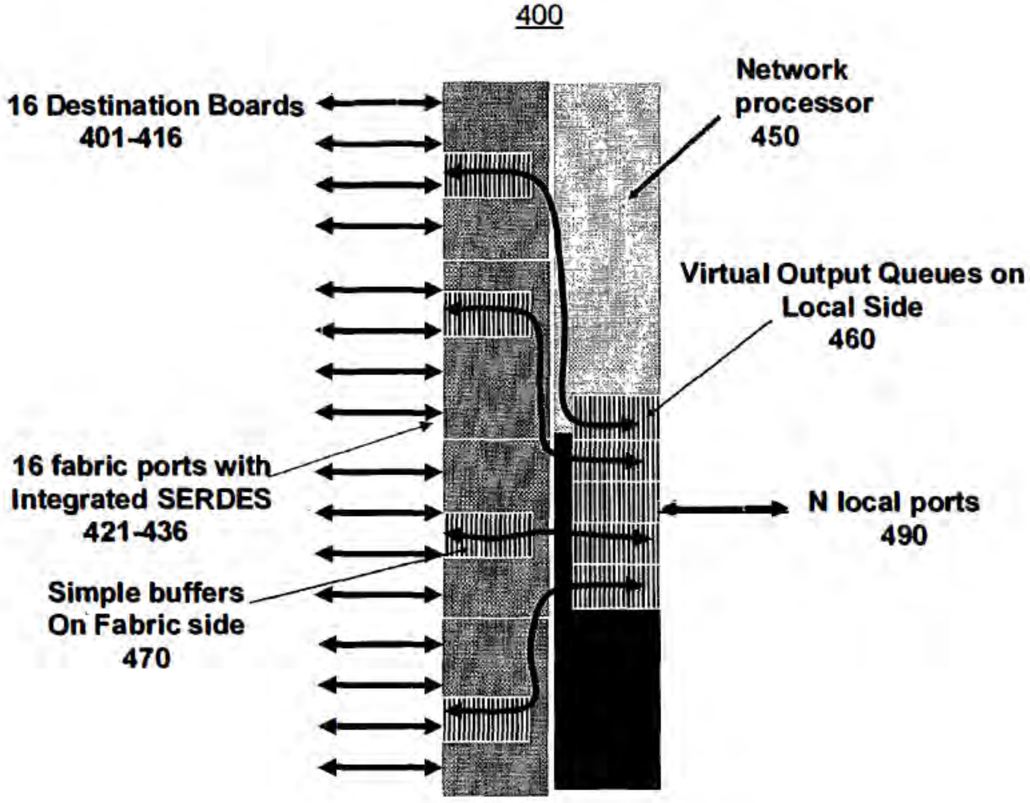
No.	’740 Patent Claim 1	Lebizay
		 Logical, Distributed Switch Passive Backplane Interconnect Fabric 220 Board 1 211 FIC 201 221 Board 2 212 FIC 202 222 Board 3 213 FIC 203 223 Board 4 214 FIC 204 224 Board 5 215 FIC 205 225 Board 6 216 FIC 206 226 Board 7 217 FIC 207 227 Board 8 218 FIC 208 228 FIG. 3 Lebizay at [0051] (“FIG. 5 shows a high-level diagram of a system utilizing a dual-counter-rotating ring topology as an optical backplane according to an embodiment of the present inven-tion. The concept is not limited to any particular number of boards in a chassis or shelf, but 3 boards is chosen to simplify the illustration. Each board 501-503 has transmit-ters 511-513 for each wavelength needed. The number of wavelengths needed is equal to the number

No.	’740 Patent Claim 1	Lebizay
		of boards minus one, in the system. In a counter-rotating ring topology, there are two physical rings, each carrying the same traffic flows but moving in opposite directions. Each board 501-503 transmits in both directions across the dual counter-rotating ring topology 520 such that that loss of any one board (due to removal or failure) will not disrupt communications with the other boards.”) Lebizay at [0059] (“FIG. 8 illustrates an overlay module according to an embodiment of the present invention. The overlay mod-ule 710 includes a series of connectors 720-N mounted to the front of the enclosure. The overlay module 710 contains the optical fibers 521 that form the dual counter-rotating ring 520. Referring to FIG. 7 and FIG. 8, each connector 720-725 has transmit 521A and receive 521B optical fibers for each direction on the counterrotation ring 520. Each connector 720-725 (720-N) has transmit 521A and receive fiber 521B in each direction to each of its nearest neighbors. Fibers 521 pass between the two end connectors 720, 735 (720, N) to complete the rings 520.”) Lebizay at Figure 8  The enclosure contains the optical fibers that form the dual counter-rotating ring. Each connector has transmit and receive fiber pairs in each direction to each of its nearest neighbors. Fibers pass between the two end connectors to compelete the rings. Each Connector has transmit and receive optical fibers for each direction on the counter-rotation ring. FIG. 8
1[e]	receiving a data frame having frame attributes	Lebizay discloses receiving a data frame having frame attributes sent between the communication network and the network node.

No.	’740 Patent Claim 1	Lebizay
	sent between the communication network and the network node:	For example, Lebizay discloses data flows of packets sent between network boards and a communication network with certain characteristics including source and destination addresses. Lebizay at [0035] (“A flow is a set of packets that all share a set of characteristics. Typically, the characteristics include the source and destination address of the packet, as well as its protocol type and possibly its priority or classification. It is important that all the packets in a flow maintain a certain sequence in which they were sent, preferably arriving at their destination in that same sequence. If they do arrive out of sequence they can be re-sequenced, or put back in order. However, it is not desireable to re-sequence packets at the end. Therefore, a good design attempts to keep all the packets in a flow in sequence all through the network so that they arrive at the far end in sequence and do not require re-sequencing.”) Lebizay at [0054] (“Since each wavelength is transmitted by only one board and consumed by only one board, each wavelength uniquely identifies both a source and destination across the optical backplane 520. This provides a substantial improve-ment in efficiency over current day solutions from the standpoint of a reduction in protocol overhead. The mesh topology is preserved through the meshing of wavelengths; each containing dedicated bandwidth between any two end points.”)
1[f]:	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	Lebizay discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. For example, Lebizay discloses processing packets based on a set of parameters and determining the path over which to send packets to their destination. The path determination includes specific local ports and backplane interconnect fabric ports. A person of ordinary skill would understand that the local ports and backplane interconnect fabric ports are selected in a single computation based on packet parameters. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the

No.	’740 Patent Claim 1	Lebizay
		Lebizay is found to not meet this limitation, selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group would have been obvious to a person having ordinary skill in the art, as explained below. Lebizay at [0040] (“Processing of packets from ingress to fabric input is shown in FIG. 4. Depicted is the flow of packets through a single FIC 400 located on a board going to 16 destination boards 401-416. A network processor 450 on the FIC 400 classifies all outgoing traffic to a particular flow. The flow defines a set of parameters including the 16 destination boards 401-416, the path by which it will get to the desti-nation board, i.e., one of the 16 fabric ports 421-436, and classification (AF, EF or BE). The individual packets end up in virtual output (send) queues 460 according to the classi-fication. There is a separate set of virtual output (send) queues 460 (AF, EF or BE) for every destination point, i.e., fabric port 421-436, per FIC. The fabric ports 421-436 and N local ports 490 have both send and receive terminals.”) Lebizay at [0041] (“Multiple virtual output (send) queues 460 are main-tained per FIC for two reasons. The primary reason is that multiple virtual output queues 460 provide a steering func-tion. After classification, the destination board 401-416 has been identified. If the packets were put back into the same (common) queue, the destination board 401-416 information would have to be carried along with the packet. This scheme is in fact done in some implementations. However, separat-ing the packets into separate queues is another way to carry the information.”) Lebizay at [0043] (“InfiniBand offers link layer Virtual Lanes (VLs) to support multiple logical channels (i.e. multiplexing) on the same physical link. Infiniband offers up to 16 virtual lanes per link. VLs provide a mechanism to avoid head-of-line blocking and the ability to support Quality of Service (QoS). The difference between a Virtual Lane and a Service Level (SL) is that a Virtual Lane is the actual logical lane (mul-tiplexed) used on a given point-to-point link. The Service Level stays constant as a packet traverses the fabric, and specifies the desired service level within a subnet. The SL (AF, EF or BE) is included in the link header, and each switch maps the SL to a VL supported by the destination link. A switch supporting a

No.	’740 Patent Claim 1	Lebizay
		limited number of virtual lanes will map the SL field to a VL it supports. Without preserving the SL, the desired SL (AF, EF or BE) would be lost in this mapping, and later in the path, a switch supporting more VLs would be unable to recover finer granularity of SLs between two packets mapped to the same VL.”) Lebizay at [0044] (“In the case of the FIC 400 shown in FIG. 4, the virtual lanes are established by classifying, sorting, and placing packets, according to service level and destination, into the virtual output queues 460. The packets are then multiplexed onto a link via a fabric port 421-436 to be transmitted to a destination board 401-416. The classifying, sorting, placing and multiplexing are carried out by the network processor 450.”) Lebizay at Figure 4

No.	’740 Patent Claim 1	Lebizay
		 400 16 Destination Boards 401-416 Network processor 450 Virtual Output Queues on Local Side 460 16 fabric ports with Integrated SERDES 421-436 N local ports 490 Simple buffers On Fabric side 470 FIG. 4 Lebizay at [0045] (“FIG. 4 also illustrates simple buffers (receive queues) 470 on the egress fabric side. The 16 fabric ports 421-436 have integrated serializer/de-serializers (SERDES). By allocating buffers (queues) to each port on both the receive and transmit sides, traffic can

No.	’740 Patent Claim 1	Lebizay
		be managed without head of line blocking or congestion. This enables each board to have a private communications channel with every other board in the system.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Lebizay in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 1[f] of Exhibit E-1 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Basso discloses using a hash function and index table to select for blade/port combinations over which to send the packet over a user port and a switch fabric link. Basso further discloses that this selection is performed based upon packet information. Basso at [0010] (“Upon a network processor receiving a packet of data, the network processor may index into a table, commonly referred to as a forwarding table, to determine the table associated with a particular logical interface as well as the next destination address. The forwarding table may comprise a plurality of entries where each entry may comprise information indicating a particular table associated with a particular logical interface as well as the next destination address. Each logical interface may be associated with a table storing a plurality of entries containing blade/ port combinations as discussed further below. In one embodiment, an entry may be indexed in the forwarding table using a destination address in the received packet header.”) Basso at [0011] (“A hash function may then be performed on the received packet to generate a hash value. In one embodiment, a hash function may be performed on the source and destination address in the packet header to generate a hash value.”) Basso at [0012] (“The hash value generated may be used to index into the table associated with a particular logical interface. Upon indexing into the table associated with the logical interface, an appropriate blade/port combination may be identified to transmit the received packet of data. In one embodiment, a blade/port combination may be selected in the indexed entry of the table associated with the logical interface by using a portion of the bits of the hashed value. The received packet may then be transmitted through the identified blade/port combination to

No.	’740 Patent Claim 1	Lebizay
		the next destination (next destination previously identified by the next destination address in the forwarding table).”) Basso at [0035] (“By logically grouping a plurality of ports 404 coupled to a particular network device into a logical interface 405, network processor 403 may be configured to transmit processed packets to that particular network device via any blade 402/port 404 combination grouped in that logical interface 405. For example, referring to FIG. 4, ports 404A-404I are physically connected to router 104B. If ports 404A-404I were logically grouped into logical interface 405, then a particular network processor 403, e.g., network processor 403A, may be configured to transmit processed packets that are determined to be transmitted to router 104B through any of ports 404A-404I in blades 402A-C, respectively. Network processor 403, e.g., network processor 403A, may be configured to transmit the processed packets to router 104B through ports 404, e.g., ports 404D-I, not in its blade 402, e.g., blade 402A, by forwarding the processed packets to switch fabric 401 which may then direct the processed packets to another appropriate physical blade 402/port 404 combination. Network processor 403, e.g., network processor 403A, may further be configured to transmit the processed packets to router 104B through any ports 404, e.g., ports 404A-C, in its blade 402, e.g., blade402A, instead of just one physical port 404 in its blade 402, e.g., blade 402A. A more detailed description of routing packets implementing logical interface(s) 405 is provided below in FIG. 5.”) Basso at [0040] (“In step 502, network processor 403, e.g., network processor 403A, may receive a packet of data from switch fabric 401. Upon receiving the packet of data, network processor 403, in step 503, may index into a table, commonly referred to as a forwarding table, to determine the table associated with a particular logical interface 405 as well as the next destination address, i.e., the next hop address. The forwarding table may comprise a plurality of entries where each entry may comprise information indicating a particular table associated with a particular logical interface 405 as well as the next destination address. Each logical interface 405 may be associated with a table storing a plurality of entries containing blade 402/port 404 combinations as discussed further below. In one embodiment, an entry may be indexed in the forwarding table using a destination address in the received packet header. It is noted that an entry may be indexed in the forwarding table usingother means and that such means would be recognized by an artisan of ordinary skill in the art. It is further noted that

No.	’740 Patent Claim 1	Lebizay
		embodiments implementing such means would fall within the scope of the present invention.”) Basso at [0041] (“In step 504, a hash function may be performed on the received packet to generate a hash value. In one embodiment, a hash function may be performed on the source and destination address in the packet header to generate a hash value. It is noted that in other embodiments a hash function may be performed on different fields, e.g., port, type of service, in the received packet to generate a hash value.”) Basso at [0042] (“In step 505, the hash value generated in step 504 may be used to index into the table associated with a particular logical interface 405 determined in step 503. Upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be identified in step 506 to transmit the received packet of data as explained below.”) Basso at [0043] (“As stated above, the table associated with a particular logical interface 405 may comprise a plurality of entries where each entry may comprise a threshold value associated with a particular blade 402/port 404 combination. The threshold value may represent a percentage of the total number of packets received by router 104A that may be transmitted through the blade 402/port 404 combination associated with that threshold value. In one embodiment, the threshold value may be updated periodically by a user, e.g., system administrator, in control of router 104, e.g., router 104A. For example, the threshold value, e.g., twenty percent of the number of packets received by router 104A, associated with a particular blade 402/port 404 combination may be updated by lowering the threshold value by one percent during each update. An example of an entry of the table associated with a particular logical interface 405 is shown in Table 1 below:

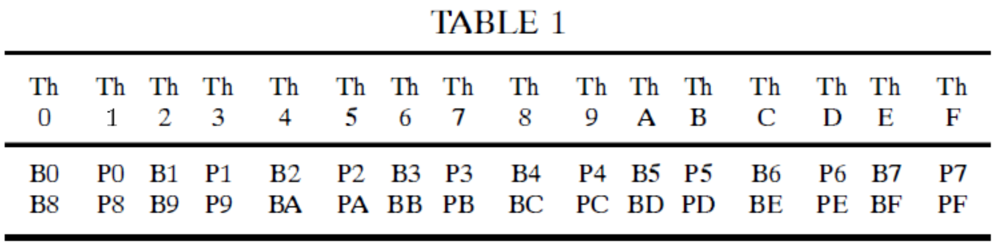
<table>
<tr><th>No.</th><th>’740 Patent Claim 1</th><th>Lebizay</th></tr>
<tr><td></td><td></td><td>


TABLE 1

Th 0	Th 1	Th 2	Th 3	Th 4	Th 5	Th 6	Th 7	Th 8	Th 9	Th A	Th B	Th C	Th D	Th E	Th F
B0	P0	B1	P1	B2	P2	B3	P3	B4	P4	B5	P5	B6	P6	B7	P7
B8	P8	B9	P9	BA	PA	BB	PB	BC	PC	BD	PD	BE	PE	BF	PF

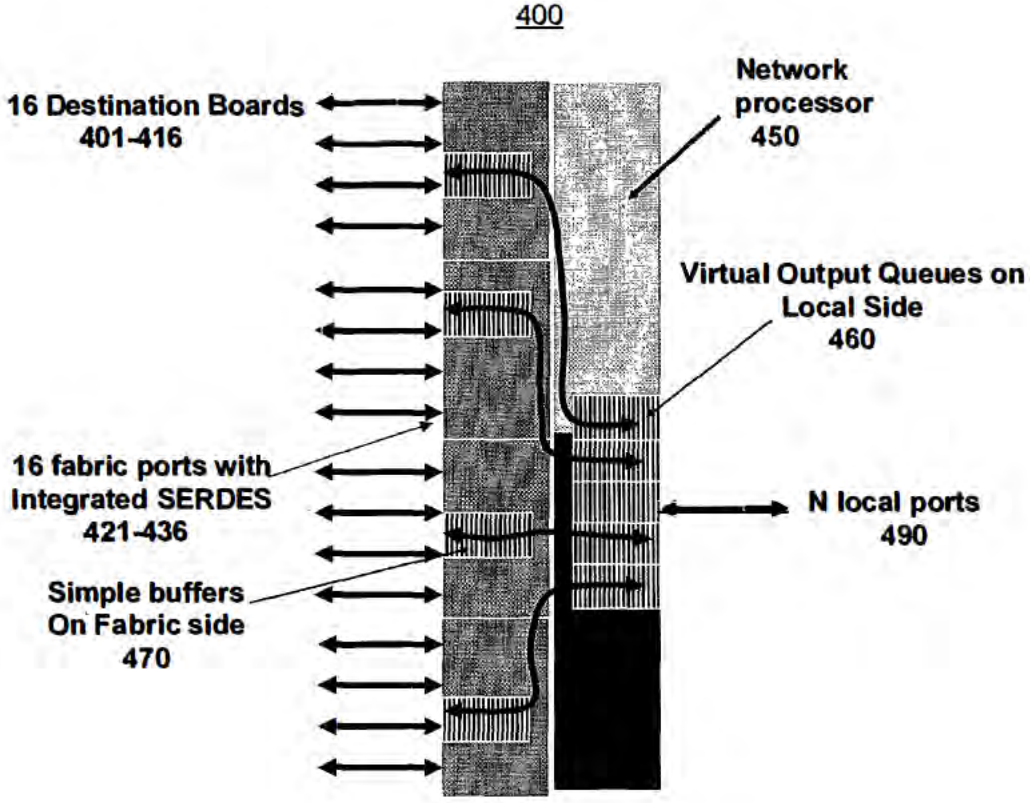
Basso at [0044] (“Table 1 above illustrates an exemplary entry in the table associated with a particular logical interface 405. Each entry may comprise a plurality of threshold values (16 threshold values in exemplary Table 1) where each threshold value is associated with a particular blade 402/port 404 combination. For example, threshold value (Th0) is associated with blade B0/port P0 combination where blade BO may refer to a particular blade 402, e.g., blade 402B, and port PO may refer to a particular port 404, e.g., port 404E. Threshold value (Thl) is associated with blade Bl/port Pl combination and so forth. As stated above, each threshold value may represent a percentage of the total number of packets received by router 104A that may be transmitted through the blade 402/port 404 combination associated with that threshold value. For example, threshold value (Th0) may represent a percentage of the total number of packets received by router 104A that may be transmitted through port PO in blade BO. If port PO refers to port 404D and blade BO refers to blade 402B, then if Th0 has a value of twenty percent, a maximum of twenty percent of the total packets received by router 104A may be transmitted through port 404D in blade 402B.”)

Basso at [0045] (“As stated above, upon indexing into the table associated with the logical interface 405 determined in step 503, an appropriate blade 402/port 404 combination may be identified in step 506 to transmit the received packet of data. In one embodiment, the hash value generated in step 504 may be used to select a particular threshold value and hence a blade 402/port 404 combination associated with the selected threshold value. In one embodiment, a portion of the bits of the hash value, e.g., most significant bits, may be used to select a particular threshold value in the entry indexed in step 505. For example, referring to Table 1, since there are 16 different threshold values in each entry of the table associated with
</td></tr>
</table>

No.	’740 Patent Claim 1	Lebizay
		logical interface 405, only four bits of the hash value generated in step 504 may be used to select a threshold value. Upon selecting a threshold value, the blade 402/port 404 combination associated with the selected threshold value may be used to transmit the received packet.”) As another example, Wiher discloses using cell header information to at each node, i.e., frame attributes, to select and route the ATM data cell over a data link and selected backplane. Wiher at 3:43-65 (“In general, in another aspect, the invention features an apparatus for communicating data cells between a data link and a backplane. The apparatus includes transceiver circuitry to transmit and receive data cells over a data link and a plurality of backplane interfaces each including at least one cell signal terminal. Each of the backplane interface is coupled to a backplane interconnection circuit. Each backplane interconnection circuit transmits and receives cells over the cell signal terminals of its associated backplane interface. The apparatus also includes de-multiplexing circuitry coupling the transceiver circuitry to each of the backplane interconnection circuits. The de-multiplexing circuitry receives a data cell from the transceiver circuitry, select a backplane interconnection circuit associated with the data cell, and provide the data cell to the selected backplane interconnection circuit for transmission over the cell signal terminals of the associated backplane interface. The apparatus also includes multiplexing circuitry coupling the plurality of backplane interconnection circuits to the transceiver circuitry. The multiplexing circuitry receives data cells from each of the backplane interconnection circuits and provide the received data cells to the transceiver circuitry.”) Wiher at 3:66-4:22 (“Implementations of the invention may include one or more of the following features. The backplane interconnection circuits may independently receive and transmit data cells over the plurality of backplane interfaces. The de-multiplexing circuitry may select a backplane interface based on data in the header field of the data cell. The apparatus may include header translation circuitry to alter header data in cells sent between the plurality of backplane interfaces and the transceiver circuitry. Each of the plurality of backplane interfaces may include separate terminals to receive cells and separate terminals to transmit cells. The terminals to transmit cells may include a first and second control terminal and at

No.	’740 Patent Claim 1	Lebizay
		least one outgoing cell data terminal. A backplane interface's backplane interconnection circuitry may accepts a signal on the first control terminal as indicating that a cell may be sent over the interface, asserts a 15 signal on the second control terminal to indicate that a cell is being transmitted, and transmits data bits of the cell on the outgoing cell data terminal. Each backplane interface may include a single outgoing cell data terminal and each bit of the cell may be serially transmitted over the single outgoing cell data terminal. Each backplane interface may include multiple outgoing cell data terminals and bits of the cell may be sent in parallel over the eight outgoing cell data terminals.”)
1[g]	sending the data frame over the selected first and second physical links,	Lebizay discloses sending the data frame over the selected first and second physical links. For example, Lebizay discloses sending packets over specific local ports and backplane interconnect fabric ports connecting the FICs and backplane. Lebizay at [0040] (“Processing of packets from ingress to fabric input is shown in FIG. 4. Depicted is the flow of packets through a single FIC 400 located on a board going to 16 destination boards 401-416. A network processor 450 on the FIC 400 classifies all outgoing traffic to a particular flow. The flow defines a set of parameters including the 16 destination boards 401-416, the path by which it will get to the desti-nation board, i.e., one of the 16 fabric ports 421-436, and classification (AF, EF or BE). The individual packets end up in virtual output (send) queues 460 according to the classi-fication. There is a separate set of virtual output (send) queues 460 (AF, EF or BE) for every destination point, i.e., fabric port 421-436, per FIC. The fabric ports 421-436 and N local ports 490 have both send and receive terminals.”) Lebizay at [0041] (“Multiple virtual output (send) queues 460 are main-tained per FIC for two reasons. The primary reason is that multiple virtual output queues 460 provide a steering func-tion. After classification, the destination board 401-416 has been identified. If the packets were put back into the same (common) queue, the destination board 401-416 information would have to be carried along with the packet. This scheme is in fact done in some

No.	’740 Patent Claim 1	Lebizay
		implementations. However, separat-ing the packets into separate queues is another way to carry the information.”) Lebizay at [0043] (“InfiniBand offers link layer Virtual Lanes (VLs) to support multiple logical channels (i.e. multiplexing) on the same physical link. Infiniband offers up to 16 virtual lanes per link. VLs provide a mechanism to avoid head-of-line blocking and the ability to support Quality of Service (QoS). The difference between a Virtual Lane and a Service Level (SL) is that a Virtual Lane is the actual logical lane (mul-tiplexed) used on a given point-to-point link. The Service Level stays constant as a packet traverses the fabric, and specifies the desired service level within a subnet. The SL (AF, EF or BE) is included in the link header, and each switch maps the SL to a VL supported by the destination link. A switch supporting a limited number of virtual lanes will map the SL field to a VL it supports. Without preserving the SL, the desired SL (AF, EF or BE) would be lost in this mapping, and later in the path, a switch supporting more VLs would be unable to recover finer granularity of SLs between two packets mapped to the same VL.”) Lebizay at [0044] (“In the case of the FIC 400 shown in FIG. 4, the virtual lanes are established by classifying, sorting, and placing packets, according to service level and destination, into the virtual output queues 460. The packets are then multiplexed onto a link via a fabric port 421-436 to be transmitted to a destination board 401-416. The classifying, sorting, placing and multiplexing are carried out by the network processor 450.”) Lebizay at Figure 4

No.	’740 Patent Claim 1	Lebizay
		 400 16 Destination Boards 401-416 Network processor 450 Virtual Output Queues on Local Side 460 16 fabric ports with Integrated SERDES 421-436 N local ports 490 Simple buffers On Fabric side 470 FIG. 4 Lebizay at [0045] (“FIG. 4 also illustrates simple buffers (receive queues) 470 on the egress fabric side. The 16 fabric ports 421-436 have integrated serializer/de-serializers (SERDES). By allocating buffers (queues) to each port on both the receive and transmit sides, traffic can

No.	’740 Patent Claim 1	Lebizay
		be managed without head of line blocking or congestion. This enables each board to have a private communications channel with every other board in the system.”)
1[h]	said sending comprising communicating along at least one of said bi-directional links.	Lebizay discloses said sending comprising communicating along at least one of said bi-directional links. *See supra at* 1[b], 1[d], 1[g].

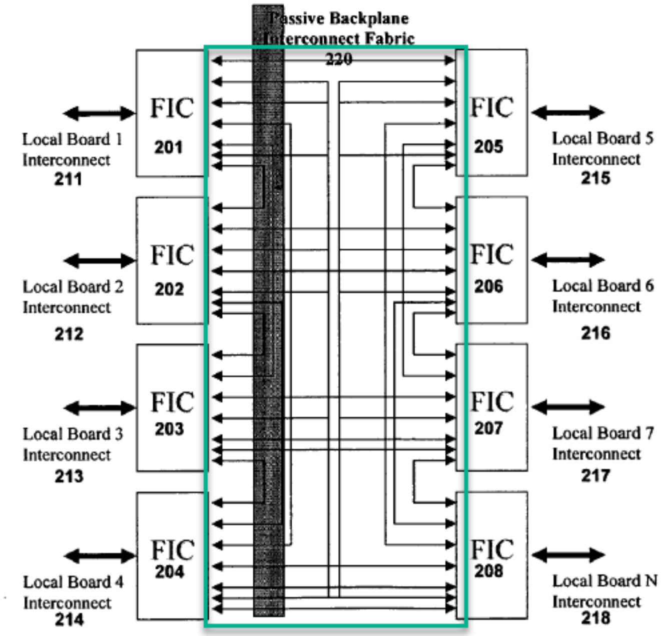
No.	’740 Patent Claim 2	Lebizay
2[a]	The method according to claim 1, wherein the network node comprises a user node, and	Lebizay discloses the method according to claim 1, wherein the network node comprises a user node. For example, Lebizay discloses a way of communicating over user devices, servers, storage devices, etc. *See supra* at Claim 1. Lebizay at [0006] (“The InfiniBand architecture is an 1/0 infrastructure technology that simplifies and speeds server-to-server connections and links to other server related systems, such as remote storage and network devices. The Infiniband fabric or mesh is the central network of connections between nodes, i.e., servers, and remote networking and storage devices within a data center. Infiniband architecture also constructs highly available data centers through multiple levels of redundancy. By connecting nodes via multiple links, Infiniband systems continue to perform even if one link fails. For increased reliability, multiple switches in a fabric provide redundant paths that allow for seamless re-routing of data through the fabric should a link between switches fail. Completely redundant fabrics may be config-ured for the highest level of reliability and may continue to perform even if an entire fabric fails.”) Lebizay at [0009] (“Infiniband uses Internet Protocol Version 6 (IPv6) headers natively, and can connect to Local Area Network (LAN) and Wide Area Network (WAN) switches and

No.	’740 Patent Claim 2	Lebizay
		routers with the TCA providing a seamless transition between the system area network and external networks. InfiniBand defines network layers up to the transport layer and offers all four combinations of reliable/unreliable and connection datagram service. The Infiniband transport protocol is used within the system area network, but other transport protocols can be accessed by sending raw packets via a TCA. TCAs provide connections to storage, fibre channel networks, and other 1/0 nodes, and include an 1/0 controller specific to the device's protocol, be it Small Computer Systems Interface (SCSI), Ethernet, etc. A TCAincludes an Infiniband protocol engine that dramatically accelerates the performance of critical Infiniband transport functions in the TCA's hard-ware, achieving aggregate internal transaction switching throughput speeds of 150 gigabits per second. TCAs are highly optimized for Infiniband target applications such as bridging from Infiniband devices (switches) to local busses i.e., Gigabit Ethernet, Fibre Channel, and Transport Control Protocol/Internet Protocol TCP/IP devices, as well as next-generation 1/0 protocols.”) Lebizay at [0034] (“Links are the connections between boards (or node to node, or switch to switch, in general). They are capable of transmitting no more than a certain amount of information at any one time, known as bandwidth. Bandwidth may be measured, for example, in bits/second. In general, there is no restriction on the number of boards (N) except that it be greater than 1.”)
2[b]	wherein sending the data frame comprises establishing a communication service between the user node and the communication network.	Lebizay discloses wherein sending the data frame comprises establishing a communication service between the user node and the communication network. For example, Lebizay discloses connecting multiple entities in a system by sending data between network devices and a communication network. Lebizay at [0006] (“The InfiniBand architecture is an 1/0 infrastructure technology that simplifies and speeds server-to-server connections and links to other server related systems, such as remote storage and network devices. The Infiniband fabric or mesh is the central network of connections between nodes, i.e., servers, and remote networking and storage devices within a data center. Infiniband architecture also constructs highly available data centers through multiple levels of redundancy. By connecting nodes via multiple links,

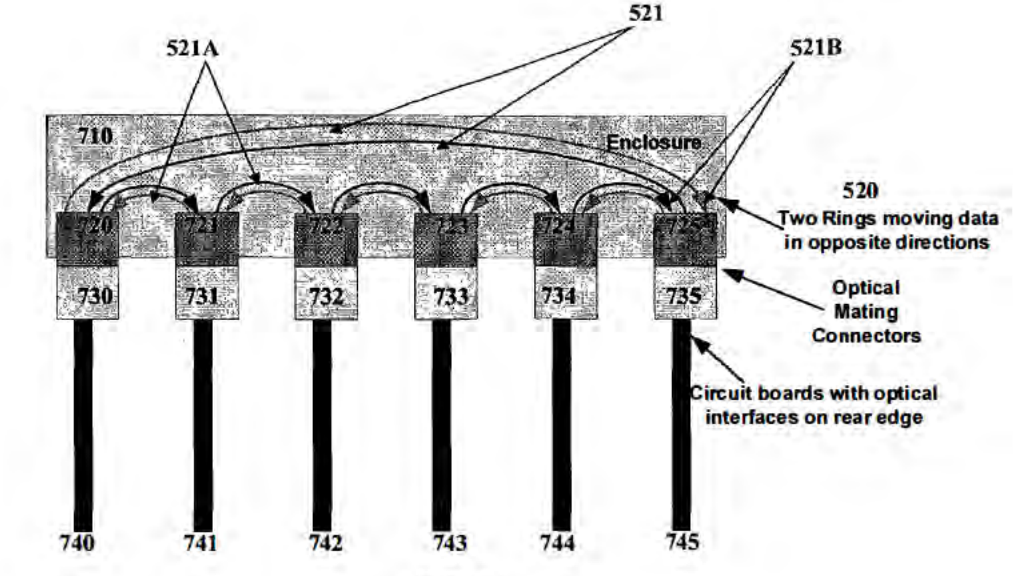
No.	’740 Patent Claim 2	Lebizay
		Infiniband systems continue to perform even if one link fails. For increased reliability, multiple switches in a fabric provide redundant paths that allow for seamless re-routing of data through the fabric should a link between switches fail. Completely redundant fabrics may be config-ured for the highest level of reliability and may continue to perform even if an entire fabric fails.”) Lebizay at [0028] (“Embodiments of the present invention exists within the context of connecting multiple entities within a system (specifically multiple boards within a chassis), using mul-tiple TCAs. The components consist of multi-port TCAs and a meshed backplane that is equipped to mount boards via connectors on the backplane. A multi-port TCA capable of performing multiple bridging functions simultaneously i.e., bridging from an Infiniband meshed backplane to multiple local busses, i.e., Gigabit Ethernet, Fibre Channel, and TCP/IP devices, is referred to as a Fabric Interconnect Chip (FIC). FIG. 1 depicts the connection of a FIC 100 that interconnects between local components on a board (local busses) 120 and a backplane mesh fabric 110.”) Lebizay at [0030] (“The FIC provides data queuing first in first out (FIFOs) on both its inbound and outbound sides for each of its multiple ports. By placing FICs on every board within a chassis that is connected to the meshed backplane, the boards can inter-communicate with each other in a "meshed" topology. Therefore, every board has a point to point connection with every other board in the backplane, making a complete cross-connect without the need for a separate switch device.”) Lebizay at [0037] (“For example, an entity may set up a Service Level Agreement (SLA) with a network service provider (the entity that is providing the network), that specifies that the entity's traffic will always have available a certain band-width (e.g., 10 Megabits per second, or 10 Mbps) and latency (e.g., less than 1 millisecond, or ms). Then, when-ever a packet is detected that comes from or is going to that entity, the packet should receive special handling. If the overall flow to the entity is currently less than 10 Mbps, then the packet should get through without being dropped and with a latency of less than 1 ms. This type of flow is said to be handled by Assured Forwarding (AF). Packets arriving when the current flows are greater than 10 Mbps will be handled differently, perhaps as Best Effort (BE) traffic (see below).”)

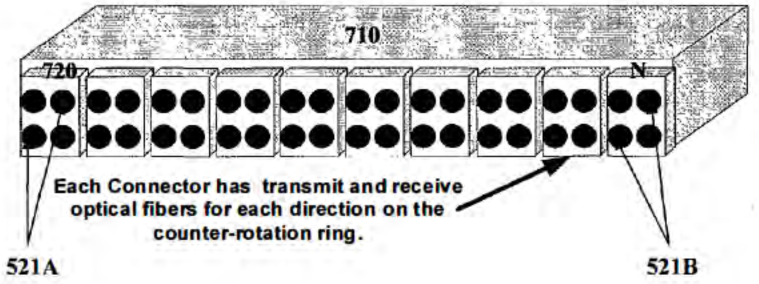
No.	’740 Patent Claim 2	Lebizay
		Lebizay at [0049] (“One alternative is to use optical fibers to create an optical backplane fabric. The optical backplane may consist of either a mesh of individual optical fibers or a redundant pair of fibers configured in a counter-rotating ring topology. Either way, fibers represent a much wider bandwidth poten-tial than copper, although a ring requires less fibers than a mesh. When using a mesh, each board will have a direct connection to every other board in the chassis using an optical fiber. Each port on a board may have a transmitter/ receiver pair connected to a single fiber.”)

No.	’740 Patent Claim 3	Lebizay
3	The method according to claim 1, wherein the second physical links comprise backplane traces formed on a backplane to which the one or more interface modules are coupled.	Lebizay discloses the method according to claim 1, wherein the second physical links comprise backplane traces formed on a backplane to which the one or more interface modules are coupled. For example, Lebizay discloses backplane connectors and traces that are formed on a backplane and connect to the FICs. *See supra* at Claim 1. Lebizay at [0028] (“Embodiments of the present invention exists within the context of connecting multiple entities within a system (specifically multiple boards within a chassis), using mul-tiple TCAs. The components consist of multi-port TCAs and a meshed backplane that is equipped to mount boards via connectors on the backplane. A multi-port TCA capable of performing multiple bridging functions simultaneously i.e., bridging from an Infiniband meshed vbackplane to multiple local busses, i.e., Gigabit Ethernet, Fibre Channel, and TCP/IP devices, is referred to as a Fabric Interconnect Chip (FIC). FIG. 1 depicts the connection of a FIC 100 that interconnects between local components on a board (local busses) 120 and a backplane mesh fabric 110.”) Lebizay at [0029] (“FIG. 2 depicts connection of multiple intercon-nected FICs. Each board 211-218 on the mesh 220 contains one FIC 201-208 . Each FIC may provide an

No.	’740 Patent Claim 3	Lebizay
		interconnection to every other board in the shelf (chassis). FIG. 2 shows an 8-way (8 board) mesh, but any size mesh is realizable, hence the 8,h board 218 is labeled "N". The lines drawn show how the traces from each FIC 'port' travels across the passive backplane interconnect fabric 220 to its corresponding 'port' on another FIC, (located on another board).”) Lebizay at Figure 2 (annotation added)  Passive Backplane Interconnect Fabric 220 FIC 201 FIC 202 FIC 203 FIC 204 FIC 205 FIC 206 FIC 207 FIC 208 Local Board 1 Interconnect 211 Local Board 2 Interconnect 212 Local Board 3 Interconnect 213 Local Board 4 Interconnect 214 Local Board 5 Interconnect 215 Local Board 6 Interconnect 216 Local Board 7 Interconnect 217 Local Board N Interconnect 218 FIG. 2 Lebizay at [0032] (“Since each FIC (201-208) has queuing (221-228) on both the inbound and outbound sides, coupled with the inherent cross-connect in the backplane 220, an extremely efficient logically distributed switch can be built at much lower cost, greater reliability, higher

No.	’740 Patent Claim 3	Lebizay
		efficiency, lower power and lower latency than a typical, centrally switched topol-ogy, or any architecture that places a switch on every board of a meshed topology.) Lebizay at [0046] (“Today, boards in a common chassis interconnect to each other over copper traces embedded in the chassis backplane. Existing backplanes use copper traces within a PCB substrate to establish the communications paths between boards. Topologies such as common bus PCI and switched star (i.e. some Ethernet approaches) are popular today. The copper traces and the electrical drivers that communicate over them are limited in terms of their capac-ity to carry data bandwidth with respect to the Local Area Network (LAN) and Wide Area Network (WAN) optical signals that these boards are intended to terminate.”) Lebizay at [0058] (“Referring to FIG. 7, the basic components of the dual counter-rotating ring optical backplane include a set of fibers 521, arranged in a dual-counter-rotating ring 520, housed in a overlay module 710 that allows fiber cavity mechanical connectors 720-725 to mount to the overlay module 710 front surface. Each mechanical connector 720-725 mates up to a mated connector 730-735 that mounts to the individual boards 740-745 in a chassis. Mechanical connectors 720-725 may also contain an OADM 541 ring adaptor previously depicted in FIG. 5 and FIG. 6.”) Lebizay at Figure 7

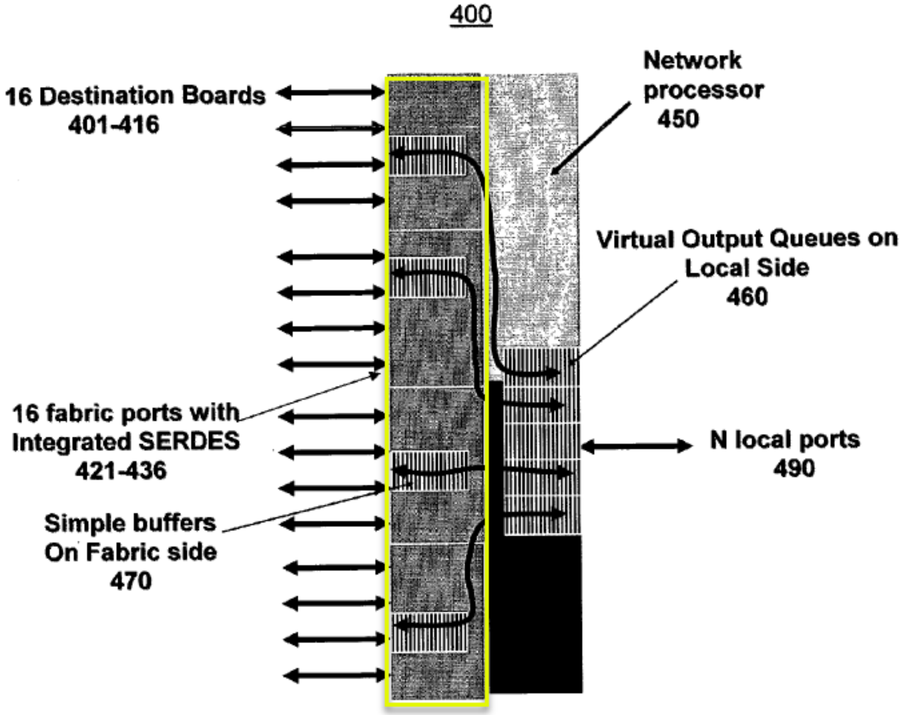
No.	’740 Patent Claim 3	Lebizay
		 521 521A 521B 710 Enclosure 520 Two Rings moving data in opposite directions 720 721 722 723 724 725 730 731 732 733 734 735 Optical Mating Connectors Circuit boards with optical interfaces on rear edge 740 741 742 743 744 745 FIG. 7 Lebizay at [0059] (“FIG. 8 illustrates an overlay module according to an embodiment of the present invention. The overlay mod-ule 710 includes a series of connectors 720-N mounted to the front of the enclosure. The overlay module 710 contains the optical fibers 521 that form the dual counter-rotating ring 520. Referring to FIG. 7 and FIG. 8, each connector 720-725 has transmit 521A and receive 521B optical fibers for each direction on the counterrotation ring 520. Each connector 720-725 (720-N) has transmit 521A and receive fiber 521B in each direction to each of its nearest neighbors. Fibers 521 pass between the two end connectors 720, 735 (720, N) to complete the rings 520.”) Lebizay at Figure 8

No.	’740 Patent Claim 3	Lebizay
		The enclosure contains the optical fibers that form the dual counter-rotating ring. Each connector has transmit and receive fiber pairs in each direction to each of its nearest neighbors. Fibers pass between the two end connectors to compelete the rings.  Each Connector has transmit and receive optical fibers for each direction on the counter-rotation ring. FIG. 8

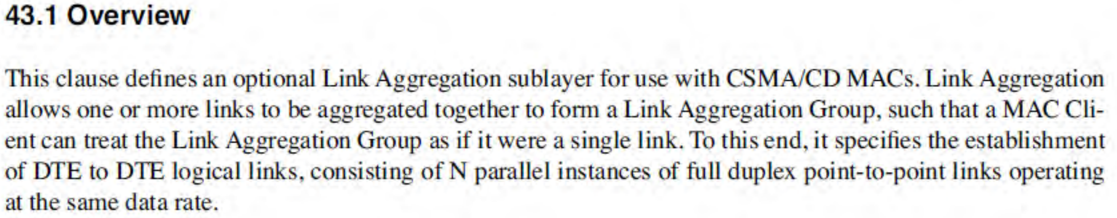
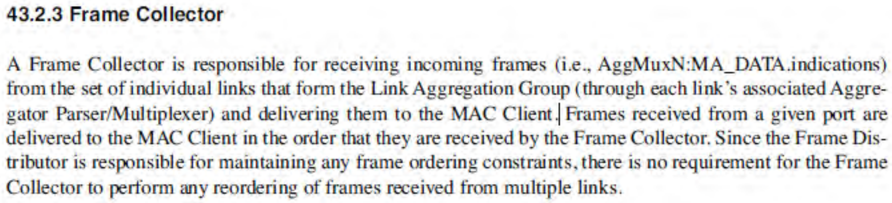
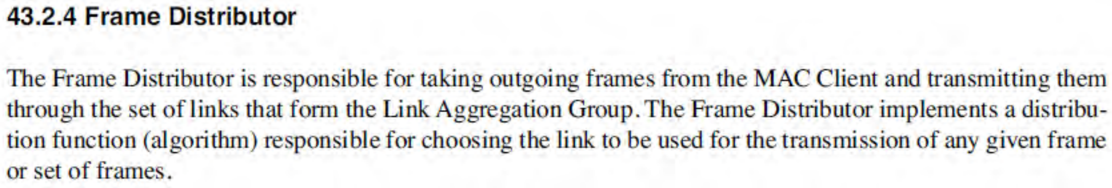
No.	’740 Patent Claim 4	Lebizay
4[preamble]	A method for communication, comprising:	Lebizay discloses a method for communication. *See supra* at 1[preamble].
4[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	Lebizay discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].

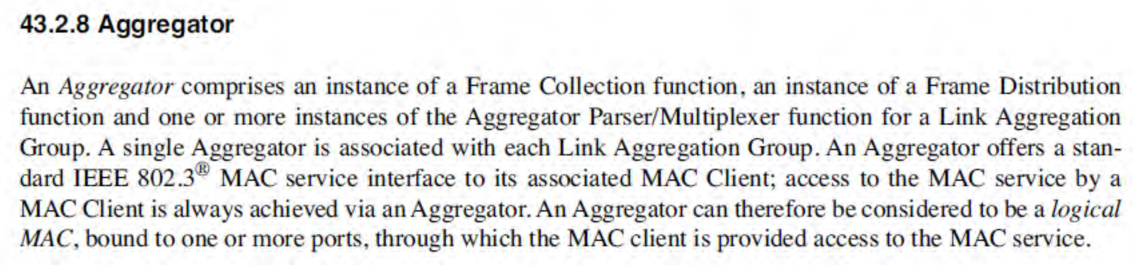
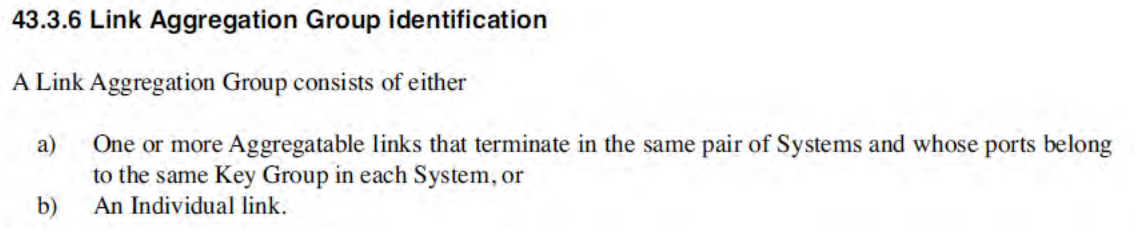
No.	’740 Patent Claim 4	Lebizay
4[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Lebizay discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra* at 1[c].
4[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	Lebizay discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra* at 1[e].
4[d]	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	Lebizay discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. *See supra* at 1[f].
4[e]	sending the data frame over the selected first and	Lebizay discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].

No.	’740 Patent Claim 4	Lebizay
	second physical links,	
4[f]	at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group.	Lebizay discloses at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group. For example, Lebizay discloses Virtual Lanes/virtual output queues that multiplex multiple ports in a single logical link. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Lebizay is found to not meet this limitation, at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group would have been obvious to a person having ordinary skill in the art, as explained below. Lebizay at [0009] (“Infiniband uses Internet Protocol Version 6 (IPv6) headers natively, and can connect to Local Area Network (LAN) and Wide Area Network (WAN) switches and routers with the TCA providing a seamless transition between the system area network and external networks. InfiniBand defines network layers up to the transport layer and offers all four combinations of reliable/unreliable and connection datagram service. The Infiniband transport protocol is used within the system area network, but other transport protocols can be accessed by sending raw packets via a TCA. TCAs provide connections to storage, fibre channel networks, and other 1/0 nodes, and include an 1/0 controller specific to the device's protocol, be it Small Computer Systems Interface (SCSI), Ethernet, etc. A TCAincludes an Infiniband protocol engine that dramatically accelerates the performance of critical Infiniband transport functions in the TCA's hard-ware, achieving aggregate internal transaction switching throughput speeds of 150 gigabits per second. TCAs are highly optimized for Infiniband target applications such as bridging from Infiniband devices (switches) to local busses i.e., Gigabit Ethernet, Fibre Channel, and Transport Control Protocol/Internet Protocol TCP/IP devices, as well as next-generation 1/0 protocols.”) Lebizay at Figure 4 (annotation added)

No.	’740 Patent Claim 4	Lebizay
		 400 16 Destination Boards 401-416 Network processor 450 Virtual Output Queues on Local Side 460 16 fabric ports with Integrated SERDES 421-436 N local ports 490 Simple buffers On Fabric side 470 FIG. 4 Lebizay at [0040] (“Processing of packets from ingress to fabric input is shown in FIG. 4. Depicted is the flow of packets through a single FIC 400 located on a board going to 16 destination boards 401-416. A network processor 450 on the FIC 400 classifies all outgoing traffic to a particular flow. The flow defines a set of parameters including the 16 destination boards 401-416, the path by which it will get to the desti-nation board, i.e., one of the 16 fabric ports 421-436, and classification (AF, EF or BE). The individual packets end up in virtual output (send) queues 460 according to the classi-fication. There is a separate set of virtual output (send) queues 460 (AF, EF

No.	’740 Patent Claim 4	Lebizay
		or BE) for every destination point, i.e., fabric port 421-436, per FIC. The fabric ports 421-436 and N local ports 490 have both send and receive terminals.”) Lebizay at [0041] (“Multiple virtual output (send) queues 460 are maintained per FIC for two reasons. The primary reason is that multiple virtual output queues 460 provide a steering function. After classification, the destination board 401-416 has been identified. If the packets were put back into the same (common) queue, the destination board 401-416 information would have to be carried along with the packet. This scheme is in fact done in some implementations. However, separating the packets into separate queues is another way to carry the information.”) Lebizay at [0043] (“InfiniBand offers link layer Virtual Lanes (VLs) to support multiple logical channels (i.e. multiplexing) on the same physical link. Infiniband offers up to 16 virtual lanes per link. VLs provide a mechanism to avoid head-of-line blocking and the ability to support Quality of Service (QoS). The difference between a Virtual Lane and a Service Level (SL) is that a Virtual Lane is the actual logical lane (mul-tiplexed) used on a given point-to-point link. The Service Level stays constant as a packet traverses the fabric, and specifies the desired service level within a subnet. The SL (AF, EF or BE) is included in the link header, and each switch maps the SL to a VL supported by the destination link. A switch supporting a limited number of virtual lanes will map the SL field to a VL it supports. Without preserving the SL, the desired SL (AF, EF or BE) would be lost in this mapping, and later in the path, a switch supporting more VLs would be unable to recover finer granularity of SLs between two packets mapped to the same VL.”) Lebizay at [0044] (“In the case of the FIC 400 shown in FIG. 4, the virtual lanes are established by classifying, sorting, and placing packets, according to service level and destination, into the virtual output queues 460. The packets are then multiplexed onto a link via a fabric port 421-436 to be transmitted to a destination board 401-416. The classifying, sorting, placing and multiplexing are carried out by the network processor 450.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Lebizay in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further

No.	’740 Patent Claim 4	Lebizay
		combination with (2) each (individually, as well as one or more together) of the references identified in element 4[f] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, IEEE 802.3 discloses the aggregation of one or more links together to form a Link Aggregation Group. IEEE 802.3 at 1465  **43.1 Overview** This clause defines an optional Link Aggregation sublayer for use with CSMA/CD MACs. Link Aggregation allows one or more links to be aggregated together to form a Link Aggregation Group, such that a MAC Client can treat the Link Aggregation Group as if it were a single link. To this end, it specifies the establishment of DTE to DTE logical links, consisting of N parallel instances of full duplex point-to-point links operating at the same data rate. IEEE 802.3 at 1470  **43.2.3 Frame Collector** A Frame Collector is responsible for receiving incoming frames (i.e., AggMuxN:MA_DATA.indications) from the set of individual links that form the Link Aggregation Group (through each link’s associated Aggregator Parser/Multiplexer) and delivering them to the MAC Client. Frames received from a given port are delivered to the MAC Client in the order that they are received by the Frame Collector. Since the Frame Distributor is responsible for maintaining any frame ordering constraints, there is no requirement for the Frame Collector to perform any reordering of frames received from multiple links. IEEE 802.3 at 1471  **43.2.4 Frame Distributor** The Frame Distributor is responsible for taking outgoing frames from the MAC Client and transmitting them through the set of links that form the Link Aggregation Group. The Frame Distributor implements a distribution function (algorithm) responsible for choosing the link to be used for the transmission of any given frame or set of frames.

No.	’740 Patent Claim 4	Lebizay
		IEEE 802.3 at 1474  **43.2.8 Aggregator** An *Aggregator* comprises an instance of a Frame Collection function, an instance of a Frame Distribution function and one or more instances of the Aggregator Parser/Multiplexer function for a Link Aggregation Group. A single Aggregator is associated with each Link Aggregation Group. An Aggregator offers a standard IEEE 802.3® MAC service interface to its associated MAC Client; access to the MAC service by a MAC Client is always achieved via an Aggregator. An Aggregator can therefore be considered to be a *logical MAC*, bound to one or more ports, through which the MAC client is provided access to the MAC service. IEEE 802.3 at 1481  **43.3.6 Link Aggregation Group identification** A Link Aggregation Group consists of either a) One or more Aggregatable links that terminate in the same pair of Systems and whose ports belong to the same Key Group in each System, or b) An Individual link. For example, Ghosh discloses aggregating physical links, including ports, into aggregate port channels that form a single logical link to increase bandwidth. Ghosh at Abstract (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggregation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent gen-eration of port channel map tables.”) Ghosh at [0004] (“Neighboring nodes in a fibre channel network are typically interconnected through multiple physical links. For example, a local fibre channel switch may be connected to a

No.	’740 Patent Claim 4	Lebizay
		remote fibre channel switch through four physical links. In many instances, it may be beneficial to aggregate some of the physical links into logical links. That is, multiple physi-cal links can be combined to form a logical interface to provide higher aggregate bandwidth, load balancing, and link redundancy. When a frame is being transmitted over a logical link, it does not matter what particular physical link is being used as long as all the frames of a given flow are transmitted through the same link. If a constituent physical link goes down, the logical link can still remain operational.”) Ghosh at [0007] (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggre-gation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent generation of port channel map tables.”) Ghosh at [0008] (“In one embodiment, a method for aggregating ports in a fibre channel fabric is provided. It is determined that a plurality of local ports at a local fibre channel switch are compatible. Identifiers for the plurality of local ports are sent to a remote fibre channel switch. The remote fibre channel switch determines if a plurality of remote ports are compat-ible, the plurality of remote ports corresponding to the plurality of local ports. An indication that one or more of the remote physical ports are compatible is received. A port channel including one or more of the local ports correspond-ing to the compatible remote ports is created.”) Ghosh at [0010] (“In another embodiment, a fibre channel network is described. The fibre channel network includes a local fibre channel switch and a remote fibre channel switch. The local fibre channel switch aggregates a compatible subset of the plurality of local ports and sends identifiers for the compat-ible subset of the plurality of local ports to the remote fibre channel switch. The remote fibre channel switch determines if a subset of the plurality of remote ports are compatible. The subset of the plurality of remote ports corresponds to the compatible subset of the plurality of local ports.”)

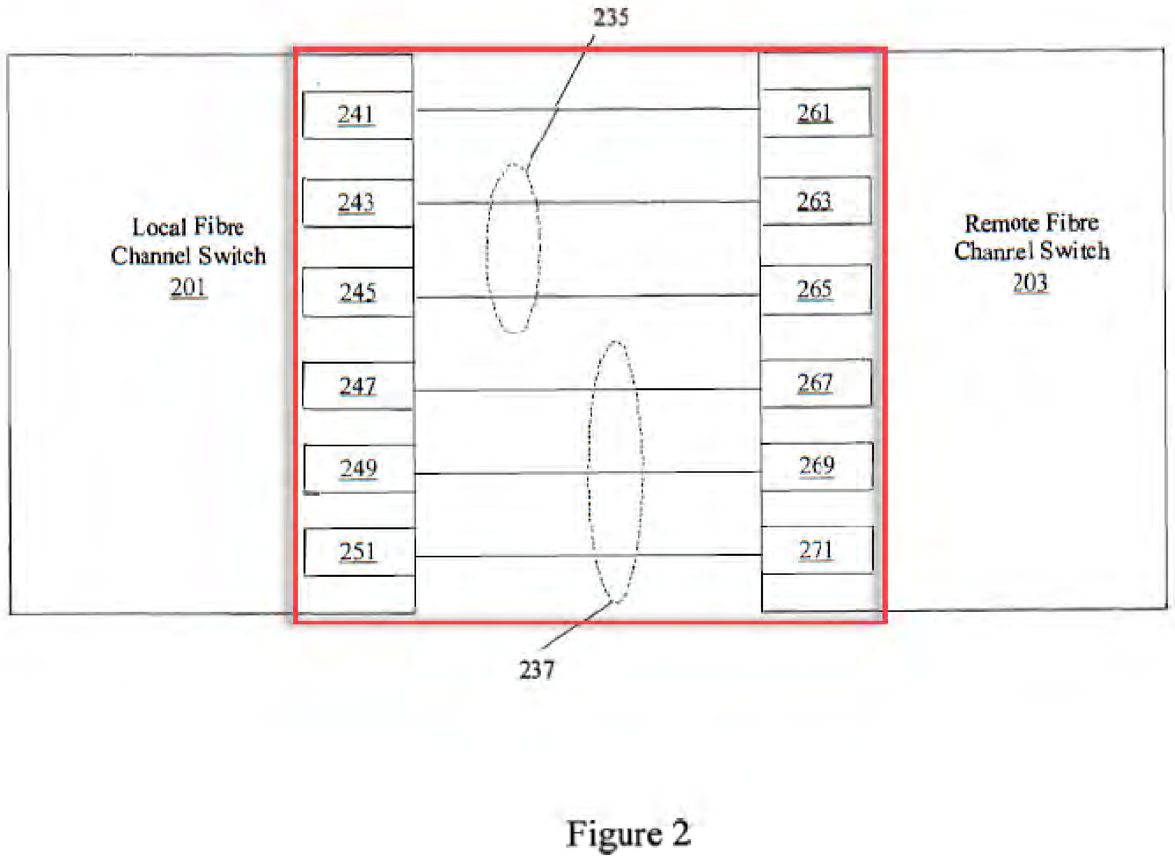
No.	’740 Patent Claim 4	Lebizay
		Ghosh at [0022] (“Switches in a fibre channel network are typically interconnected using multiple physical links. The physical links connecting a pair of switches allows transmission of data and control signals. In some instances, it is useful to aggregate multiple physical links into a logical link. Physi-cal links are also referred to herein as physical interfaces and channels while logical links are also referred to herein as logical interfaces and port channels. For example, a local switch may be connected to a remote switch through four physical links. Instead of having to transmit data through a particular physical link, the physical links can be aggregated to form one or more logical links. In one example, all four physical links are aggregated into a single logical link. Instead of having data transmitted through a particular physical link, the data can merely be transmitted over a particular logical link without regard to the particular physi-cal interface used. Aggregating physical links into a logical link allows for higher aggregated bandwidth, load balancing, and link redundancy. For example, if a particular physical link fails or is overloaded, data can still be transmitted over the logical link.”) Ghosh at [0029] (“FIG. 2 is a diagrammatic representation showing links between two switches, such as two fibre channel switches shown in FIG. 1. A local fibre channel switch 201 includes local ports 241, 243, 245, 247, 249, and 251. A remote fibre channel switch 203 includes remote ports 261, 263, 265, 267, 269, and 271. Local port 241 is coupled to remote port 261 through an individual physical link or channel. Connected ports are also referred to herein as peer ports. Local port 243 is coupled to remote port 263 and local port 245 is coupled to remote port 265. The two resulting physical links are aggregated to form port channel 235. Local ports 247, 249, and 251 are coupled to remote ports 267, 269, and 271 respectively. The three resulting physical links are aggregated to form port channel 237.”) Ghosh at [0030] (“According to various embodiments, local fibre channel switch 201 and remote fibre channel switch both have associated identifiers. In some examples, the identifiers are globally unique identifiers such as a global switch world wide names (WWNs). Each local port 241, 243, 245, 247, 249, and 251 and each remote port 261,263,265,267, 269, and 271 can also be associated with identifiers. In some examples, the identifiers are port WWNs. The port WWNs are typically used for debugging or identifying the peer port in alert or warning messages. However, according to various embodiments, the techniques of the present invention use WWNs as globally unique

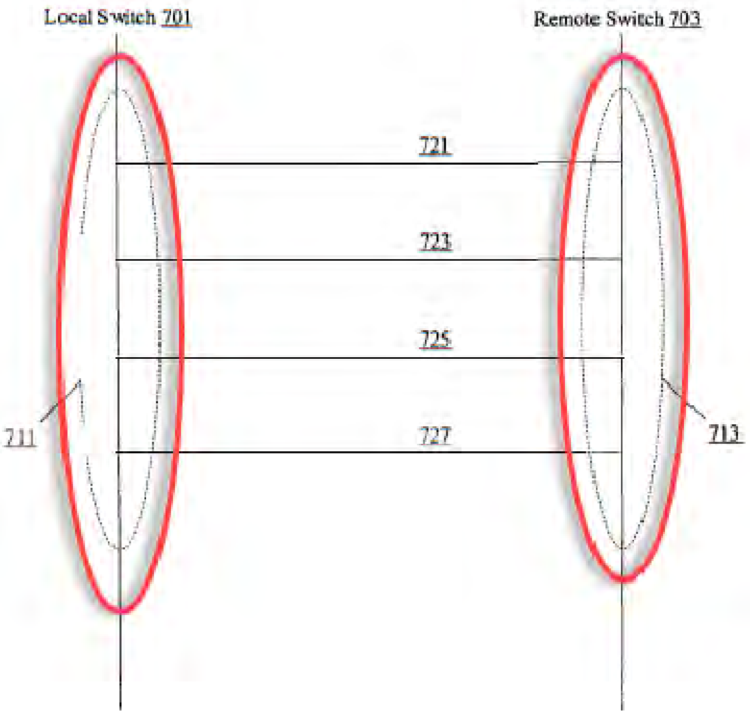
No.	’740 Patent Claim 4	Lebizay
		identifiers to aggregate ports instead of using compatibility keys which are only locally unique. Compatibility keys are mechanisms typically used by other protocols such as Ethernet for aggregation.”) Ghosh at [0033] (“A variety of parameters can be used to aggregate physical ports. FIG. 3 is a flow process diagram showing one technique for aggregating physical ports into a logical port. And 301, it is determined if auto create functionality is enabled. According to various embodiments, auto create functionality allows automatic configuration and detection of compatible physical ports as well as aggregation into one or more logical ports. Auto creation does not require user intervention. In other examples, administrators can manu-ally arrange ports for aggregation.”) Ghosh at [0037] (“FIG. 4 is an exchange diagram showing one example of a bring up procedure used for a port creating a new port channel. A local switch 401 is coupled to a remote switch 403. The local switch 401 includes a physical port Al coupled to physical port Bl included in remote switch 403. When two peer ports Al and Bl are being aggregated into a port channel, the peer switches 401 and 403 typically already know the world wide names of the individual physical peer ports. However, the peer switches only know the world wide name of their own logical port or port channel. That is, both switches have the individual physical link configured, but the link is not yet part of a port channel. At 421, a local switch 401 sends a synchronize (sync) message 411 to the remote switch 403 to begin the process of creating a port channel including ports Al and Bl.”) Ghosh at [0038] (“In some examples, the sync message 411 includes a local port channel identifier and a remote port channel identifier. In one particular example, the local port channel identifier is set to the world wide name of the local port channel assigned by the local switch 401. The remote port channel identifier is left blank to indicate that the port Al is being aggregated as part of a new port channel. The sync message 411 can also include other parameters such as channel status, channel model, or channel intent.”) Ghosh at [0042] (“When two peer ports A2 and B2 are aggregated into a port channel Cl, the peer switches 501 and 503 typically already know the world wide names of the indi-vidual physical peer ports A2 and B2 as well as the world wide name information of the port channel Cl.

No.	’740 Patent Claim 4	Lebizay
		Conse-quently, the port channel is already successfully established. According to various embodiments, local switch 501 and remote switch 503 perform parameter checking to ensure that the new physical port A2 and B2 can be safely added to the existing port channel Cl. At 521, a local switch can check configuration parameters to ensure that physical ports Al and A2 at the local switch 501 are compatible. The compatibility checking can be performed anytime. In some examples, compatibility checking is checked before a local switch 501 sends a synchronize (sync) message 511 to the remote switch 503 to begin the process of aggregating ports A2 and B2 into the port channel.) Ghosh at [0043] (“In some examples, the sync message 511 includes local port channel identifier and a remote port channel identifier. In one particular example, the local port channel identifier is set to the world wide name of the local port channel assigned by the local switch 501. The remote port channel identifier is filled with the existing port channel identifier to indicate that the port A2 is being aggregated into existing port channel C2. The sync message 511 can also include other parameters such as channel status, channel model, or channel intent.”) Ghosh at [0044] (“At 531, remote switch 503 uses the information received from the local switch 501 to verify port B2 is compatible with other port in port channel C2. In one example, configuration parameters associated with B2 are checked against configuration parameters associated with Bl. The remote switch 503 can also check if the port B2 is already assigned to a different port channel. If the port B2 is compatible with port Bl, the remote switch 503 can proceed and send a sync accept message 513 in response to the sync message 511 to indicate that the port B2 can be aggregated into the port channel. The sync accept message indicates that a port channel can now be modified. At 523, local switch 501 uses the information to update its own port channel database. However, the port channel may not yet be fully operational until the hardware configuration is completed. The local switch 501 continues hardware configuration such as line card configuration to make the port A2 part of the port channel Cl. An acknowledgment 527 is sent and received by remote switch 503 at 529. In some examples, the local switch 501 sends a commit signal 515 when hardware configuration is complete.”)

No.	’740 Patent Claim 4	Lebizay
		Ghosh at [0045] (“The remote switch 503 receives the commit signal at 533 and begins its own hardware configuration. On completion of its hardware configuration, remote switch 503 sends out a commit accept signal 517 to indicate to local switch 501 that hardware configuration is completed. According to various embodiments, local switch 501 receives the commit accept signal 517 and notifies relevant applications that the port channel is now fully operational at 525 and that port A2 has been aggregated into port channel Cl. The local switch 501 can also send out an acknowledge message 519. When the remote switch 503 receives the acknowledge, it notifies relevant applications that the port channel is operational at 535 and that port B2 has been aggregated into port channel Cl. In one embodiments, the techniques of the present invention contemplate using a two phase SYNC and COMMIT mechanism similar to the mechanism used in EPP.”) Ghosh at [0046] (“FIGS. 4 and 5 show examples of ports being aggregated into a port channel. At a particular switch, ports can be selected for aggregation into a port channel in a variety of manners. FIG. 6 is an exchange diagram showing automatic selection of ports at a switch for aggregation into a port channel. A local switch 601 is coupled to a remote switch 603. In one example, the local switch 601 includes physical ports Al, A2, A3, and A4 while remote switch 603 includes physical ports Bl, B2, B3, and B4. No port chan-nels have been formed.”) Ghosh at [0049] (“At 631, remote switch 603 uses the information received from the local switch 601 to verify that the peer ports of Al, A2, and A4 are compatible. That is, ports Bl, B2, and B4 are checked for compatibility. In one example, only ports Bl and B2 may be compatible, and consequently only ports Al, A2, Bl, and B2 can be included in the port channel. In another example, ports Bl, B2, and B4 are compatible, so ports Al, A2, A4, Bl, B2, and B4 can be aggregated into port channel Cl. According to various embodiments, if the port B2 is compatible with port Bl, the remote switch 603 can proceed and send a sync accept message 613 in response to the sync message 611 to indicate that the port B2 can be aggregated into the port channel. It should be noted that remote switch 603 can send a list indicating that ports B2 and B4 are compatible with Bl. However, the remote switch 603 sends only one compatible port B2 back for several reasons, and in the process of selection compatible port channels get priority over com-patible individual ports.”)

No.	’740 Patent Claim 4	Lebizay
		Ghosh at [0050] (“One reason is that aggregation mechanisms and techniques can be implemented more elegantly by handling ports on an individual basis. Any individual port will either start a new port channel, be added to an existing port channel, or operate stand alone. There is no need to keep track of groups of ports to be aggregated. Another reason is that fewer ports need to be locked if only a single port is being aggregated at any one time. The sync accept message indicates that a port channel can now be modified. At 623, local switch 601 receives the information and recognizes that Al and A2 can now be aggregated into port channel Cl. However, the port channel may not yet be fully operational until the hardware configuration is completed. An acknowl-edgment 627 is sent and received by remote switch 603 at 629. In some examples, the local switch 601 sends a commit signal 615 when hardware configuration is complete.”) Ghosh at [0051] (“The remote switch 603 receives the commit signal at 633 to create port channel Cl including ports Bl and B2. Hardware configuration can now be performed. On comple-tion of its hardware configuration, remote switch 603 sends out a commit accept signal 617 to indicate to local switch 601 that hardware configuration is completed. According to various embodiments, local switch 601 receives the commit accept signal 617 and notifies relevant applications that the port channel is now fully operational at 625 and that ports Al and A2 have been aggregated into port channel Cl. The local switch 601 can also send out an acknowledge message 619. When the remote switch 603 receives the acknowledge, it notifies relevant applications that the port channel is fully operational at 635 and that ports Bl and B2 have been aggregated into port channel Cl.”) Ghosh at [0053] (“FIG. 7 is a diagrammatic representation showing synchronous aggregation of ports into a port channel. A local switch 701 is coupled to a remote switch 703 through links 721, 723, 725, and 727. According to various embodiments, the links are being aggregated into port channel 711 at the local switch 701 and port channel 713 at the remote switch 703 in a synchronous manner. That is the peer ports corre-sponding to each link are brought up in the same order at both the local switch 701 and the remote switch 703.”) Ghosh at Figure 2 (annotation added)

No.	’740 Patent Claim 4	Lebizay
		 235 241 243 245 247 249 251 261 263 265 267 269 271 Local Fibre Channel Switch 201 Remote Fibre Chanrel Switch 203 237 Figure 2 Ghosh at Figure 7 (annotation added)

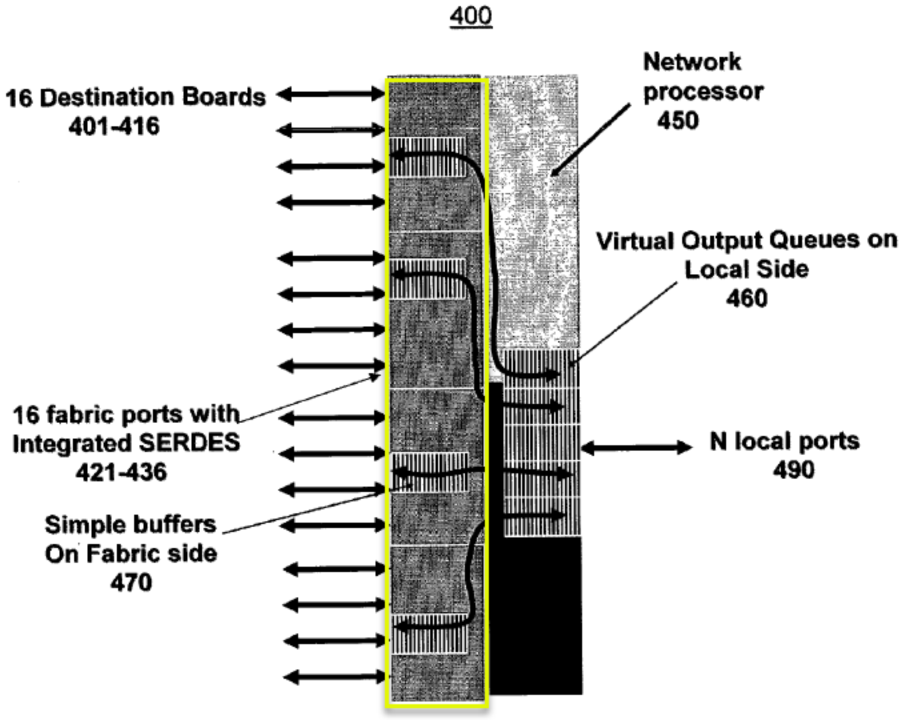
No.	’740 Patent Claim 4	Lebizay
		 Local Switch 701 Remote Switch 703 721 723 725 727 711 713 Figure 7

No.	’740 Patent Claim 5	Lebizay
5[preamble]	A method for communication, comprising:	Lebizay discloses a method for communication. *See supra at* 1[preamble].
5[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	Lebizay discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].
5[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Lebizay discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra at* 1[c].
5[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	Lebizay discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra at* 1[e].
5[d]	selecting, in a single	Lebizay discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group.

No.	’740 Patent Claim 5	Lebizay
	computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	*See supra at* 1[f].
5[e]	sending the data frame over the selected first and second physical links,	Lebizay discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].
5[f]	coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node.	Lebizay discloses coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node. For example, Lebizay discloses connecting network boards, i.e., network nodes, to FICs using Virtual Lanes/virtual output queues that multiplex multiple ports in a single logical link that aggregates bandwidth. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Lebizay is found to not meet this limitation, coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node would have been obvious to a person having ordinary skill in the art, as explained below Lebizay at [0009] (“Infiniband uses Internet Protocol Version 6 (IPv6) headers natively, and can connect to Local Area Network (LAN) and Wide Area Network (WAN) switches and routers with the TCA providing a seamless transition between the system area network and external networks.

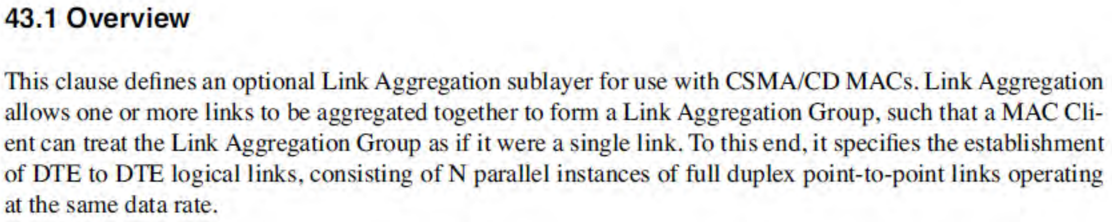
No.	’740 Patent Claim 5	Lebizay
		InfiniBand defines network layers up to the transport layer and offers all four combinations of reliable/unreliable and connection datagram service. The Infiniband transport protocol is used within the system area network, but other transport protocols can be accessed by sending raw packets via a TCA. TCAs provide connections to storage, fibre channel networks, and other 1/0 nodes, and include an 1/0 controller specific to the device's protocol, be it Small Computer Systems Interface (SCSI), Ethernet, etc. A TCAincludes an Infiniband protocol engine that dramatically accelerates the performance of critical Infiniband transport functions in the TCA's hard-ware, achieving aggregate internal transaction switching throughput speeds of 150 gigabits per second. TCAs are highly optimized for Infiniband target applications such as bridging from Infiniband devices (switches) to local busses i.e., Gigabit Ethernet, Fibre Channel, and Transport Control Protocol/Internet Protocol TCP/IP devices, as well as next-generation 1/0 protocols.”) Lebizay at [0034] (“Links are the connections between boards (or node to node, or switch to switch, in general). They are capable of transmitting no more than a certain amount of information at any one time, known as bandwidth. Bandwidth may be measured, for example, in bits/second. In general, there is no restriction on the number of boards (N) except that it be greater than 1.”) Lebizay at [0037] (“For example, an entity may set up a Service Level Agreement (SLA) with a network service provider (the entity that is providing the network), that specifies that the entity's traffic will always have available a certain band-width (e.g., 10 Megabits per second, or 10 Mbps) and latency (e.g., less than 1 millisecond, or ms). Then, when-ever a packet is detected that comes from or is going to that entity, the packet should receive special handling. If the overall flow to the entity is currently less than 10 Mbps, then the packet should get through without being dropped and with a latency of less than 1 ms. This type of flow is said to be handled by Assured Forwarding (AF). Packets arriving when the current flows are greater than 10 Mbps will be handled differently, perhaps as Best Effort (BE) traffic (see below).”) Lebizay at [0038] (“As another example, a router may be set up to recognize certain types of flows as real-time flows. Real-time flows are characterized by the idea that if the packet doesn't arrive in time it might as well not arrive at all. For example, a packet of voice data in a telephone conversation has to be available at the receiver when it is needed, or it is useless. Too late, and it

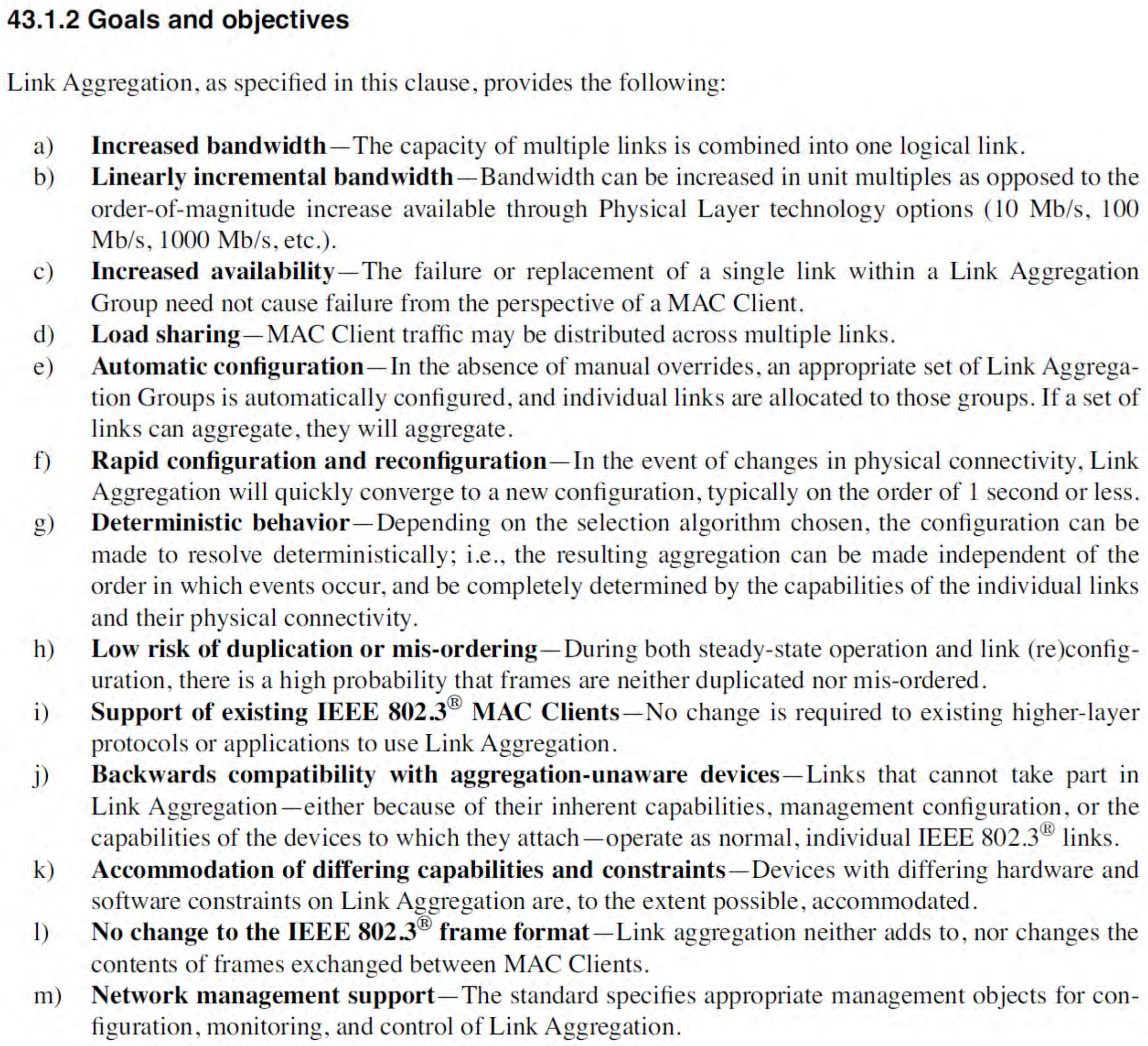
No.	’740 Patent Claim 5	Lebizay
		cannot be used and will just be discarded. So real-time traffic (a stream of packets) belong-ing to a voice conversation should be handled by a class of behavior known as Expedited Forwarding (EF). A packet handled this way will be forwarded very quickly (with low latency). Hopefully, the variation in latency (known as jitter) will also be low. As a tradeoff, packets in such a stream may be simply dropped if their aggregate bandwidth exceeds a certain threshold. Also, a SLA covering such packets may be expensive to the buyer because providing this kind of service requires that a router have features that make it expensive to build.”) Lebizay at Figure 4 (annotation added)

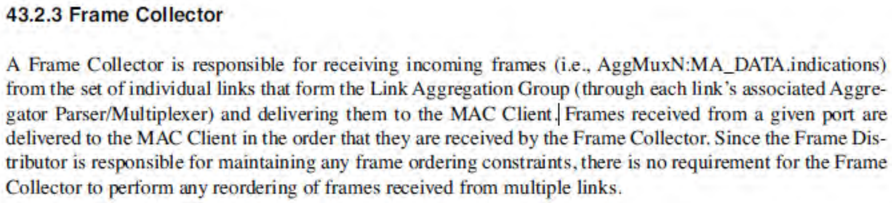
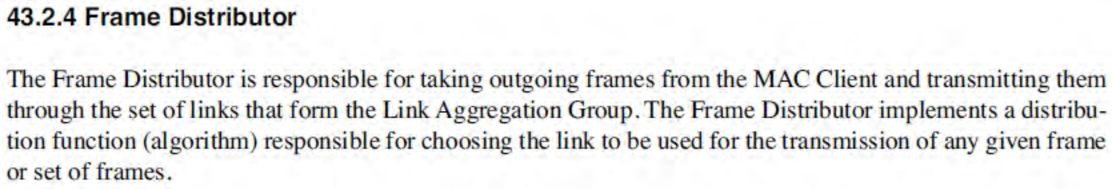
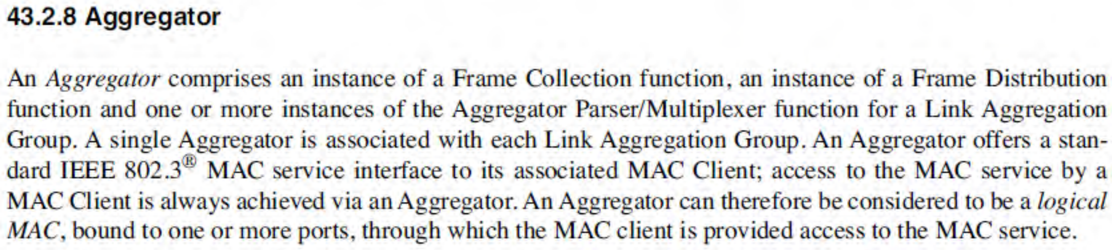
No.	’740 Patent Claim 5	Lebizay
		 400 16 Destination Boards 401-416 Network processor 450 Virtual Output Queues on Local Side 460 16 fabric ports with Integrated SERDES 421-436 N local ports 490 Simple buffers On Fabric side 470 FIG. 4 Lebizay at [0040] (“Processing of packets from ingress to fabric input is shown in FIG. 4. Depicted is the flow of packets through a single FIC 400 located on a board going to 16 destination boards 401-416. A network processor 450 on the FIC 400 classifies all outgoing traffic to a particular flow. The flow defines a set of parameters including the 16 destination boards 401-416, the path by which it will get to the desti-nation board, i.e., one of the 16 fabric ports 421-436, and classification (AF, EF or BE). The individual packets end up in virtual output (send) queues 460 according to the classi-fication. There is a separate set of virtual output (send) queues 460 (AF, EF or BE) for every

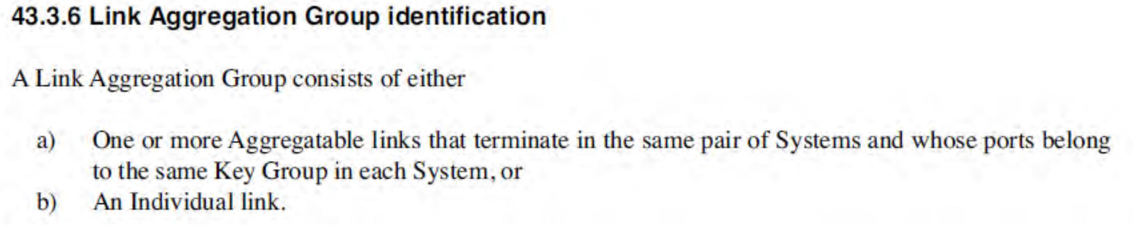
No.	’740 Patent Claim 5	Lebizay
		destination point, i.e., fabric port 421-436, per FIC. The fabric ports 421-436 and N local ports 490 have both send and receive terminals.”) Lebizay at [0041] (“Multiple virtual output (send) queues 460 are maintained per FIC for two reasons. The primary reason is that multiple virtual output queues 460 provide a steering function. After classification, the destination board 401-416 has been identified. If the packets were put back into the same (common) queue, the destination board 401-416 information would have to be carried along with the packet. This scheme is in fact done in some implementations. However, separating the packets into separate queues is another way to carry the information.”) Lebizay at [0043] (“InfiniBand offers link layer Virtual Lanes (VLs) to support multiple logical channels (i.e. multiplexing) on the same physical link. Infiniband offers up to 16 virtual lanes per link. VLs provide a mechanism to avoid head-of-line blocking and the ability to support Quality of Service (QoS). The difference between a Virtual Lane and a Service Level (SL) is that a Virtual Lane is the actual logical lane (mul-tiplexed) used on a given point-to-point link. The Service Level stays constant as a packet traverses the fabric, and specifies the desired service level within a subnet. The SL (AF, EF or BE) is included in the link header, and each switch maps the SL to a VL supported by the destination link. A switch supporting a limited number of virtual lanes will map the SL field to a VL it supports. Without preserving the SL, the desired SL (AF, EF or BE) would be lost in this mapping, and later in the path, a switch supporting more VLs would be unable to recover finer granularity of SLs between two packets mapped to the same VL.”) Lebizay at [0044] (“In the case of the FIC 400 shown in FIG. 4, the virtual lanes are established by classifying, sorting, and placing packets, according to service level and destination, into the virtual output queues 460. The packets are then multiplexed onto a link via a fabric port 421-436 to be transmitted to a destination board 401-416. The classifying, sorting, placing and multiplexing are carried out by the network processor 450.”) Lebizay at [0046] (“Today, boards in a common chassis interconnect to each other over copper traces embedded in the chassis backplane. Existing backplanes use copper traces within a PCB

No.	’740 Patent Claim 5	Lebizay
		substrate to establish the communications paths between boards. Topologies such as common bus PCI and switched star (i.e. some Ethernet approaches) are popular today. The copper traces and the electrical drivers that communicate over them are limited in terms of their capac-ity to carry data bandwidth with respect to the Local Area Network (LAN) and Wide Area Network (WAN) optical signals that these boards are intended to terminate.”) Lebizay at [0050] (“The issue with using a ring, however, is how to map the addressing of multiple boards across these fibers. One solution is to employ Wavelength Division Multiplex-ing (WDM). A WDM optical mesh defines a meshed-topology in the wavelength space as opposed to the physical fiber space. By utilizing multiple discrete lambda-waves as optical carriers such that by meshing dedicated optical wavelengths between every two boards, layer 2 protocols are eliminated, thereby creating a dramatic improvement in the efficiency of the transport. Today, every packet transport requires a protocol that allows the end point (and interme-diate points) to decipher the intended path (or consumer) of the packet. This protocol increases the amount of overhead required in the packet bus, allowing less room for actual data to be sent. By moving the protocol into the wavelength of the actual optical signal, the destination is implied by the wavelength and no additional bandwidth needs to be sur-rendered on the signal to provide this information. This makes the efficiency of the transport better and also speeds the routing of the packet through the network. In addition, the use of optical interconnects in a backplane environment greatly increases chassis bandwidth as well as reducing electrical radiation that often accompanies copper intercon-nects. The components involved include an optical back-plane in a physical ring topology, and the necessary trans-mitters and receivers for the size of the installation (i.e., number of slots in the chassis). In addition, optical add/drop multiplexer devices are required.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Lebizay in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 5[f] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references.

No.	’740 Patent Claim 5	Lebizay
		For example, IEEE 802.3 discloses connecting a MAC Client to the aggregator, i.e., coupling the network node to the one or more interface modules, by aggregating one or more links together to form a Link Aggregation Group in order to increase bandwidth. IEEE 802.3 at 1465  **43.1 Overview** This clause defines an optional Link Aggregation sublayer for use with CSMA/CD MACs. Link Aggregation allows one or more links to be aggregated together to form a Link Aggregation Group, such that a MAC Client can treat the Link Aggregation Group as if it were a single link. To this end, it specifies the establishment of DTE to DTE logical links, consisting of N parallel instances of full duplex point-to-point links operating at the same data rate. IEEE 802.3 at 1465

No.	’740 Patent Claim 5	Lebizay
		 **43.1.2 Goals and objectives** Link Aggregation, as specified in this clause, provides the following: a) **Increased bandwidth**—The capacity of multiple links is combined into one logical link. b) **Linearly incremental bandwidth**—Bandwidth can be increased in unit multiples as opposed to the order-of-magnitude increase available through Physical Layer technology options (10 Mb/s, 100 Mb/s, 1000 Mb/s, etc.). c) **Increased availability**—The failure or replacement of a single link within a Link Aggregation Group need not cause failure from the perspective of a MAC Client. d) **Load sharing**—MAC Client traffic may be distributed across multiple links. e) **Automatic configuration**—In the absence of manual overrides, an appropriate set of Link Aggregation Groups is automatically configured, and individual links are allocated to those groups. If a set of links can aggregate, they will aggregate. f) **Rapid configuration and reconfiguration**—In the event of changes in physical connectivity, Link Aggregation will quickly converge to a new configuration, typically on the order of 1 second or less. g) **Deterministic behavior**—Depending on the selection algorithm chosen, the configuration can be made to resolve deterministically; i.e., the resulting aggregation can be made independent of the order in which events occur, and be completely determined by the capabilities of the individual links and their physical connectivity. h) **Low risk of duplication or mis-ordering**—During both steady-state operation and link (re)configuration, there is a high probability that frames are neither duplicated nor mis-ordered. i) **Support of existing IEEE 802.3® MAC Clients**—No change is required to existing higher-layer protocols or applications to use Link Aggregation. j) **Backwards compatibility with aggregation-unaware devices**—Links that cannot take part in Link Aggregation—either because of their inherent capabilities, management configuration, or the capabilities of the devices to which they attach—operate as normal, individual IEEE 802.3® links. k) **Accommodation of differing capabilities and constraints**—Devices with differing hardware and software constraints on Link Aggregation are, to the extent possible, accommodated. l) **No change to the IEEE 802.3® frame format**—Link aggregation neither adds to, nor changes the contents of frames exchanged between MAC Clients. m) **Network management support**—The standard specifies appropriate management objects for configuration, monitoring, and control of Link Aggregation.

No.	’740 Patent Claim 5	Lebizay
		IEEE 802.3 at 1470  43.2.3 Frame Collector A Frame Collector is responsible for receiving incoming frames (i.e., AggMuxN:MA_DATA.indications) from the set of individual links that form the Link Aggregation Group (through each link’s associated Aggregator Parser/Multiplexer) and delivering them to the MAC Client. Frames received from a given port are delivered to the MAC Client in the order that they are received by the Frame Collector. Since the Frame Distributor is responsible for maintaining any frame ordering constraints, there is no requirement for the Frame Collector to perform any reordering of frames received from multiple links. IEEE 802.3 at 1471  43.2.4 Frame Distributor The Frame Distributor is responsible for taking outgoing frames from the MAC Client and transmitting them through the set of links that form the Link Aggregation Group. The Frame Distributor implements a distribution function (algorithm) responsible for choosing the link to be used for the transmission of any given frame or set of frames. IEEE 802.3 at 1474  43.2.8 Aggregator An *Aggregator* comprises an instance of a Frame Collection function, an instance of a Frame Distribution function and one or more instances of the Aggregator Parser/Multiplexer function for a Link Aggregation Group. A single Aggregator is associated with each Link Aggregation Group. An Aggregator offers a standard IEEE 802.3® MAC service interface to its associated MAC Client; access to the MAC service by a MAC Client is always achieved via an Aggregator. An Aggregator can therefore be considered to be a *logical MAC*, bound to one or more ports, through which the MAC client is provided access to the MAC service. IEEE 802.3 at 1481

No.	’740 Patent Claim 5	Lebizay
		 **43.3.6 Link Aggregation Group identification** A Link Aggregation Group consists of either a) One or more Aggregatable links that terminate in the same pair of Systems and whose ports belong to the same Key Group in each System, or b) An Individual link. As another example, Ghosh discloses aggregating physical links, including ports, into aggregate port channels that form a single logical link to increase bandwidth, load balancing, and link redundancy. Ghosh at Abstract (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggregation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent gen-eration of port channel map tables.”) Ghosh at [0004] (“Neighboring nodes in a fibre channel network are typically interconnected through multiple physical links. For example, a local fibre channel switch may be connected to a remote fibre channel switch through four physical links. In many instances, it may be beneficial to aggregate some of the physical links into logical links. That is, multiple physi-cal links can be combined to form a logical interface to provide higher aggregate bandwidth, load balancing, and link redundancy. When a frame is being transmitted over a logical link, it does not matter what particular physical link is being used as long as all the frames of a given flow are transmitted through the same link. If a constituent physical link goes down, the logical link can still remain operational.”) Ghosh at [0007] (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggre-gation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences

No.	’740 Patent Claim 5	Lebizay
		with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent generation of port channel map tables.”) Ghosh at [0008] (“In one embodiment, a method for aggregating ports in a fibre channel fabric is provided. It is determined that a plurality of local ports at a local fibre channel switch are compatible. Identifiers for the plurality of local ports are sent to a remote fibre channel switch. The remote fibre channel switch determines if a plurality of remote ports are compat-ible, the plurality of remote ports corresponding to the plurality of local ports. An indication that one or more of the remote physical ports are compatible is received. A port channel including one or more of the local ports correspond-ing to the compatible remote ports is created.”) Ghosh at [0010] (“In another embodiment, a fibre channel network is described. The fibre channel network includes a local fibre channel switch and a remote fibre channel switch. The local fibre channel switch aggregates a compatible subset of the plurality of local ports and sends identifiers for the compat-ible subset of the plurality of local ports to the remote fibre channel switch. The remote fibre channel switch determines if a subset of the plurality of remote ports are compatible. The subset of the plurality of remote ports corresponds to the compatible subset of the plurality of local ports.”) Ghosh at [0022] (“Switches in a fibre channel network are typically interconnected using multiple physical links. The physical links connecting a pair of switches allows transmission of data and control signals. In some instances, it is useful to aggregate multiple physical links into a logical link. Physi-cal links are also referred to herein as physical interfaces and channels while logical links are also referred to herein as logical interfaces and port channels. For example, a local switch may be connected to a remote switch through four physical links. Instead of having to transmit data through a particular physical link, the physical links can be aggregated to form one or more logical links. In one example, all four physical links are aggregated into a single logical link. Instead of having data transmitted through a particular physical link, the data can merely be transmitted over a particular logical link without regard to the particular physi-cal interface used. Aggregating physical links into a logical link allows for higher aggregated bandwidth, load balancing, and link

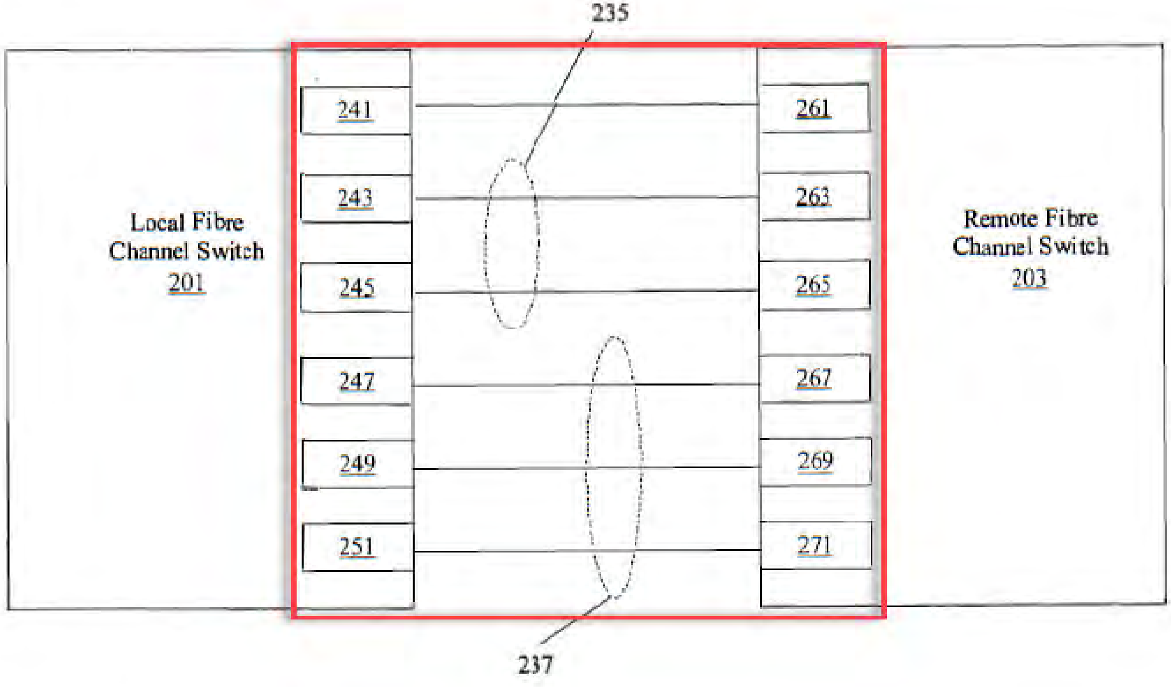
No.	’740 Patent Claim 5	Lebizay
		redundancy. For example, if a particular physical link fails or is overloaded, data can still be transmitted over the logical link.”) Ghosh at [0029] (“FIG. 2 is a diagrammatic representation showing links between two switches, such as two fibre channel switches shown in FIG. 1. A local fibre channel switch 201 includes local ports 241, 243, 245, 247, 249, and 251. A remote fibre channel switch 203 includes remote ports 261, 263, 265, 267, 269, and 271. Local port 241 is coupled to remote port 261 through an individual physical link or channel. Connected ports are also referred to herein as peer ports. Local port 243 is coupled to remote port 263 and local port 245 is coupled to remote port 265. The two resulting physical links are aggregated to form port channel 235. Local ports 247, 249, and 251 are coupled to remote ports 267, 269, and 271 respectively. The three resulting physical links are aggregated to form port channel 237.”) Ghosh at [0030] (“According to various embodiments, local fibre channel switch 201 and remote fibre channel switch both have associated identifiers. In some examples, the identifiers are globally unique identifiers such as a global switch world wide names (WWNs). Each local port 241, 243, 245, 247, 249, and 251 and each remote port 261,263,265,267, 269, and 271 can also be associated with identifiers. In some examples, the identifiers are port WWNs. The port WWNs are typically used for debugging or identifying the peer port in alert or warning messages. However, according to various embodiments, the techniques of the present invention use WWNs as globally unique identifiers to aggregate ports instead of using compatibility keys which are only locally unique. Compatibility keys are mechanisms typically used by other protocols such as Ethernet for aggregation.”) Ghosh at [0033] (“A variety of parameters can be used to aggregate physical ports. FIG. 3 is a flow process diagram showing one technique for aggregating physical ports into a logical port. And 301, it is determined if auto create functionality is enabled. According to various embodiments, auto create functionality allows automatic configuration and detection of compatible physical ports as well as aggregation into one or more logical ports. Auto creation does not require user intervention. In other examples, administrators can manu-ally arrange ports for aggregation.”)

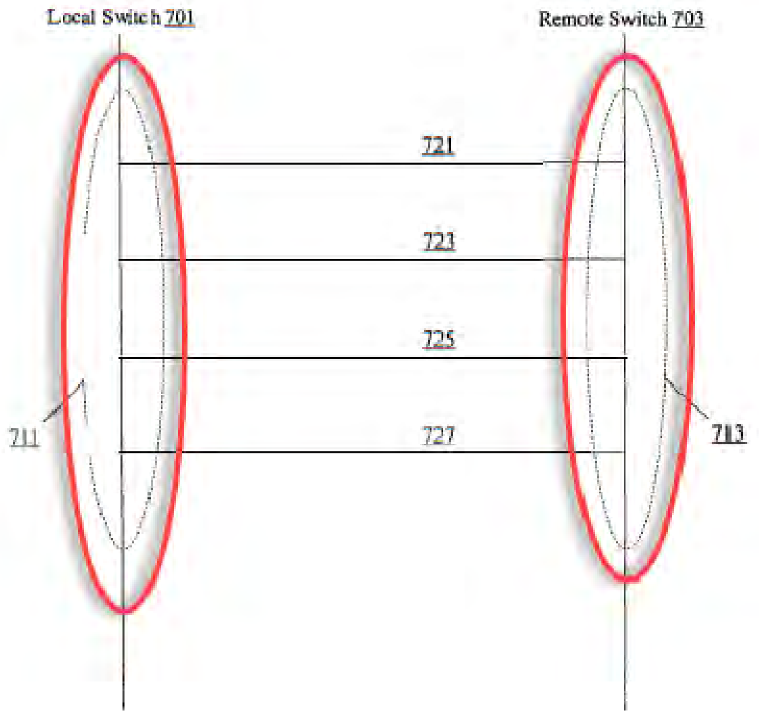
No.	’740 Patent Claim 5	Lebizay
		Ghosh at [0037] (“FIG. 4 is an exchange diagram showing one example of a bring up procedure used for a port creating a new port channel. A local switch 401 is coupled to a remote switch 403. The local switch 401 includes a physical port Al coupled to physical port Bl included in remote switch 403. When two peer ports Al and Bl are being aggregated into a port channel, the peer switches 401 and 403 typically already know the world wide names of the individual physical peer ports. However, the peer switches only know the world wide name of their own logical port or port channel. That is, both switches have the individual physical link configured, but the link is not yet part of a port channel. At 421, a local switch 401 sends a synchronize (sync) message 411 to the remote switch 403 to begin the process of creating a port channel including ports Al and Bl.”) Ghosh at [0038] (“In some examples, the sync message 411 includes a local port channel identifier and a remote port channel identifier. In one particular example, the local port channel identifier is set to the world wide name of the local port channel assigned by the local switch 401. The remote port channel identifier is left blank to indicate that the port Al is being aggregated as part of a new port channel. The sync message 411 can also include other parameters such as channel status, channel model, or channel intent.”) Ghosh at [0042] (“When two peer ports A2 and B2 are aggregated into a port channel Cl, the peer switches 501 and 503 typically already know the world wide names of the indi-vidual physical peer ports A2 and B2 as well as the world wide name information of the port channel Cl. Conse-quently, the port channel is already successfully established. According to various embodiments, local switch 501 and remote switch 503 perform parameter checking to ensure that the new physical port A2 and B2 can be safely added to the existing port channel Cl. At 521, a local switch can check configuration parameters to ensure that physical ports Al and A2 at the local switch 501 are compatible. The compatibility checking can be performed anytime. In some examples, compatibility checking is checked before a local switch 501 sends a synchronize (sync) message 511 to the remote switch 503 to begin the process of aggregating ports A2 and B2 into the port channel.) Ghosh at [0043] (“In some examples, the sync message 511 includes local port channel identifier and a remote port channel identifier. In one particular example, the local port channel identifier is

No.	’740 Patent Claim 5	Lebizay
		set to the world wide name of the local port channel assigned by the local switch 501. The remote port channel identifier is filled with the existing port channel identifier to indicate that the port A2 is being aggregated into existing port channel C2. The sync message 511 can also include other parameters such as channel status, channel model, or channel intent.”) Ghosh at [0044] (“At 531, remote switch 503 uses the information received from the local switch 501 to verify port B2 is compatible with other port in port channel C2. In one example, configuration parameters associated with B2 are checked against configuration parameters associated with Bl. The remote switch 503 can also check if the port B2 is already assigned to a different port channel. If the port B2 is compatible with port Bl, the remote switch 503 can proceed and send a sync accept message 513 in response to the sync message 511 to indicate that the port B2 can be aggregated into the port channel. The sync accept message indicates that a port channel can now be modified. At 523, local switch 501 uses the information to update its own port channel database. However, the port channel may not yet be fully operational until the hardware configuration is completed. The local switch 501 continues hardware configuration such as line card configuration to make the port A2 part of the port channel Cl. An acknowledgment 527 is sent and received by remote switch 503 at 529. In some examples, the local switch 501 sends a commit signal 515 when hardware configuration is complete.”) Ghosh at [0045] (“The remote switch 503 receives the commit signal at 533 and begins its own hardware configuration. On completion of its hardware configuration, remote switch 503 sends out a commit accept signal 517 to indicate to local switch 501 that hardware configuration is completed. According to various embodiments, local switch 501 receives the commit accept signal 517 and notifies relevant applications that the port channel is now fully operational at 525 and that port A2 has been aggregated into port channel Cl. The local switch 501 can also send out an acknowledge message 519. When the remote switch 503 receives the acknowledge, it notifies relevant applications that the port channel is operational at 535 and that port B2 has been aggregated into port channel Cl. In one embodiments, the techniques of the present invention contemplate using a two phase SYNC and COMMIT mechanism similar to the mechanism used in EPP.”)

No.	’740 Patent Claim 5	Lebizay
		Ghosh at [0046] (“FIGS. 4 and 5 show examples of ports being aggregated into a port channel. At a particular switch, ports can be selected for aggregation into a port channel in a variety of manners. FIG. 6 is an exchange diagram showing automatic selection of ports at a switch for aggregation into a port channel. A local switch 601 is coupled to a remote switch 603. In one example, the local switch 601 includes physical ports Al, A2, A3, and A4 while remote switch 603 includes physical ports Bl, B2, B3, and B4. No port chan-nels have been formed.”) Ghosh at [0049] (“At 631, remote switch 603 uses the information received from the local switch 601 to verify that the peer ports of Al, A2, and A4 are compatible. That is, ports Bl, B2, and B4 are checked for compatibility. In one example, only ports Bl and B2 may be compatible, and consequently only ports Al, A2, Bl, and B2 can be included in the port channel. In another example, ports Bl, B2, and B4 are compatible, so ports Al, A2, A4, Bl, B2, and B4 can be aggregated into port channel Cl. According to various embodiments, if the port B2 is compatible with port Bl, the remote switch 603 can proceed and send a sync accept message 613 in response to the sync message 611 to indicate that the port B2 can be aggregated into the port channel. It should be noted that remote switch 603 can send a list indicating that ports B2 and B4 are compatible with Bl. However, the remote switch 603 sends only one compatible port B2 back for several reasons, and in the process of selection compatible port channels get priority over com-patible individual ports.”) Ghosh at [0050] (“One reason is that aggregation mechanisms and techniques can be implemented more elegantly by handling ports on an individual basis. Any individual port will either start a new port channel, be added to an existing port channel, or operate stand alone. There is no need to keep track of groups of ports to be aggregated. Another reason is that fewer ports need to be locked if only a single port is being aggregated at any one time. The sync accept message indicates that a port channel can now be modified. At 623, local switch 601 receives the information and recognizes that Al and A2 can now be aggregated into port channel Cl. However, the port channel may not yet be fully operational until the hardware configuration is completed. An acknowl-edgment 627 is sent and received by remote switch 603 at 629. In some examples, the local switch 601 sends a commit signal 615 when hardware configuration is complete.”)

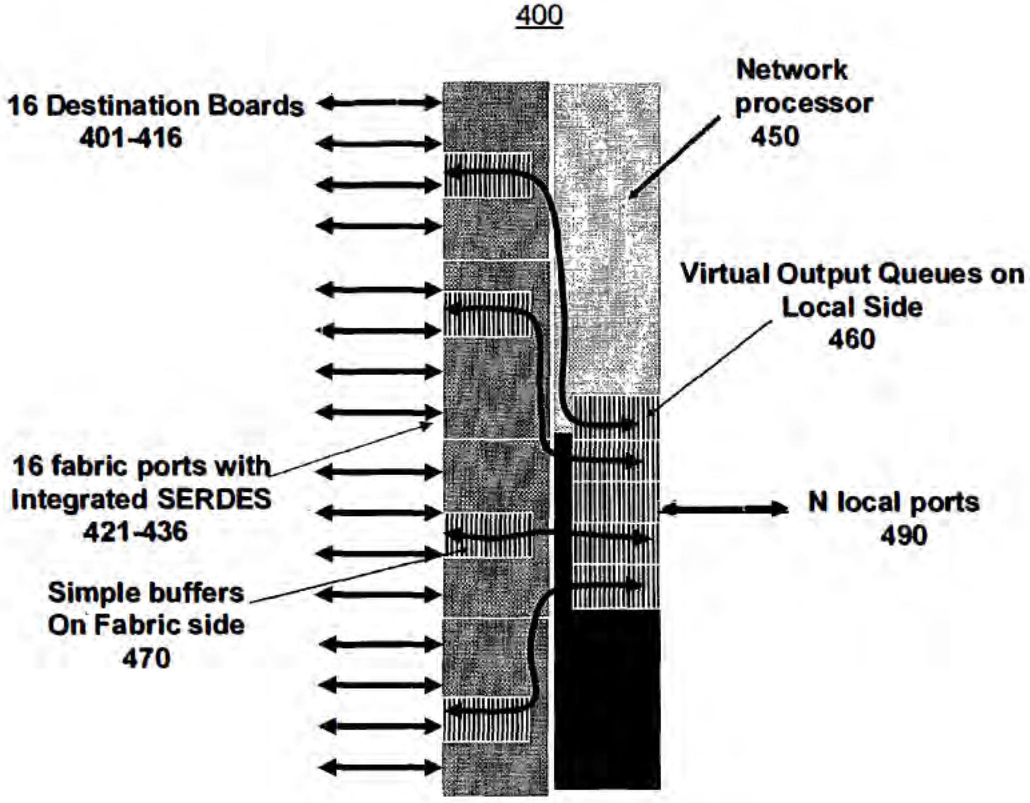
No.	’740 Patent Claim 5	Lebizay
		Ghosh at [0051] (“The remote switch 603 receives the commit signal at 633 to create port channel Cl including ports Bl and B2. Hardware configuration can now be performed. On comple-tion of its hardware configuration, remote switch 603 sends out a commit accept signal 617 to indicate to local switch 601 that hardware configuration is completed. According to various embodiments, local switch 601 receives the commit accept signal 617 and notifies relevant applications that the port channel is now fully operational at 625 and that ports Al and A2 have been aggregated into port channel Cl. The local switch 601 can also send out an acknowledge message 619. When the remote switch 603 receives the acknowledge, it notifies relevant applications that the port channel is fully operational at 635 and that ports Bl and B2 have been aggregated into port channel Cl.”) Ghosh at [0053] (“FIG. 7 is a diagrammatic representation showing synchronous aggregation of ports into a port channel. A local switch 701 is coupled to a remote switch 703 through links 721, 723, 725, and 727. According to various embodiments, the links are being aggregated into port channel 711 at the local switch 701 and port channel 713 at the remote switch 703 in a synchronous manner. That is the peer ports corre-sponding to each link are brought up in the same order at both the local switch 701 and the remote switch 703.”) Ghosh at Figure 2 (annotation added)

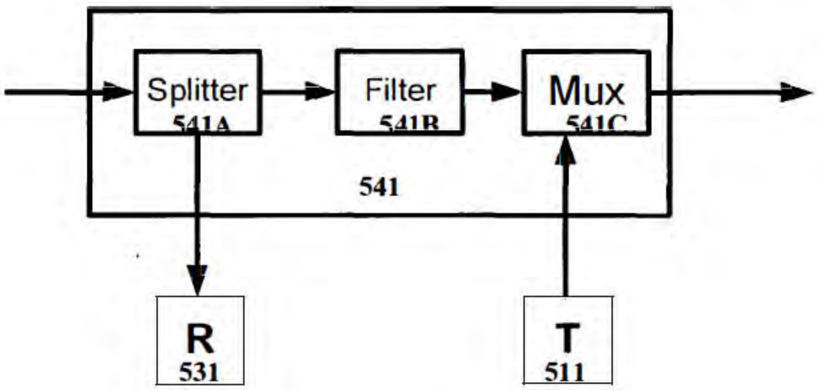
No.	’740 Patent Claim 5	Lebizay
		 235 241 243 245 247 249 251 261 263 265 267 269 271 Local Fibre Channel Switch 201 Remote Fibre Channel Switch 203 237 Figure 2 Ghosh at Figure 7 (annotation added)

No.	’740 Patent Claim 5	Lebizay
		 Local Switch 701 Remote Switch 703 721 723 725 727 711 713 Figure 7

No.	’740 Patent Claim 6	Lebizay
6	The method according to claim 1, wherein coupling each of the one or more interface modules to the	Lebizay discloses the method according to claim 1, wherein coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node.

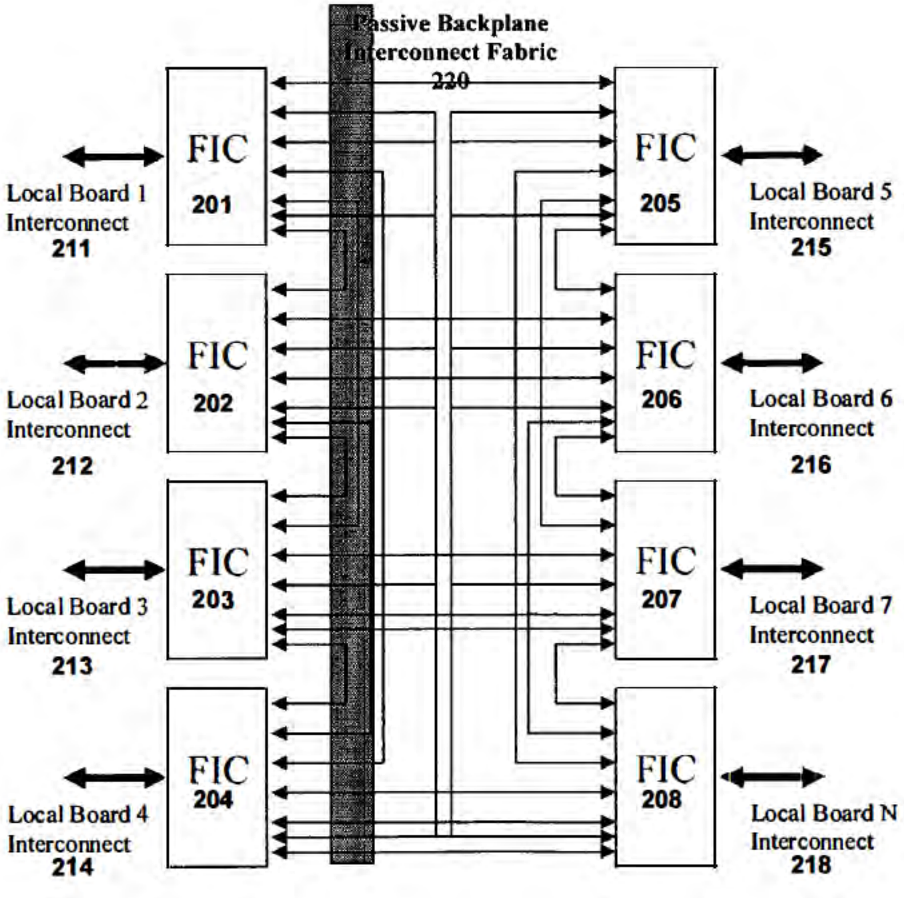
No.	’740 Patent Claim 6	Lebizay
	communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node.	For example, Lebizay discloses an optical add/drop multiplexer that multiplexes and demultiplexes data packets sent between the network boards and network. *See supra* at Claim 1. Lebizay at [0043] (“InfiniBand offers link layer Virtual Lanes (VLs) to support multiple logical channels (i.e. multiplexing) on the same physical link. Infiniband offers up to 16 virtual lanes per link. VLs provide a mechanism to avoid head-of-line blocking and the ability to support Quality of Service (QoS). The difference between a Virtual Lane and a Service Level (SL) is that a Virtual Lane is the actual logical lane (mul-tiplexed) used on a given point-to-point link. The Service Level stays constant as a packet traverses the fabric, and specifies the desired service level within a subnet. The SL (AF, EF or BE) is included in the link header, and each switch maps the SL to a VL supported by the destination link. A switch supporting a limited number of virtual lanes will map the SL field to a VL it supports. Without preserving the SL, the desired SL (AF, EF or BE) would be lost in this mapping, and later in the path, a switch supporting more VLs would be unable to recover finer granularity of SLs between two packets mapped to the same VL.”) Lebizay at [0044] (“In the case of the FIC 400 shown in FIG. 4, the virtual lanes are established by classifying, sorting, and placing packets, according to service level and destination, into the virtual output queues 460. The packets are then multiplexed onto a link via a fabric port 421-436 to be transmitted to a destination board 401-416. The classifying, sorting, placing and multiplexing are carried out by the network processor 450.”) Lebizay at Figure 4

No.	’740 Patent Claim 6	Lebizay
		 400 16 Destination Boards 401-416 Network processor 450 Virtual Output Queues on Local Side 460 16 fabric ports with Integrated SERDES 421-436 N local ports 490 Simple buffers On Fabric side 470 FIG. 4 Lebizay at [0055] (“Between multiplexing and demultiplexing points in a WDM system, there is an area in which multiple wave-lengths exist (i.e., the dual counter-rotating rings). It is necessary to remove or insert one or more wavelengths at points along the dual-counter-

No.	’740 Patent Claim 6	Lebizay
		rotating rings. Demultiplexing must be done before the optical signal is detected, because photo detectors are inherently broadband devices that cannot selectively detect a single wavelength. An optical add/drop multiplexer (OADM) performs this function. The OADM can add, pass, and remove wavelengths. There are two general types of OADMs. The first is a fixed device that is physically configured to drop specific predetermined wave-lengths, while adding others. The second type is reconfig-urable and capable of dynamically selecting which wave-lengths are added and dropped.”) Lebizay at [0057] (“FIG. 6 illustrates a ring adaptor utilizing an OADM according to an embodiment of the present inven-tion. The blocks show how the ring adapter 541 shown previously in FIG. 5 may consist of a splitter 541A to route traffic towards the receiver 531, a filter 541B to eliminate wavelengths generated by the transmitters and a multiplexer 541C to add the new wavelengths received from the trans-mitters 511 back into the optical stream.”) Lebizay at Figure 6  Splitter 541A Filter 541B Mux 541C 541 R 531 T 511 FIG. 6

No.	’740 Patent Claim 7	Lebizay
7	The method according to claim 1, wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the first and second physical links.	Lebizay discloses the method according to claim 1, wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the first and second physical links. For example, Lebizay discloses determining a path to send data across fabric ports, local ports, and interconnects by distributing the data packets using a queuing mechanism. *See supra* at Claim 1. Lebizay at [0026] (“All switch devices have a given amount of ineffi-ciency based upon their queuing logic. Most often, this leads to head-of-line blocking in the switch because multiple contenders are vying for a single bottle neck. However, by placing multiple queues both inbound and outbound, for each port on the switch, these head-of-line blocking laten-cies can be largely removed. A single device that incorpo-rates such a queuing model is both costly and large, almost too large and expensive to put into a single piece of usable silicon. However, by distributing this functionality across multiple components, using the inherent cross-connect of the meshed backplane, an efficient logical distributed switch may be built at much lower cost and higher efficiency.”) Lebizay at [0027] (“Therefore, a simpler solution is to transport data from a TCA on a board directly to the TCA on another board through a meshed backplane. Communication from TCA to TCA requires putting more ports on each individual TCA(16 for example) along with the appropriate input and output buffering on each TCA. This proposed solution essentially creates one large logical distributed switch that may be more efficient in many ways, i.e., less costly, lower power, etc.”) Lebizay at [0031] (“In addition, by incorporating the queuing logic on both the inbound and outbound sides of the FIC, coupled with the inherent cross-connect of the mesh, a logical, distributed switch is created, without really having a switch device. FIG. 3 illustrates the same 8-way mesh depicted in FIG. 2, highlighting the logic switch which is created by distributing the switch function across FICs 201-208 on all the boards 211-218. Each multi-port FIC 201-

No.	’740 Patent Claim 7	Lebizay
		208 is shown having inbound and outbound queues 221-228 for each port. In total, what is depicted is a 8-way logical distributed switch, where each local bus (board 211-218) represents an access point on the logical switch.”) Lebizay at [0032] (“Since each FIC (201-208) has queuing (221-228) on both the inbound and outbound sides, coupled with the inherent cross-connect in the backplane 220, an extremely efficient logically distributed switch can be built at much lower cost, greater reliability, higher efficiency, lower power and lower latency than a typical, centrally switched topol-ogy, or any architecture that places a switch on every board of a meshed topology.”) Lebizay at Figure 2

No.	’740 Patent Claim 7	Lebizay
		 Passive Backplane Interconnect Fabric 220 FIC 201 Local Board 1 Interconnect 211 FIC 202 Local Board 2 Interconnect 212 FIC 203 Local Board 3 Interconnect 213 FIC 204 Local Board 4 Interconnect 214 FIC 205 Local Board 5 Interconnect 215 FIC 206 Local Board 6 Interconnect 216 FIC 207 Local Board 7 Interconnect 217 FIC 208 Local Board N Interconnect 218 FIG. 2

No.	’740 Patent Claim 8	Lebizay
8	The method according to claim 1, wherein selecting the first and second physical links comprises applying a mapping function to the at least one of the frame attributes.	Lebizay discloses the method according to claim 1, wherein selecting the first and second physical links comprises applying a mapping function to the at least one of the frame attributes. For example, Lebizay discloses determining a path to send data across fabric ports, local ports, and interconnects by mapping to specific links based on data packet information. *See supra* at Claim 1. Lebizay at [0009] (“Infiniband uses Internet Protocol Version 6 (IPv6) headers natively, and can connect to Local Area Network (LAN) and Wide Area Network (WAN) switches and routers with the TCA providing a seamless transition between the system area network and external networks. InfiniBand defines network layers up to the transport layer and offers all four combinations of reliable/unreliable and connection datagram service. The Infiniband transport protocol is used within the system area network, but other transport protocols can be accessed by sending raw packets via a TCA. TCAs provide connections to storage, fibre channel networks, and other 1/0 nodes, and include an 1/0 controller specific to the device's protocol, be it Small Computer Systems Interface (SCSI), Ethernet, etc. A TCAincludes an Infiniband protocol engine that dramatically accelerates the performance of critical Infiniband transport functions in the TCA's hardware, achieving aggregate internal transaction switching throughput speeds of 150 gigabits per second. TCAs are highly optimized for Infiniband target applications such as bridging from Infiniband devices (switches) to local busses i.e., Gigabit Ethernet, Fibre Channel, and Transport Control Protocol/Internet Protocol TCP/IP devices, as well as next-generation 1/0 protocols.”) Lebizay at [0028] (“Embodiments of the present invention exists within the context of connecting multiple entities within a system (specifically multiple boards within a chassis), using mul-tiple TCAs. The components consist of multi-port TCAs and a meshed backplane that is equipped to mount boards via connectors on the backplane. A multi-port TCA capable of performing multiple bridging functions simultaneously i.e., bridging from an Infiniband meshed backplane to multiple local busses, i.e., Gigabit Ethernet, Fibre Channel, and TCP/IP devices, is referred to as a Fabric Interconnect Chip (FIC). FIG. 1 depicts the connection of a

No.	’740 Patent Claim 8	Lebizay
		FIC 100 that interconnects between local components on a board (local busses) 120 and a backplane mesh fabric 110.”) Lebizay at [0035] (“A flow is a set of packets that all share a set of characteristics. Typically, the characteristics include the source and destination address of the packet, as well as its protocol type and possibly its priority or classification. It is important that all the packets in a flow maintain a certain sequence in which they were sent, preferably arriving at their destination in that same sequence. If they do arrive out of sequence they can be re-sequenced, or put back in order. However, it is not desireable to re-sequence packets at the end. Therefore, a good design attempts to keep all the packets in a flow in sequence all through the network so that they arrive at the far end in sequence and do not require re-sequencing.”) Lebizay at [0043] (“InfiniBand offers link layer Virtual Lanes (VLs) to support multiple logical channels (i.e. multiplexing) on the same physical link. Infiniband offers up to 16 virtual lanes per link. VLs provide a mechanism to avoid head-of-line blocking and the ability to support Quality of Service (QoS). The difference between a Virtual Lane and a Service Level (SL) is that a Virtual Lane is the actual logical lane (mul-tiplexed) used on a given point-to-point link. The Service Level stays constant as a packet traverses the fabric, and specifies the desired service level within a subnet. The SL (AF, EF or BE) is included in the link header, and each switch maps the SL to a VL supported by the destination link. A switch supporting a limited number of virtual lanes will map the SL field to a VL it supports. Without preserving the SL, the desired SL (AF, EF or BE) would be lost in this mapping, and later in the path, a switch supporting more VLs would be unable to recover finer granularity of SLs between two packets mapped to the same VL.”) Lebizay at [0050] (“The issue with using a ring, however, is how to map the addressing of multiple boards across these fibers. One solution is to employ Wavelength Division Multiplex-ing (WDM). A WDM optical mesh defines a meshed-topology in the wavelength space as opposed to the physical fiber space. By utilizing multiple discrete lambda-waves as optical carriers such that by meshing dedicated optical wavelengths between every two boards, layer 2 protocols are eliminated, thereby creating a dramatic improvement in the efficiency of the transport. Today, every packet transport requires a protocol that allows the end point (and

No.	’740 Patent Claim 8	Lebizay
		interme-diate points) to decipher the intended path (or consumer) of the packet. This protocol increases the amount of overhead required in the packet bus, allowing less room for actual data to be sent. By moving the protocol into the wavelength of the actual optical signal, the destination is implied by the wavelength and no additional bandwidth needs to be sur-rendered on the signal to provide this information. This makes the efficiency of the transport better and also speeds the routing of the packet through the network. In addition, the use of optical interconnects in a backplane environment greatly increases chassis bandwidth as well as reducing electrical radiation that often accompanies copper intercon-nects. The components involved include an optical back-plane in a physical ring topology, and the necessary trans-mitters and receivers for the size of the installation (i.e., number of slots in the chassis). In addition, optical add/drop multiplexer devices are required.”)

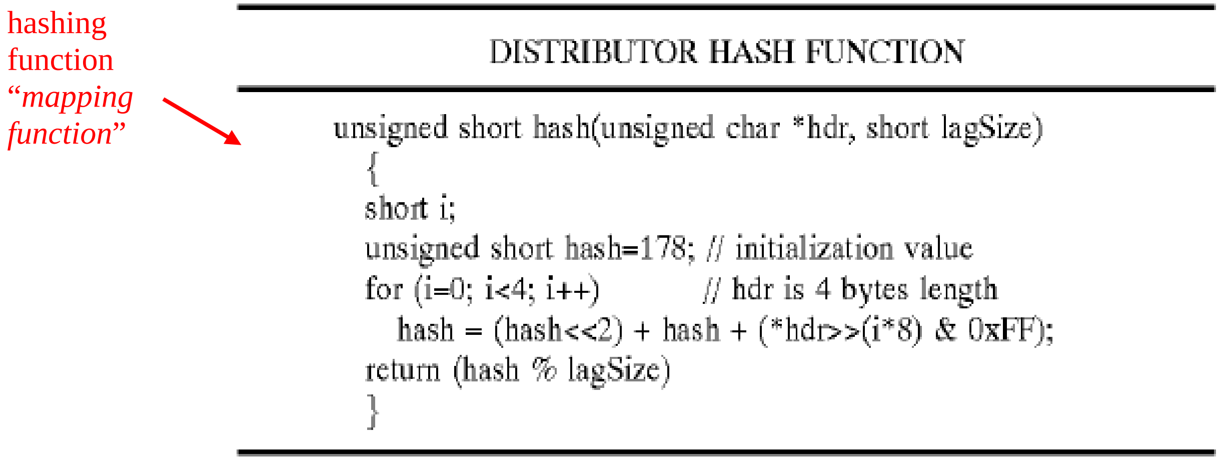
No.	’740 Patent Claim 9	Lebizay
9	The method according to claim 8, wherein applying the mapping function comprises applying a hashing function.	Lebizay discloses the method according to claim 8, wherein applying the mapping function comprises applying a hashing function. For example, Lebizay discloses a packet traversing the fabric in which the switch maps its transmission to a destination link. A person of ordinary skill in the art would understand that mapping often involves a hash function. A person of ordinary skill in the art would understand that applying a hash function includes determining parameters responsive to the system and packet features, and generating a result. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Lebizay is found to not meet this limitation, wherein applying the mapping function comprises applying a hashing function would have been obvious to a person having ordinary skill in the art, as explained below *See supra* at Claim 8.

No.	’740 Patent Claim 9	Lebizay
		Lebizay at [0043] (“InfiniBand offers link layer Virtual Lanes (VLs) to support multiple logical channels (i.e. multiplexing) on the same physical link. Infiniband offers up to 16 virtual lanes per link. VLs provide a mechanism to avoid head-of-line blocking and the ability to support Quality of Service (QoS). The difference between a Virtual Lane and a Service Level (SL) is that a Virtual Lane is the actual logical lane (mul-tiplexed) used on a given point-to-point link. The Service Level stays constant as a packet traverses the fabric, and specifies the desired service level within a subnet. The SL (AF, EF or BE) is included in the link header, and each switch maps the SL to a VL supported by the destination link. A switch supporting a limited number of virtual lanes will map the SL field to a VL it supports. Without preserving the SL, the desired SL (AF, EF or BE) would be lost in this mapping, and later in the path, a switch supporting more VLs would be unable to recover finer granularity of SLs between two packets mapped to the same VL.”) Lebizay at [0050] (“The issue with using a ring, however, is how to map the addressing of multiple boards across these fibers. One solution is to employ Wavelength Division Multiplex-ing (WDM). A WDM optical mesh defines a meshed-topology in the wavelength space as opposed to the physical fiber space. By utilizing multiple discrete lambda-waves as optical carriers such that by meshing dedicated optical wavelengths between every two boards, layer 2 protocols are eliminated, thereby creating a dramatic improvement in the efficiency of the transport. Today, every packet transport requires a protocol that allows the end point (and interme-diate points) to decipher the intended path (or consumer) of the packet. This protocol increases the amount of overhead required in the packet bus, allowing less room for actual data to be sent. By moving the protocol into the wavelength of the actual optical signal, the destination is implied by the wavelength and no additional bandwidth needs to be sur-rendered on the signal to provide this information. This makes the efficiency of the transport better and also speeds the routing of the packet through the network. In addition, the use of optical interconnects in a backplane environment greatly increases chassis bandwidth as well as reducing electrical radiation that often accompanies copper intercon-nects. The components involved include an optical back-plane in a physical ring topology, and the necessary trans-mitters and receivers for the size of the installation (i.e., number of slots in the chassis). In addition, optical add/drop multiplexer devices are required.”)

No.	’740 Patent Claim 9	Lebizay
		Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Lebizay in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 9 of Exhibit E-1 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses applying a distributor hash function. Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”)

No.	’740 Patent Claim 9	Lebizay
		Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”) Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:

No.	’740 Patent Claim 9	Lebizay

		Bruckman at Table 1 (annotated)
		
		 hashing function
		“mapping function”
		DISTRIBUTOR HASH FUNCTION
		`unsigned short hash(unsigned char *hdr, short lagSize)`
		`{`
		`short i;`
		`unsigned short hash=178; // initialization value`
		`for (i=0; i<4; i++)            // hdr is 4 bytes length`
		`   hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);`
		`return (hash % lagSize)`
		`}`
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		As another example, Solomon discloses applying a mapping function which comprises a hashing function performed by the mapper.
		Solomon at [0024] (“In another embodiment, switching the data packets includes mapping the data packets to the selected port responsively to the label. Additionally or alternatively, map-ping the data packets includes applying a hashing function to the label so as to determine a number of the selected port, and choosing the label includes applying an inverse of the hashing function to the number of the selected port.”)

No.	’740 Patent Claim 9	Lebizay
		Solomon at [0048] (“The mapping function typically uses MPLS label 52 for mapping, since the MPLS label uniquely identifies MPLS tunnel 28, and it is required that all MPLS packets belonging to the same tunnel be switched through the same physical port 24. Additionally or alternatively, the mapping function uses a "PW" label (pseudo wire label, formerly known as a virtual connection, or VC label), which is optionally added to MPLS header 50. The PW label com-prises information that the egress node requires for deliver-ing the packet to its destination, and is optionally added during the encapsulation of MPLS packets. Additional details regarding the VC label can be found in an IETF draft by Martini et al. entitled "Encapsulation Methods for Trans-port of Ethernet Frames Over IP/MPLS Networks" (IETF draft-ietf-pwe3-ethernet-encap-07.txt, May, 2004), which is incorporated herein by reference. In some embodiments, mapper 34 applies a hashing function to the MPLS and/or PW label, as will be described below.”) Solomon at [0059] (“In this method, the mapping function used by mapper 34 of switch A is a hashing function. Various hashing functions are known in the art, and any suitable hashing function may be used in mapper 34. Since the hashing operation is performed for each packet, it is desirable to have a hashing function that is computationally simple.”) Solomon at [0060] (“As mentioned above, the hashing function typically hashes the value of MPLS label 52 to determine the selected physical port, as the MPLS label uniquely identifies tunnel 28. For example, the following hashing function may be used by mapper 34: Selected port number=l+((MPLS label) mod N), wherein N denotes the number of physical Ethernet ports in LAG group 25, and "mod" denotes the modulus operator. Assuming the values of MPLS labels are distrib-uted uniformly over a certain range, this function achieves a uniform distribution of port allocations for the different MPLS labels. It can also be seen that all packets carrying the same MPLS label (in other words-belonging to the same MPLS tunnel) will be mapped to the same physical port.”) Solomon at [0065] (“Mapper 34 of switch A maps each received packet to the selected physical port of LAG group 25 using the hashing function, at a hashing step 90. Mapper 34 extracts the MPLS label from each received packet and uses the hashing function to calculate the serial number

No.	’740 Patent Claim 9	Lebizay
		of the selected physical port, which was selected by the CAC processor at step 82. Following the numerical example given above, the mapper extracts MPLS label=65647 from the packet. Substituting this value and N=3 into the hashing function gives: Selected port number=1+(65647 mod 3)=2, which is indeed the port number selected in the example above.”)

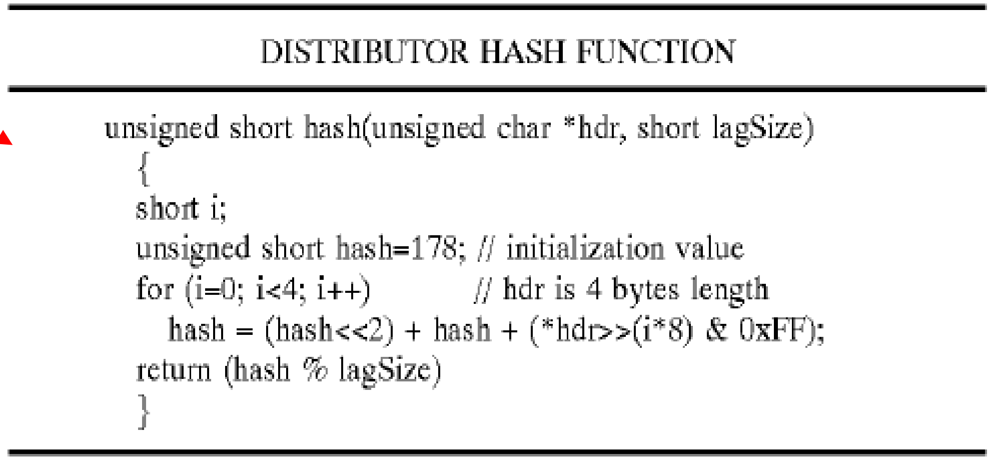
No.	’740 Patent Claim 10	Lebizay
10[a]	The method according to claim 9, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second physical links,	Lebizay discloses the method according to claim 9, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second physical links. For example, Lebizay mapping packets traversing the fabric to its destination link. A person of ordinary skill in the art would understand that mapping includes applying a hash function by determining parameters responsive to the system and packet features, and generating a result. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Lebizay is found to not meet this limitation, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second physical links would have been obvious to a person having ordinary skill in the art, as explained below *See supra at* Claim 9. Lebizay at [0043] (“InfiniBand offers link layer Virtual Lanes (VLs) to support multiple logical channels (i.e. multiplexing) on the same physical link. Infiniband offers up to 16 virtual lanes per link. VLs provide a mechanism to avoid head-of-line blocking and the ability to support Quality of Service (QoS). The difference between a Virtual Lane and a Service Level (SL) is that a Virtual Lane is the actual logical lane (mul-tiplexed) used on a given point-to-point link. The Service Level stays constant as a packet traverses the fabric, and specifies the desired service level within a subnet. The SL (AF, EF or BE) is included in the link header, and each switch maps the SL to a

No.	’740 Patent Claim 10	Lebizay
		VL supported by the destination link. A switch supporting a limited number of virtual lanes will map the SL field to a VL it supports. Without preserving the SL, the desired SL (AF, EF or BE) would be lost in this mapping, and later in the path, a switch supporting more VLs would be unable to recover finer granularity of SLs between two packets mapped to the same VL.”) Lebizay at [0050] (“The issue with using a ring, however, is how to map the addressing of multiple boards across these fibers. One solution is to employ Wavelength Division Multiplex-ing (WDM). A WDM optical mesh defines a meshed-topology in the wavelength space as opposed to the physical fiber space. By utilizing multiple discrete lambda-waves as optical carriers such that by meshing dedicated optical wavelengths between every two boards, layer 2 protocols are eliminated, thereby creating a dramatic improvement in the efficiency of the transport. Today, every packet transport requires a protocol that allows the end point (and interme-diate points) to decipher the intended path (or consumer) of the packet. This protocol increases the amount of overhead required in the packet bus, allowing less room for actual data to be sent. By moving the protocol into the wavelength of the actual optical signal, the destination is implied by the wavelength and no additional bandwidth needs to be sur-rendered on the signal to provide this information. This makes the efficiency of the transport better and also speeds the routing of the packet through the network. In addition, the use of optical interconnects in a backplane environment greatly increases chassis bandwidth as well as reducing electrical radiation that often accompanies copper intercon-nects. The components involved include an optical back-plane in a physical ring topology, and the necessary trans-mitters and receivers for the size of the installation (i.e., number of slots in the chassis). In addition, optical add/drop multiplexer devices are required.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Lebizay in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[a] of Exhibit E-1 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses applying a distributor hash function to the frame information which includes determining a number of the plurality of physical links.

No.	’740 Patent Claim 10	Lebizay
		Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In

No.	’740 Patent Claim 10	Lebizay
		the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”) Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I: Bruckman at Table 1 (annotated)

No.	’740 Patent Claim 10	Lebizay

		hashing function “*mapping function*”
		 
		 DISTRIBUTOR HASH FUNCTION
		 unsigned short hash(unsigned char *hdr, short lagSize)
		 {
		 short i;
		 unsigned short hash=178; // initialization value
		 for (i=0; i<4; i++) // hdr is 4 bytes length
		 hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		 return (hash % lagSize)
		 }
		 Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		 For example, Solomon discloses applying a distributor hash function to the frame information which includes determining a number of the plurality of physical links.
		 Solomon at [0024] (“In another embodiment, switching the data packets includes mapping the data packets to the selected port responsively to the label. Additionally or alternatively, map-ping the data packets includes applying a hashing function to the label so as to determine a number of the selected port, and choosing the label includes applying an inverse of the hashing function to the number of the selected port.”)

No.	’740 Patent Claim 10	Lebizay
		Solomon at [0048] (“The mapping function typically uses MPLS label 52 for mapping, since the MPLS label uniquely identifies MPLS tunnel 28, and it is required that all MPLS packets belonging to the same tunnel be switched through the same physical port 24. Additionally or alternatively, the mapping function uses a "PW" label (pseudo wire label, formerly known as a virtual connection, or VC label), which is optionally added to MPLS header 50. The PW label com-prises information that the egress node requires for deliver-ing the packet to its destination, and is optionally added during the encapsulation of MPLS packets. Additional details regarding the VC label can be found in an IETF draft by Martini et al. entitled "Encapsulation Methods for Trans-port of Ethernet Frames Over IP/MPLS Networks" (IETF draft-ietf-pwe3-ethernet-encap-07.txt, May, 2004), which is incorporated herein by reference. In some embodiments, mapper 34 applies a hashing function to the MPLS and/or PW label, as will be described below.”) Solomon at [0059] (“In this method, the mapping function used by mapper 34 of switch A is a hashing function. Various hashing functions are known in the art, and any suitable hashing function may be used in mapper 34. Since the hashing operation is performed for each packet, it is desirable to have a hashing function that is computationally simple.”) Solomon at [0060] (“As mentioned above, the hashing function typically hashes the value of MPLS label 52 to determine the selected physical port, as the MPLS label uniquely identifies tunnel 28. For example, the following hashing function may be used by mapper 34: Selected port number=l+((MPLS label) mod N), wherein N denotes the number of physical Ethernet ports in LAG group 25, and "mod" denotes the modulus operator. Assuming the values of MPLS labels are distrib-uted uniformly over a certain range, this function achieves a uniform distribution of port allocations for the different MPLS labels. It can also be seen that all packets carrying the same MPLS label (in other words-belonging to the same MPLS tunnel) will be mapped to the same physical port.”) Solomon at [0065] (“Mapper 34 of switch A maps each received packet to the selected physical port of LAG group 25 using the hashing function, at a hashing step 90. Mapper 34 extracts the MPLS label from each received packet and uses the hashing function to calculate the serial number of the selected physical port, which was selected by the CAC processor at step 82. Following the

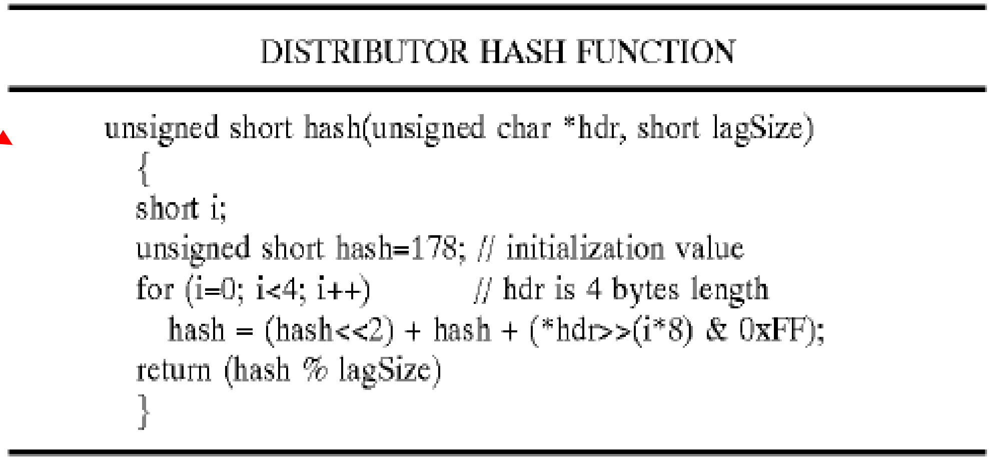
No.	’740 Patent Claim 10	Lebizay
		numerical example given above, the mapper extracts MPLS label=65647 from the packet. Substituting this value and N=3 into the hashing function gives: Selected port number=1+(65647 mod 3)=2, which is indeed the port number selected in the example above.”)
10[b]	applying the hashing function to the at least one of the frame attributes to produce a hashing key,	Lebizay discloses applying the hashing function to the at least one of the frame attributes to produce a hashing key. For example, Lebizay mapping packets traversing the fabric to its destination link. A person of ordinary skill in the art would understand that mapping includes applying a hash function by determining parameters responsive to the system and packet features, and generating a result. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Lebizay is found to not meet this limitation, applying the hashing function to the at least one of the frame attributes to produce a hashing key would have been obvious to a person having ordinary skill in the art, as explained below. Lebizay at [0043] (“InfiniBand offers link layer Virtual Lanes (VLs) to support multiple logical channels (i.e. multiplexing) on the same physical link. Infiniband offers up to 16 virtual lanes per link. VLs provide a mechanism to avoid head-of-line blocking and the ability to support Quality of Service (QoS). The difference between a Virtual Lane and a Service Level (SL) is that a Virtual Lane is the actual logical lane (mul-tiplexed) used on a given point-to-point link. The Service Level stays constant as a packet traverses the fabric, and specifies the desired service level within a subnet. The SL (AF, EF or BE) is included in the link header, and each switch maps the SL to a VL supported by the destination link. A switch supporting a limited number of virtual lanes will map the SL field to a VL it supports. Without preserving the SL, the desired SL (AF, EF or BE) would be lost in this mapping, and later in the path, a switch supporting more VLs would be unable to recover finer granularity of SLs between two packets mapped to the same VL.”) Lebizay at [0050] (“The issue with using a ring, however, is how to map the addressing of multiple boards across these fibers. One solution is to employ Wavelength Division Multiplex-ing (WDM). A WDM optical mesh defines a meshed-topology in the wavelength space as opposed to the

No.	’740 Patent Claim 10	Lebizay
		physical fiber space. By utilizing multiple discrete lambda-waves as optical carriers such that by meshing dedicated optical wavelengths between every two boards, layer 2 protocols are eliminated, thereby creating a dramatic improvement in the efficiency of the transport. Today, every packet transport requires a protocol that allows the end point (and interme-diate points) to decipher the intended path (or consumer) of the packet. This protocol increases the amount of overhead required in the packet bus, allowing less room for actual data to be sent. By moving the protocol into the wavelength of the actual optical signal, the destination is implied by the wavelength and no additional bandwidth needs to be sur-rendered on the signal to provide this information. This makes the efficiency of the transport better and also speeds the routing of the packet through the network. In addition, the use of optical interconnects in a backplane environment greatly increases chassis bandwidth as well as reducing electrical radiation that often accompanies copper intercon-nects. The components involved include an optical back-plane in a physical ring topology, and the necessary trans-mitters and receivers for the size of the installation (i.e., number of slots in the chassis). In addition, optical add/drop multiplexer devices are required.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Lebizay in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[b] of Exhibit E-1 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses applying a distributor hash function to the frame information which includes determining a number of the plurality of physical links. Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its

No.	’740 Patent Claim 10	Lebizay
		reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card

No.	’740 Patent Claim 10	Lebizay
		32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”) Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I: Bruckman at Table 1 (annotated)

No.	’740 Patent Claim 10	Lebizay

		hashing function *“mapping function”*
		
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		   hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		For example, Alexander discloses applying a distributor hash function to the frame information which includes determining a hash key based on packet information.
		Alexander at 3:1-40 (“The hash function is preferably selected such that suc-cessive application of the hash function to all source and destination addresses expected to be seen by the Ethernet switch will produce a lowest value hash key, a highest value hash key, and a group of hash keys having intermediate values distributed evenly between the lowest and highest values.

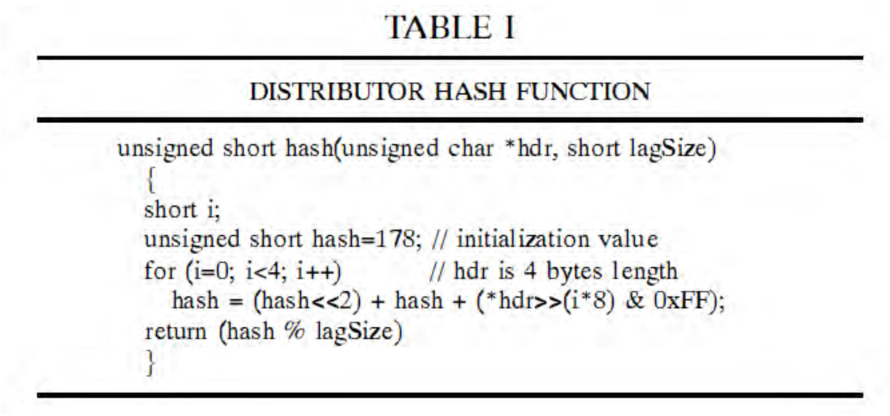
No.	’740 Patent Claim 10	Lebizay
		The distribution table contains a separate port identifier look-up table for each aggregated grouping of outgoing ports. Advantageously, the hash key is an N bit hash key; and, each port identifier look-up table contains 2^N entries occupying 2^N consecutive locations, with each entry being an identifier of a particular one of the physical outgoing ports. Identifiers for particular outgoing ports are retrieved from the distribution table by extracting first and second N bit hash keys which form part of the retrieved destination and source address contexts respectively. The hash keys are combined to form an N bit connection identifier. The port identifier look-up table corresponding to the aggregated grouping represented by the retrieved destination address is selected, and the entry at the table location corresponding to the value of the N bit connection identifier is retrieved. If the address look-up table does not contain a destination address corresponding to the extracted destination address then first and second hash keys are produced by applying a hash function to the extracted source and destination addresses respectively. The hash keys are combined to form an N bit connection identifier. The incoming port on which the packet containing the extracted source address was received is identified. All of the aggregated groupings are scanned to identify all outgoing ports to which packets may be directed from the incoming port on which the packet was received. For each one of those outgoing ports, the port identifier look-up table corresponding to the aggregated grouping containing that outgoing port is selected, the entry at the table location corresponding to the value of the N bit connection identifier is retrieved, and the received packet is queued for outgoing transmission on the outgoing port corresponding to the retrieved entry.”) Alexander at 5:10-35 (“If a packet arrives bearing a source Ethernet MAC address that was not found in look-up table 12 by address resolution unit 10, learning function 16 is invoked to update look-up table 12 with the new address (i.e. processing branches along the "No" exit from FIG. 2, block 36). Learning function 16 first computes a hash function on the source Ethernet MAC address, generating an N-bit hash key ("partial connection identifier") from the 48-bit MAC address, where N is some small integer in the range of 3 to 8 (FIG. 2, block 38). The physical port on which the packet arrived is then determined. If the physical port is found to be associated with an aggregate group (i.e., it is one of a set of ports that have been bound into a single logical port), then the logical identifier assigned to the aggregate group is also determined. The hash key is then

No.	’740 Patent Claim 10	Lebizay
		stored into address look-up table 12 in conjunction with the actual Ethernet MAC address and the port identifier (FIG. 2, block 40). The physical port identifier is used if the port is not part of an aggregate group (i.e. if processing branched along the "No" exit from block 30 and through block 32), while the logical identifier is used for ports that have been aggregated (i.e. if processing branched along the "Yes" exit from block 30 and through block 34). The hash key and port identifier are considered to form the "context" for the given MAC address.”) Alexander at 5:36-46 (“The hash function should be selected to ensure an even distribution of hash key values over the range of MAC addresses that are expected to be seen by the Ethernet switch. As a specific example, the EXACT™ Ethernet switch system employs an exclusive-OR based hash function, wherein the 48-bit MAC address is divided into 16-bit blocks, which are then exclusive-ORed together to form a single 16-bit number; the 3 least significant bits (LSBs) of this number are taken to produce a 3-bit hash key. Other schemes such as CRC-based or checksum-based hashes may also be used.”) Alexander at 6:49-65 (“If the context information for the destination address indicates, however, that the target is an aggregate group (i.e. if processing branches along the "Yes" exit from FIG. 2, block 42) then the logical identifier assigned to the aggregate group is retrieved and is used to select the proper look-up table contained within the distribution table data structure. The hash keys (partial connection identifiers) stored into the contexts for the source and destination MAC addresses are obtained from address resolution unit 10 and combined to generate a "connection identifier" with the same number of bits (FIG. 2, block 44). (In the EXACT™ Ethernet switch, a Boolean exclusive-OR operation is used to combine the hash keys without increasing the number of bits.) This connection identifier is then used to index into the selected look-up table, and finally retrieve an actual physical port index on which the packet must be transmitted (FIG. 2, block 46).”)
10[c]	calculating a modulo of a division operation of the hashing key	Lebizay discloses calculating a modulo of a division operation of the hashing key by the hashing size.

No.	’740 Patent Claim 10	Lebizay
	by the hashing size, and	For example, Lebizay mapping packets traversing the fabric to its destination link. A person of ordinary skill in the art would understand that mapping includes applying a hash function by determining parameters responsive to the system and packet features, and generating a result. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Lebizay is found to not meet this limitation, calculating a modulo of a division operation of the hashing key by the hashing size would have been obvious to a person having ordinary skill in the art, as explained below. Lebizay at [0043] (“InfiniBand offers link layer Virtual Lanes (VLs) to support multiple logical channels (i.e. multiplexing) on the same physical link. Infiniband offers up to 16 virtual lanes per link. VLs provide a mechanism to avoid head-of-line blocking and the ability to support Quality of Service (QoS). The difference between a Virtual Lane and a Service Level (SL) is that a Virtual Lane is the actual logical lane (mul-tiplexed) used on a given point-to-point link. The Service Level stays constant as a packet traverses the fabric, and specifies the desired service level within a subnet. The SL (AF, EF or BE) is included in the link header, and each switch maps the SL to a VL supported by the destination link. A switch supporting a limited number of virtual lanes will map the SL field to a VL it supports. Without preserving the SL, the desired SL (AF, EF or BE) would be lost in this mapping, and later in the path, a switch supporting more VLs would be unable to recover finer granularity of SLs between two packets mapped to the same VL.”) Lebizay at [0050] (“The issue with using a ring, however, is how to map the addressing of multiple boards across these fibers. One solution is to employ Wavelength Division Multiplex-ing (WDM). A WDM optical mesh defines a meshed-topology in the wavelength space as opposed to the physical fiber space. By utilizing multiple discrete lambda-waves as optical carriers such that by meshing dedicated optical wavelengths between every two boards, layer 2 protocols are eliminated, thereby creating a dramatic improvement in the efficiency of the transport. Today, every packet transport requires a protocol that allows the end point (and interme-diate points) to decipher the intended path (or consumer) of the packet. This protocol increases the amount of overhead required in the packet bus, allowing less room for actual data to be sent. By moving the protocol into the wavelength of the actual optical signal, the destination is implied by the wavelength and no additional bandwidth needs to be sur-rendered on the signal to provide this

No.	’740 Patent Claim 10	Lebizay
		information. This makes the efficiency of the transport better and also speeds the routing of the packet through the network. In addition, the use of optical interconnects in a backplane environment greatly increases chassis bandwidth as well as reducing electrical radiation that often accompanies copper intercon-nects. The components involved include an optical back-plane in a physical ring topology, and the necessary trans-mitters and receivers for the size of the installation (i.e., number of slots in the chassis). In addition, optical add/drop multiplexer devices are required.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Lebizay in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[c] of Exhibit E-1 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses distributing data frames over physical links and traces based on a hash function involving a division operation (%). Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0025] (“Typically, setting the protection policy includes determining a maximum number of the physical links that may fail while the logical link continues to provide at least the guaranteed bandwidth for the connection. In one embodiment, the guaranteed bandwidth is a bandwidth B, and the plurality of physical links consists of N links, and the maximum number is an integer P, and the link bandwidth allocated to each of the links is no less than B/(N-P). Conveying the data may further include managing the transmission of the data responsively to an actual number X of the physical links that have failed so that the guaranteed bandwidth on each of the links is limited to B/(N-X), X P, and an excess bandwidth on the physical links over the guaranteed bandwidth is available for other connections.”)

No.	’740 Patent Claim 10	Lebizay

		Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”)
		Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:
		
		TABLE I
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		”)
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)

No.	’740 Patent Claim 10	Lebizay
		Bruckman at [0067] (“A similar problem may arise if there is a failure in a link in an aggregation group or in one of a number of line cards serving the aggregation group. In this case, to maintain the bandwidth allocation B made by CAC 44, each of the remaining links in the group must now carry, on average, B/(N-M) traffic, wherein M is the number of links in the group that are out of service. If only BIN has been allocated to each link, the remaining active links may not have sufficient bandwidth to continue to provide the bandwidth that has been guaranteed to the connections that they are required to carry. A similar problem arises with respect to loading of traces 52. For example, if there is a failure in LC2 or in one of links 30 in group 36 that connect to LC2, the trace connecting the multiplexer 50 in LCl will have to carry a substantially larger share of the bandwidth, or even all of the bandwidth, that is allocated to the connection in ques-tion.”) Bruckman at [0068] (“FIG. 3 is a flow chart that schematically illustrates a method for dealing with these problems of fluctuating bandwidth requirements, in accordance with an embodiment of the present invention. In order to provide sufficient bandwidth for failure protection, CAC 44 uses a safety margin based on a protection parameter P, which is assigned at a protection setting step 60. P represents the maximum number of links in the group that can be out of service while still permitting the aggregation group to provide a given connection with the bandwidth that has been guaranteed to the connection. CAC 44 will then allocate at least B/(N-P) bandwidth to each link in the group, so that if P links fail, the group still provides total bandwidth of (N-P)*B/(N-P)= B. Setting P=l is equivalent to 1:N protection, so that the group will be unaffected by failure of a single link. In the example of group 36, shown in FIG. 2, setting P=2 will give both facility and equipment protection, i.e., the group will be unaffected not only by failure of a link, but also by failure of one of line cards 34. In the extreme case, in which P=N-1, CAC 44 will allocate the full bandwidth B on each link in the group.”) As another example, Singh discloses determining a ratio between the number of ingress and egress links and the number of links carrying data to the backplane and using a modulo to correspond to the channel’s link number.

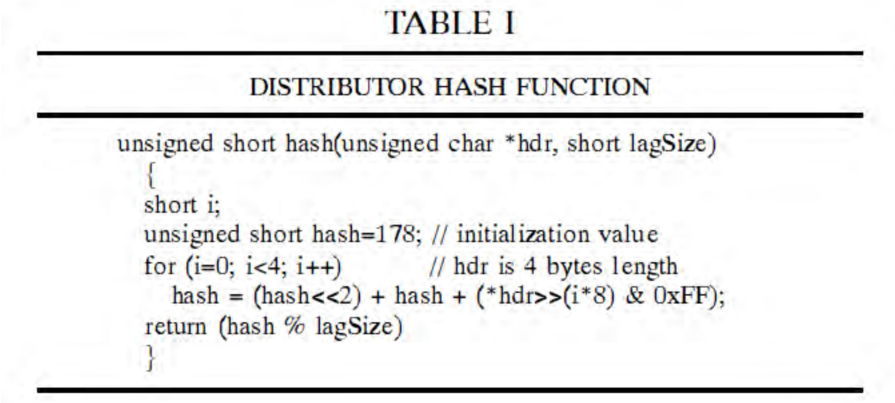
No.	’740 Patent Claim 10	Lebizay
		Singh at 9:30-43 (“The ratio between the number of line ingress links and the number of links carrying data to the backplane gives the backplane speedup for the system. In this example, there are 10 ingress links into the MS and 20 links (2 backplane channels) carrying that data to the backplane. This gives a backplane speedup of 2x. As another example, with 8 ingress links and 12 backplane links, there is a speedup of 1.5x. It should be noted that in addition to the backplane speedup, there is also an ingress/egress speedup. With 10 ingress links capable of carrying 2 Gbps each of raw data, this presents a 20 Gbps interface to the MS. An OC-192 only has approximately 10 Gbps worth of data. Taking into account cell overhead and cell quantization inefficiencies, there still remains excess capacity in the links.”) Singh at 11:29-38 (“FIG. 9 is a diagram illustrating link to channel assignments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a channel, whether it is a backplane channel or an ingress/egress channel. There is no logical relationship between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.”) Singh at 15:15-39 (“The number of crossbars that are required in a system is dependent on how many links are being used to create the backplane channels. There should be an even number of crossbars and they would be divided evenly across the switch cards. The following equation, for most cases, pro-vides the correct number of crossbars: # of Crossbars=(# links per ingress channelx# of ingress channels per portx# of port cardsx speedup)/32. For the 8x8 configuration, the # of crossbars should be multiplied by (4x# of iMS)/(# backplane channels per port card). The number of port cards should be rounded up to the nearest supported configuration, i.e. 8, 16, or 32. The speedup in the case of crossbars should be the fractional speedup that is desired.

No.	’740 Patent Claim 10	Lebizay
		Example to determine the number of arbiters and cross-bars for the following system: 4 channel port cards (40 Gbps) 8 links per channel 16 port cards Speedup=l.5 # of arbiters=(4x2x2)/2=8 # of crossbars=(8x4x16xl.5)/32=24. This would give 3crossbars per arbiter.”) Singh at 16:28-44 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 16x16 and 32x32 is the organization of the switchplane. The port card remains the same. Backplane channels 1 and 2 are used for the backplane connectivity. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 16, correspond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”) Singh at 17:31-49 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 8x8 and 16x16 is the organi-zation of the switchplane. The port card remains the same. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Links 0-7 and 24-31 on the arbiters would not be used and would be powered off. Links 0-7 and 24-31 on the crossbars would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Backplane channels 1 and 2 are used for the backplane connectivity. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 8, corre-spond to the port card numbers. Link numbers on the

No.	’740 Patent Claim 10	Lebizay
		MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”)
10[d]	selecting the first and second physical links responsively to the modulo.	Lebizay discloses selecting the first and second physical links responsively to the modulo. For example, Lebizay mapping packets traversing the fabric to its destination link. A person of ordinary skill in the art would understand that mapping includes applying a hash function by determining parameters responsive to the system and packet features, and generating a result. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Lebizay is found to not meet this limitation, selecting the first and second physical links responsively to the modulo would have been obvious to a person having ordinary skill in the art, as explained below. Lebizay at [0043] (“InfiniBand offers link layer Virtual Lanes (VLs) to support multiple logical channels (i.e. multiplexing) on the same physical link. Infiniband offers up to 16 virtual lanes per link. VLs provide a mechanism to avoid head-of-line blocking and the ability to support Quality of Service (QoS). The difference between a Virtual Lane and a Service Level (SL) is that a Virtual Lane is the actual logical lane (mul-tiplexed) used on a given point-to-point link. The Service Level stays constant as a packet traverses the fabric, and specifies the desired service level within a subnet. The SL (AF, EF or BE) is included in the link header, and each switch maps the SL to a VL supported by the destination link. A switch supporting a limited number of virtual lanes will map the SL field to a VL it supports. Without preserving the SL, the desired SL (AF, EF or BE) would be lost in this mapping, and later in the path, a switch supporting more VLs would be unable to recover finer granularity of SLs between two packets mapped to the same VL.”) Lebizay at [0050] (“The issue with using a ring, however, is how to map the addressing of multiple boards across these fibers. One solution is to employ Wavelength Division Multiplex-ing (WDM). A WDM optical mesh defines a meshed-topology in the wavelength space as opposed to the physical fiber space. By utilizing multiple discrete lambda-waves as optical carriers such that by meshing dedicated optical wavelengths between every two boards, layer 2 protocols are eliminated, thereby creating a dramatic improvement in the efficiency of the transport. Today,

No.	’740 Patent Claim 10	Lebizay
		every packet transport requires a protocol that allows the end point (and interme-diate points) to decipher the intended path (or consumer) of the packet. This protocol increases the amount of overhead required in the packet bus, allowing less room for actual data to be sent. By moving the protocol into the wavelength of the actual optical signal, the destination is implied by the wavelength and no additional bandwidth needs to be sur-rendered on the signal to provide this information. This makes the efficiency of the transport better and also speeds the routing of the packet through the network. In addition, the use of optical interconnects in a backplane environment greatly increases chassis bandwidth as well as reducing electrical radiation that often accompanies copper intercon-nects. The components involved include an optical back-plane in a physical ring topology, and the necessary trans-mitters and receivers for the size of the installation (i.e., number of slots in the chassis). In addition, optical add/drop multiplexer devices are required.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Lebizay in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[d] of Exhibit E-1 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses distributing data frames over physical links and traces based on a hash function involving a division operation (%). Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0025] (“Typically, setting the protection policy includes determining a maximum number of the physical links that may fail while the logical link continues to provide at least the guaranteed bandwidth for the connection. In one embodiment, the guaranteed bandwidth is a bandwidth B, and the plurality of physical links consists of N links, and the maximum number is

No.	’740 Patent Claim 10	Lebizay

		an integer P, and the link bandwidth allocated to each of the links is no less than B/(N-P). Conveying the data may further include managing the transmission of the data responsively to an actual number X of the physical links that have failed so that the guaranteed bandwidth on each of the links is limited to B/(N-X), X P, and an excess bandwidth on the physical links over the guaranteed bandwidth is available for other connections.”)
		Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”)
		Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:
		
		TABLE I
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		”)

No.	’740 Patent Claim 10	Lebizay
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”) Bruckman at [0067] (“A similar problem may arise if there is a failure in a link in an aggregation group or in one of a number of line cards serving the aggregation group. In this case, to maintain the bandwidth allocation B made by CAC 44, each of the remaining links in the group must now carry, on average, B/(N-M) traffic, wherein M is the number of links in the group that are out of service. If only BIN has been allocated to each link, the remaining active links may not have sufficient bandwidth to continue to provide the bandwidth that has been guaranteed to the connections that they are required to carry. A similar problem arises with respect to loading of traces 52. For example, if there is a failure in LC2 or in one of links 30 in group 36 that connect to LC2, the trace connecting the multiplexer 50 in LCl will have to carry a substantially larger share of the bandwidth, or even all of the bandwidth, that is allocated to the connection in ques-tion.”) Bruckman at [0068] (“FIG. 3 is a flow chart that schematically illustrates a method for dealing with these problems of fluctuating bandwidth requirements, in accordance with an embodiment of the present invention. In order to provide sufficient bandwidth for failure protection, CAC 44 uses a safety margin based on a protection parameter P, which is assigned at a protection setting step 60. P represents the maximum number of links in the group that can be out of service while still permitting the aggregation group to provide a given connection with the bandwidth that has been guaranteed to the connection. CAC 44 will then allocate at least B/(N-P) bandwidth to each link in the group, so that if P links fail, the group still provides total bandwidth of (N-P)*B/(N-P)= B. Setting P=l is equivalent to 1:N protection, so that the group will be unaffected by failure of a single link. In the example of group 36, shown in FIG. 2, setting P=2 will give both facility and equipment protection, i.e., the group will be unaffected not only by failure of a link, but also

No.	’740 Patent Claim 10	Lebizay
		by failure of one of line cards 34. In the extreme case, in which P=N-1, CAC 44 will allocate the full bandwidth B on each link in the group.”) As another example, Singh discloses determining a ratio between the number of ingress and egress links and the number of links carrying data to the backplane and using a modulo to correspond to the channel’s link number. Singh at 9:30-43 (“The ratio between the number of line ingress links and the number of links carrying data to the backplane gives the backplane speedup for the system. In this example, there are 10 ingress links into the MS and 20 links (2 backplane channels) carrying that data to the backplane. This gives a backplane speedup of 2x. As another example, with 8 ingress links and 12 backplane links, there is a speedup of 1.5x. It should be noted that in addition to the backplane speedup, there is also an ingress/egress speedup. With 10 ingress links capable of carrying 2 Gbps each of raw data, this presents a 20 Gbps interface to the MS. An OC-192 only has approximately 10 Gbps worth of data. Taking into account cell overhead and cell quantization inefficiencies, there still remains excess capacity in the links.”) Singh at 11:29-38 (“FIG. 9 is a diagram illustrating link to channel assignments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a channel, whether it is a backplane channel or an ingress/egress channel. There is no logical relationship between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.”) Singh at 15:15-39 (“The number of crossbars that are required in a system is dependent on how many links are being used to create the backplane channels. There should be an even number of crossbars and they would be divided evenly across the switch cards. The following equation, for most cases, pro-vides the correct number of crossbars:

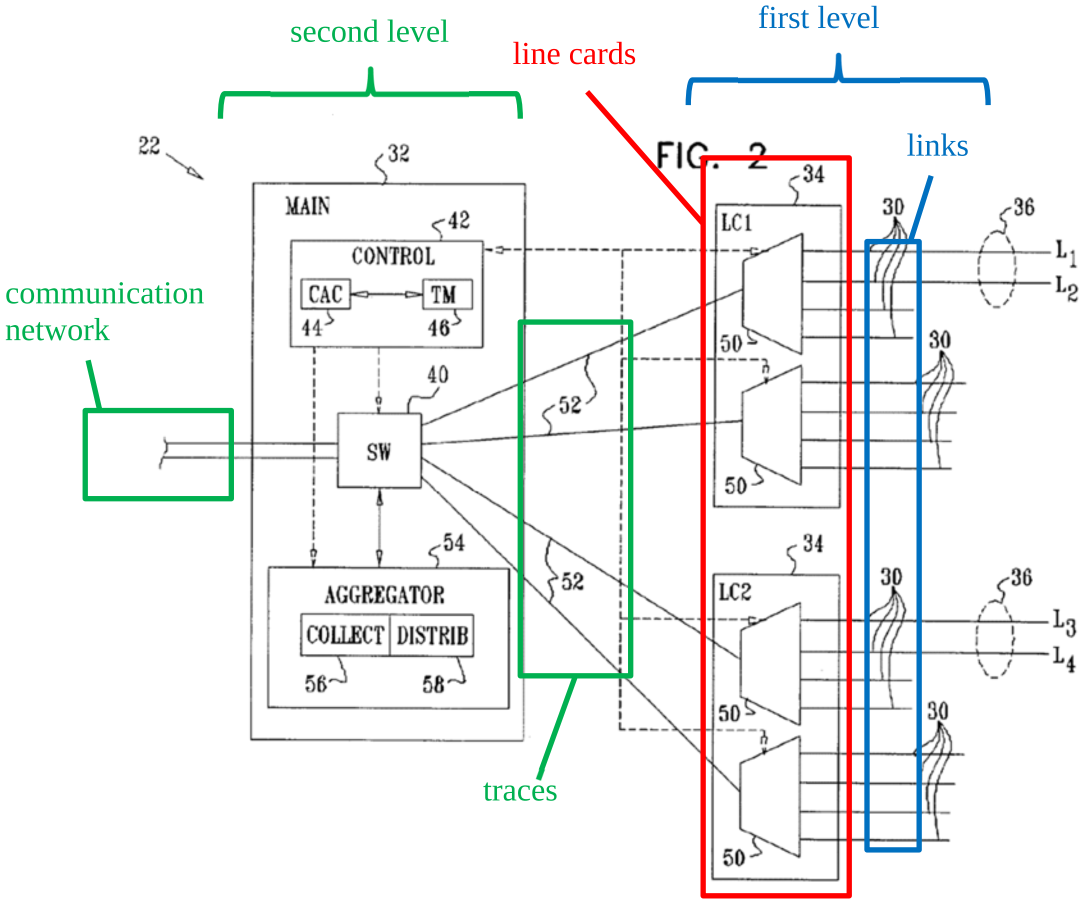
No.	’740 Patent Claim 10	Lebizay
		# of Crossbars=(# links per ingress channelx# of ingress channels per portx# of port cardsx speedup)/32. For the 8x8 configuration, the # of crossbars should be multiplied by (4x# of iMS)/(# backplane channels per port card). The number of port cards should be rounded up to the nearest supported configuration, i.e. 8, 16, or 32. The speedup in the case of crossbars should be the fractional speedup that is desired. Example to determine the number of arbiters and cross-bars for the following system: 4 channel port cards (40 Gbps) 8 links per channel 16 port cards Speedup=l.5 # of arbiters=(4x2x2)/2=8 # of crossbars=(8x4x16xl.5)/32=24. This would give 3crossbars per arbiter.”) Singh at 16:28-44 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 16x16 and 32x32 is the organization of the switchplane. The port card remains the same. Backplane channels 1 and 2 are used for the backplane connectivity. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 16, correspond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”) Singh at 17:31-49 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 8x8 and 16x16 is the

No.	’740 Patent Claim 10	Lebizay
		organi-zation of the switchplane. The port card remains the same. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Links 0-7 and 24-31 on the arbiters would not be used and would be powered off. Links 0-7 and 24-31 on the crossbars would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Backplane channels 1 and 2 are used for the backplane connectivity. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 8, corre-spond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”)

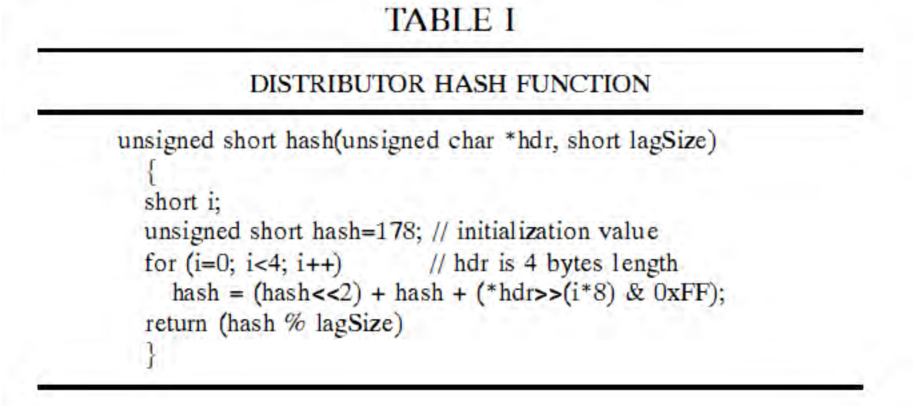
No.	’740 Patent Claim 11	Lebizay
11	The method according to claim 10, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo.	Lebizay discloses the method according to claim 10, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo. For example, Lebizay mapping packets traversing the fabric to its destination link. A person of ordinary skill in the art would understand that mapping includes applying a hash function by determining parameters responsive to the system and packet features, and generating a result. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Lebizay is found to not meet this limitation, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo would have been obvious to a person having ordinary skill in the art, as explained below. *See supra* Claim 10

No.	’740 Patent Claim 11	Lebizay
		Lebizay at [0043] (“InfiniBand offers link layer Virtual Lanes (VLs) to support multiple logical channels (i.e. multiplexing) on the same physical link. Infiniband offers up to 16 virtual lanes per link. VLs provide a mechanism to avoid head-of-line blocking and the ability to support Quality of Service (QoS). The difference between a Virtual Lane and a Service Level (SL) is that a Virtual Lane is the actual logical lane (mul-tiplexed) used on a given point-to-point link. The Service Level stays constant as a packet traverses the fabric, and specifies the desired service level within a subnet. The SL (AF, EF or BE) is included in the link header, and each switch maps the SL to a VL supported by the destination link. A switch supporting a limited number of virtual lanes will map the SL field to a VL it supports. Without preserving the SL, the desired SL (AF, EF or BE) would be lost in this mapping, and later in the path, a switch supporting more VLs would be unable to recover finer granularity of SLs between two packets mapped to the same VL.”) Lebizay at [0050] (“The issue with using a ring, however, is how to map the addressing of multiple boards across these fibers. One solution is to employ Wavelength Division Multiplex-ing (WDM). A WDM optical mesh defines a meshed-topology in the wavelength space as opposed to the physical fiber space. By utilizing multiple discrete lambda-waves as optical carriers such that by meshing dedicated optical wavelengths between every two boards, layer 2 protocols are eliminated, thereby creating a dramatic improvement in the efficiency of the transport. Today, every packet transport requires a protocol that allows the end point (and interme-diate points) to decipher the intended path (or consumer) of the packet. This protocol increases the amount of overhead required in the packet bus, allowing less room for actual data to be sent. By moving the protocol into the wavelength of the actual optical signal, the destination is implied by the wavelength and no additional bandwidth needs to be sur-rendered on the signal to provide this information. This makes the efficiency of the transport better and also speeds the routing of the packet through the network. In addition, the use of optical interconnects in a backplane environment greatly increases chassis bandwidth as well as reducing electrical radiation that often accompanies copper intercon-nects. The components involved include an optical back-plane in a physical ring topology, and the necessary trans-mitters and receivers for the size of the installation (i.e., number of slots in the chassis). In addition, optical add/drop multiplexer devices are required.”)

No.	’740 Patent Claim 11	Lebizay
		Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Lebizay in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 11 of Exhibit E-1 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses distributing data frames over physical links and traces based on a division operation in the hash function, involving specific byte lengths of the frame information. Bruckman at Figure 2 (annotated)

No.	’740 Patent Claim 11	Lebizay
		 second level line cards first level links communication network traces 22 32 MAIN 42 CONTROL CAC TM 44 46 40 SW 54 AGGREGATOR COLLECT DISTRIB 56 58 52 52 34 LC1 50 50 30 30 36 L_1 L_2 34 LC2 50 50 30 30 36 L_3 L_4 FIG. 2 Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:

No.	’740 Patent Claim 11	Lebizay

		 TABLE I
		DISTRIBUTOR HASH FUNCTION
		`unsigned short hash(unsigned char *hdr, short lagSize)`
		`{`
		`short i;`
		`unsigned short hash=178; // initialization value`
		`for (i=0; i<4; i++)          // hdr is 4 bytes length`
		`   hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);`
		`return (hash % lagSize)`
		`}`
		”)
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		As another example, Solomon discloses using a subset of bits to encode for the selected physical port, involving specific byte lengths of the frame information.
		Solomon at [0054] (“Having selected a physical port, RSVP-TE proces-sor 30 of switch A now generates a suitable MPLS label, at a label generation step 64. The preceding node upstream of switch A will subsequently attach this MPLS label to all MPLS packets transmitted through tunnel 28 to switch A. The label is assigned, in conjunction with the mapping function of mapper 34, so as to ensure that all MPLS packets carrying this label are switched through the physical port that was selected for this tunnel at step 62. For this purpose, RSVP-TE processor 30 of switch A dedicates a sub-set of the bits of MPLS label 52 to encode the serial number of the selected physical port. For example, the four least-signifi-cant bits of MPLS label 52 may be used for encoding the selected port number. This configuration is suitable for representing LAG groups

No.	’740 Patent Claim 11	Lebizay
		having up to 16 physical ports (N<16). The remaining bits of MPLS label 52 may be chosen at random or using any suitable method known in the art.”) Solomon at [0056] (“Mapper 34 of switch A maps the received packets belonging to tunnel 28 to the selected physical Ethernet port at a mapping step 70. For this purpose, mapper 34 extracts the MPLS label from each received packet and decodes the selected physical port number from the dedicated sub-set of bits, such as the four LSB, as described in step 64 above. The decoded value is used for mapping the packet to the selected physical port, which was allocated by the CAC processor at step 62 above. In the four-bit example described above, the mapping function may be written explicitly as: Selected port number=((MPLS label) and (0x0000F)), wherein "and" denotes the "bitwise and" operator.”)

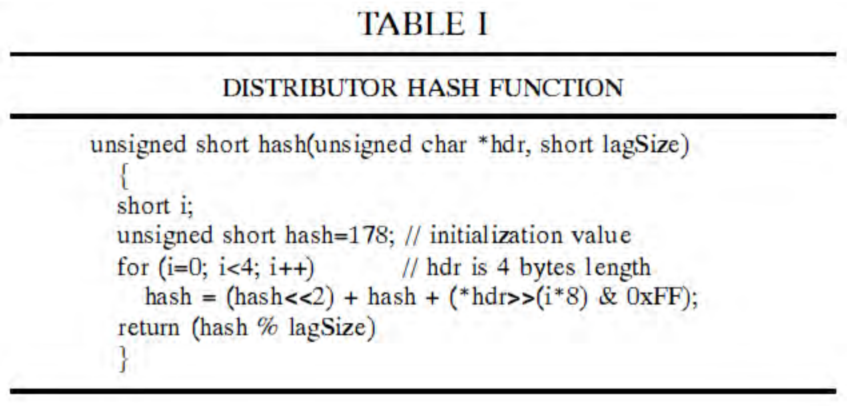
No.	’740 Patent Claim 12	Lebizay
12	The method according to claim 1, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination	Lebizay discloses the method according to claim 1, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port. For example, Lebizay discloses data packet characteristics including specific source and destination addresses corresponding to specific protocols. A person of ordinary skill in the art would understand that such frame information could include at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port, which are used to route the frame according to the port channel map table. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Ghosh is found to not meet this limitation, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer

No.	’740 Patent Claim 12	Lebizay
	MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port.	4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port would have been obvious to a person having ordinary skill in the art, as explained below. *See supra* at Claim 1. Lebizay at [0009] (“Infiniband uses Internet Protocol Version 6 (IPv6) headers natively, and can connect to Local Area Network (LAN) and Wide Area Network (WAN) switches and routers with the TCA providing a seamless transition between the system area network and external networks. InfiniBand defines network layers up to the transport layer and offers all four combinations of reliable/unreliable and connection datagram service. The Infiniband transport protocol is used within the system area network, but other transport protocols can be accessed by sending raw packets via a TCA. TCAs provide connections to storage, fibre channel networks, and other 1/0 nodes, and include an 1/0 controller specific to the device's protocol, be it Small Computer Systems Interface (SCSI), Ethernet, etc. A TCAincludes an Infiniband protocol engine that dramatically accelerates the performance of critical Infiniband transport functions in the TCA's hardware, achieving aggregate internal transaction switching throughput speeds of 150 gigabits per second. TCAs are highly optimized for Infiniband target applications such as bridging from Infiniband devices (switches) to local busses i.e., Gigabit Ethernet, Fibre Channel, and Transport Control Protocol/Internet Protocol TCP/IP devices, as well as next-generation 1/0 protocols.”) Lebizay at [0028] (“Embodiments of the present invention exists within the context of connecting multiple entities within a system (specifically multiple boards within a chassis), using mul-tiple TCAs. The components consist of multi-port TCAs and a meshed backplane that is equipped to mount boards via connectors on the backplane. A multi-port TCA capable of performing multiple bridging functions simultaneously i.e., bridging from an Infiniband meshed backplane to multiple local busses, i.e., Gigabit Ethernet, Fibre Channel, and TCP/IP devices, is referred to as a Fabric Interconnect Chip (FIC). FIG. 1 depicts the connection of a

No.	’740 Patent Claim 12	Lebizay
		FIC 100 that interconnects between local components on a board (local busses) 120 and a backplane mesh fabric 110.”) Lebizay at [0035] (“A flow is a set of packets that all share a set of characteristics. Typically, the characteristics include the source and destination address of the packet, as well as its protocol type and possibly its priority or classification. It is important that all the packets in a flow maintain a certain sequence in which they were sent, preferably arriving at their destination in that same sequence. If they do arrive out of sequence they can be re-sequenced, or put back in order. However, it is not desireable to re-sequence packets at the end. Therefore, a good design attempts to keep all the packets in a flow in sequence all through the network so that they arrive at the far end in sequence and do not require re-sequencing.”) Lebizay at [0043] (“InfiniBand offers link layer Virtual Lanes (VLs) to support multiple logical channels (i.e. multiplexing) on the same physical link. Infiniband offers up to 16 virtual lanes per link. VLs provide a mechanism to avoid head-of-line blocking and the ability to support Quality of Service (QoS). The difference between a Virtual Lane and a Service Level (SL) is that a Virtual Lane is the actual logical lane (mul-tiplexed) used on a given point-to-point link. The Service Level stays constant as a packet traverses the fabric, and specifies the desired service level within a subnet. The SL (AF, EF or BE) is included in the link header, and each switch maps the SL to a VL supported by the destination link. A switch supporting a limited number of virtual lanes will map the SL field to a VL it supports. Without preserving the SL, the desired SL (AF, EF or BE) would be lost in this mapping, and later in the path, a switch supporting more VLs would be unable to recover finer granularity of SLs between two packets mapped to the same VL.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Lebizay in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 12 of Exhibit E-3 renders the claim, including the present limitation, obvious. Below is one such example.

No.	’740 Patent Claim 12	Lebizay
		For example, Bruckman discloses frame information including header fields, source addresses, destination addresses, ports, etc. Bruckman specifically discloses Ethernet frame information including source MAC address, destination MAC address, reception port, type of destination address, Ethernet Length/Type value, and higher layer protocol information. Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”)

No.	’740 Patent Claim 12	Lebizay

		Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria:
		[0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link.
		[0060] The hash function uses all the specified input data from the frame headers.
		[0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values
		[0062] The hash function generates very different hash values for similar data.”)
		
		Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:
		
		TABLE I
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		  {
		  short i;
		  unsigned short hash=178; // initialization value
		  for (i=0; i<4; i++) // hdr is 4 bytes length
		    hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		  return (hash % lagSize)
		  }
		”)
		
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively,

118

No.	’740 Patent Claim 12	Lebizay
		dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)

No.	’740 Patent Claim 13	Lebizay
13[preamble]	A method for communication, comprising:	Lebizay discloses a method for communication. *See supra at* 1[preamble].
13[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	Lebizay discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].
13[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Lebizay discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra at* 1[c].
13[c]	receiving a data frame having frame attributes sent between the communication	Lebizay discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra at* 1[e].

No.	’740 Patent Claim 13	Lebizay
	network and the network node:	
13[d]	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	Lebizay discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. *See supra at* 1[f].
13[e]	sending the data frame over the selected first and second physical links,	Lebizay discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].
13[f]	coupling the network node to the one or more interface modules and	Lebizay discloses coupling the network node to the one or more interface modules. *See supra at* 1[a].
13[g]	coupling each of the one or more interface modules to the communication network comprising	Lebizay discloses coupling each of the one or more interface modules to the communication network. *See supra at* 1[c].
13[h]	specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak	Lebizay discloses specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node. For example, Lebizay discloses bandwidth considerations that are specific to a network service provider. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement

No.	’740 Patent Claim 13	Lebizay
	information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node, and	Disclosures, this limitation is met. To the extent that the Lebizay is found to not meet this limitation, coupling each of the one or more interface modules to the communication network comprising specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node would have been obvious to a person having ordinary skill in the art, as explained below. Lebizay at [0037] (“For example, an entity may set up a Service Level Agreement (SLA) with a network service provider (the entity that is providing the network), that specifies that the entity's traffic will always have available a certain band-width (e.g., 10 Megabits per second, or 10 Mbps) and latency (e.g., less than 1 millisecond, or ms). Then, when-ever a packet is detected that comes from or is going to that entity, the packet should receive special handling. If the overall flow to the entity is currently less than 10 Mbps, then the packet should get through without being dropped and with a latency of less than 1 ms. This type of flow is said to be handled by Assured Forwarding (AF). Packets arriving when the current flows are greater than 10 Mbps will be handled differently, perhaps as Best Effort (BE) traffic (see below).”) Lebizay at [0038] (“As another example, a router may be set up to recognize certain types of flows as real-time flows. Real-time flows are characterized by the idea that if the packet doesn't arrive in time it might as well not arrive at all. For example, a packet of voice data in a telephone conversation has to be available at the receiver when it is needed, or it is useless. Too late, and it cannot be used and will just be discarded. So real-time traffic (a stream of packets) belong-ing to a voice conversation should be handled by a class of behavior known as Expedited Forwarding (EF). A packet handled this way will be forwarded very quickly (with low latency). Hopefully, the variation in latency (known as jitter) will also be low. As a tradeoff, packets in such a stream may be simply dropped if their aggregate bandwidth exceeds a certain threshold. Also, a SLA covering such packets may be expensive to the buyer because providing this kind of service requires that a router have features that make it expensive to build.”)

No.	’740 Patent Claim 13	Lebizay
		Lebizay at [0046] (“Today, boards in a common chassis interconnect to each other over copper traces embedded in the chassis backplane. Existing backplanes use copper traces within a PCB substrate to establish the communications paths between boards. Topologies such as common bus PCI and switched star (i.e. some Ethernet approaches) are popular today. The copper traces and the electrical drivers that communicate over them are limited in terms of their capac-ity to carry data bandwidth with respect to the Local Area Network (LAN) and Wide Area Network (WAN) optical signals that these boards are intended to terminate.”) Lebizay at [0050] (“The issue with using a ring, however, is how to map the addressing of multiple boards across these fibers. One solution is to employ Wavelength Division Multiplex-ing (WDM). A WDM optical mesh defines a meshed-topology in the wavelength space as opposed to the physical fiber space. By utilizing multiple discrete lambda-waves as optical carriers such that by meshing dedicated optical wavelengths between every two boards, layer 2 protocols are eliminated, thereby creating a dramatic improvement in the efficiency of the transport. Today, every packet transport requires a protocol that allows the end point (and interme-diate points) to decipher the intended path (or consumer) of the packet. This protocol increases the amount of overhead required in the packet bus, allowing less room for actual data to be sent. By moving the protocol into the wavelength of the actual optical signal, the destination is implied by the wavelength and no additional bandwidth needs to be sur-rendered on the signal to provide this information. This makes the efficiency of the transport better and also speeds the routing of the packet through the network. In addition, the use of optical interconnects in a backplane environment greatly increases chassis bandwidth as well as reducing electrical radiation that often accompanies copper intercon-nects. The components involved include an optical back-plane in a physical ring topology, and the necessary trans-mitters and receivers for the size of the installation (i.e., number of slots in the chassis). In addition, optical add/drop multiplexer devices are required.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Lebizay in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 13[h] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references.

No.	’740 Patent Claim 13	Lebizay
		For example, Bruckman discloses specifying certain committed, excess, and guaranteed bandwidths, including CIR, EIR, and PIR, respectively. Bruckman at [0013] (“Service level agreements between network service providers and customers commonly specify a certain com-mitted bandwidth, or committed information rate (CIR), which the service provider guarantees to provide to the customer at all times, regardless of bandwidth stress on the network. Additionally or alternatively, the agreement may specify an excess bandwidth, which is available to the customer when network traffic permits. The excess band-width is typically used by customers for lower-priority services, which do not require committed bandwidth. The network service provider may guarantee the customer a certain minimum excess bandwidth, or excess information rate (EIR), in order to avoid starvation of such services in case of bandwidth stress. In general, the bandwidth guaran-teed by a service provider, referred to as the peak informa-tion rate (PIR), may include either CIR, or EIR, or both CIR and EIR (in which case PIR=CIR+EIR). The term "guaran-teed bandwidth," as used in the context of the present patent application and in the claims, includes all these types of guaranteed bandwidth.”) As another example, Solomon discloses a service property of a guaranteed bandwidth, sometimes denoted as CIR-Committed Information Rate and PIR-Peak Information Rate. Solomon at [0023] (“In another embodiment, establishing the path includes receiving an indication of a requested service property of the flow, and selecting the port includes assign-ing the port to the flow so as to comply with the requested service property. In a disclosed embodiment, the requested service property includes at least one of a guaranteed bandwidth, a peak bandwidth and a class-of-service. Addi-tionally or alternatively, assigning the port includes selecting the port having a maximum available bandwidth out of the plurality of aggregated ports. Further additionally or alter-natively, assigning the port includes selecting the port hav-ing a minimum available bandwidth out of the plurality of aggregated ports, which is still greater than or equal to the guaranteed bandwidth.”)

No.	’740 Patent Claim 13	Lebizay
		Solomon at [0050] (“The method of FIG. 3 begins when the preceding node asks to establish a part of tunnel 28 (comprising one or more hops) for sending MPLS packets to MPLS/LAG switch 26 A. The preceding node requests and then receives the MPLS label, which it will subsequently attach to all packets that are sent to MPLS/LAG switch 26 labeledA. The preceding node sends downstream an RSVP-TE PATH mes-sage augmented with a LABEL_REQUEST object, as defined by RSVP-TE, to MPLS/LAG switch A, at a label requesting step 60. The PATH message typically comprises information regarding service properties that are requested for tunnel 28. The service properties may comprise a guar-anteed bandwidth (sometimes denoted CIR-Committed Information Rate) and a peak bandwidth (sometimes denoted PIR-Peak Information Rate), as well as a requested CoS (Class of Service-a measure of packet priority).”)
13[i]	allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements.	Lebizay discloses allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements. For examples, Lebizay discloses network service provider specified bandwidth considerations over ports and backplane connectors. Lebizay at [0021] (“Since VIA was only intended to be used for com-munication across the physical servers of a cluster (in other words across high-bandwidth links with very high reliabil-ity), the specification eliminates much of the standard net-work protocol code that deals with special cases. Also, because of the well-defined environment of operation, the message exchange protocol was defined to avoid kernel mode interaction and allow for access to the NIC from user mode. Finally, because of the direct access to the NIC, unnecessary copying of the data into kernel buffers was also eliminated since the user is able to directly transfer data from user-space to the NIC. In addition to the standard send/ receive operations that are typically available in a network-ing library, the VIA provides Remote Direct Memory Access (RDMA) operations where the initiator of the operation specifies both the source and destination of a data transfer, resulting in zero-copy data transfers with minimum involve-ment of the CPUs.”)

No.	’740 Patent Claim 13	Lebizay
		Lebizay at [0034] (“Links are the connections between boards (or node to node, or switch to switch, in general). They are capable of transmitting no more than a certain amount of information at any one time, known as bandwidth. Bandwidth may be measured, for example, in bits/second. In general, there is no restriction on the number of boards (N) except that it be greater than 1.”) Lebizay at [0037] (“For example, an entity may set up a Service Level Agreement (SLA) with a network service provider (the entity that is providing the network), that specifies that the entity's traffic will always have available a certain band-width (e.g., 10 Megabits per second, or 10 Mbps) and latency (e.g., less than 1 millisecond, or ms). Then, when-ever a packet is detected that comes from or is going to that entity, the packet should receive special handling. If the overall flow to the entity is currently less than 10 Mbps, then the packet should get through without being dropped and with a latency of less than 1 ms. This type of flow is said to be handled by Assured Forwarding (AF). Packets arriving when the current flows are greater than 10 Mbps will be handled differently, perhaps as Best Effort (BE) traffic (see below).”) Lebizay at [0038] (“As another example, a router may be set up to recognize certain types of flows as real-time flows. Real-time flows are characterized by the idea that if the packet doesn't arrive in time it might as well not arrive at all. For example, a packet of voice data in a telephone conversation has to be available at the receiver when it is needed, or it is useless. Too late, and it cannot be used and will just be discarded. So real-time traffic (a stream of packets) belong-ing to a voice conversation should be handled by a class of behavior known as Expedited Forwarding (EF). A packet handled this way will be forwarded very quickly (with low latency). Hopefully, the variation in latency (known as jitter) will also be low. As a tradeoff, packets in such a stream may be simply dropped if their aggregate bandwidth exceeds a certain threshold. Also, a SLA covering such packets may be expensive to the buyer because providing this kind of service requires that a router have features that make it expensive to build.”) Lebizay at [0048] (“As communications rates grow, the need for faster back plane technologies also grows. The limits of perfor-mance on copper traces within a PCB substrate will soon be

No.	’740 Patent Claim 13	Lebizay
		reached giving rise to the need to investigate optical inter-connects for the backplane. Optical fibers may be employed to provide increased bandwidth while using only a few fibers in a small space. Copper traces, on the other hand, often have to contain multiple (parallel) interconnects in order to achieve a limited bandwidth.”)

No.	’740 Patent Claim 14	Lebizay
14[preamble]	A method for connecting user ports to a communication network, comprising:	Lebizay discloses a method for connecting user ports to a communication network. For example, Lebizay discloses steps for ports to a communication network for data transfer. Lebizay at Abstract (“Multiple boards are connected within a chassis, using a multi-port Target Channel Adapter (TCA). Data is transported from a TCA on a board directly to a TCA on another board through a meshed backplane. The meshed backplane is equipped to mount boards via connectors and may consist of a fabric of copper conductors or optical fibers. Commu-nication from TCA to TCA requires placing ports on each individual TCA along with the appropriate input and output buffering. A multi-port TCA capable of performing multiple bridging functions simultaneously i.e., bridging from a high speed serial meshed backplane to multiple local busses, i.e., Gigabit Ethernet, Fibre Channel, and TCP/IP devices, is referred to as a Fabric Interconnect Chip (FIC).”) Lebizay at [0022] (“The InfiniBand Architecture uses basically the VIA primitives for its operation at the transport layer. In order for an application to communicate with another application over the InfiniBand it must first create a work queue that consists of a queue pair (QP). In order for the application to execute an operation, it must place a work queue element (WQE) in the work queue. From there, the operation is picked-up for execution by the channel adapter. Therefore, the Work Queue forms the communications medium between appli-cations and the channel adapter, relieving the operating system from having to deal with this responsibility.”)

No.	’740 Patent Claim 14	Lebizay
		Lebizay at [0030] (“The FIC provides data queuing first in first out (FIFOs) on both its inbound and outbound sides for each of its multiple ports. By placing FICs on every board within a chassis that is connected to the meshed backplane, the boards can inter-communicate with each other in a "meshed" topology. Therefore, every board has a point to point connection with every other board in the backplane, making a complete cross-connect without the need for a separate switch device.”)
14[a]	coupling the user ports to one or more user interface modules;	Lebizay discloses coupling the user ports to one or more user interface modules. *See supra at* 1[a].
14[b]	coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel,	Lebizay discloses coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel. *See supra at* 1[c], 3.
14[c]	at least one of said backplane traces being bi-directional and operative to communicate in both an upstream direction and a downstream direction;	Lebizay discloses at least one of said backplane traces being bi-directional and operative to communicate in both an upstream direction and a downstream direction. *See supra at* 14[b], 1[d].
14[d]	receiving data frames sent between the user ports and the	Lebizay discloses receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes.

No.	’740 Patent Claim 14	Lebizay
	communication network, the data frames having respective frame attributes;	*See supra at* 14[a], 1[e].
14[e]	for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces; and	Lebizay discloses for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces. *See supra at* 14[b], 1[f].
14[f]	sending the data frame over the selected backplane trace;	Lebizay discloses sending the data frame over the selected backplane trace. *See supra at* 14[e], 1[g].
14[g]	said sending comprising communicating along said at least one of said backplane traces.	Lebizay discloses said sending comprising communicating along said at least one of said backplane traces. *See supra at* 14[f], 1[h].

No.	’740 Patent Claim 15	Lebizay
15[preamble]	A method for connecting user ports to a communication network, comprising:	Lebizay discloses a method for connecting user ports to a communication network. *See supra at* 14[preamble].

No.	’740 Patent Claim 15	Lebizay
15[a]	coupling the user ports to one or more user interface modules;	Lebizay discloses coupling the user ports to one or more user interface modules. *See supra at* 14[a].
15[b]	coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel;	Lebizay discloses coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel. *See supra at* 14[b].
15[c]	receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes;	Lebizay discloses receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes. *See supra at* 14[d].
15[d]	for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces; and	Lebizay discloses for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces. *See supra at* 14[e].
15[e]	sending the data frame over the	Lebizay discloses sending the data frame over the selected backplane trace.

No.	’740 Patent Claim 15	Lebizay
	selected backplane trace,	*See supra at* 14[f].
15[f]	at least some of the backplane traces being aggregated into an Ethernet link aggregation (LAG) group.	Lebizay discloses at least some of the backplane traces being aggregated into an Ethernet link aggregation (LAG) group. *See supra at* 15[e], 4[f], 3.

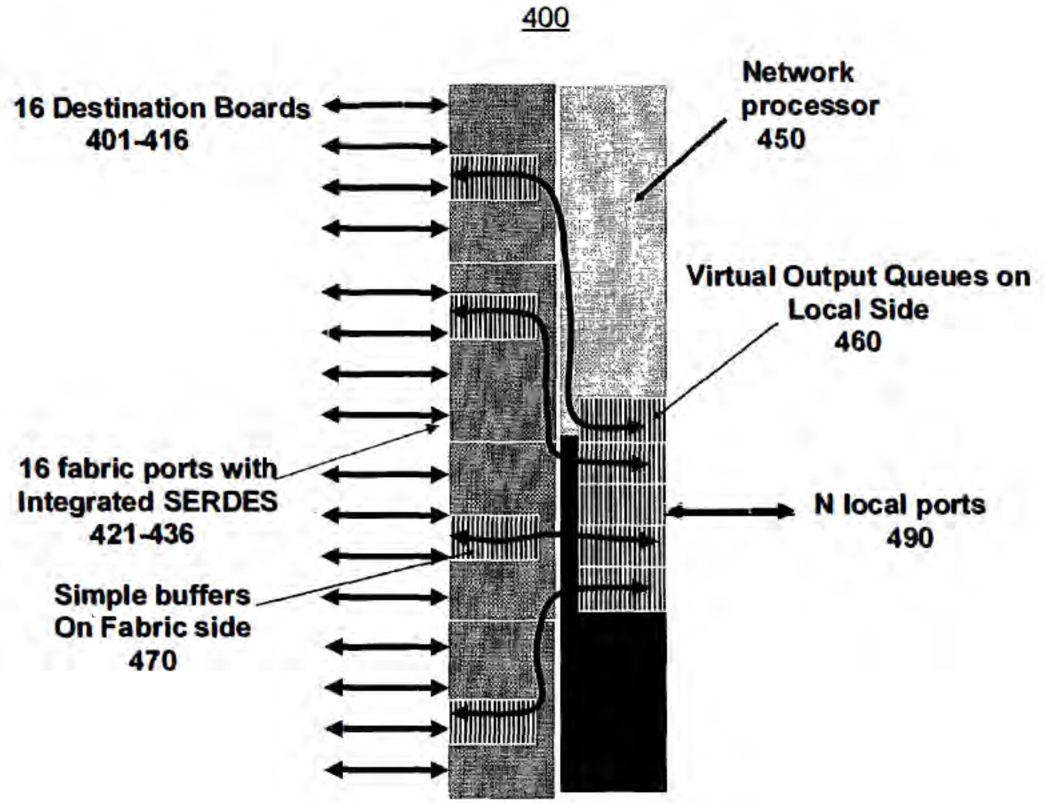
No.	’740 Patent Claim 16	Lebizay
16	The method according to claim 14, wherein selecting the backplane trace comprises applying a hashing function to the at least one of the frame attributes.	Lebizay discloses the method according to claim 14, wherein selecting the backplane trace comprises applying a hashing function to the at least one of the frame attributes. *See supra* at 14, 9, 8.

No.	’740 Patent Claim 17	Lebizay
17[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Lebizay discloses apparatus for connecting a network node with a communication network. For example, Lebizay discloses a communication system connecting nodes to a communication network for data transfer. Lebizay at Abstract (“Multiple boards are connected within a chassis, using a multi-port Target Channel Adapter (TCA). Data is transported from a TCA on a board directly to a TCA on another board through a meshed backplane. The meshed backplane is equipped to mount boards via connectors and may consist of a fabric of copper conductors or optical fibers.

No.	’740 Patent Claim 17	Lebizay
		Commu-nication from TCA to TCA requires placing ports on each individual TCA along with the appropriate input and output buffering. A multi-port TCA capable of performing multiple bridging functions simultaneously i.e., bridging from a high speed serial meshed backplane to multiple local busses, i.e., Gigabit Ethernet, Fibre Channel, and TCP/IP devices, is referred to as a Fabric Interconnect Chip (FIC).”) Lebizay at [0022] (“The InfiniBand Architecture uses basically the VIA primitives for its operation at the transport layer. In order for an application to communicate with another application over the InfiniBand it must first create a work queue that consists of a queue pair (QP). In order for the application to execute an operation, it must place a work queue element (WQE) in the work queue. From there, the operation is picked-up for execution by the channel adapter. Therefore, the Work Queue forms the communications medium between appli-cations and the channel adapter, relieving the operating system from having to deal with this responsibility.”) Lebizay at [0030] (“The FIC provides data queuing first in first out (FIFOs) on both its inbound and outbound sides for each of its multiple ports. By placing FICs on every board within a chassis that is connected to the meshed backplane, the boards can inter-communicate with each other in a "meshed" topology. Therefore, every board has a point to point connection with every other board in the backplane, making a complete cross-connect without the need for a separate switch device.”)
17[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network,	Lebizay discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. For example, Lebizay discloses multiple interconnected FICs that process data flows with specific characteristics between nodes and the network. Lebizay at [0028] (“Embodiments of the present invention exists within the context of connecting multiple entities within a system (specifically multiple boards within a chassis), using mul-tiple TCAs. The components consist of multi-port TCAs and a meshed backplane

No.	’740 Patent Claim 17	Lebizay
		that is equipped to mount boards via connectors on the backplane. A multi-port TCA capable of performing multiple bridging functions simultaneously i.e., bridging from an Infiniband meshed backplane to multiple local busses, i.e., Gigabit Ethernet, Fibre Channel, and TCP/IP devices, is referred to as a Fabric Interconnect Chip (FIC). FIG. 1 depicts the connection of a FIC 100 that interconnects between local components on a board (local busses) 120 and a backplane mesh fabric 110.”) Lebizay at [0029] (“FIG. 2 depicts connection of multiple interconnected FICs. Each board 211-218 on the mesh 220 contains one FIC 201-208 . Each FIC may provide an interconnection to every other board in the shelf (chassis). FIG. 2 shows an 8-way (8 board) mesh, but any size mesh is realizable, hence the 8,h board 218 is labeled "N". The lines drawn show how the traces from each FIC 'port' travels across the passive backplane interconnect fabric 220 to its corresponding 'port' on another FIC, (located on another board).”) Lebizay at [0030] (“The FIC provides data queuing first in first out (FIFOs) on both its inbound and outbound sides for each of its multiple ports. By placing FICs on every board within a chassis that is connected to the meshed backplane, the boards can inter-communicate with each other in a "meshed" topology. Therefore, every board has a point to point connection with every other board in the backplane, making a complete cross-connect without the need for a separate switch device.”) Lebizay at [0031] (“In addition, by incorporating the queuing logic on both the inbound and outbound sides of the FIC, coupled with the inherent cross-connect of the mesh, a logical, distributed switch is created, without really having a switch device. FIG. 3 illustrates the same 8-way mesh depicted in FIG. 2, highlighting the logic switch which is created by distributing the switch function across FICs 201-208 on all the boards 211-218. Each multi-port FIC 201-208 is shown having inbound and outbound queues 221-228 for each port. In total, what is depicted is a 8-way logical distributed switch, where each local bus (board 211-218) represents an access point on the logical switch.”)

No.	’740 Patent Claim 17	Lebizay
		Lebizay at [0040] (“Processing of packets from ingress to fabric input is shown in FIG. 4. Depicted is the flow of packets through a single FIC 400 located on a board going to 16 destination boards 401-416. A network processor 450 on the FIC 400 classifies all outgoing traffic to a particular flow. The flow defines a set of parameters including the 16 destination boards 401-416, the path by which it will get to the desti-nation board, i.e., one of the 16 fabric ports 421-436, and classification (AF, EF or BE). The individual packets end up in virtual output (send) queues 460 according to the classi-fication. There is a separate set of virtual output (send) queues 460 (AF, EF or BE) for every destination point, i.e., fabric port 421-436, per FIC. The fabric ports 421-436 and N local ports 490 have both send and receive terminals.”) Lebizay at Figure 4

No.	’740 Patent Claim 17	Lebizay
		 400 16 Destination Boards 401-416 Network processor 450 Virtual Output Queues on Local Side 460 16 fabric ports with Integrated SERDES 421-436 N local ports 490 Simple buffers On Fabric side 470 FIG. 4

No.	’740 Patent Claim 17	Lebizay
17[b]	at least one of said interface modules being operative to communicate in both an upstream direction and a downstream direction;	Lebizay discloses at least one of said interface modules being operative to communicate in both an upstream direction and a downstream direction. For example, Lebizay discloses multiple interconnected FICs that communicate in both inbound and outbound directions. Lebizay at [0028] (“Embodiments of the present invention exists within the context of connecting multiple entities within a system (specifically multiple boards within a chassis), using mul-tiple TCAs. The components consist of multi-port TCAs and a meshed backplane that is equipped to mount boards via connectors on the backplane. A multi-port TCA capable of performing multiple bridging functions simultaneously i.e., bridging from an Infiniband meshed backplane to multiple local busses, i.e., Gigabit Ethernet, Fibre Channel, and TCP/IP devices, is referred to as a Fabric Interconnect Chip (FIC). FIG. 1 depicts the connection of a FIC 100 that interconnects between local components on a board (local busses) 120 and a backplane mesh fabric 110.”) Lebizay at [0029] (“FIG. 2 depicts connection of multiple intercon-nected FICs. Each board 211-218 on the mesh 220 contains one FIC 201-208 . Each FIC may provide an interconnection to every other board in the shelf (chassis). FIG. 2 shows an 8-way (8 board) mesh, but any size mesh is realizable, hence the 8,h board 218 is labeled "N". The lines drawn show how the traces from each FIC 'port' travels across the passive backplane interconnect fabric 220 to its corresponding 'port' on another FIC, (located on another board).”) Lebizay at [0030] (“The FIC provides data queuing first in first out (FIFOs) on both its inbound and outbound sides for each of its multiple ports. By placing FICs on every board within a chassis that is connected to the meshed backplane, the boards can inter-communicate with each other in a "meshed" topology. Therefore, every board has a point to point connection with every other board in the backplane, making a complete cross-connect without the need for a separate switch device.”)

No.	’740 Patent Claim 17	Lebizay
		Lebizay at [0031] (“In addition, by incorporating the queuing logic on both the inbound and outbound sides of the FIC, coupled with the inherent cross-connect of the mesh, a logical, distributed switch is created, without really having a switch device. FIG. 3 illustrates the same 8-way mesh depicted in FIG. 2, highlighting the logic switch which is created by distributing the switch function across FICs 201-208 on all the boards 211-218. Each multi-port FIC 201-208 is shown having inbound and outbound queues 221-228 for each port. In total, what is depicted is a 8-way logical distributed switch, where each local bus (board 211-218) represents an access point on the logical switch.”) Lebizay at [0032] (“Since each FIC (201-208) has queuing (221-228) on both the inbound and outbound sides, coupled with the inherent cross-connect in the backplane 220, an extremely efficient logically distributed switch can be built at much lower cost, greater reliability, higher efficiency, lower power and lower latency than a typical, centrally switched topol-ogy, or any architecture that places a switch on every board of a meshed topology.”)
17[c]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Lebizay discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 1[a].
17[d]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Lebizay discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 1[c].

No.	’740 Patent Claim 17	Lebizay
17[e]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame;	Lebizay discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 1[f].
17[f]	at least one of said first physical links and at least one of said second links being bi-directional links operative to communicate in both said upstream direction and said downstream direction.	Lebizay discloses at least one of said first physical links and at least one of said second links being bi-directional links operative to communicate in both said upstream direction and said downstream direction. *See supra at* 1[b], 1[d].

No.	’740 Patent Claim 18	Lebizay
18[a]	The apparatus according to claim 17, and comprising a backplane to which the one or more interface modules are coupled,	Lebizay discloses the apparatus according to claim 17, and comprising a backplane to which the one or more interface modules are coupled. *See supra at* 3, 17.
18[b]	wherein the second physical links comprise backplane traces formed on the backplane.	Lebizay discloses wherein the second physical links comprise backplane traces formed on the backplane. *See supra at* 3, 17.

No.	’740 Patent Claim 19	Lebizay
19[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Lebizay discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
19[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	Lebizay discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].

No.	’740 Patent Claim 19	Lebizay
19[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Lebizay discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].
19[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Lebizay discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].
19[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second	Lebizay discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].

No.	’740 Patent Claim 19	Lebizay
	group over which to send the data frame,	
19[e]	at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group.	Lebizay discloses at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group. *See supra at* 4[f].

No.	’740 Patent Claim 20	Lebizay
20[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Lebizay discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
20[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	Lebizay discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].
20[b]	a first group of first physical links arranged in parallel so as to couple the network node to the	Lebizay discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].

No.	’740 Patent Claim 20	Lebizay
	one or more interface modules;	
20[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Lebizay discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].
20[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,	Lebizay discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].
20[e]	two or more of the first physical links being aggregated into an external Ethernet	Lebizay discloses two or more of the first physical links being aggregated into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node.

No.	’740 Patent Claim 20	Lebizay
	link aggregation (LAG) group so as to increase a data bandwidth provided to the network node.	*See supra at* 19[e], 5[f].

No.	’740 Patent Claim 21	Lebizay
21	The apparatus according to claim 17, and comprising a multiplexer, which is arranged to perform at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node.	Lebizay discloses the apparatus according to claim 17, and comprising a multiplexer, which is arranged to perform at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node. *See supra at* 6, 17.

No.	’740 Patent Claim 22	Lebizay
22	The apparatus according to claim	Lebizay discloses the apparatus according to claim 17, wherein the control module is arranged to balance a frame data rate among at least some of the first and second physical links.

No.	’740 Patent Claim 22	Lebizay
	17, wherein the control module is arranged to balance a frame data rate among at least some of the first and second physical links.	*See supra at* 7, 17.

No.	’740 Patent Claim 23	Lebizay
23	The apparatus according to claim 17, wherein the control module is arranged to apply a mapping function to the at least one of the frame attributes so as to select the first and second physical links.	Lebizay discloses the apparatus according to claim 17, wherein the control module is arranged to apply a mapping function to the at least one of the frame attributes so as to select the first and second physical links. *See supra at* 8, 17.

No.	’740 Patent Claim 24	Lebizay
24	The apparatus according to claim 23, wherein the mapping function comprises a hashing function.	Lebizay discloses the apparatus according to claim 23, wherein the mapping function comprises a hashing function. *See supra at* 9, 23.

No.	’740 Patent Claim 25	Lebizay
25[a]	The apparatus according to claim 24, wherein the control module is arranged to determine a hashing size responsively to a number of at least some of the first and second physical links,	Lebizay discloses the apparatus according to claim 24, wherein the control module is arranged to determine a hashing size responsively to a number of at least some of the first and second physical links. *See supra at* 10[a], 24.
25[b]	to apply the hashing function to the at least one of the frame attributes to produce a hashing key,	Lebizay discloses to apply the hashing function to the at least one of the frame attributes to produce a hashing key. *See supra at* 10[b].
25[c]	to calculate a modulo of a division operation of the hashing key by the hashing size, and	Lebizay discloses to calculate a modulo of a division operation of the hashing key by the hashing size. *See supra at* 10[c].
25[d]	to select the first and second physical links responsively to the modulo.	Lebizay discloses to select the first and second physical links responsively to the modulo. *See supra at* 10[d].

No.	’740 Patent Claim 26	Lebizay
26	The apparatus according to claim 25, wherein the control module is arranged to select the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo.	Lebizay discloses the apparatus according to claim 25, wherein the control module is arranged to select the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo. *See supra at* 11, 25.

No.	’740 Patent Claim 27	Lebizay
27	The apparatus according to claim 17, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC)	Lebizay discloses the apparatus according to claim 17, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port. *See supra at* 12, 17.

No.	’740 Patent Claim 27	Lebizay
	address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port.	

No.	’740 Patent Claim 28	Lebizay
28[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Lebizay discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
28[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	Lebizay discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].
28[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Lebizay discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].

No.	’740 Patent Claim 28	Lebizay
28[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Lebizay discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].
28[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,	Lebizay discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].
28[e]	the communication network being arranged to provide a communication service to the network node,	Lebizay discloses the communication network being arranged to provide a communication service to the network node. *See supra at* 2[b].

No.	’740 Patent Claim 28	Lebizay
28[f]	the service having specified bandwidth requirements comprising at least one of a committed information rate (CR), a peak information rate (PIR) and an excess information rate (EIR), and	Lebizay discloses the service having specified bandwidth requirements comprising at least one of a committed information rate (CR), a peak information rate (PIR) and an excess information rate (EIR). *See supra at* 13[i].
28[g]	the first and second groups of physical links being dimensioned to provide an allocated bandwidth for the communication service responsively to the bandwidth requirements.	Lebizay discloses the first and second groups of physical links being dimensioned to provide an allocated bandwidth for the communication service responsively to the bandwidth requirements. *See supra at* 13[j].

No.	’740 Patent Claim 29	Lebizay
29[preamble]	Apparatus for connecting user ports to a communication network, comprising:	Lebizay discloses apparatus for connecting user ports to a communication network. *See supra at* 17[preamble], 14[preamble].
29[a]	one or more user interface modules coupled to the user	Lebizay discloses one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network.

No.	’740 Patent Claim 29	Lebizay
	ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network,	*See supra at* 17[a], 14[a].
29[b]	at least one of said user interface modules being bi-directional and operative to communicate in both an upstream direction and a downstream direction;	Lebizay discloses at least one of said user interface modules being bi-directional and operative to communicate in both an upstream direction and a downstream direction. *See supra at* 17[b], 14[c].
29[c]	a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the	Lebizay discloses a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network. *See supra at* 14[b]-[e].

No.	’740 Patent Claim 29	Lebizay
	communication network,	
29[d]	at least one of said backplane traces being bi-directional and operative to communicate in both said upstream direction and said downstream direction; and	Lebizay discloses at least one of said backplane traces being bi-directional and operative to communicate in both said upstream direction and said downstream direction. *See supra at* 14[c], 17[b].
29[e]	a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame.	Lebizay discloses a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame. *See supra at* 14[e], 17[e].

No.	’740 Patent Claim 30	Lebizay
30[preamble]	Apparatus for connecting user ports to a communication network, comprising:	Lebizay discloses apparatus for connecting user ports to a communication network. *See supra at* 29[preamble].

No.	’740 Patent Claim 30	Lebizay
30[a]	one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network;	Lebizay discloses one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network. *See supra at* 29[a].
30[b]	a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network;	Lebizay discloses a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network. *See supra at* 29[c].
30[c]	a control module, which is arranged to select, for each data frame, responsively to at least one of the	Lebizay discloses a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame. *See supra at* 29[e].

No.	’740 Patent Claim 30	Lebizay
	frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame;	
30[d]	at least some of the backplane traces are aggregated into an Ethernet link aggregation (LAG) group.	Lebizay discloses at least some of the backplane traces are aggregated into an Ethernet link aggregation (LAG) group. *See supra at* 4[f], 15[f].

No.	’740 Patent Claim 31	Lebizay
31	The apparatus according to claim 29, wherein the control module is arranged to apply a hashing function to the at least one of the frame attributes so as to select the backplane trace.	Lebizay discloses the apparatus according to claim 29, wherein the control module is arranged to apply a hashing function to the at least one of the frame attributes so as to select the backplane trace. *See supra at* 16, 29, 30[c].

EXHIBIT C-10
Defendant's Preliminary Invalidity Contentions
Orckit Corporation v. Cisco Systems, Inc., 2:22-cv-00276-JRG-RSP

Chart for U.S. Patent 7,545,740 ("the '740 Patent")
U.S. Patent No. 6,081,530 to Wiher et al. ("Wiher '530")

As shown in the chart below, all Asserted Claims of the '740 Patent are invalid under (1) 35 U.S.C. §§ 102 (a), (b), (e), and (g) because Wiher '530 meets each element of those claims, and/or (2) 35 U.S.C. § 103 because Wiher '530 renders those claims obvious either alone, or in combination with the knowledge of a person having ordinary skill in the art, and in further combination with the references specifically identified below and in the following claim chart and/or one or more references identified in Defendant's Preliminary Invalidity Contentions. The following quotations and diagrams come from Wiher '530 titled "Transmission of ATM Cells", which was filed on November 24, 1997, and issued on June 27, 2000.

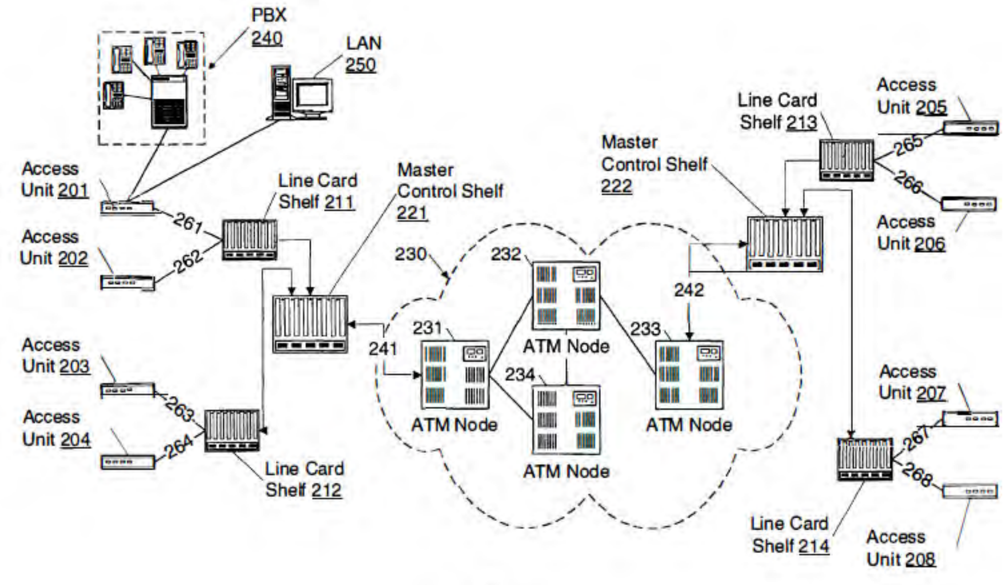
Motivations to combine the disclosures in Wiher '530 with disclosures in other publications known in the art, as explained in this chart, include at least the similarity in subject matter between the references to the extent they concern methods of data communication systems, and specifically to methods and systems for link aggregation in a data communication network. Insofar as the references cite other patents or publications, or suggest additional changes, one of ordinary skill in the art would look beyond a single reference to other references in the field.

These invalidity contentions are based on Defendant's present understanding of the asserted claims, and Orckit's apparent construction of the claims in its November 3, 2022 Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1, and Orckit's January 19, 2023 First Amended Disclosure of Asserted Claims and Infringement Contentions Pursuant to P.R. 3-1 (Orckit's "Infringement Disclosures"), which is deficient at least insofar as it fails to cite any documents or identify accused structures, acts, or materials in the Accused Products with particularity. Defendant does not agree with Orckit's application of the claims, or that the claims satisfy the requirements of 35 U.S.C. § 112. Defendant's contentions herein are not, and should in no way be seen as, admissions or adoptions as to any particular claim scope or construction, or as any admission that any particular element is met by any accused product in any particular way. Defendant objects to any attempt to imply claim construction from this chart. Defendant's prior art invalidity contentions are made in a variety of alternatives and do not represent Defendant's agreement or view as to the meaning, definiteness, written description support for, or enablement of any claim contained therein.

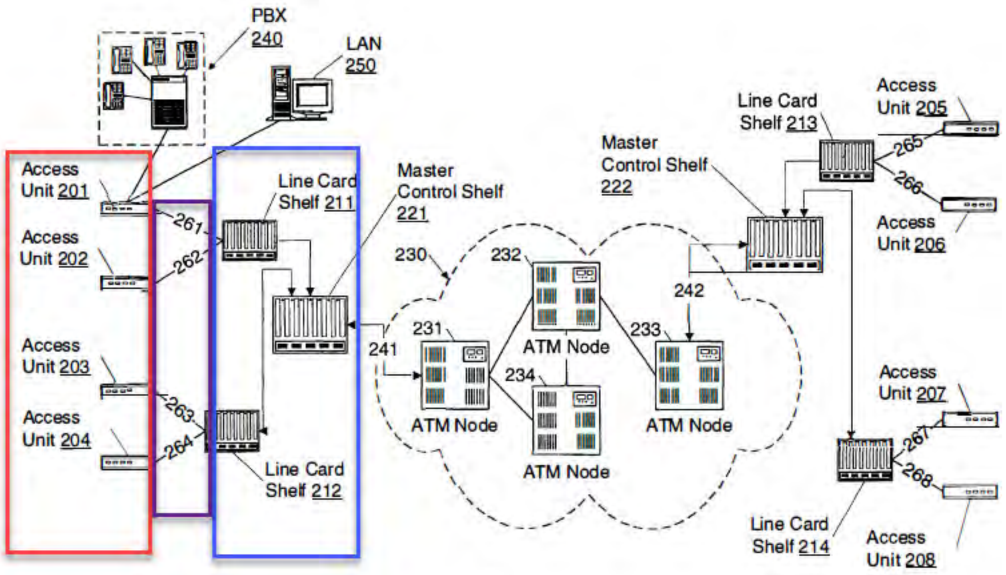
The following contentions are subject to revision and amendment pursuant to Federal Rule of Civil Procedure 26(e), the Local Rules, and the Orders of record in this matter subject to further investigation and discovery regarding the prior art and the Court's construction of the claims at issue.

No.	'740 Patent Claim 1	Wiher '530
1[preamble]	A method for communication, comprising:	Wiher '530 discloses a method for communication. For example, Wiher '530 discloses a method of transmitting data cells via circuitry. Wiher '530 at Abstract ("A system and method of transmitting data cells are dis-closed. The system includes a data transmitting and receiving unit including transceiver circuitry, a main backplane interface, and backplane interconnection circuitry. The transceiver circuitry transmits and receive data cells over the data link, the main backplane interface provides physical interconnection to the backplane, and the backplane inter-connection circuitry transmits and receives cells. The main backplane interface including at least one cell signal termi-nal and at least one operations data signal terminal. The operations data signal terminals are separate from the cell signal terminals. The operations data signal terminals and the cell signal terminals are configured to connect to mating connectors on a backplane. Backplane interconnection cir-cuitry couples the transceiver circuitry to the main back-plane interface. The interconnection circuitry receives data cells from the transceiver circuitry and transmit them over cell signal terminals, receives data cells from the cell signal terminals and provide them to the transceiver circuitry for transmission over the first data link, and transmit and receive operations data over the operations data signal terminals. The method includes asserting a signal indicating the pri-ority of a cell to be transmitted over the backplane, receiving on a second backplane signal line a signal that the apparatus may begin transmitting the data cell, and transmitting bits of the data cell on a third backplane signal line after receiving the signal that the apparatus may begin transmitting the data cell.") Wiher '530 at 1:4-9 ("The present invention relates to the transmission of asyn-chronous transfer mode (ATM) cells. Asynchronous transfer mode (ATM) data transfer is a communication technology in which fixed-size packets of data, known as "cells," are transferred between ATM switch-ing devices ("switches").")

No.	’740 Patent Claim 1	Wiher ’530
		Wiher ’530 at 3:25-34 (“In general, in another aspect, the invention features a method of sending a data cell over backplane signal lines. The method includes asserting on a first backplane signal line a signal indicating the priority of a cell to be transmitted over the backplane. Receiving on a second backplane signal line a signal that the apparatus may begin transmitting the data cell, and transmitting bits of the data cell on a third backplane signal line after receiving the signal that the apparatus may begin transmitting the data cell.”) Wiher ’530 at 3:35-42 (“In general, in another aspect, the invention features a method of receiving a data cell. The method includes asserting on a first backplane signal line a signal identifying an addressable apparatus port that is ready to receive a data cell. Receiving on a second signal line a signal indicating that a data cell is being transferred to the apparatus, and receiving on a third signal line bits of the data cell being transferred to the apparatus.”) Wiher ’530 at 4:45-57 (“In general, in another aspect, the invention features a method of transmitting a data cell over a backplane. The method includes receiving a data cell over a data link interface. Examining header information in the received data cell. Selecting one of a plurality of backplane cell interfaces. Receiving on a first signal line of the selected interface a signal indicating that a data cell may be transmitted on the interface. Transmitting on a second signal line of the selected interface a signal indicating that transfer of the data cell is occurring, and transmitting bits of the data cell on a third signal line of the selected interface. Implementations of the invention may include selecting based on a port address in a data cell header.”) Wiher ’530 at Figure 2

No.	’740 Patent Claim 1	Wiher ’530
		 PBX 240 LAN 250 Access Unit 201 Access Unit 202 Access Unit 203 Access Unit 204 261 262 263 264 Line Card Shelf 211 Line Card Shelf 212 Master Control Shelf 221 241 230 231 232 233 234 ATM Node ATM Node ATM Node ATM Node 242 Master Control Shelf 222 Line Card Shelf 213 Access Unit 205 265 266 Access Unit 206 Access Unit 207 267 268 Line Card Shelf 214 Access Unit 208 Fig. 2
1[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel,	Wiher ’530 discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. For example, Wiher ’530 discloses an ATM network in which network access equipment units are connected to line cards in line card shelves via parallel line card ports or loops. Wiher ’530 at 6:26-42 (“Network access equipment 201-208 may combine data from multiple sources. For example, data from a LAN 250 and circuit oriented traffic, such as a Tl connection from a private branch exchange phone system (PBX) 240, may each be converted to ATM cells at network access equipment 201. ATM cells corresponding to LAN 250 and PBX 240 data are multiplexed together and sent by the network access equipment 201 over media 261 to a line card in a line card shelf 211. VPI and VCI information in transmitted ATM cells is used to uniquely identify data sources and destina-tions at, for example, network access

No.	’740 Patent Claim 1	Wiher ’530
		equipment 201, line card shelf 211, master control shelf 221, and within the ATM network 230. For example, by assigning a unique VPI/VCI value to ATM cells transporting LAN 250 data and different VPI/VCI value to cells transporting PBX 240 data, inde-pendent routing and logical separation of the PBX 240 and LAN 250 data can be maintained. Wiher ’530 at 6:43-64 (“ATM cells originating at network access equipment 201-208 are sent over transmission loops 261-268 between the network access equipment and line cards in line card shelves 211-214. A loop 261-268 may be, for example, a digital subscriber line operated over a twisted wire pair connection. A loop 261-268 terminates at a line card in a line card shelf 211-214. A line card shelf 211-214 may house multiple line cards. Each line card terminates one or more loop connections to network access equipment 201-208. A master control shelf 221, 222 is connected to one or more line card shelves. For example, line card shelves 211 and 212 are connected to master control shelf 221 and line card shelves 213 and 214 are connected to master control shelf 222. A master control shelf 221, 222 is a card shelf that controls and regulates the flow of data between line card shelves and a trunk interface 241, 242. A trunk interface 241, 242 provides a trunk connection 241, 242 between the master control shelf 221, 222 and the ATM network 230. Trunk interface 241, 242 is, for example, a 45 Mbit/second T-3 interface or a standard 155 Mbit/second fiber optic synchronous optical network optical carrier level 3 concatenated data (SONET OC-3c) interface.”) Wiher ’530 at 6:65-7:14 (“In an ATM network, a particular VPI/VCI value in a cell header is used to route a cell between the switching ports of two connected nodes, but the particular VPI/VCI value does not provide for routing through multiple nodes. To route a cell from one endpoint to another endpoint through multiple nodes in an ATM network, VPI/VCI information must be translated at each node. Thus, to route an ATM cell, a node performs the following steps: 1) an incoming cell’s VPI/VCI information is read, 2) a node output port providing cell transport to a destination node is determined based on the VPI/VCI information in the incoming cell’s header; 3) the node replaces the cell’s VPINCI information with new VPI/VCI information for routing through the destination node, and 4) the node forwards the cell through the deter-mined output port to the destination node. The destination node repeats this process until the cell reaches its final destination.”)

No.	’740 Patent Claim 1	Wiher ’530
		Wiher ’530 at 7:47-58 (“ATM cells are sent between an ATM network access unit 201-208 and a line card in a line card shelf 211-214 over a wire loop 261-268. FIG. 3 illustrates a line card shelf 300 having, for example, twelve line cards 301-312. Each line card 301-312 terminates, for example, two subscriber loop connections to network access units 201-208. The line cards support, for example, high bit rate digital subscriber line (HDSL), asymmetric digital subscriber line (ADSL), or a rate adaptive digital subscriber line (RADSL) data transmission over the subscriber loops. A line card shelf 300 also includes a main line card shelf multiplexer (LSM) 330 and a backup LSM 340.”) Wiher ’530 at Figure 2 (annotated added)  PBX 240 LAN 250 Access Unit 201 Access Unit 202 Access Unit 203 Access Unit 204 261 262 263 264 Line Card Shelf 211 Line Card Shelf 212 Master Control Shelf 221 230 231 232 233 234 241 242 ATM Node ATM Node ATM Node ATM Node Master Control Shelf 222 Line Card Shelf 213 Line Card Shelf 214 Access Unit 205 Access Unit 206 Access Unit 207 Access Unit 208 265 266 267 268 Fig. 2

No.	’740 Patent Claim 1	Wiher ’530
		Wiher ’530 at 11:40-57 (“ATM cells are transferred from a LSM to a line card by the exchange of the LC-RR, LSM-SR, and LSM-DATA signals on the LSM to line card cell transport signal lines 611. FIGS. 7A and 7B illustrate timing and modulation of the LC-RR, LSM-SR, and LSM-DATA signals. The LC-RR signal is sent from the line card to the LSM to indicate line card ports that are ready to receive ATM cell transfers. FIG. 7A is a LC-RR signal timing diagram for a line card supporting two line card ports. The LC-RR signal is modu-lated to periodically transmit a framing indicator and port status information from the line card to the LSM. The framing indicator is sent by asserting the LC-RR signal for one clock cycle at, for example, sixteen clock cycle inter-vals. During each clock cycle following the framing indicator, port status information may be sent from the line card to the LSM. Port status information is sent by asserting or de-asserting the LC-RR signal during a clock period that is unique for each port on the line card.”) Wiher ’530 at 12:4-12 (“A line card supporting more than two subscriber loops will typically have additional ports. For example, a line card supporting four subscriber loops may receive data from the LSM at four line card ports. A line card with more than two ports will convey additional port status information follow-ing the illustrated ‘P2’ indicator period in FIG. 7A. In an alternative line card implementation, VPI/VCI information rather than a line card port identifier, may be used to identify the particular destination subscriber loop.”)
1[b]	at least one of said first physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction	Wiher ’530 discloses at least one of said first physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction. For example, Wiher ’530 discloses line card ports or loops in which data cells may be transmitted or received in both directions. A person of ordinary skill in the art would understand that at least one of Wiher ’530’s line card ports can transmit data in both an upstream and a downstream direction. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. Wiher ’530 at 6:65-7:14 (“In an ATM network, a particular VPI/VCI value in a cell header is used to route a cell between the switching ports of two connected nodes, but the particular

No.	’740 Patent Claim 1	Wiher ’530
		VPINCI value does not provide for routing through multiple nodes. To route a cell from one endpoint to another endpoint through multiple nodes in an ATM network, VPI/VCI information must be translated at each node. Thus, to route an ATM cell, a node performs the following steps: 1) an incoming cell's VPINCI information is read, 2) a node output port providing cell transport to a destination node is determined based on the VPI/VCI information in the incoming cell's header; 3) the node replaces the cell's VPINCI information with new VPI/VCI information for routing through the destination node, and 4) the node forwards the cell through the deter-mined output port to the destination node. The destination node repeats this process until the cell reaches its final destination.”) Wiher ’530 at 10:5-24 (“In various implementations, transceivers 643 and 644 implement, for example, Digital Subscriber Line (DSL), Integrated Services Digital Network (ISDN), Rate Adaptive Digital Subscriber Line (RADSL), High Bit Rate Digital Subscriber Line (HDSL), Asymmetric Digital Subscriber Line (ADSL) modulation, or other digital modulation tech-nique. Line card circuitry 643-647 may be implemented in one or more integrated circuit chips, may include discrete circuit components, and may include additional functional-ity. In the line card 600, each transceiver 643 and 644 has a two-wire coupling to loop interface 632 over which both transmit and receive signals may be sent. In alternate implementations, transceivers 643 and 644 may have addi-tional signal couplings to provide, for example, a transmis-sion and reception of a four-wire service, or may provide a four-wire transceiver coupling for use with external hybrid circuitry that adapts the four-wire coupling to a two-wire loop interface. Still other transceiver-to-loop signal coupling arrangements may be used depending on the particular transceiver type and line card application.”) Wiher ’530 at 12:4-12 (“A line card supporting more than two subscriber loops will typically have additional ports. For example, a line card supporting four subscriber loops may receive data from the LSM at four line card ports. A line card with more than two ports will convey additional port status information follow-ing the illustrated 'P2' indicator period in FIG. 7A. In an alternative line card implementation, VPI/VCI information, rather than a line card port identifier, may be used to identify the particular destination subscriber loop.”)

No.	’740 Patent Claim 1	Wiher ’530
		Wiher ’530 at 12:13-31 (“When a line card port is ready to receive an ATM cell, a cell may be transferred using the LSM-SR and LSM-DATA signals. FIG. 7B is an exemplary signal timing diagram illustrating states of the LSM-SR and LSM-DATA signals during an ATM cell transfer from the LSM to the line card. When a LSM is ready to send a cell to a waiting line card port, the LSM asserts (low) the LSM-SR signal and simultaneously begins modulating data over the interface 611 using the LSM-DATA signal. For example, at clock 4, the LSM asserts the LSM-SR signal and begins a serial transfer of a data cells by modulating the LSM-DATA signal. To send a '1' bit value, the LSM-DATA signal is asserted (high) during a clock cycle, and to send a '0' bit value the LSM-DATA signal is de-asserted (low) during a clock cycle. To send a 53-byte (424 bit) ATM cell, the LSM-DATA signal is modulated for a 424 clock cycle period. After transmission of the ATM cell, the LSM-SR signal is de-asserted. When the LSM-SR signal is de-asserted, the LSM-DATA signal is not sampled.”)
1[c]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel,	Wiher ’530 discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. For example, Wiher ’530 discloses line cards in a line card shelf connected to the ATM network using backplane interfaces, such as physical backplane interconnections. Wiher ’530 at 1:36-57 (“In general, in one aspect, the invention features an appa-ratus for communicating data cells between a data link and a backplane. The apparatus includes transceiver circuitry, a main backplane interface, and backplane interconnection circuitry. The transceiver circuitry transmits and receive data cells over the data link, the main backplane interface provides physical interconnection to the backplane, and the backplane interconnection circuitry transmits and receives cells. The main backplane interface including at least one cell signal terminal and at least one operations data signal terminal. The operations data signal terminals are separate from the cell signal terminals. The operations data signal terminals and the cell signal terminals are configured to connect to mating connectors on a backplane. Backplane interconnection circuitry couples the transceiver circuitry to the main backplane interface. The interconnection circuitry receives data cells from the transceiver

<table>
<tr><th>No.</th><th>’740 Patent Claim 1</th><th>Wiher ’530</th></tr>
<tr><td></td><td></td><td>

circuitry and trans-mit them over cell signal terminals, receives data cells from the cell signal terminals and provide them to the transceiver circuitry for transmission over the first data link, and trans-mit and receive operations data over the operations data signal terminals.”)

Wiher ’530 at 2:10-18 (“Implementations of the invention may also include one or more of the following features. The apparatus may be implemented on a card configured to be plugged into a backplane having backplane mating connectors correspond-ing to the separate terminal connectors. A fiber-optic data link interface may couple the transceiver circuitry to a synchronous optical network (SONET) data link. A high bit-rate digital subscriber line (HDSL) data link interface may be coupled to the transceiver circuitry.”)

Wiher ’530 at Figure 4 (annotation added)

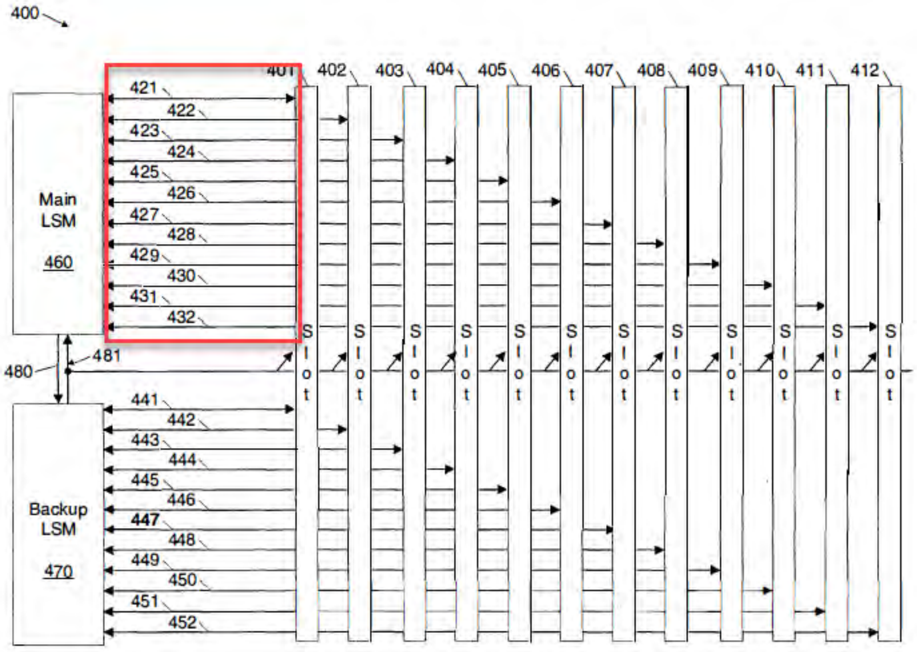


Fig. 4

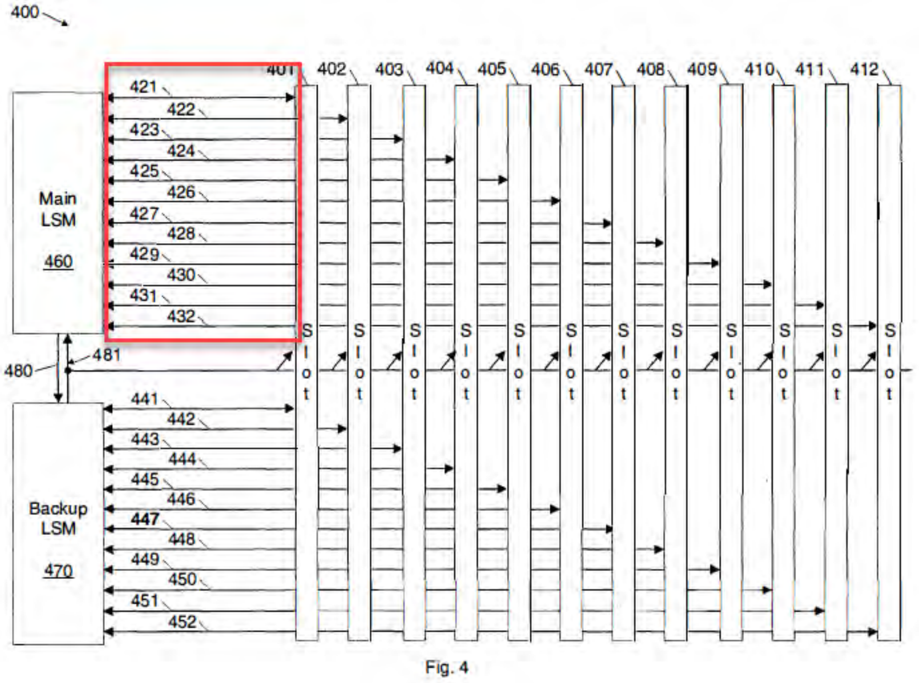
</td></tr>
</table>

No.	’740 Patent Claim 1	Wiher ’530
		Wiher ’530 at 7:59-8:22 (“A line card shelf 300 includes a line card shelf backplane. FIG. 4 illustrates a line card shelf backplane 400. The line card shelf backplane 400 has twelve line card slots 401-412 and two line card shelf multiplexer (LSM) slots 460 and 470. A line card slot 401-412 is the point at which a line card is coupled to conductive signal paths on the backplane 400. LSM slots 460 and 470 are the points at which LSM's are coupled to line card shelf backplane signal paths. Main LSM signal paths 421-432 and backup LSM signal paths 441-452 couple line card slots 401-412 to the main LSM slot 460 and the backup LSM slot 470. Each line card slot 401-412 is connected over a dedicated signal path to main LSM slot 460 and over a second dedicated path to backup LSM slot 470. For example, line card slot 401 is connected to main LSM slot 460 over signal path 421 and to backup LSM slot 470 over signal path 441, while line card slot 402 is connected to main LSM slot 460 over signal path 422 and to backup LSM slot 470 over signal path 442. Line card slots 403-412 are similarly connected to the main LSM slot 460 over signal paths 423-432 and to backup LSM slot 470 over signal paths 443-452, respectively. A line card shelf backplane 400 also has LSM status signal paths 480 and 481. Status signal path 480 provides a single conductive path over which main LSM 460 conveys its status to the backup LSM 470. Status information sent from the main LSM to the backup LSM indicates whether the main LSM is in an active state or whether the main LSM is in a standby or failure state. Similarly, status signal path 481 provides a single shared conductive path over which backup LSM 470 conveys its status (active or standby) to the main LSM 460 and to line cards in each of the line card slots 401-412.”) Wiher ’530 at 8:23-33 (“Line cards and LSMs include interfaces to couple signals to and from the backplane 400. FIG. 5 is a functional diagram of a LSM. The LSM 500 has line card interfaces 501-512 for communicating signals with line cards, a status input interface 531 for receiving a status signal from another LSM, a status output interface 532 for sending a status signal indicative of the LSM's current operating condition, a timing output interface 533 to output reference clock signals and an interface 530 to a master line adapter (MLA). The LSM includes, for example, circuitry 520-528 to process signals on interfaces 501-512 and 530-533.”)

No.	’740 Patent Claim 1	Wiher ’530
		Wiher ’530 at 10:38-62 (“The described coupling between line cards, LSMs, and backplane signal paths provides each line card with dedicated connections to two LSMs. For example, when a line card 600 is inserted in line card slot 401, the line card's main LSM interface 610 is coupled to main LSM line card interface 501 by backplane signal paths 421, the line card's backup LSM interface 620 is coupled to backup LSM line card interface 501 by backplane signal paths 441, the line card's LSM status input interface 631 is coupled to the backup LSM's status output interface 532 by backplane signal paths 481 and the line card's subscriber loop interface 632 is coupled to subscriber loop signal paths. Similarly, when a line card 600 is inserted in line card slot 402, the line card's main LSM interface 610 is coupled to main LSM line card interface 502 by backplane signal paths 422, the line card's backup LSM interface 620 is coupled to backup LSM line card interface 502 by backplane signal paths 442, the line card's LSM status input interface 631 is coupled to the backup LSM's status interface 532 by backplane signal paths 481 and the line card's subscriber loop interface 632 is coupled to subscriber loop signal paths. Line cards inserted in slots 403-412 are similarly connected to main LSM interfaces 503-512, backup LSM interfaces 503-512, backup LSM status output interface 532, and subscriber loop connections.”) Wiher ’530 at 10:63-11:9 (“A line card's main LSM interface 610 includes LSM to line card cell transfer signal lines 611 and line card to LSM cell transfer signal lines 612. Signal lines 611 and 612 are coupled to complementary signal lines at line card interfaces 501-512 (FIG. 5) of a LSM by backplane signal paths 421-432 (FIG. 4). Signals modulated over signal lines 611 are used to receive ATM cells sent from the main LSM. Signals modulated over signal lines 612 are used to send ATM cells to the main LSM. Signals exchanged over signal lines 611 and 612 are modulated in reference to a clock signal that is, for example, a 12.5 megahertz (MHz) clock signal received from a LSM, and signals may be asserted or de-asserted on the rising edge of a clock pulse and sampled on the falling edge of the clock pulse.”) Wiher ’530 at 18:33-60 (“A trunk card includes, for example, circuitry 1122-1128 to process signals on interfaces 1101-1106 and on interfaces 1129-1135. MLA interface control circuitry 1122 processes ATM cell transfer signals exchanged over interfaces 1101-1106 and controls the transmission of ATM cells between the trunk card and MLAs. Interface control cir-cuitry 1122 may include ATM cell buffers to temporarily store cells received from, or being

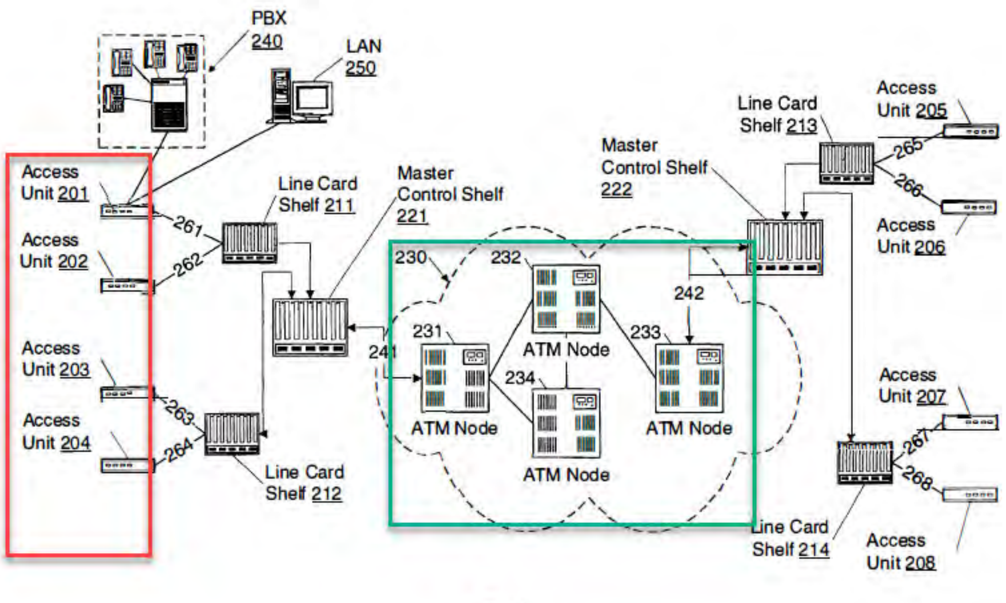
No.	’740 Patent Claim 1	Wiher ’530
		transmitted to, MLAs over interfaces 1101-1106. Cell multiplexer/de-multiplexer circuitry 1123 exchanges ATM cells with the MLA interface control circuitry 1122 and determines the flow of ATM cells between the control circuitry 1122 and header translation circuitry 1124. Additionally, the multiplexer/de-multiplexer circuitry 1123 may extract OAMP cells arriving from the trunk interface 1125 and direct them to processor 1128 and insert OAMP cells destined to the trunk interface 1125 from the processor 1128. Header translation circuitry 1124 translates header information in ATM cells arriving from or destined to the trunk interface 1125. Header translation circuitry 1124 may access header translation programs and data stored in RAM and ROM memory 1127. For example, processor 1128 may store VPI/VCI header field translation information in memory 1127 for use by header translation circuitry 1124. Header translation circuitry 1124 exchanges ATM cells with trunk interface circuitry 1125. Trunk inter-face circuitry 1125 provides a trunk connection to an ATM network. The trunk connection is, for example, a standard 45 Megabit per second T3 trunk connection. Wiher ’530 at 30:56-31:6 (“In various implementations, either ITU-standard 53-byte ATM cells, non-standard ATM cells, or both ITU-standard and non-standard ATM cells may be employed. For example, a non-standard 54-byte cell may be formed by adding an additional parity byte to a standard 53-byte cell. This additional parity byte may provide for a parity check of the preceding 53 bytes ITU-standard ATM cell. Such a 54 byte cell may be sent, for example, between the trunk card and MLA, between the MLA and LSM, and between the LSM and each line card while an ITU-standard 53-byte ATM cell may be sent between the trunk card and an ATM network and between the line card and network access equipment at a customer's premises. Additionally, ATM cells may include non-standard header fields. For example, the fifth byte of the ATM cell, used for header error control information in a ITU-standard ATM cell, can be used for line card port identifying information in cells transmitted between the LSM and the line card.”)
1[d]	at least one of said second physical links being a bi-directional link operative to communicate in both	Wiher ’530 discloses at least one of said second physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction. For example, Wiher ’530 discloses backplane interfaces that may receive or transmit data cells in both directions. A person of ordinary skill in the art would understand that at least one of

No.	’740 Patent Claim 1	Wiher ’530
	an upstream direction and a downstream direction;	Wiher ’530’s backplane interfaces can transmit data in both an upstream and a downstream direction. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. Wiher ’530 at 3:65-4:22 (“Implementations of the invention may include one or more of the following features. The backplane interconnection circuits may independently receive and transmit data cells over the plurality of backplane interfaces. The de-multiplexing circuitry may select a backplane interface based on data in the header field of the data cell. The apparatus may include header translation circuitry to alter header data in cells sent between the plurality of backplane interfaces and the transceiver circuitry. Each of the plurality of backplane interfaces may include separate terminals to receive cells and separate terminals to transmit cells. The terminals to transmit cells may include a first and second control terminal and at least one outgoing cell data terminal. A backplane interface's backplane interconnection circuitry may accepts a signal on the first control terminal as indicating that a cell may be sent over the interface, asserts a signal on the second control terminal to indicate that a cell is being transmitted, and transmits data bits of the cell on the outgoing cell data terminal. Each backplane interface may include a single outgoing cell data terminal and each bit of the cell may be serially transmitted over the single outgoing cell data terminal. Each backplane interface may include multiple outgoing cell data terminals and bits of the cell may be sent in parallel over the eight outgoing cell data terminals.”) Wiher ’530 at 4:45-57 (“In general, in another aspect, the invention features a method of transmitting a data cell over a backplane. The method includes receiving a data cell over a data link interface. Examining header information in the received data cell. Selecting one of a plurality of backplane cell interfaces. Receiving on a first signal line of the selected interface a signal indicating that a data cell may be transmitted on the interface. Transmitting on a second signal line of the selected interface a signal indicating that transfer of the data cell is occurring, and transmitting bits of the data cell on a third signal line of the selected interface. Implementations of the invention may include selecting based on a port address in a data cell header.”) Wiher ’530 at Figure 4 (annotation added)

No.	’740 Patent Claim 1	Wiher ’530
		 400 401 402 403 404 405 406 407 408 409 410 411 412 421 422 423 424 425 426 427 428 429 430 431 432 Main LSM 460 480 481 441 442 443 444 445 446 447 448 449 450 451 452 Backup LSM 470 Slot Fig. 4
1[e]	receiving a data frame having frame attributes sent between the communication network and the network node:	Wiher ’530 discloses receiving a data frame having frame attributes sent between the communication network and the network node. For example, Wiher ’530 discloses receiving ATM cells with specific identifying characteristics between the ATM network and network access equipment. Wiher ’530 at 1:58-2:9 (“Implementations of the invention may include one or more of the following features. The apparatus may include data cell header translation circuitry that alters header field data in cells exchanged between the backplane interconnection circuitry and the transceiver circuitry. Altering of data cell headers by the data cell header translation circuitry

No.	’740 Patent Claim 1	Wiher ’530
		may be determined by the operations data received over the operations data signal terminals. The invention may include an operations processor that transmits and receives operations, administration, maintenance, and provisioning (OAMP) data over the operations data signal terminals using the high level data link control (HDLC) protocol. Implementations may include a processor coupled to the back-plane interconnection circuitry and to the transceiver circuitry and configured to receive variable-length data transmissions over the operations data signal terminals, convert the variable-length data to fixed-length cells, and provides the fixed-length cells to the transceiver circuitry for transmission over the data link.”) Wiher ’530 at 5:25-43 (“FIG. lA illustrates an ATM cell having a 53-byte format as defined by the ITU. The ATM cell 100 includes a header field 101 and a payload field 102. An ITU-standard header field 101 may be either a user-network interface header or a network-network interface header. FIGS. lB and lC illustrate, respectively, a user-network interface header 125 and a network-network interface header 150. In general, ATM cells having a user-network interface header 125 are sent between ATM network access equipment that is located at an endpoint of an ATM connection and ATM network switching equipment ("nodes"). Cells having a network-network interface header 150 are sent between nodes in the ATM network, i.e., from non-endpoint to non-endpoint ATM cell switching equipment. User-network interface headers 125 and network-network interface headers 150 include multiple information sub-fields and differ in the information contained in the first byte ("Byte l") of cell header 125 and cell header 150.”) Wiher ’530 at 5:44-65 (“A user-network header 125 (FIG. 1B) includes a four-bit generic flow control (GFC) field, an eight-bit virtual path identifier (VPI) field, a sixteen-bit virtual channel identifier (VCI) field, a three-bit payload type identifier (PTI) field, a one-bit cell loss priority (CLP) field and an eight-bit header error control (HEC) field. The GFC field carries information to assist in controlling the flow of ATM cells over the user-network interface. The VPI field identifies a virtual path and the VCI field identifies the virtual channel for routing the ATM cell through a network. The PTI field identifies whether the cell contains user or network management related information. The CLP field indicates the cell loss priority. If the value of the CLP field is 1, the cell may be subject to discard, depending on network conditions

No.	’740 Patent Claim 1	Wiher ’530
		such as a network overload. If the value of the CLP field is 0, the cell has high priority and therefore ATM nodes should allocate sufficient network resources to prevent cell discard and ensure transport of the cell. The HEC field contains header error control information to detect transmission errors in header 101 information. Additional information on these standard header fields can be found in ATM User-Network Interface Specification Version 3.1, ATM Forum, 1994.”) Wiher ’530 at 6:65-7:14 (“In an ATM network, a particular VPI/VCI value in a cell header is used to route a cell between the switching ports of two connected nodes, but the particular VPI/VCI value does not provide for routing through multiple nodes. To route a cell from one endpoint to another endpoint through multiple nodes in an ATM network, VPI/VCI information must be translated at each node. Thus, to route an ATM cell, a node performs the following steps: 1) an incoming cell's VPI/VCI information is read, 2) a node output port providing cell transport to a destination node is determined based on the VPI/VCI information in the incoming cell's header; 3) the node replaces the cell's VPINCI information with new VPI/VCI information for routing through the destination node, and 4) the node forwards the cell through the determined output port to the destination node. The destination node repeats this process until the cell reaches its final destination.”) Wiher ’530 at Figure 2 (annotation added)

No.	’740 Patent Claim 1	Wiher ’530
		 PBX 240 LAN 250 Access Unit 201 Access Unit 202 Access Unit 203 Access Unit 204 261 262 263 264 Line Card Shelf 211 Line Card Shelf 212 Master Control Shelf 221 241 230 231 232 233 234 242 ATM Node ATM Node ATM Node ATM Node Master Control Shelf 222 Line Card Shelf 213 Access Unit 205 265 266 Access Unit 206 Access Unit 207 267 268 Access Unit 208 Line Card Shelf 214 Fig. 2
1[f]:	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	Wiher ’530 discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. For example, Wiher ’530 discloses using cell header information to at each node to select and route the ATM cell to a selected backplane interface and port. A person of ordinary skill in the art would understand that the routing of ATM cells over a port/loop and backplane interconnection involves a single computation based on frame attributes. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met.

No.	’740 Patent Claim 1	Wiher ’530
		Wiher ’530 at 3:43-65 (“In general, in another aspect, the invention features an apparatus for communicating data cells between a data link and a backplane. The apparatus includes transceiver circuitry to transmit and receive data cells over a data link and a plurality of backplane interfaces each including at least one cell signal terminal. Each of the backplane interface is coupled to a backplane interconnection circuit. Each backplane interconnection circuit transmits and receives cells over the cell signal terminals of its associated backplane interface. The apparatus also includes de-multiplexing circuitry coupling the transceiver circuitry to each of the backplane interconnection circuits. The de-multiplexing circuitry receives a data cell from the transceiver circuitry, select a backplane interconnection circuit associated with the data cell, and provide the data cell to the selected backplane interconnection circuit for transmission over the cell signal terminals of the associated backplane interface. The apparatus also includes multiplexing circuitry coupling the plurality of backplane interconnection circuits to the transceiver circuitry. The multiplexing circuitry receives data cells from each of the backplane interconnection circuits and provide the received data cells to the transceiver circuitry.”) Wiher ’530 at 3:66-4:22 (“Implementations of the invention may include one or more of the following features. The backplane interconnection circuits may independently receive and transmit data cells over the plurality of backplane interfaces. The de-multiplexing circuitry may select a backplane interface based on data in the header field of the data cell. The apparatus may include header translation circuitry to alter header data in cells sent between the plurality of backplane interfaces and the transceiver circuitry. Each of the plurality of backplane interfaces may include separate terminals to receive cells and separate terminals to transmit cells. The terminals to transmit cells may include a first and second control terminal and at least one outgoing cell data terminal. A backplane interface's backplane interconnection circuitry may accepts a signal on the first control terminal as indicating that a cell may be sent over the interface, asserts a 15 signal on the second control terminal to indicate that a cell is being transmitted, and transmits data bits of the cell on the outgoing cell data terminal. Each backplane interface may include a single outgoing cell data terminal and each bit of the cell may be serially transmitted over the single outgoing cell data terminal. Each backplane interface may include multiple outgoing cell data terminals and bits of the cell may be sent in parallel over the eight outgoing cell data terminals.”)

No.	’740 Patent Claim 1	Wiher ’530
		Wiher ’530 at 4:45-58 (“In general, in another aspect, the invention features a method of transmitting a data cell over a backplane. The method includes receiving a data cell over a data link interface. Examining header information in the received data cell. Selecting one of a plurality of backplane cell interfaces. Receiving on a first signal line of the selected interface a signal indicating that a data cell may be transmitted on the interface. Transmitting on a second signal line of the selected interface a signal indicating that transfer of the data cell is occurring, and transmitting bits of the data cell on a third signal line of the selected interface. Implementations of the invention may include selecting based on a port address in a data cell header.”) Wiher ’530 at 6:65-7:14 (“In an ATM network, a particular VPI/VCI value in a cell header is used to route a cell between the switching ports of two connected nodes, but the particular VPI/VCI value does not provide for routing through multiple nodes. To route a cell from one endpoint to another endpoint through multiple nodes in an ATM network, VPI/VCI information must be translated at each node. Thus, to route an ATM cell, a node performs the following steps: 1) an incoming cell's VPI/VCI information is read, 2) a node output port providing cell transport to a destination node is determined based on the VPI/VCI information in the incoming cell's header; 3) the node replaces the cell's VPI/VCI information with new VPI/VCI information for routing through the destination node, and 4) the node forwards the cell through the determined output port to the destination node. The destination node repeats this process until the cell reaches its final destination.”)
1[g]	sending the data frame over the selected first and second physical links,	Wiher ’530 discloses sending the data frame over the selected first and second physical links. For example, Wiher ’530 discloses transmitting the ATM cell over the selected backplane interface and port/loop. Wiher ’530 at 3:43-65 (“In general, in another aspect, the invention features an apparatus for communicating data cells between a data link and a backplane. The apparatus includes transceiver circuitry to transmit and receive data cells over a data link and a plurality of

No.	’740 Patent Claim 1	Wiher ’530
		backplane interfaces each including at least one cell signal terminal. Each of the backplane interface is coupled to a backplane interconnection circuit. Each backplane interconnection circuit transmits and receives cells over the cell signal terminals of its associated backplane interface. The apparatus also includes de-multiplexing cir-cuitry coupling the transceiver circuitry to each of the backplane interconnection circuits. The de-multiplexing circuitry receives a data cell from the transceiver circuitry, select a backplane interconnection circuit associated with the data cell, and provide the data cell to the selected backplane interconnection circuit for transmission over the cell signal terminals of the associated backplane interface. The apparatus also includes multiplexing circuitry coupling the plurality of backplane interconnection circuits to the transceiver circuitry. The multiplexing circuitry receives data cells from each of the backplane interconnection cir-cuits and provide the received data cells to the transceiver circuitry.”) Wiher ’530 at 3:66-4:22 (“Implementations of the invention may include one or more of the following features. The backplane interconnection circuits may independently receive and transmit data cells over the plurality of backplane interfaces. The de-multiplexing circuitry may select a backplane interface based on data in the header field of the data cell. The apparatus may include header translation circuitry to alter header data in cells sent between the plurality of backplane interfaces and the transceiver circuitry. Each of the plurality of backplane interfaces may include separate terminals to receive cells and separate terminals to transmit cells. The terminals to transmit cells may include a first and second control terminal and at least one outgoing cell data terminal. A backplane interface's backplane interconnection circuitry may accepts a signal on the first control terminal as indicating that a cell may be sent over the interface, asserts a 15 signal on the second control terminal to indicate that a cell is being transmitted, and transmits data bits of the cell on the outgoing cell data terminal. Each backplane interface may include a single outgoing cell data terminal and each bit of the cell may be serially transmitted over the single outgoing cell data terminal. Each backplane interface may include multiple outgoing cell data terminals and bits of the cell may be sent in parallel over the eight outgoing cell data terminals.”) Wiher ’530 at 4:45-58 (“In general, in another aspect, the invention features a method of transmitting a data cell over a backplane. The method includes receiving a data cell over a data

No.	’740 Patent Claim 1	Wiher ’530
		link interface. Examining header information in the received data cell. Selecting one of a plurality of backplane cell interfaces. Receiving on a first signal line of the selected interface a signal indicating that a data cell may be transmitted on the interface. Transmitting on a second signal line of the selected interface a signal indicating that transfer of the data cell is occurring, and transmitting bits of the data cell on a third signal line of the selected interface. Implementations of the invention may include selecting based on a port address in a data cell header.”) Wiher ’530 at 6:65-7:14 (“In an ATM network, a particular VPI/VCI value in a cell header is used to route a cell between the switching ports of two connected nodes, but the particular VPI/VCI value does not provide for routing through multiple nodes. To route a cell from one endpoint to another endpoint through multiple nodes in an ATM network, VPI/VCI information must be translated at each node. Thus, to route an ATM cell, a node performs the following steps: 1) an incoming cell's VPI/VCI information is read, 2) a node output port providing cell transport to a destination node is determined based on the VPI/VCI information in the incoming cell's header; 3) the node replaces the cell's VPINCI information with new VPI/VCI information for routing through the destination node, and 4) the node forwards the cell through the deter-mined output port to the destination node. The destination node repeats this process until the cell reaches its final destination.”)
1[h]	said sending comprising communicating along at least one of said bi-directional links.	Wiher ’530 discloses said sending comprising communicating along at least one of said bi-directional links. *See supra at* 1[b], 1[d], 1[g].

No.	’740 Patent Claim 2	Wiher ’530
2[a]	The method according to claim 1, wherein the network node	Wiher ’530 discloses the method according to claim 1, wherein the network node comprises a user node.

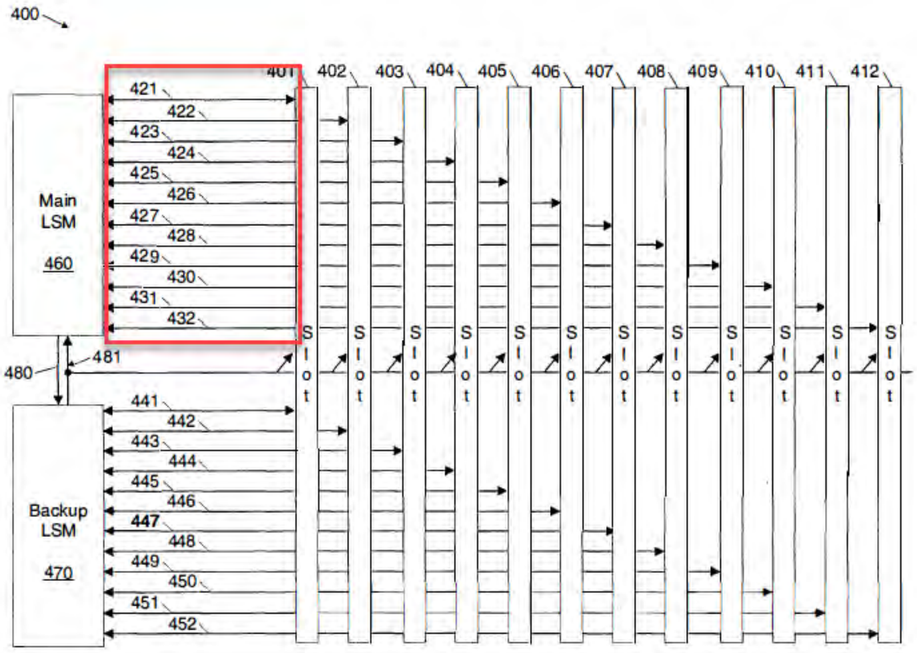
No.	’740 Patent Claim 2	Wiher ’530
	comprises a user node, and	For example, Wiher ’530 discloses network access equipment units that interface with the user in a user network. Wiher ’530 at Claim 1. Wiher ’530 at 6:8-25 (“FIG. 2 is an exemplary ATM network. ATM cells can be used to establish a communication path between network access equipment 201-208. Network access equipment 201-208 forms the originating and terminating point in an ATM communication and may convert non-ATM data traffic into an ATM format. Conversion of non-ATM data traffic to ATM cells is provided by ATM adaptation layer (AAL) services. Standard AAL services are defined in Bellcore publication GR-1113-CO RE, Asynchronous Transfer Mode and ATM Adaptation Layer (AAL) Protocols, 1994. AAL services may be used, for example, to convert a 1.544 megabit per second continuous bit rate (CBR) circuit- oriented Tl connection to an ATM virtual circuit connection or to convert variable-length packet data traffic originating on a local area network (LAN) to ATM cells for transport on an ATM network. ATM cells are sent from network access equipment to the ATM network using a user-network interface header 125 (FIG. 1B).”) Wiher ’530 at 7:15-45 (“For example, consider an ATM cell that is to be transmitted from network access equipment 201 to network access equipment 206. The cell may traverse a path between access equipment 201, line card shelf 211, master control shelf 221, node 231, node 232, node 233, master control shelf 222, line card shelf 213 and access equipment 206. Prior to transmission of ATM cells from access equipment 201 to 206, VPINCI translation information is established at each point in the path between 201 to 206. VPI/VCI trans-lation information may be established by exchanging special ATM cells providing information to control processors in the various network nodes. Next, at network access equipment 201, cells are formatted with user-network interface headers and are assigned a VPINCI value. The assigned VPI/VCI value allows routing between line card shelf 211 input and output ports, but not through, for example, the master control shelves 221 and 222, nodes 231-233, line card shelf 213 or access equipment 206. To accomplish cell transport between access equipment 201 and 206, VPI/VCI transla-tion information is established at each point along the path between access equipment 201 to 206. Thus, for example, when the cell is received at 211, the

No.	’740 Patent Claim 2	Wiher ’530
		VPINCI information in the user-network interface header is replaced with new routing information to allow routing through master control shelf 221. When the cell is received by the master control shelf 221, the routing information is replaced by new VPI/VCI information to allow routing through node 231. Similarly, header translation occurs at node 232, node 233, master control shelf 222 and at line card shelf 213. Once established, VPI/VCI translation information persist until the communication path between network access equipment endpoints 201 and 206 is no longer needed.”)
2[b]	wherein sending the data frame comprises establishing a communication service between the user node and the communication network.	Wiher ’530 discloses wherein sending the data frame comprises establishing a communication service between the user node and the communication network. For example, Wiher ’530 discloses transmitting ATM cells between network access equipment and the ATM network as part of ATM communication path. Wiher ’530 at 6:8-25 (“FIG. 2 is an exemplary ATM network. ATM cells can be used to establish a communication path between network access equipment 201-208. Network access equipment 201-208 forms the originating and terminating point in an ATM communication and may convert non-ATM data traffic into an ATM format. Conversion of non-ATM data traffic to ATM cells is provided by ATM adaptation layer (AAL) services. Standard AAL services are defined in Bellcore publication GR-1113-CO RE, Asynchronous Transfer Mode and ATM Adaptation Layer (AAL) Protocols, 1994. AAL services may be used, for example, to convert a 1.544 megabit per second continuous bit rate (CBR) circuit-oriented Tl connection to an ATM virtual circuit connection or to convert variable-length packet data traffic originating on a local area network (LAN) to ATM cells for transport on an ATM network. ATM cells are sent from network access equipment to the ATM network using a user-network interface header 125 (FIG. 1B).”) Wiher ’530 at 7:15-45 (“For example, consider an ATM cell that is to be transmitted from network access equipment 201 to network access equipment 206. The cell may traverse a path between access equipment 201, line card shelf 211, master control shelf 221, node 231, node 232, node 233, master control shelf 222, line card shelf 213 and access equipment 206. Prior

No.	’740 Patent Claim 2	Wiher ’530
		to transmission of ATM cells from access equipment 201 to 206, VPINCI translation information is established at each point in the path between 201 to 206. VPI/VCI trans-lation information may be established by exchanging special ATM cells providing information to control processors in the various network nodes. Next, at network access equipment 201, cells are formatted with user-network interface headers and are assigned a VPINCI value. The assigned VPI/VCI value allows routing between line card shelf 211 input and output ports, but not through, for example, the master control shelves 221 and 222, nodes 231-233, line card shelf 213 or access equipment 206. To accomplish cell transport between access equipment 201 and 206, VPI/VCI transla-tion information is established at each point along the path between access equipment 201 to 206. Thus, for example, when the cell is received at 211, the VPINCI information in the user-network interface header is replaced with new routing information to allow routing through master control shelf 221. When the cell is received by the master control shelf 221, the routing information is replaced by new VPI/VCI information to allow routing through node 231. Similarly, header translation occurs at node 232, node 233, master control shelf 222 and at line card shelf 213. Once established, VPINCI translation information persist until the communication path between network access equipment endpoints 201 and 206 is no longer needed.”)

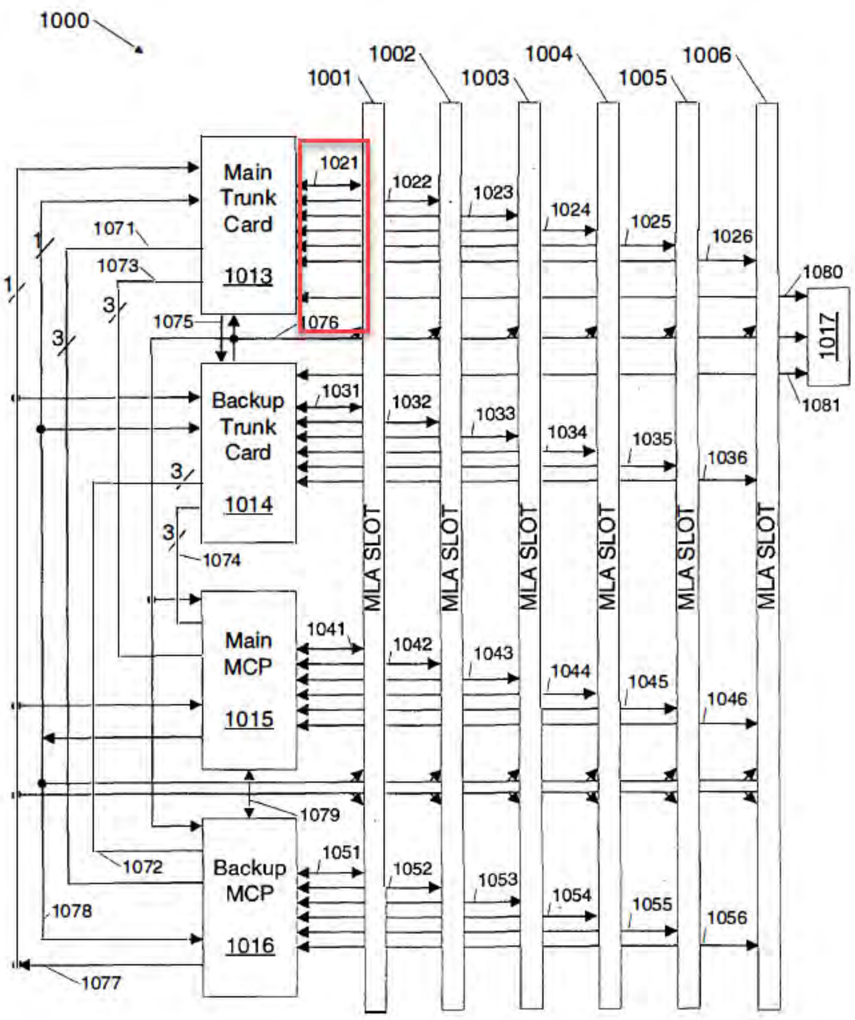
No.	’740 Patent Claim 3	Wiher ’530
3	The method according to claim 1, wherein the second physical links comprise backplane traces formed on a backplane to which the one or more interface modules are coupled.	Wiher ’530 discloses the method according to claim 1, wherein the second physical links comprise backplane traces formed on a backplane to which the one or more interface modules are coupled. For example, Wiher ’530 discloses backplane interconnection circuitry that form on the backplane which is coupled to line card shelves. Wiher ’530 at 1:36-57 (“In general, in one aspect, the invention features an appa-ratus for communicating data cells between a data link and a backplane. The apparatus includes transceiver circuitry, a main backplane interface, and backplane interconnection circuitry. The transceiver circuitry transmits and receive data cells over the data link, the main backplane

No.	’740 Patent Claim 3	Wiher ’530
		interface provides physical interconnection to the backplane, and the backplane interconnection circuitry transmits and receives cells. The main backplane interface including at least one cell signal terminal and at least one operations data signal terminal. The operations data signal terminals are separate from the cell signal terminals. The operations data signal terminals and the cell signal terminals are configured to connect to mating connectors on a backplane. Backplane interconnection circuitry couples the transceiver circuitry to the main backplane interface. The interconnection circuitry receives data cells from the transceiver circuitry and trans-mit them over cell signal terminals, receives data cells from the cell signal terminals and provide them to the transceiver circuitry for transmission over the first data link, and trans-mit and receive operations data over the operations data signal terminals.”) Wiher ’530 at 2:10-18 (“Implementations of the invention may also include one or more of the following features. The apparatus may be implemented on a card configured to be plugged into a backplane having backplane mating connectors correspond-ing to the separate terminal connectors. A fiber-optic data link interface may couple the transceiver circuitry to a synchronous optical network (SONET) data link. A high bit-rate digital subscriber line (HDSL) data link interface may be coupled to the transceiver circuitry.”) Wiher ’530 at Figure 4 (annotation added)

No.	’740 Patent Claim 3	Wiher ’530
		 400 401 402 403 404 405 406 407 408 409 410 411 412 Main LSM 460 421 422 423 424 425 426 427 428 429 430 431 432 480 481 Slot Backup LSM 470 441 442 443 444 445 446 447 448 449 450 451 452 Fig. 4 Wiher ’530 at 7:59-8:22 (“A line card shelf 300 includes a line card shelf backplane. FIG. 4 illustrates a line card shelf backplane 400. The line card shelf backplane 400 has twelve line card slots 401-412 and two line card shelf multiplexer (LSM) slots 460 and 470. A line card slot 401-412 is the point at which a line card is coupled to conductive signal paths on the backplane 400. LSM slots 460 and 470 are the points at which LSM's are coupled to line card shelf backplane signal paths. Main LSM signal paths 421-432 and backup LSM signal paths 441-452 couple line card slots 401-412 to the main LSM slot 460 and the backup LSM slot 470. Each line card slot 401-412 is connected over a dedicated signal path to main LSM slot 460 and over a second dedicated path to backup LSM slot 470. For example, line card slot 401 is connected to main LSM slot 460 over signal path 421 and to backup LSM slot 470 over signal path 441, while line card slot 402 is connected to main LSM slot 460 over signal path 422 and to backup LSM slot 470 over signal path 442. Line card slots 403-412 are similarly

No.	’740 Patent Claim 3	Wiher ’530
		connected to the main LSM slot 460 over signal paths 423-432 and to backup LSM slot 470 over signal paths 443-452, respectively. A line card shelf backplane 400 also has LSM status signal paths 480 and 481. Status signal path 480 provides a single conductive path over which main LSM 460 conveys its status to the backup LSM 470. Status information sent from the main LSM to the backup LSM indicates whether the main LSM is in an active state or whether the main LSM is in a standby or failure state. Similarly, status signal path 481 provides a single shared conductive path over which backup LSM 470 conveys its status (active or standby) to the main LSM 460 and to line cards in each of the line card slots 401-412.”) Wiher ’530 at 8:23-33 (“Line cards and LSMs include interfaces to couple signals to and from the backplane 400. FIG. 5 is a functional diagram of a LSM. The LSM 500 has line card interfaces 501-512 for communicating signals with line cards, a status input interface 531 for receiving a status signal from another LSM, a status output interface 532 for sending a status signal indicative of the LSM's current operating condition, a timing output interface 533 to output reference clock signals and an interface 530 to a master line adapter (MLA). The LSM includes, for example, circuitry 520-528 to process signals on interfaces 501-512 and 530-533.”) Wiher ’530 at 17:19-46 (“Trunk cards, MLAs, and MCPs are interconnected by a MCS backplane. FIG. 10 is a diagram of a MCS backplane. The backplane 1000 has six MLA slots 1001-1006, a main trunk card slot 1013, a backup trunk card slot 1014, a main MCP slot 1015 and a backup MCP slot 1016, and trunk interface 1017. MLA slots 1001-1006, trunk card slots 1013 and 1014 and MCP slots 1015 and 1016 are, respectively, the points at which MLAs, trunk cards, and MCPs are coupled to backplane signal paths. Trunk interface 1017 is the point at which an external trunk, such as a standard telephony T3 trunk, or a SONE T OC-3c fiber optic trunk is coupled to the backplane. The trunk interface 1017 includes switching circuitry, such as a electromagnetic relay, transistor switch-ing circuitry, or optical switching elements and receives a trunk card status signal from the backup trunk card over status signal path 1076. Trunk interface 1017 couples either the main trunk card or the backup trunk card to the external trunk based on the status signal on signal path 1076. The MCS backplane may also include slots to connect main and backup network management processors (NMPs) (not shown). NMPs may be used to connect the MCS to an external network management system and to

No.	’740 Patent Claim 3	Wiher ’530
		exchange OAMP data between LSMs, trunk cards, and MLAs. Additionally, a MCS may include slots for a main and a backup high-quality clock reference (HQR) signal generator. The main and the backup HQR provides a clock signal timing synchronized to an external network clock reference. Signals from the main and backup HQR may be sent over backplane signal paths to each LSM, MLA, and trunk card.”) Wiher ’530 at 17:47-67 (“Each MLA slot is connected by one of signal paths 1021-1026 to the main trunk card slot 1013, by one of signal paths 1031-1036 to the backup trunk card slot 1014, by one of signal paths 1041-1046 to the main MCP slot 1015 and by one of signal paths 1051-1056 to the backup MCP slot 1016. For example, MLA slot 1001 is connected to the main trunk card slot 1013 by signal paths 1021, to the backup trunk card slot 1014 by signal paths 1031, to the main MCP by signal paths 1041 and to the backup MCP by signal paths ss 1051. Similarly, MLA slot 1002 is connected to the main trunk card slot 1013 by signal paths 1022, to the backup trunk card slot 1014 by signal paths 1032, to the main MCP by signal paths 1042 and to the backup MCP by signal paths 1052. Each of signal paths 1021-1026 and 1031-1036 includes twenty conductive signal lines for the parallel transmission of ATM cells between a MLA and trunk card and, for example, 25 MHz, 19.44 MHz and 8 KHz trunk card to MLA clock signal lines. Signal paths 1041-1046 and 1051-1056 each include three conductive signal lines for the serial transmission of control data between a MLA and a MCP.”) Wiher ’530 at Figure 10 (annotation added)

No.	’740 Patent Claim 3	Wiher ’530
		 1000 1001 1002 1003 1004 1005 1006 Main Trunk Card 1013 Backup Trunk Card 1014 Main MCP 1015 Backup MCP 1016 1017 1021 1022 1023 1024 1025 1026 1031 1032 1033 1034 1035 1036 1041 1042 1043 1044 1045 1046 1051 1052 1053 1054 1055 1056 1071 1072 1073 1074 1075 1076 1077 1078 1079 1080 1081 1 1 3 3 3 3 3 MLA SLOT MLA SLOT MLA SLOT MLA SLOT MLA SLOT MLA SLOT Fig. 10

No.	’740 Patent Claim 4	Wiher ’530
4[preamble]	A method for communication, comprising:	Wiher ’530 discloses a method for communication. *See supra* at 1[preamble].
4[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	Wiher ’530 discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].
4[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Wiher ’530 discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra* at 1[c].
4[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	Wiher ’530 discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra* at 1[e].

No.	’740 Patent Claim 4	Wiher ’530
4[d]	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	Wiher ’530 discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. *See supra* at 1[f].
4[e]	sending the data frame over the selected first and second physical links,	Wiher ’530 discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].
4[f]	at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group.	Wiher ’530 discloses at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group. For example, Wiher ’530 discloses a trunk card with circuitry to provide a trunk connection. Wiher ’530 further discloses loop connections comprised of a number of port addresses. Thus, a person of ordinary skill in the art would understand that both a trunk connection and loops of port addresses comprise a grouping of first and second groups of physical links. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Wiher ’530 is found to not meet this limitation, at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group would have been obvious to a person having ordinary skill in the art, as explained below. Wiher ’530 at 6:43-64 (“ATM cells originating at network access equipment 201-208 are sent over transmission loops 261-268 between the network access equipment and line cards in line card shelves 211-214. A loop 261-268 may be, for example, a digital subscriber line operated over a

No.	’740 Patent Claim 4	Wiher ’530
		twisted wire pair connection. A loop 261-268 terminates at a line card in a line card shelf 211-214. A line card shelf 211-214 may house multiple line cards. Each line card terminates one or more loop connections to network access equipment 201-208. A master control shelf 221, 222 is connected to one or more line card shelves. For example, line card shelves 211 and 212 are connected to master control shelf 221 and line card shelves 213 and 214 are connected to master control shelf 222. A master control shelf 221, 222 is a card shelf that controls and regulates the flow of data between line card shelves and a trunk interface 241, 242. A trunk interface 241, 242 provides a trunk connection 241, 242 between the master control shelf 221, 222 and the ATM network 230. Trunk interface 241, 242 is, for example, a 45 Mbit/second T-3 interface or a standard 155 Mbit/second fiber optic synchronous optical network optical carrier level 3 concatenated data (SONET OC-3c) interface.”) Wiher ’530 at 9:43-10:4 (“FIG. 6 is a functional diagram of a line card. A line card 600 provides a signal termination point for subscriber loop data link connections. A line card 600 has a main LSM interface 610, a backup LSM interface 620, a LSM status input interface 631, and a subscriber loop data link interface 632. Each interface 610, 620, 631, 632 includes one or more signal lines over which electrically modulated signals are exchanged. To process signals on interfaces 610, 620, 631, and 632, a line card 600 includes line card circuitry such as a processor 645, line card to LSM backplane interface circuitry 647, and transceiver circuitry 643 and 644. The processor 645 may include integrated memory storage or may include interfaces to memory 646. The processor 645 may control line card to LSM communications, power management for line card circuitry, line card initialization, operations, maintenance, and provisioning. The processor 645 is, for example, a Motorola MC68360 processor. Back-plane interface circuitry 647 receives and transmits signals over the line card shelf backplane 400, multiplexes and de-multiplexes ATM cells exchanged between the LSM and line card transceivers 643, 644 and may buffer cell traffic in memory 646. The backplane interface circuitry 647 is imple-mented using, for example, an Altera FLEX l0K program-mable logic device, a field-programmable gate array, or other processing circuitry. Backplane interface circuitry 647 is coupled to transceiver circuitry 643, 644. Transceivers 643, 644 provide for modulation and demodulation of data over the subscriber loop interface 632 using a digital data modulation technology.”)

<table>
<tr><th>No.</th><th>’740 Patent Claim 4</th><th>Wiher ’530</th></tr>
<tr><td></td><td></td><td>
Wiher ’530 at Figure 6

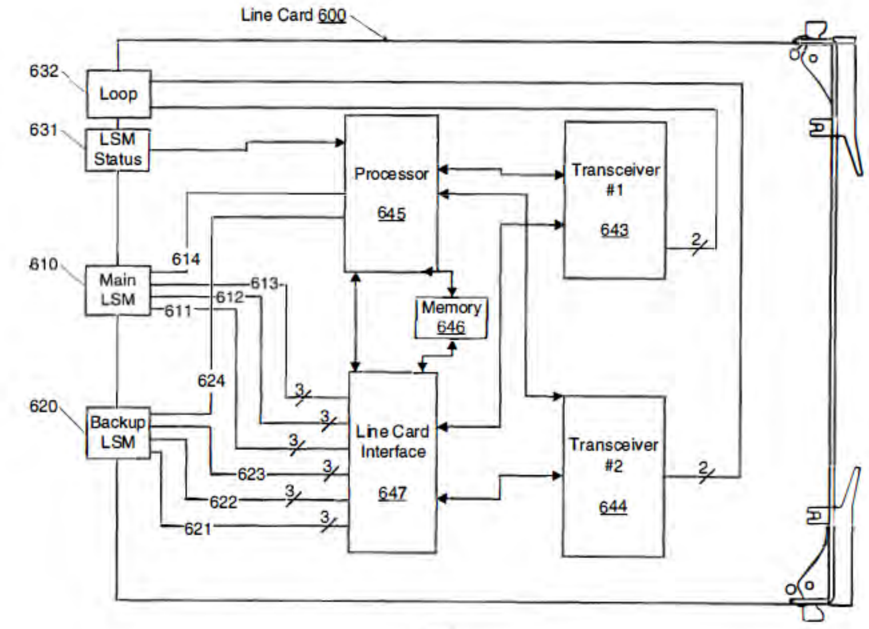
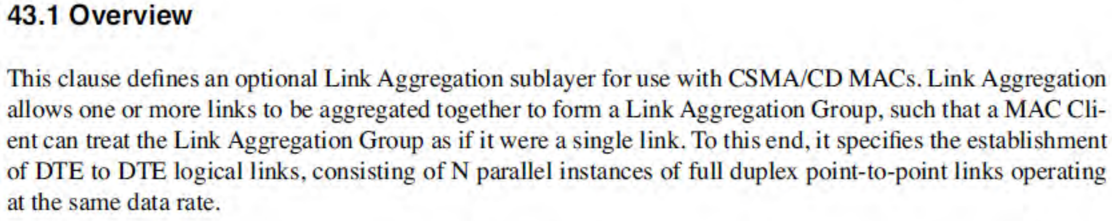
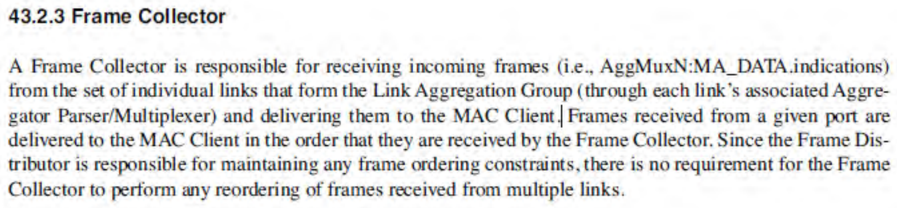
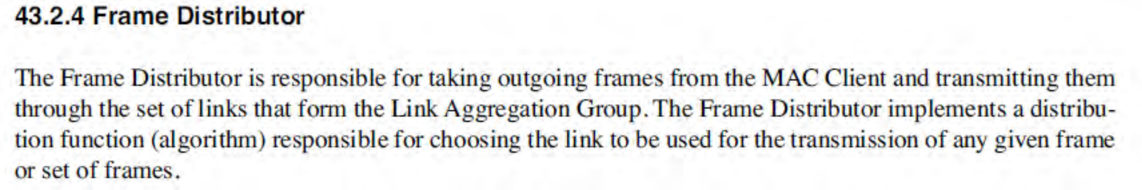
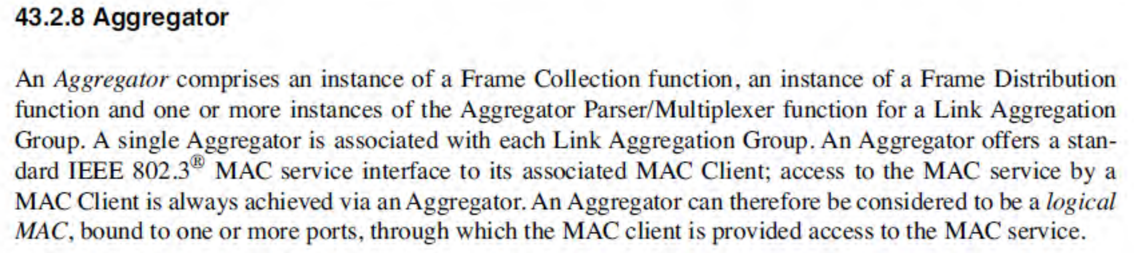
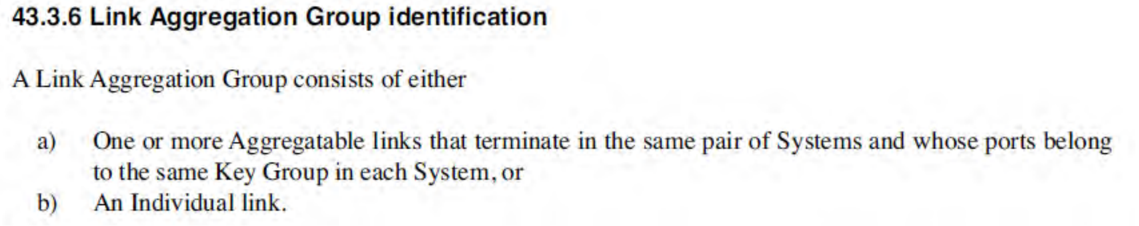


Fig. 6

Wiher ’530 at 10:38-10:62 (“The described coupling between line cards, LSMs, and backplane signal paths provides each line card with dedicated connections to two LSMs. For example, when a line card 600 is inserted in line card slot 401, the line card's main LSM interface 610 is coupled to main LSM line card interface 501 by backplane signal paths 421, the line card's backup LSM interface 620 is coupled to backup LSM line card interface 501 by backplane signal paths 441, the line card's LSM status input interface 631 is coupled to the backup LSM's status output interface 532 by backplane signal paths 481 and the line card's subscriber loop interface 632 is coupled to subscriber loop signal paths. Similarly, when a line card 600 is inserted in line card slot 402, the line card's main LSM interface 610 is coupled to main LSM line card interface 502 by backplane signal paths 422, the line card's backup LSM interface 620 is coupled to backup LSM line card interface 502 by backplane signal paths 442, the line card's LSM status input interface 631 is
</td></tr>
</table>

No.	’740 Patent Claim 4	Wiher ’530
		coupled to the backup LSM's status interface 532 by backplane signal paths 481 and the line card's subscriber loop interface 632 is coupled to subscriber loop signal paths. Line cards inserted in slots 403-412 are similarly connected to main LSM interfaces 503-512, backup LSM interfaces 503-512, backup LSM status output interface 532, and subscriber loop connections.”) Wiher ’530 at 10:63-11:9 (“A line card's main LSM interface 610 includes LSM to line card cell transfer signal lines 611 and line card to LSM cell transfer signal lines 612. Signal lines 611 and 612 are coupled to complementary signal lines at line card interfaces 501-512 (FIG. 5) of a LSM by backplane signal paths 421-432 (FIG. 4). Signals modulated over signal lines 611 are used to receive ATM cells sent from the main LSM. Signals modulated over signal lines 612 are used to send ATM cells to the main LSM. Signals exchanged over signal lines 611 and 612 are modulated in reference to a clock signal that is, for example, a 12.5 megahertz (MHz) clock signal received from a LSM, and signals may be asserted or de-asserted on the rising edge of a clock pulse and sampled on the falling edge of the clock pulse.”) Wiher ’530 at 18:33-60 (“A trunk card includes, for example, circuitry 1122-1128 to process signals on interfaces 1101-1106 and on interfaces 1129-1135. MLA interface control circuitry 1122 process ATM cell transfer signals exchanged over interfaces 1101-1106 and controls the transmission of ATM cells between the trunk card and MLAs. Interface control cir-cuitry 1122 may include ATM cell buffers to temporarily store cells received from, or being transmitted to, MLAs over interfaces 1101-1106. Cell multiplexer/de-multiplexer circuitry 1123 exchanges ATM cells with the MLA interface control circuitry 1122 and determines the flow of ATM cells between the control circuitry 1122 and header translation circuitry 1124. Additionally, the multiplexer/de-multiplexer circuitry 1123 may extract OAMP cells arriving from the trunk interface 1125 and direct them to processor 1128 and insert OAMP cells destined to the trunk interface 1125 from the processor 1128. Header translation circuitry 1124 translates header information in ATM cells arriving from or destined to the trunk interface 1125. Header translation circuitry 1124 may access header translation programs and data stored in RAM and ROM memory 1127. For example, processor 1128 may store VPI/VCI header field translation information in memory 1127 for use by header translation circuitry 1124. Header translation circuitry 1124 exchanges ATM cells with trunk interface circuitry 1125. Trunk inter-face circuitry 1125 provides a trunk connection to an

No.	’740 Patent Claim 4	Wiher ’530
		ATM network. The trunk connection is, for example, a standard 45 Megabit per second T3 trunk connection.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Wiher ’530 in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 4[f] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, IEEE 802.3 discloses the aggregation of one or more links together to form a Link Aggregation Group. IEEE 802.3 at 1465  **43.1 Overview** This clause defines an optional Link Aggregation sublayer for use with CSMA/CD MACs. Link Aggregation allows one or more links to be aggregated together to form a Link Aggregation Group, such that a MAC Client can treat the Link Aggregation Group as if it were a single link. To this end, it specifies the establishment of DTE to DTE logical links, consisting of N parallel instances of full duplex point-to-point links operating at the same data rate. IEEE 802.3 at 1470  **43.2.3 Frame Collector** A Frame Collector is responsible for receiving incoming frames (i.e., AggMuxN:MA_DATA.indications) from the set of individual links that form the Link Aggregation Group (through each link’s associated Aggregator Parser/Multiplexer) and delivering them to the MAC Client. Frames received from a given port are delivered to the MAC Client in the order that they are received by the Frame Collector. Since the Frame Distributor is responsible for maintaining any frame ordering constraints, there is no requirement for the Frame Collector to perform any reordering of frames received from multiple links. IEEE 802.3 at 1471

No.	’740 Patent Claim 4	Wiher ’530
		 **43.2.4 Frame Distributor** The Frame Distributor is responsible for taking outgoing frames from the MAC Client and transmitting them through the set of links that form the Link Aggregation Group. The Frame Distributor implements a distribution function (algorithm) responsible for choosing the link to be used for the transmission of any given frame or set of frames. IEEE 802.3 at 1474  **43.2.8 Aggregator** An *Aggregator* comprises an instance of a Frame Collection function, an instance of a Frame Distribution function and one or more instances of the Aggregator Parser/Multiplexer function for a Link Aggregation Group. A single Aggregator is associated with each Link Aggregation Group. An Aggregator offers a standard IEEE 802.3® MAC service interface to its associated MAC Client; access to the MAC service by a MAC Client is always achieved via an Aggregator. An Aggregator can therefore be considered to be a *logical MAC*, bound to one or more ports, through which the MAC client is provided access to the MAC service. IEEE 802.3 at 1481  **43.3.6 Link Aggregation Group identification** A Link Aggregation Group consists of either a) One or more Aggregatable links that terminate in the same pair of Systems and whose ports belong to the same Key Group in each System, or b) An Individual link. For example, Ghosh discloses aggregating physical links, including ports, into aggregate port channels that form a single logical link to increase bandwidth. Ghosh at Abstract (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggregation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange

No.	’740 Patent Claim 4	Wiher ’530
		sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent gen-eration of port channel map tables.”) Ghosh at [0004] (“Neighboring nodes in a fibre channel network are typically interconnected through multiple physical links. For example, a local fibre channel switch may be connected to a remote fibre channel switch through four physical links. In many instances, it may be beneficial to aggregate some of the physical links into logical links. That is, multiple physi-cal links can be combined to form a logical interface to provide higher aggregate bandwidth, load balancing, and link redundancy. When a frame is being transmitted over a logical link, it does not matter what particular physical link is being used as long as all the frames of a given flow are transmitted through the same link. If a constituent physical link goes down, the logical link can still remain operational.”) Ghosh at [0007] (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggre-gation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent generation of port channel map tables.”) Ghosh at [0008] (“In one embodiment, a method for aggregating ports in a fibre channel fabric is provided. It is determined that a plurality of local ports at a local fibre channel switch are compatible. Identifiers for the plurality of local ports are sent to a remote fibre channel switch. The remote fibre channel switch determines if a plurality of remote ports are compat-ible, the plurality of remote ports corresponding to the plurality of local ports. An indication that one or more of the remote physical ports are compatible is received. A port channel including one or more of the local ports correspond-ing to the compatible remote ports is created.”) Ghosh at [0010] (“In another embodiment, a fibre channel network is described. The fibre channel network includes a local fibre channel switch and a remote fibre channel switch. The local fibre channel switch aggregates a compatible subset of the plurality of local ports and sends identifiers for the compat-ible subset of the plurality of local ports to the remote fibre channel switch. The

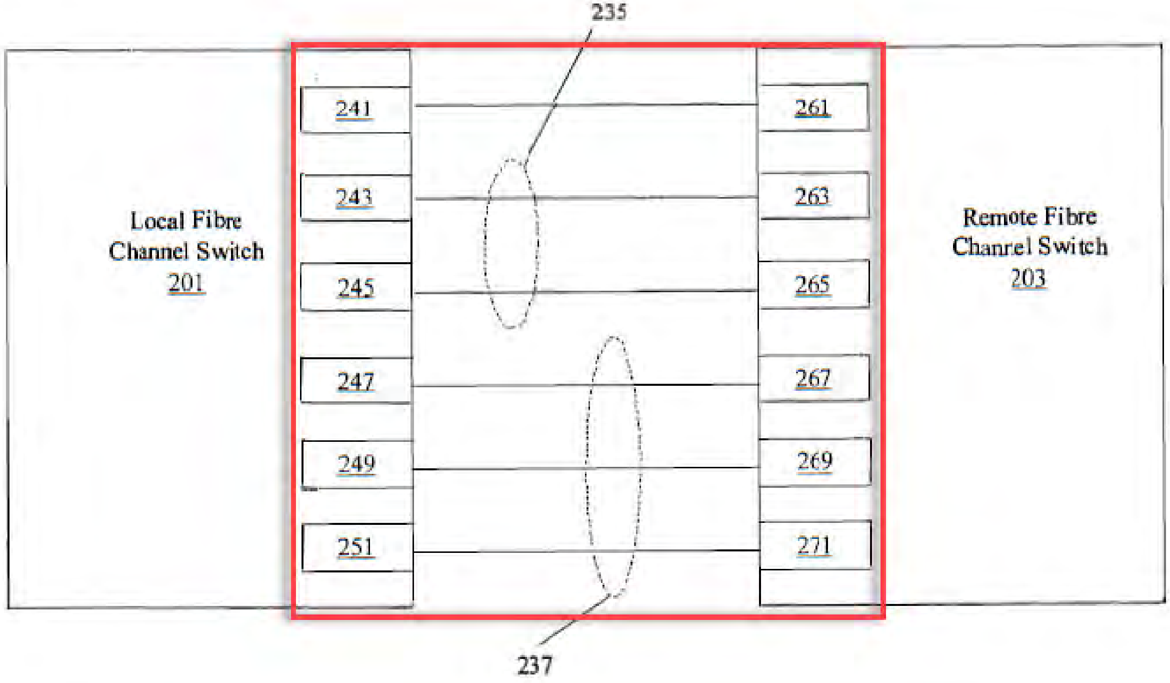
No.	’740 Patent Claim 4	Wiher ’530
		remote fibre channel switch determines if a subset of the plurality of remote ports are compatible. The subset of the plurality of remote ports corresponds to the compatible subset of the plurality of local ports.”) Ghosh at [0022] (“Switches in a fibre channel network are typically interconnected using multiple physical links. The physical links connecting a pair of switches allows transmission of data and control signals. In some instances, it is useful to aggregate multiple physical links into a logical link. Physi-cal links are also referred to herein as physical interfaces and channels while logical links are also referred to herein as logical interfaces and port channels. For example, a local switch may be connected to a remote switch through four physical links. Instead of having to transmit data through a particular physical link, the physical links can be aggregated to form one or more logical links. In one example, all four physical links are aggregated into a single logical link. Instead of having data transmitted through a particular physical link, the data can merely be transmitted over a particular logical link without regard to the particular physi-cal interface used. Aggregating physical links into a logical link allows for higher aggregated bandwidth, load balancing, and link redundancy. For example, if a particular physical link fails or is overloaded, data can still be transmitted over the logical link.”) Ghosh at [0029] (“FIG. 2 is a diagrammatic representation showing links between two switches, such as two fibre channel switches shown in FIG. 1. A local fibre channel switch 201 includes local ports 241, 243, 245, 247, 249, and 251. A remote fibre channel switch 203 includes remote ports 261, 263, 265, 267, 269, and 271. Local port 241 is coupled to remote port 261 through an individual physical link or channel. Connected ports are also referred to herein as peer ports. Local port 243 is coupled to remote port 263 and local port 245 is coupled to remote port 265. The two resulting physical links are aggregated to form port channel 235. Local ports 247, 249, and 251 are coupled to remote ports 267, 269, and 271 respectively. The three resulting physical links are aggregated to form port channel 237.”) Ghosh at [0030] (“According to various embodiments, local fibre channel switch 201 and remote fibre channel switch both have associated identifiers. In some examples, the identifiers are globally unique identifiers such as a global switch world wide names (WWNs). Each local port 241, 243,

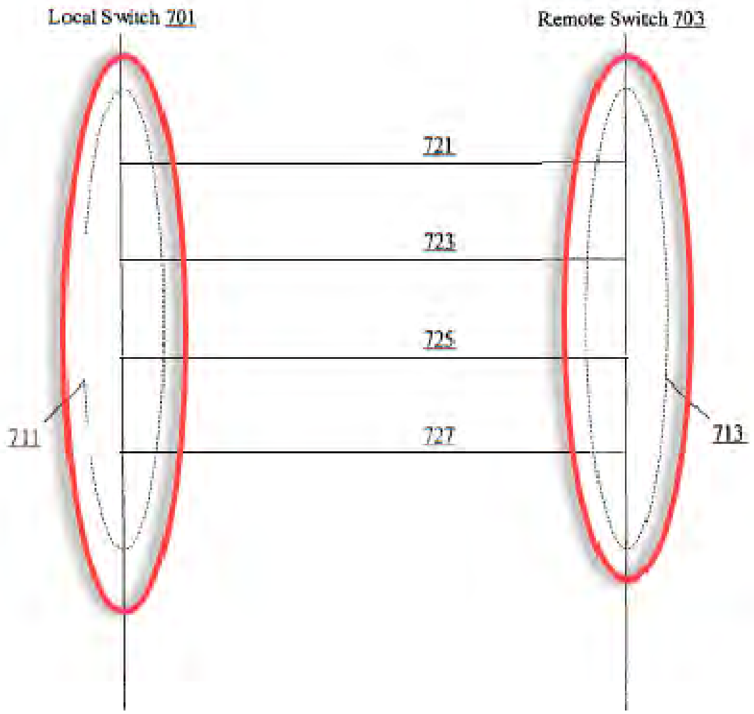
No.	’740 Patent Claim 4	Wiher ’530
		245, 247, 249, and 251 and each remote port 261,263,265,267, 269, and 271 can also be associated with identifiers. In some examples, the identifiers are port WWNs. The port WWNs are typically used for debugging or identifying the peer port in alert or warning messages. However, according to various embodiments, the techniques of the present invention use WWNs as globally unique identifiers to aggregate ports instead of using compatibility keys which are only locally unique. Compatibility keys are mechanisms typically used by other protocols such as Ethernet for aggregation.”) Ghosh at [0033] (“A variety of parameters can be used to aggregate physical ports. FIG. 3 is a flow process diagram showing one technique for aggregating physical ports into a logical port. And 301, it is determined if auto create functionality is enabled. According to various embodiments, auto create functionality allows automatic configuration and detection of compatible physical ports as well as aggregation into one or more logical ports. Auto creation does not require user intervention. In other examples, administrators can manu-ally arrange ports for aggregation.”) Ghosh at [0037] (“FIG. 4 is an exchange diagram showing one example of a bring up procedure used for a port creating a new port channel. A local switch 401 is coupled to a remote switch 403. The local switch 401 includes a physical port Al coupled to physical port Bl included in remote switch 403. When two peer ports Al and Bl are being aggregated into a port channel, the peer switches 401 and 403 typically already know the world wide names of the individual physical peer ports. However, the peer switches only know the world wide name of their own logical port or port channel. That is, both switches have the individual physical link configured, but the link is not yet part of a port channel. At 421, a local switch 401 sends a synchronize (sync) message 411 to the remote switch 403 to begin the process of creating a port channel including ports Al and Bl.”) Ghosh at [0038] (“In some examples, the sync message 411 includes a local port channel identifier and a remote port channel identifier. In one particular example, the local port channel identifier is set to the world wide name of the local port channel assigned by the local switch 401. The remote port channel identifier is left blank to indicate that the port Al is being aggregated as part of a new port channel. The sync message 411 can also include other parameters such as channel status, channel model, or channel intent.”)

No.	’740 Patent Claim 4	Wiher ’530
		Ghosh at [0042] (“When two peer ports A2 and B2 are aggregated into a port channel Cl, the peer switches 501 and 503 typically already know the world wide names of the indi-vidual physical peer ports A2 and B2 as well as the world wide name information of the port channel Cl. Conse-quently, the port channel is already successfully established. According to various embodiments, local switch 501 and remote switch 503 perform parameter checking to ensure that the new physical port A2 and B2 can be safely added to the existing port channel Cl. At 521, a local switch can check configuration parameters to ensure that physical ports Al and A2 at the local switch 501 are compatible. The compatibility checking can be performed anytime. In some examples, compatibility checking is checked before a local switch 501 sends a synchronize (sync) message 511 to the remote switch 503 to begin the process of aggregating ports A2 and B2 into the port channel.) Ghosh at [0043] (“In some examples, the sync message 511 includes local port channel identifier and a remote port channel identifier. In one particular example, the local port channel identifier is set to the world wide name of the local port channel assigned by the local switch 501. The remote port channel identifier is filled with the existing port channel identifier to indicate that the port A2 is being aggregated into existing port channel C2. The sync message 511 can also include other parameters such as channel status, channel model, or channel intent.”) Ghosh at [0044] (“At 531, remote switch 503 uses the information received from the local switch 501 to verify port B2 is compatible with other port in port channel C2. In one example, configuration parameters associated with B2 are checked against configuration parameters associated with Bl. The remote switch 503 can also check if the port B2 is already assigned to a different port channel. If the port B2 is compatible with port Bl, the remote switch 503 can proceed and send a sync accept message 513 in response to the sync message 511 to indicate that the port B2 can be aggregated into the port channel. The sync accept message indicates that a port channel can now be modified. At 523, local switch 501 uses the information to update its own port channel database. However, the port channel may not yet be fully operational until the hardware configuration is completed. The local switch 501 continues hardware configuration such as line card configuration to make the port A2 part of the port channel Cl. An acknowledgment 527 is

No.	’740 Patent Claim 4	Wiher ’530
		sent and received by remote switch 503 at 529. In some examples, the local switch 501 sends a commit signal 515 when hardware configuration is complete.”) Ghosh at [0045] (“The remote switch 503 receives the commit signal at 533 and begins its own hardware configuration. On completion of its hardware configuration, remote switch 503 sends out a commit accept signal 517 to indicate to local switch 501 that hardware configuration is completed. According to various embodiments, local switch 501 receives the commit accept signal 517 and notifies relevant applications that the port channel is now fully operational at 525 and that port A2 has been aggregated into port channel Cl. The local switch 501 can also send out an acknowledge message 519. When the remote switch 503 receives the acknowledge, it notifies relevant applications that the port channel is operational at 535 and that port B2 has been aggregated into port channel Cl. In one embodiments, the techniques of the present invention contemplate using a two phase SYNC and COMMIT mechanism similar to the mechanism used in EPP.”) Ghosh at [0046] (“FIGS. 4 and 5 show examples of ports being aggregated into a port channel. At a particular switch, ports can be selected for aggregation into a port channel in a variety of manners. FIG. 6 is an exchange diagram showing automatic selection of ports at a switch for aggregation into a port channel. A local switch 601 is coupled to a remote switch 603. In one example, the local switch 601 includes physical ports Al, A2, A3, and A4 while remote switch 603 includes physical ports Bl, B2, B3, and B4. No port chan-nels have been formed.”) Ghosh at [0049] (“At 631, remote switch 603 uses the information received from the local switch 601 to verify that the peer ports of Al, A2, and A4 are compatible. That is, ports Bl, B2, and B4 are checked for compatibility. In one example, only ports Bl and B2 may be compatible, and consequently only ports Al, A2, Bl, and B2 can be included in the port channel. In another example, ports Bl, B2, and B4 are compatible, so ports Al, A2, A4, Bl, B2, and B4 can be aggregated into port channel Cl. According to various embodiments, if the port B2 is compatible with port Bl, the remote switch 603 can proceed and send a sync accept message 613 in response to the sync message 611 to indicate that the port B2 can be aggregated into the port channel. It should be noted that remote switch 603 can send a list indicating that ports B2 and B4 are compatible with Bl.

No.	’740 Patent Claim 4	Wiher ’530
		However, the remote switch 603 sends only one compatible port B2 back for several reasons, and in the process of selection compatible port channels get priority over com-patible individual ports.”) Ghosh at [0050] (“One reason is that aggregation mechanisms and techniques can be implemented more elegantly by handling ports on an individual basis. Any individual port will either start a new port channel, be added to an existing port channel, or operate stand alone. There is no need to keep track of groups of ports to be aggregated. Another reason is that fewer ports need to be locked if only a single port is being aggregated at any one time. The sync accept message indicates that a port channel can now be modified. At 623, local switch 601 receives the information and recognizes that Al and A2 can now be aggregated into port channel Cl. However, the port channel may not yet be fully operational until the hardware configuration is completed. An acknowl-edgment 627 is sent and received by remote switch 603 at 629. In some examples, the local switch 601 sends a commit signal 615 when hardware configuration is complete.”) Ghosh at [0051] (“The remote switch 603 receives the commit signal at 633 to create port channel Cl including ports Bl and B2. Hardware configuration can now be performed. On comple-tion of its hardware configuration, remote switch 603 sends out a commit accept signal 617 to indicate to local switch 601 that hardware configuration is completed. According to various embodiments, local switch 601 receives the commit accept signal 617 and notifies relevant applications that the port channel is now fully operational at 625 and that ports Al and A2 have been aggregated into port channel Cl. The local switch 601 can also send out an acknowledge message 619. When the remote switch 603 receives the acknowledge, it notifies relevant applications that the port channel is fully operational at 635 and that ports Bl and B2 have been aggregated into port channel Cl.”) Ghosh at [0053] (“FIG. 7 is a diagrammatic representation showing synchronous aggregation of ports into a port channel. A local switch 701 is coupled to a remote switch 703 through links 721, 723, 725, and 727. According to various embodiments, the links are being aggregated into port channel 711 at the local switch 701 and port channel 713 at the remote switch 703 in a synchronous manner. That is the peer ports corre-sponding to each link are brought up in the same order at both the local switch 701 and the remote switch 703.”)

No.	’740 Patent Claim 4	Wiher ’530
		Ghosh at Figure 2 (annotation added)  235 Local Fibre Channel Switch 201 241 243 245 247 249 251 261 263 265 267 269 271 Remote Fibre Chanrel Switch 203 237 Figure 2 Ghosh at Figure 7 (annotation added)

No.	’740 Patent Claim 4	Wiher ’530
		 Local Switch 701 Remote Switch 703 721 723 725 727 711 713 Figure 7

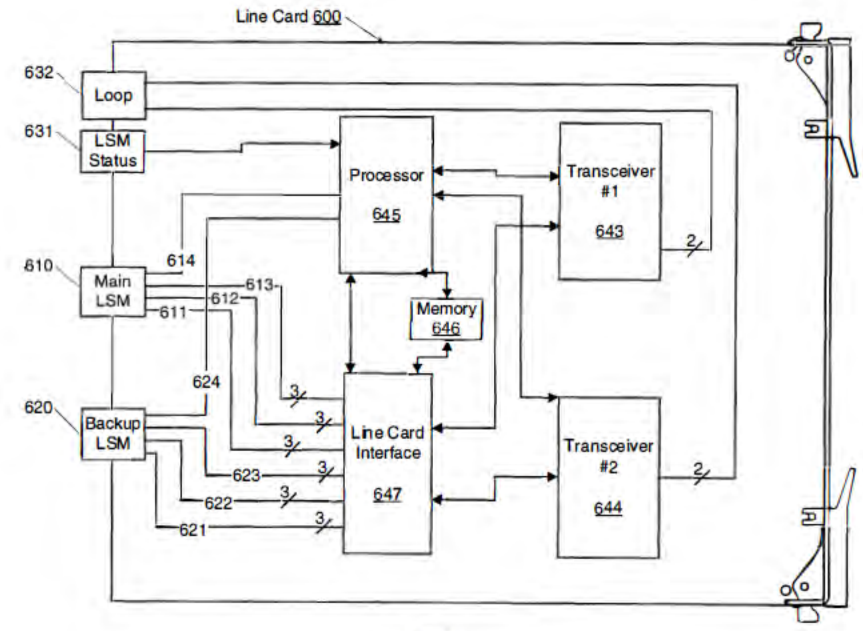
No.	’740 Patent Claim 5	Wiher ’530
5[preamble]	A method for communication, comprising:	Wiher ’530 discloses a method for communication. *See supra at* 1[preamble].

No.	’740 Patent Claim 5	Wiher ’530
5[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	Wiher ’530 discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].
5[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Wiher ’530 discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra at* 1[c].
5[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	Wiher ’530 discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra at* 1[e].
5[d]	selecting, in a single computation based on at least one of the frame attributes, a first	Wiher ’530 discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. *See supra at* 1[f].

No.	’740 Patent Claim 5	Wiher ’530
	physical link out of the first group and a second physical link out of the second group; and	
5[e]	sending the data frame over the selected first and second physical links,	Wiher ’530 discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].
5[f]	coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node.	Wiher ’530 discloses coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node. For example, Wiher ’530 discloses a trunk card with circuitry to provide a trunk connection. Wiher ’530 further discloses loop connections comprised of a number of port addresses. Thus, a person of ordinary skill in the art would understand that both a trunk connection and loops of port addresses comprise a grouping of first and second groups of physical links to increase data bandwidth. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Wiher ’530 is found to not meet this limitation, coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node would have been obvious to a person having ordinary skill in the art, as explained below. Wiher ’530 at 3:43-65 (“In general, in another aspect, the invention features an apparatus for communicating data cells between a data link and a backplane. The apparatus includes transceiver circuitry to transmit and receive data cells over a data link and a plurality of backplane interfaces each including at least one cell signal terminal. Each of the backplane interface is coupled to a backplane interconnection circuit. Each backplane interconnection circuit transmits and receives

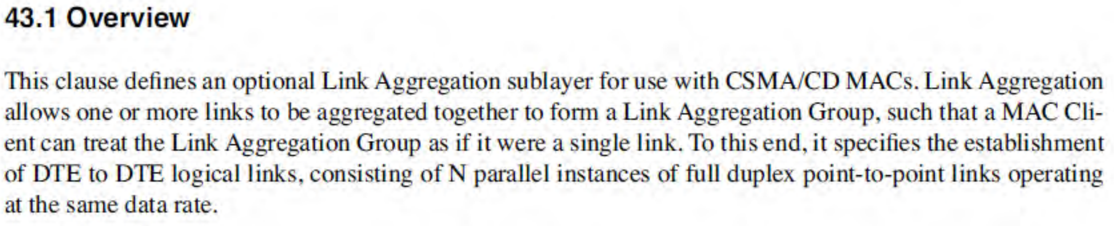
No.	’740 Patent Claim 5	Wiher ’530
		cells over the cell signal terminals of its associated backplane interface. The apparatus also includes de-multiplexing cir-cuitry coupling the transceiver circuitry to each of the backplane interconnection circuits. The de-multiplexing circuitry receives a data cell from the transceiver circuitry, select a backplane interconnection circuit associated with the data cell, and provide the data cell to the selected backplane interconnection circuit for transmission over the cell signal terminals of the associated backplane interface. The apparatus also includes multiplexing circuitry coupling the plurality of backplane interconnection circuits to the transceiver circuitry. The multiplexing circuitry receives data cells from each of the backplane interconnection cir-cuits and provide the received data cells to the transceiver circuitry.”) Wiher ’530 at 6:43-64 (“ATM cells originating at network access equipment 201-208 are sent over transmission loops 261-268 between the network access equipment and line cards in line card shelves 211-214. A loop 261-268 may be, for example, a digital subscriber line operated over a twisted wire pair connection. A loop 261-268 terminates at a line card in a line card shelf 211-214. A line card shelf 211-214 may house multiple line cards. Each line card terminates one or more loop connections to network access equipment 201-208. A master control shelf 221, 222 is connected to one or more line card shelves. For example, line card shelves 211 and 212 are connected to master control shelf 221 and line card shelves 213 and 214 are connected to master control shelf 222. A master control shelf 221, 222 is a card shelf that controls and regulates the flow of data between line card shelves and a trunk interface 241, 242. A trunk interface 241, 242 provides a trunk connection 241, 242 between the master control shelf 221, 222 and the ATM network 230. Trunk interface 241, 242 is, for example, a 45 Mbit/second T-3 interface or a standard 155 Mbit/second fiber optic synchronous optical network optical carrier level 3 concatenated data (SONET OC-3c) interface.”) Wiher ’530 at 9:43-10:4 (“FIG. 6 is a functional diagram of a line card. A line card 600 provides a signal termination point for subscriber loop data link connections. A line card 600 has a main LSM interface 610, a backup LSM interface 620, a LSM status input interface 631, and a subscriber loop data link interface 632. Each interface 610, 620, 631, 632 includes one or more signal lines over which electrically modulated signals are exchanged. To process signals on interfaces 610, 620, 631, and 632, a line card 600 includes line card circuitry such as a processor 645, line card to LSM

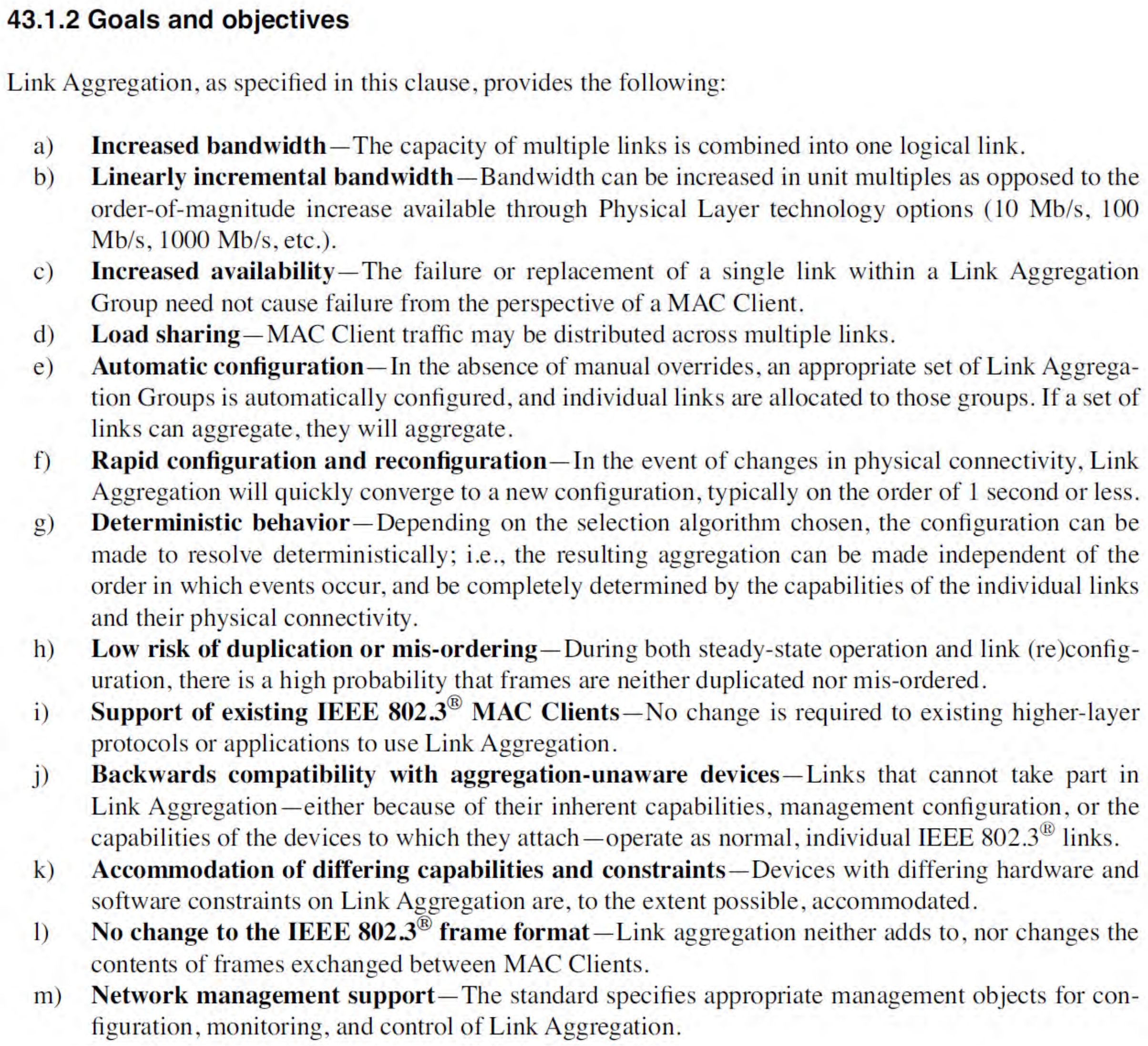
No.	’740 Patent Claim 5	Wiher ’530
		backplane interface circuitry 647, and transceiver circuitry 643 and 644. The processor 645 may include integrated memory storage or may include interfaces to memory 646. The processor 645 may control line card to LSM communications, power management for line card circuitry, line card initialization, operations, maintenance, and provisioning. The processor 645 is, for example, a Motorola MC68360 processor. Back-plane interface circuitry 647 receives and transmits signals over the line card shelf backplane 400, multiplexes and de-multiplexes ATM cells exchanged between the LSM and line card transceivers 643, 644 and may buffer cell traffic in memory 646. The backplane interface circuitry 647 is imple-mented using, for example, an Altera FLEX l0K program-mable logic device, a field-programmable gate array, or other processing circuitry. Backplane interface circuitry 647 is coupled to transceiver circuitry 643, 644. Transceivers 643, 644 provide for modulation and demodulation of data over the subscriber loop interface 632 using a digital data modulation technology.”) Wiher ’530 at Figure 6

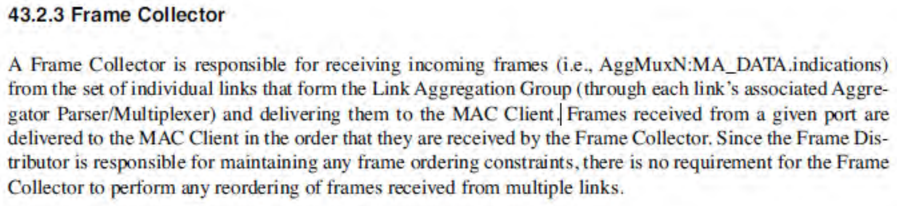
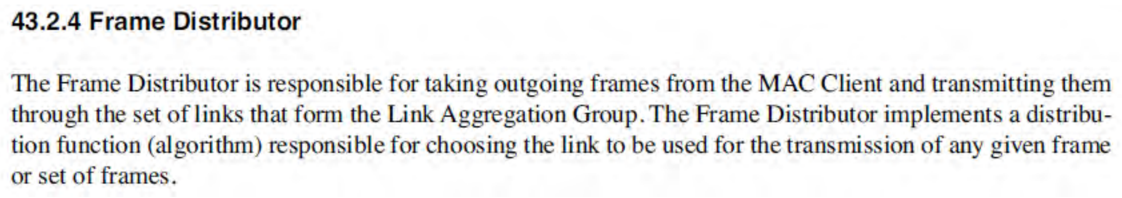
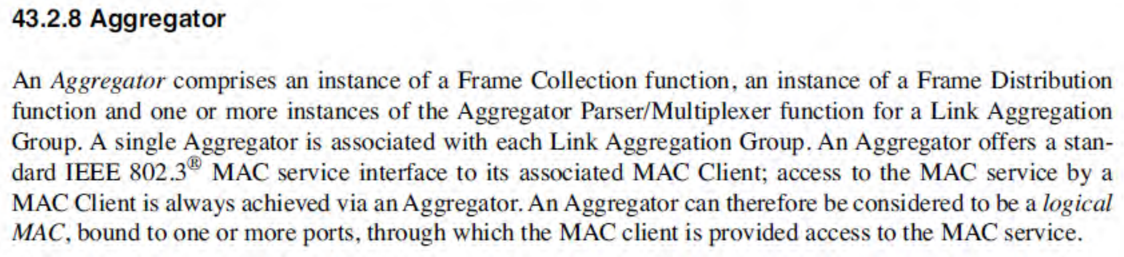
No.	’740 Patent Claim 5	Wiher ’530
		 Line Card 600 632 Loop 631 LSM Status Processor 645 Transceiver #1 643 2 610 Main LSM 614 613 612 611 Memory 646 624 3 620 Backup LSM Line Card Interface 647 Transceiver #2 644 623 622 621 Fig. 6 Wiher ’530 at 10:38-10:62 (“The described coupling between line cards, LSMs, and backplane signal paths provides each line card with dedicated connections to two LSMs. For example, when a line card 600 is inserted in line card slot 401, the line card's main LSM interface 610 is coupled to main LSM line card interface 501 by backplane signal paths 421, the line card's backup LSM interface 620 is coupled to backup LSM line card interface 501 by backplane signal paths 441, the line card's LSM status input interface 631 is coupled to the backup LSM's status output interface 532 by backplane signal paths 481 and the line card's subscriber loop interface 632 is coupled to subscriber loop signal paths. Similarly, when a line card 600 is inserted in line card slot 402, the line card's main LSM interface 610 is coupled to main LSM line card interface 502 by backplane signal paths 422, the line card's backup LSM interface 620 is coupled to backup LSM line card interface 502 by backplane signal paths 442, the line card's LSM status input interface 631 is coupled to the backup LSM's status interface 532 by backplane signal paths 481 and the line card's

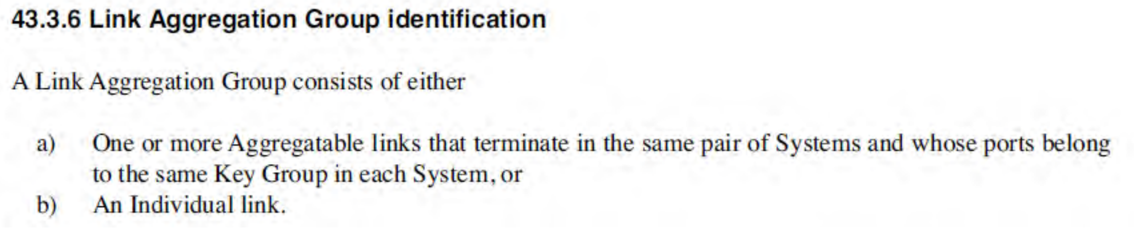
No.	’740 Patent Claim 5	Wiher ’530
		subscriber loop interface 632 is coupled to subscriber loop signal paths. Line cards inserted in slots 403-412 are similarly connected to main LSM interfaces 503-512, backup LSM interfaces 503-512, backup LSM status output interface 532, and subscriber loop connections.”) Wiher ’530 at 10:63-11:9 (“A line card's main LSM interface 610 includes LSM to line card cell transfer signal lines 611 and line card to LSM cell transfer signal lines 612. Signal lines 611 and 612 are coupled to complementary signal lines at line card interfaces 501-512 (FIG. 5) of a LSM by backplane signal paths 421-432 (FIG. 4). Signals modulated over signal lines 611 are used to receive ATM cells sent from the main LSM. Signals modulated over signal lines 612 are used to send ATM cells to the main LSM. Signals exchanged over signal lines 611 and 612 are modulated in reference to a clock signal that is, for example, a 12.5 megahertz (MHz) clock signal received from a LSM, and signals may be asserted or de-asserted on the rising edge of a clock pulse and sampled on the falling edge of the clock pulse.”) Wiher ’530 at 11:40-57 (“ATM cells are transferred from a LSM to a line card by the exchange of the LC-RR, LSM-SR, and LSM-DATA signals on the LSM to line card cell transport signal lines 611. FIGS. 7A and 7B illustrate timing and modulation of the LC-RR, LSM-SR, and LSM-DATA signals. The LC-RR signal is sent from the line card to the LSM to indicate line card ports that are ready to receive ATM cell transfers. FIG. 7A is a LC-RR signal timing diagram for a line card supporting two line card ports. The LC-RR signal is modu-lated to periodically transmit a framing indicator and port status information from the line card to the LSM. The framing indicator is sent by asserting the LC-RR signal for one clock cycle at, for example, sixteen clock cycle inter-vals. During each clock cycle following the framing indicator, port status information may be sent from the line card to the LSM. Port status information is sent by asserting or de-asserting the LC-RR signal during a clock period that is unique for each port on the line card.”) Wiher ’530 at 11:58-12:3 (“Referring to FIG. 7A, in the exemplary LC-RR timing diagram 700, the LC-RR signal is asserted when in a low voltage state. At clock 1, a framing indicator "F" is sent by asserting the LC-RR signal. At clock 2, the LC-RR signal is de-asserted indicating that the first line card port "Pl" is not ready to receive data. At clock 3, the LC-RR signal is asserted indicating that the second line card port "P2" is ready to receive data. During the clock periods between the

No.	’740 Patent Claim 5	Wiher ’530
		port status information and the subsequent framing indicator, i.e., periods 4 to 16, the LC-RR signal remains de-asserted. At clock 17 the framing indicator is again asserted and followed, at clocks 18 and 19, by updated port information.”) Wiher ’530 at 12:32-63 (“ATM cells are transferred from the line card to the LSM by the exchange of LC-SR, LSM-RR, and LC-DATA signals on the line card to LSM cell transfer signal lines 612. FIG. 7C is a line card to LSM cell transport interface signal timing diagram. The signal timing diagram 750 illustrates timing and modulation of LC-SR, LSM-RR, and LC-DATA signals exchanged during an ATM cell transfer from the line card to a LSM. At clock cycle 2, the line card indicates that it is ready to transfer a data cell to a LSM by asserting (low) the LC-SR signal. At clock cycle 3, the LSM indicates that it is ready to accept data from the line card by asserting (low) the LSM-RR signal. Note that the LSM need not assert the LSM-RR signal immediately after receipt of the LC-SR signal, but rather the LSM may delay assertion of LSM-RR until it is ready to receive the data transfer. Following the LSM's assertion of the LSM-RR signal, the line card waits for two clock cycles (clock 3 and clock 4) before the start of data transfer. This two clock cycle delay facilitates backplane signal timing and LSM to line card synchronization. In alternative implementations, this two clock cycle delay period may be reduced or increased depending on, for example, backplane signal propagation characteristics and required LSM and line card circuitry response times. Following the two clock cycle delay period, the line card begins a serial transfer of data by modulating the LC-DATA signal. For example, FIG. 7C shows the transfer of a '1' bit during clock cycles 5, 8, 9, 11,426,427, and 428 by asserting (high) the LC-DATA signal and the transfer of a '0' bit during clock cycles 6, 7, 10, and 12 by de-asserting the LC-DATA signal. In alternative embodiments, the LSM-SR may be de-asserted once a cell transfer has begun, for example, at clock cycle 5 of FIG. 7B.”) Wiher ’530 at 29:27-40 (“The foregoing line card shelf and master control shelf descriptions are illustrative. In alternative implementations, a line card shelf may be constructed to hold up to X line cards, where X is a number that may be equal to, greater than, or lesser than the number of line card slots described herein. Line card shelves with a greater or lesser number of line card slots will have a corresponding increase or decrease in the number of LSM signal paths between the main LSM and line card slots and between the backup LSM and line card slots. Similarly, LSMs may

No.	’740 Patent Claim 5	Wiher ’530
		include a greater or lesser number of line card interfaces to correspond to an increase or decrease in the number of line card slots in a line card shelf. Similarly, a line card shelf may be constructed to hold up to Y LSMs.”) Wiher ’530 at 30:45-55 (“Clock frequencies at line card, LSM, MLA, trunk card, and MCP interfaces may differ from those described herein. For example, a MCS may include MLA control interfaces having a 256 KHz clock frequency allowing data transfer between the MCS and MLA at 256 Kbits/second and trunk cards may include cell transport interfaces having a 50 MHz clock frequency allowing data transfer between the trunk card and the MLA at 50 Mbytes/second. Clock frequencies may be varied depending on, for example, desired data transfer rates, signal propagation constraints, and circuitry response times.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Wiher ’530 in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 5[f] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, IEEE 802.3 discloses connecting a MAC Client to the aggregator by aggregating one or more links together to form a Link Aggregation Group in order to increase bandwidth. IEEE 802.3 at 1465  **43.1 Overview** This clause defines an optional Link Aggregation sublayer for use with CSMA/CD MACs. Link Aggregation allows one or more links to be aggregated together to form a Link Aggregation Group, such that a MAC Client can treat the Link Aggregation Group as if it were a single link. To this end, it specifies the establishment of DTE to DTE logical links, consisting of N parallel instances of full duplex point-to-point links operating at the same data rate. IEEE 802.3 at 1465

No.	’740 Patent Claim 5	Wiher ’530
		 **43.1.2 Goals and objectives** Link Aggregation, as specified in this clause, provides the following: a) **Increased bandwidth**—The capacity of multiple links is combined into one logical link. b) **Linearly incremental bandwidth**—Bandwidth can be increased in unit multiples as opposed to the order-of-magnitude increase available through Physical Layer technology options (10 Mb/s, 100 Mb/s, 1000 Mb/s, etc.). c) **Increased availability**—The failure or replacement of a single link within a Link Aggregation Group need not cause failure from the perspective of a MAC Client. d) **Load sharing**—MAC Client traffic may be distributed across multiple links. e) **Automatic configuration**—In the absence of manual overrides, an appropriate set of Link Aggregation Groups is automatically configured, and individual links are allocated to those groups. If a set of links can aggregate, they will aggregate. f) **Rapid configuration and reconfiguration**—In the event of changes in physical connectivity, Link Aggregation will quickly converge to a new configuration, typically on the order of 1 second or less. g) **Deterministic behavior**—Depending on the selection algorithm chosen, the configuration can be made to resolve deterministically; i.e., the resulting aggregation can be made independent of the order in which events occur, and be completely determined by the capabilities of the individual links and their physical connectivity. h) **Low risk of duplication or mis-ordering**—During both steady-state operation and link (re)configuration, there is a high probability that frames are neither duplicated nor mis-ordered. i) **Support of existing IEEE 802.3® MAC Clients**—No change is required to existing higher-layer protocols or applications to use Link Aggregation. j) **Backwards compatibility with aggregation-unaware devices**—Links that cannot take part in Link Aggregation—either because of their inherent capabilities, management configuration, or the capabilities of the devices to which they attach—operate as normal, individual IEEE 802.3® links. k) **Accommodation of differing capabilities and constraints**—Devices with differing hardware and software constraints on Link Aggregation are, to the extent possible, accommodated. l) **No change to the IEEE 802.3® frame format**—Link aggregation neither adds to, nor changes the contents of frames exchanged between MAC Clients. m) **Network management support**—The standard specifies appropriate management objects for configuration, monitoring, and control of Link Aggregation.

No.	’740 Patent Claim 5	Wiher ’530
		IEEE 802.3 at 1470  **43.2.3 Frame Collector** A Frame Collector is responsible for receiving incoming frames (i.e., AggMuxN:MA_DATA.indications) from the set of individual links that form the Link Aggregation Group (through each link’s associated Aggregator Parser/Multiplexer) and delivering them to the MAC Client. Frames received from a given port are delivered to the MAC Client in the order that they are received by the Frame Collector. Since the Frame Distributor is responsible for maintaining any frame ordering constraints, there is no requirement for the Frame Collector to perform any reordering of frames received from multiple links. IEEE 802.3 at 1471  **43.2.4 Frame Distributor** The Frame Distributor is responsible for taking outgoing frames from the MAC Client and transmitting them through the set of links that form the Link Aggregation Group. The Frame Distributor implements a distribution function (algorithm) responsible for choosing the link to be used for the transmission of any given frame or set of frames. IEEE 802.3 at 1474  **43.2.8 Aggregator** An *Aggregator* comprises an instance of a Frame Collection function, an instance of a Frame Distribution function and one or more instances of the Aggregator Parser/Multiplexer function for a Link Aggregation Group. A single Aggregator is associated with each Link Aggregation Group. An Aggregator offers a standard IEEE 802.3® MAC service interface to its associated MAC Client; access to the MAC service by a MAC Client is always achieved via an Aggregator. An Aggregator can therefore be considered to be a *logical MAC*, bound to one or more ports, through which the MAC client is provided access to the MAC service. IEEE 802.3 at 1481

No.	’740 Patent Claim 5	Wiher ’530
		 **43.3.6 Link Aggregation Group identification** A Link Aggregation Group consists of either a) One or more Aggregatable links that terminate in the same pair of Systems and whose ports belong to the same Key Group in each System, or b) An Individual link. As another example, Ghosh discloses aggregating physical links, including ports, into aggregate port channels that form a single logical link to increase bandwidth, load balancing, and link redundancy. Ghosh at Abstract (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggregation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent gen-eration of port channel map tables.”) Ghosh at [0004] (“Neighboring nodes in a fibre channel network are typically interconnected through multiple physical links. For example, a local fibre channel switch may be connected to a remote fibre channel switch through four physical links. In many instances, it may be beneficial to aggregate some of the physical links into logical links. That is, multiple physi-cal links can be combined to form a logical interface to provide higher aggregate bandwidth, load balancing, and link redundancy. When a frame is being transmitted over a logical link, it does not matter what particular physical link is being used as long as all the frames of a given flow are transmitted through the same link. If a constituent physical link goes down, the logical link can still remain operational.”) Ghosh at [0007] (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggre-gation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences

No.	’740 Patent Claim 5	Wiher ’530
		with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent generation of port channel map tables.”) Ghosh at [0008] (“In one embodiment, a method for aggregating ports in a fibre channel fabric is provided. It is determined that a plurality of local ports at a local fibre channel switch are compatible. Identifiers for the plurality of local ports are sent to a remote fibre channel switch. The remote fibre channel switch determines if a plurality of remote ports are compat-ible, the plurality of remote ports corresponding to the plurality of local ports. An indication that one or more of the remote physical ports are compatible is received. A port channel including one or more of the local ports correspond-ing to the compatible remote ports is created.”) Ghosh at [0010] (“In another embodiment, a fibre channel network is described. The fibre channel network includes a local fibre channel switch and a remote fibre channel switch. The local fibre channel switch aggregates a compatible subset of the plurality of local ports and sends identifiers for the compat-ible subset of the plurality of local ports to the remote fibre channel switch. The remote fibre channel switch determines if a subset of the plurality of remote ports are compatible. The subset of the plurality of remote ports corresponds to the compatible subset of the plurality of local ports.”) Ghosh at [0022] (“Switches in a fibre channel network are typically interconnected using multiple physical links. The physical links connecting a pair of switches allows transmission of data and control signals. In some instances, it is useful to aggregate multiple physical links into a logical link. Physi-cal links are also referred to herein as physical interfaces and channels while logical links are also referred to herein as logical interfaces and port channels. For example, a local switch may be connected to a remote switch through four physical links. Instead of having to transmit data through a particular physical link, the physical links can be aggregated to form one or more logical links. In one example, all four physical links are aggregated into a single logical link. Instead of having data transmitted through a particular physical link, the data can merely be transmitted over a particular logical link without regard to the particular physi-cal interface used. Aggregating physical links into a logical link allows for higher aggregated bandwidth, load balancing, and link

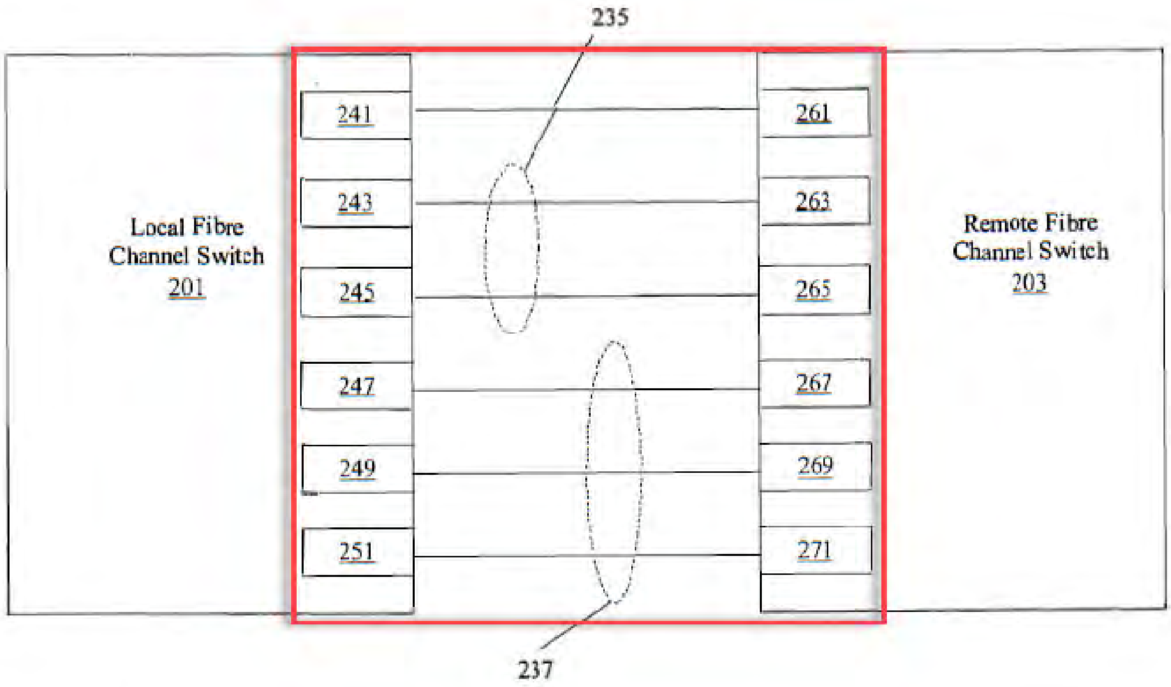
No.	’740 Patent Claim 5	Wiher ’530
		redundancy. For example, if a particular physical link fails or is overloaded, data can still be transmitted over the logical link.”) Ghosh at [0029] (“FIG. 2 is a diagrammatic representation showing links between two switches, such as two fibre channel switches shown in FIG. 1. A local fibre channel switch 201 includes local ports 241, 243, 245, 247, 249, and 251. A remote fibre channel switch 203 includes remote ports 261, 263, 265, 267, 269, and 271. Local port 241 is coupled to remote port 261 through an individual physical link or channel. Connected ports are also referred to herein as peer ports. Local port 243 is coupled to remote port 263 and local port 245 is coupled to remote port 265. The two resulting physical links are aggregated to form port channel 235. Local ports 247, 249, and 251 are coupled to remote ports 267, 269, and 271 respectively. The three resulting physical links are aggregated to form port channel 237.”) Ghosh at [0030] (“According to various embodiments, local fibre channel switch 201 and remote fibre channel switch both have associated identifiers. In some examples, the identifiers are globally unique identifiers such as a global switch world wide names (WWNs). Each local port 241, 243, 245, 247, 249, and 251 and each remote port 261,263,265,267, 269, and 271 can also be associated with identifiers. In some examples, the identifiers are port WWNs. The port WWNs are typically used for debugging or identifying the peer port in alert or warning messages. However, according to various embodiments, the techniques of the present invention use WWNs as globally unique identifiers to aggregate ports instead of using compatibility keys which are only locally unique. Compatibility keys are mechanisms typically used by other protocols such as Ethernet for aggregation.”) Ghosh at [0033] (“A variety of parameters can be used to aggregate physical ports. FIG. 3 is a flow process diagram showing one technique for aggregating physical ports into a logical port. And 301, it is determined if auto create functionality is enabled. According to various embodiments, auto create functionality allows automatic configuration and detection of compatible physical ports as well as aggregation into one or more logical ports. Auto creation does not require user intervention. In other examples, administrators can manu-ally arrange ports for aggregation.”)

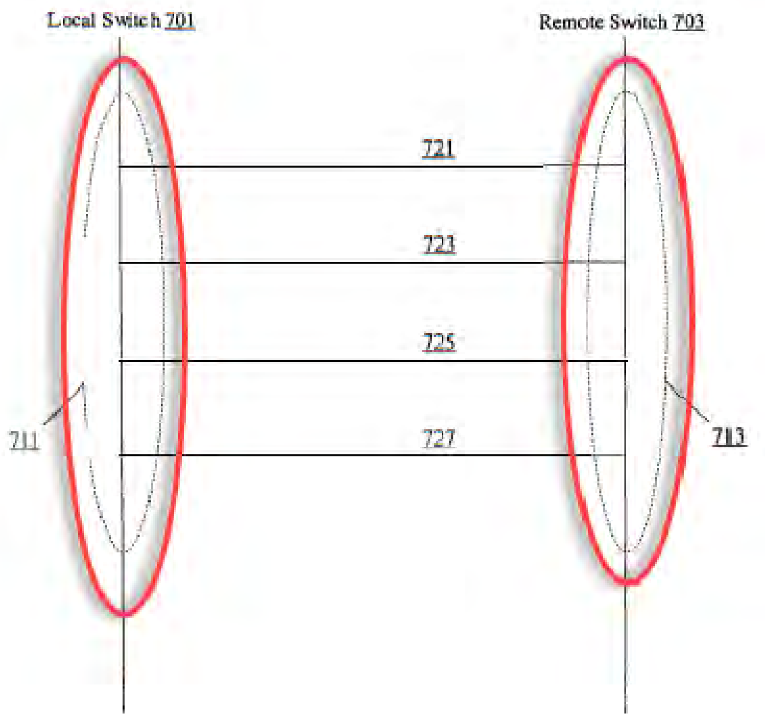
No.	’740 Patent Claim 5	Wiher ’530
		Ghosh at [0037] (“FIG. 4 is an exchange diagram showing one example of a bring up procedure used for a port creating a new port channel. A local switch 401 is coupled to a remote switch 403. The local switch 401 includes a physical port Al coupled to physical port Bl included in remote switch 403. When two peer ports Al and Bl are being aggregated into a port channel, the peer switches 401 and 403 typically already know the world wide names of the individual physical peer ports. However, the peer switches only know the world wide name of their own logical port or port channel. That is, both switches have the individual physical link configured, but the link is not yet part of a port channel. At 421, a local switch 401 sends a synchronize (sync) message 411 to the remote switch 403 to begin the process of creating a port channel including ports Al and Bl.”) Ghosh at [0038] (“In some examples, the sync message 411 includes a local port channel identifier and a remote port channel identifier. In one particular example, the local port channel identifier is set to the world wide name of the local port channel assigned by the local switch 401. The remote port channel identifier is left blank to indicate that the port Al is being aggregated as part of a new port channel. The sync message 411 can also include other parameters such as channel status, channel model, or channel intent.”) Ghosh at [0042] (“When two peer ports A2 and B2 are aggregated into a port channel Cl, the peer switches 501 and 503 typically already know the world wide names of the indi-vidual physical peer ports A2 and B2 as well as the world wide name information of the port channel Cl. Conse-quently, the port channel is already successfully established. According to various embodiments, local switch 501 and remote switch 503 perform parameter checking to ensure that the new physical port A2 and B2 can be safely added to the existing port channel Cl. At 521, a local switch can check configuration parameters to ensure that physical ports Al and A2 at the local switch 501 are compatible. The compatibility checking can be performed anytime. In some examples, compatibility checking is checked before a local switch 501 sends a synchronize (sync) message 511 to the remote switch 503 to begin the process of aggregating ports A2 and B2 into the port channel.) Ghosh at [0043] (“In some examples, the sync message 511 includes local port channel identifier and a remote port channel identifier. In one particular example, the local port channel identifier is

No.	’740 Patent Claim 5	Wiher ’530
		set to the world wide name of the local port channel assigned by the local switch 501. The remote port channel identifier is filled with the existing port channel identifier to indicate that the port A2 is being aggregated into existing port channel C2. The sync message 511 can also include other parameters such as channel status, channel model, or channel intent.”) Ghosh at [0044] (“At 531, remote switch 503 uses the information received from the local switch 501 to verify port B2 is compatible with other port in port channel C2. In one example, configuration parameters associated with B2 are checked against configuration parameters associated with Bl. The remote switch 503 can also check if the port B2 is already assigned to a different port channel. If the port B2 is compatible with port Bl, the remote switch 503 can proceed and send a sync accept message 513 in response to the sync message 511 to indicate that the port B2 can be aggregated into the port channel. The sync accept message indicates that a port channel can now be modified. At 523, local switch 501 uses the information to update its own port channel database. However, the port channel may not yet be fully operational until the hardware configuration is completed. The local switch 501 continues hardware configuration such as line card configuration to make the port A2 part of the port channel Cl. An acknowledgment 527 is sent and received by remote switch 503 at 529. In some examples, the local switch 501 sends a commit signal 515 when hardware configuration is complete.”) Ghosh at [0045] (“The remote switch 503 receives the commit signal at 533 and begins its own hardware configuration. On completion of its hardware configuration, remote switch 503 sends out a commit accept signal 517 to indicate to local switch 501 that hardware configuration is completed. According to various embodiments, local switch 501 receives the commit accept signal 517 and notifies relevant applications that the port channel is now fully operational at 525 and that port A2 has been aggregated into port channel Cl. The local switch 501 can also send out an acknowledge message 519. When the remote switch 503 receives the acknowledge, it notifies relevant applications that the port channel is operational at 535 and that port B2 has been aggregated into port channel Cl. In one embodiments, the techniques of the present invention contemplate using a two phase SYNC and COMMIT mechanism similar to the mechanism used in EPP.”)

No.	’740 Patent Claim 5	Wiher ’530
		Ghosh at [0046] (“FIGS. 4 and 5 show examples of ports being aggregated into a port channel. At a particular switch, ports can be selected for aggregation into a port channel in a variety of manners. FIG. 6 is an exchange diagram showing automatic selection of ports at a switch for aggregation into a port channel. A local switch 601 is coupled to a remote switch 603. In one example, the local switch 601 includes physical ports Al, A2, A3, and A4 while remote switch 603 includes physical ports Bl, B2, B3, and B4. No port chan-nels have been formed.”) Ghosh at [0049] (“At 631, remote switch 603 uses the information received from the local switch 601 to verify that the peer ports of Al, A2, and A4 are compatible. That is, ports Bl, B2, and B4 are checked for compatibility. In one example, only ports Bl and B2 may be compatible, and consequently only ports Al, A2, Bl, and B2 can be included in the port channel. In another example, ports Bl, B2, and B4 are compatible, so ports Al, A2, A4, Bl, B2, and B4 can be aggregated into port channel Cl. According to various embodiments, if the port B2 is compatible with port Bl, the remote switch 603 can proceed and send a sync accept message 613 in response to the sync message 611 to indicate that the port B2 can be aggregated into the port channel. It should be noted that remote switch 603 can send a list indicating that ports B2 and B4 are compatible with Bl. However, the remote switch 603 sends only one compatible port B2 back for several reasons, and in the process of selection compatible port channels get priority over com-patible individual ports.”) Ghosh at [0050] (“One reason is that aggregation mechanisms and techniques can be implemented more elegantly by handling ports on an individual basis. Any individual port will either start a new port channel, be added to an existing port channel, or operate stand alone. There is no need to keep track of groups of ports to be aggregated. Another reason is that fewer ports need to be locked if only a single port is being aggregated at any one time. The sync accept message indicates that a port channel can now be modified. At 623, local switch 601 receives the information and recognizes that Al and A2 can now be aggregated into port channel Cl. However, the port channel may not yet be fully operational until the hardware configuration is completed. An acknowl-edgment 627 is sent and received by remote switch 603 at 629. In some examples, the local switch 601 sends a commit signal 615 when hardware configuration is complete.”)

No.	’740 Patent Claim 5	Wiher ’530
		Ghosh at [0051] (“The remote switch 603 receives the commit signal at 633 to create port channel Cl including ports Bl and B2. Hardware configuration can now be performed. On comple-tion of its hardware configuration, remote switch 603 sends out a commit accept signal 617 to indicate to local switch 601 that hardware configuration is completed. According to various embodiments, local switch 601 receives the commit accept signal 617 and notifies relevant applications that the port channel is now fully operational at 625 and that ports Al and A2 have been aggregated into port channel Cl. The local switch 601 can also send out an acknowledge message 619. When the remote switch 603 receives the acknowledge, it notifies relevant applications that the port channel is fully operational at 635 and that ports Bl and B2 have been aggregated into port channel Cl.”) Ghosh at [0053] (“FIG. 7 is a diagrammatic representation showing synchronous aggregation of ports into a port channel. A local switch 701 is coupled to a remote switch 703 through links 721, 723, 725, and 727. According to various embodiments, the links are being aggregated into port channel 711 at the local switch 701 and port channel 713 at the remote switch 703 in a synchronous manner. That is the peer ports corre-sponding to each link are brought up in the same order at both the local switch 701 and the remote switch 703.”) Ghosh at Figure 2 (annotation added)

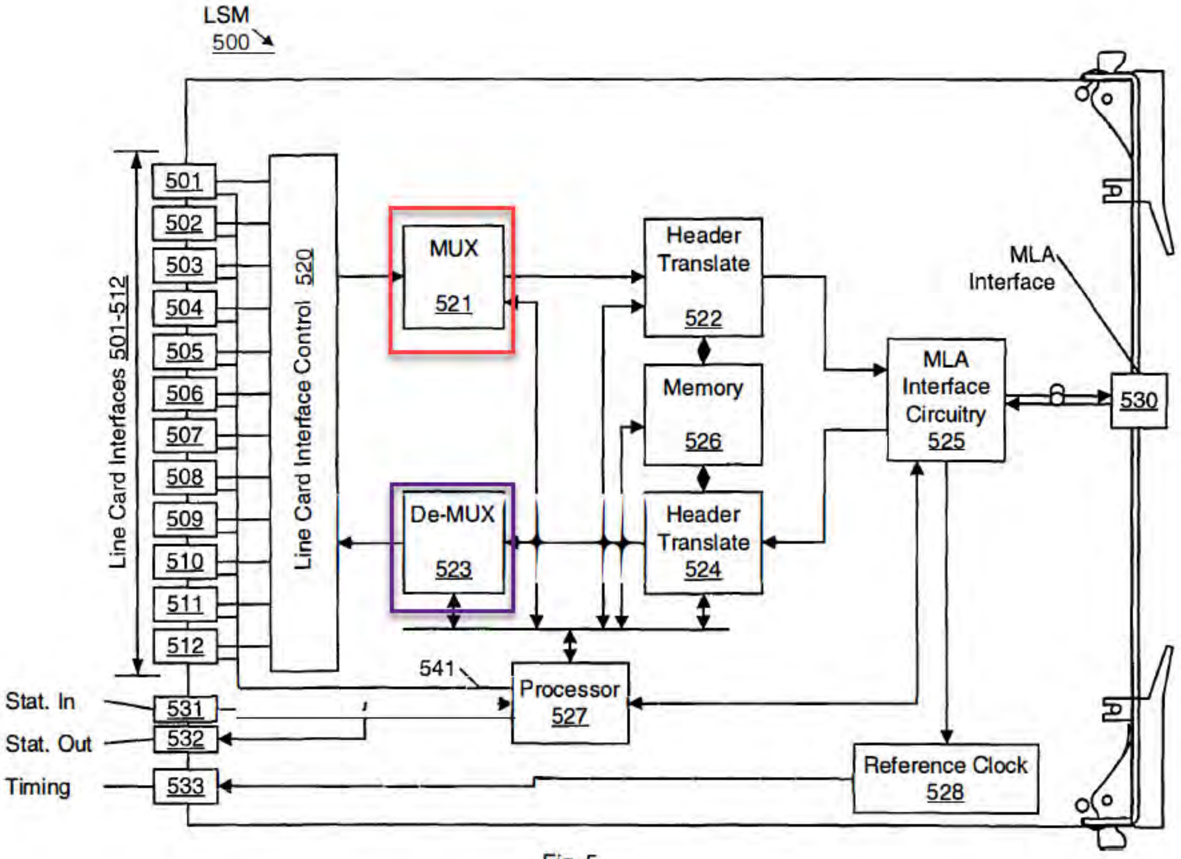
No.	’740 Patent Claim 5	Wiher ’530
		 235 241 243 245 247 249 251 261 263 265 267 269 271 Local Fibre Channel Switch 201 Remote Fibre Channel Switch 203 237 Figure 2 Ghosh at Figure 7 (annotation added)

No.	’740 Patent Claim 5	Wiher ’530
		 Local Switch 701 Remote Switch 703 721 723 725 727 711 713 Figure 7

No.	’740 Patent Claim 6	Wiher ’530
6	The method according to claim 1, wherein coupling each of the one or more interface	Wiher ’530 discloses the method according to claim 1, wherein coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and

No.	’740 Patent Claim 6	Wiher ’530
	modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node.	demultiplexing downstream data frames sent from the communication network to the network node. For example, Wiher ’530 discloses multiplexing and demultiplexing circuitry to transmit and receive ATM data cells over a data link between the ATM network and network access equipment. *See supra* Claim 1. Wiher ’530 at 3:43-65 (“In general, in another aspect, the invention features an apparatus for communicating data cells between a data link and a backplane. The apparatus includes transceiver circuitry to transmit and receive data cells over a data link and a plurality of backplane interfaces each including at least one cell signal terminal. Each of the backplane interface is coupled to a backplane interconnection circuit. Each backplane interconnection circuit transmits and receives cells over the cell signal terminals of its associated backplane interface. The apparatus also includes de-multiplexing circuitry coupling the transceiver circuitry to each of the backplane interconnection circuits. The de-multiplexing circuitry receives a data cell from the transceiver circuitry, select a backplane interconnection circuit associated with the data cell, and provide the data cell to the selected backplane interconnection circuit for transmission over the cell signal terminals of the associated backplane interface. The apparatus also includes multiplexing circuitry coupling the plurality of backplane interconnection circuits to the transceiver circuitry. The multiplexing circuitry receives data cells from each of the backplane interconnection circuits and provide the received data cells to the transceiver circuitry.”) Wiher ’530 at 6:26-42 (“Network access equipment 201-208 may combine data from multiple sources. For example, data from a LAN 250 and circuit oriented traffic, such as a Tl connection from a private branch exchange phone system (PBX) 240, may each be converted to ATM cells at network access equipment 201. ATM cells corresponding to LAN 250 and PBX 240 data are multiplexed together and sent by the network access equipment 201 over media 261 to a line card in a line card shelf 211. VPI and VCI information in transmitted ATM cells is used to uniquely identify data sources and destinations at, for example, network access equipment 201,

No.	’740 Patent Claim 6	Wiher ’530
		line card shelf 211, master control shelf 221, and within the ATM network 230. For example, by assigning a unique VPI/VCI value to ATM cells transporting LAN 250 data and different VPI/VCI value to cells transporting PBX 240 data, inde-pendent routing and logical separation of the PBX 240 and LAN 250 data can be maintained.”) Wiher ’530 at 8:34-52 (“Line card interface control circuitry 520 processes ATM cell transfer signals exchanged over interfaces 501-512 and buffers ATM cells being sent to and received from line cards. Line card interface circuitry 520 may be implemented as a single integrated circuit to process signals on all of the line card interfaces or may be implemented as separate circuit components each processing, for example, signals on a single line card interface. Cells received by the interface control circuitry 520 from a line card may be temporarily buffered by the control circuitry 520. Received cells are subsequently selected by multiplexer circuitry 521 for processing by line card circuitry 522-527. Line card interfaces 501-512 also contain signal paths 541 over which HDLC formatted control data is exchanged between the LSM and line cards. Signals exchanged between the interface circuitry 520 and each of the line card interfaces 501-512 correspond to signals exchanged by the line card over signal paths 611 and 612 to a main LSM or over 621 and 622 to a backup LSM (FIG. 6).”) Wiher ’530 at 9:10-27 (“The LSM's MLA interface circuitry 525 may also receive ATM cells from a MLA. The MLA interface circuitry 525 may extract OAMP cells arriving from a MLA and send them to the processor 527. Data cells arriving at the interface 525 and destined to a line card are sent to header translation circuitry 524 which may perform ATM cell header manipu-lations. Header translation circuitry 524 performs VPINCI header field translation and other ATM cell header manipu-lation functions. Header translation circuitry 524 may deter-mine appropriate header manipulations based on, for example, programs and translation tables stored in RAM and ROM memory 526. Following header translation, data cells may be sent to de-multiplexer circuitry 523. The de-multiplexer circuitry controls the flow of ATM cells to interface control circuitry 520. Processor 527 may also send OAMP cells to the de-multiplexer 523 for transfer to line cards. Interface control circuitry 520 then transmits the ATM cell to a line card.”) Wiher ’530 at Figure 5 (annotations added)

No.	’740 Patent Claim 6	Wiher ’530
		 LSM 500 Line Card Interfaces 501-512 501 502 503 504 505 506 507 508 509 510 511 512 Line Card Interface Control 520 MUX 521 De-MUX 523 Header Translate 522 Memory 526 Header Translate 524 MLA Interface MLA Interface Circuitry 525 530 541 Processor 527 Stat. In Stat. Out Timing 531 532 533 Reference Clock 528 Fig. 5 Wiher ’530 at 9:43-10:3 (“FIG. 6 is a functional diagram of a line card. A line card 600 provides a signal termination point for subscriber loop data link connections. A line card 600 has a main LSM interface 610, a backup LSM interface 620, a LSM status input interface 631, and a subscriber loop data link interface 632. Each interface 610, 620, 631, 632 includes one or more signal lines over which electrically modulated signals are exchanged. To process signals on

No.	’740 Patent Claim 6	Wiher ’530
		interfaces 610, 620, 631, and 632, a line card 600 includes line card circuitry such as a processor 645, line card to LSM backplane interface circuitry 647, and transceiver circuitry 643 and 644. The processor 645 may include integrated memory storage or may include interfaces to memory 646. The processor 645 may control line card to LSM communications, power management for line card circuitry, line card initialization, operations, maintenance, and provisioning. The processor 645 is, for example, a Motorola MC68360 processor. Back-plane interface circuitry 647 receives and transmits signals over the line card shelf backplane 400, multiplexes and de-multiplexes ATM cells exchanged between the LSM and line card transceivers 643, 644 and may buffer cell traffic in memory 646. The backplane interface circuitry 647 is imple-mented using, for example, an Altera FLEX l0K program-mable logic device, a field-programmable gate array, or other processing circuitry. Backplane interface circuitry 647 is coupled to transceiver circuitry 643, 644. Transceivers 643, 644 provide for modulation and demodulation of data over the subscriber loop interface 632 using a digital data modulation technology.”) Wiher ’530 at 14:1-21 (“In an exemplary implementation, line cards and line shelf multiplexers may be interconnected by a line card shelf backplane that includes twenty-four line card slots and two LSM slots. Each line card slot may include electrical connectors manufactured by FCI/Burndy as part number HMl W53DPR000H9. Line card slots receive corresponding line card mating connectors. Each LSM slot may include two HMl W53DPR000H9 connectors and one HMl W52DPR000H9 connector and receive LSM cards having three corresponding mating connectors. Line card shelf backplane interconnections for this exemplary imple-mentation are detailed in Table 1. In Table 1, the main LSM slot includes connectors labeled L325, L425, and L525, the backup LSM slot includes connectors labeled L326, L426, and L526, and the twenty-four line card slots include con-nectors labeled L201 through L224, respectively. Connec-tors L325, L425, L326, L426, and L201 through L224 have, for example, FCI/Burndy connectors having 120 electrical contacts ("pins") and part number HM1W53DPR000H9. Connector L525 and L526 are FCI/Burndy connectors hav-ing 60 pins and part number HM1W52DPR000H9.”) Wiher ’530 at 18:33-60 (“A trunk card includes, for example, circuitry 1122-1128 to process signals on interfaces 1101-1106 and on interfaces 1129-1135. MLA interface control circuitry 1122 processes ATM cell transfer signals exchanged over interfaces 1101-1106 and controls the transmission of ATM cells between the trunk card and MLAs. Interface control cir-cuitry

No.	’740 Patent Claim 6	Wiher ’530
		1122 may include ATM cell buffers to temporarily store cells received from, or being transmitted to, MLAs over interfaces 1101-1106. Cell multiplexer/de-multiplexer circuitry 1123 exchanges ATM cells with the MLA interface control circuitry 1122 and determines the flow of ATM cells between the control circuitry 1122 and header translation circuitry 1124. Additionally, the multiplexer/de-multiplexer circuitry 1123 may extract OAMP cells arriving from the trunk interface 1125 and direct them to processor 1128 and insert OAMP cells destined to the trunk interface 1125 from the processor 1128. Header translation circuitry 1124 translates header information in ATM cells arriving from or destined to the trunk interface 1125. Header translation circuitry 1124 may access header translation programs and data stored in RAM and ROM memory 1127. For example, processor 1128 may store VPI/VCI header field translation information in memory 1127 for use by header translation circuitry 1124. Header translation circuitry 1124 exchanges ATM cells with trunk interface circuitry 1125. Trunk inter-face circuitry 1125 provides a trunk connection to an ATM network. The trunk connection is, for example, a standard 45 Megabit per second T3 trunk connection.”)

No.	’740 Patent Claim 7	Wiher ’530
7	The method according to claim 1, wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the first and second physical links.	Wiher ’530 discloses the method according to claim 1, wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the first and second physical links. For example, Wiher ’530 discloses balancing a ATM cell transmission rates based on clock frequencies at various interfaces in which first and second links are selected. Thus a person of ordinary skill in the art would understand that delaying and altering the timing of transmission is in an effort to balance a frame data rate. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Wiher ’530 is found to not meet this limitation, wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the first and second physical links would have been obvious to a person having ordinary skill in the art, as explained below.

No.	’740 Patent Claim 7	Wiher ’530
		See supra Claim 1. Wiher ’530 at 30:45-55 (“Clock frequencies at line card, LSM, MLA, trunk card, and MCP interfaces may differ from those described herein. For example, a MCS may include MLA control interfaces having a 256 KHz clock frequency allowing data transfer between the MCS and MLA at 256 Kbits/second and trunk cards may include cell transport interfaces having a 50 MHz clock frequency allowing data transfer between the trunk card and the MLA at 50 Mbytes/second. Clock frequencies may be varied depending on, for example, desired data transfer rates, signal propagation constraints, and circuitry response times.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Wiher ’530 in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 7 of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Ghosh discloses aggregating physical links, including ports, into aggregate port channels that form a single logical link to increase bandwidth, load balancing, and link redundancy. Ghosh at Abstract (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggregation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent gen-eration of port channel map tables.”) Ghosh at [0004] (“Neighboring nodes in a fibre channel network are typically interconnected through multiple physical links. For example, a local fibre channel switch may be connected to a remote fibre channel switch through four physical links. In many instances, it may be beneficial to aggregate some of the physical links into logical links. That is, multiple physi-cal links can be combined to form a logical interface to provide higher aggregate bandwidth, load

No.	’740 Patent Claim 7	Wiher ’530
		balancing, and link redundancy. When a frame is being transmitted over a logical link, it does not matter what particular physical link is being used as long as all the frames of a given flow are transmitted through the same link. If a constituent physical link goes down, the logical link can still remain operational.”) Ghosh at [0007] (“According to the present invention, methods and apparatus are provided to allow efficient and effective aggre-gation of ports into port channels in a fibre channel network. A local fibre channel switch can automatically identify compatible ports and initiate exchange sequences with a remote fibre channel switch to aggregate ports into port channels. Ports can be aggregated synchronously to allow consistent generation of port channel map tables.”) Ghosh at [0008] (“In one embodiment, a method for aggregating ports in a fibre channel fabric is provided. It is determined that a plurality of local ports at a local fibre channel switch are compatible. Identifiers for the plurality of local ports are sent to a remote fibre channel switch. The remote fibre channel switch determines if a plurality of remote ports are compat-ible, the plurality of remote ports corresponding to the plurality of local ports. An indication that one or more of the remote physical ports are compatible is received. A port channel including one or more of the local ports correspond-ing to the compatible remote ports is created.”) Ghosh at [0010] (“In another embodiment, a fibre channel network is described. The fibre channel network includes a local fibre channel switch and a remote fibre channel switch. The local fibre channel switch aggregates a compatible subset of the plurality of local ports and sends identifiers for the compat-ible subset of the plurality of local ports to the remote fibre channel switch. The remote fibre channel switch determines if a subset of the plurality of remote ports are compatible. The subset of the plurality of remote ports corresponds to the compatible subset of the plurality of local ports.”) Ghosh at [0022] (“Switches in a fibre channel network are typically interconnected using multiple physical links. The physical links connecting a pair of switches allows transmission of data and control signals. In some instances, it is useful to aggregate multiple physical links into a logical link. Physi-cal links are also referred to herein as physical interfaces and channels

No.	’740 Patent Claim 7	Wiher ’530
		while logical links are also referred to herein as logical interfaces and port channels. For example, a local switch may be connected to a remote switch through four physical links. Instead of having to transmit data through a particular physical link, the physical links can be aggregated to form one or more logical links. In one example, all four physical links are aggregated into a single logical link. Instead of having data transmitted through a particular physical link, the data can merely be transmitted over a particular logical link without regard to the particular physi-cal interface used. Aggregating physical links into a logical link allows for higher aggregated bandwidth, load balancing, and link redundancy. For example, if a particular physical link fails or is overloaded, data can still be transmitted over the logical link.”) Ghosh at [0029] (“FIG. 2 is a diagrammatic representation showing links between two switches, such as two fibre channel switches shown in FIG. 1. A local fibre channel switch 201 includes local ports 241, 243, 245, 247, 249, and 251. A remote fibre channel switch 203 includes remote ports 261, 263, 265, 267, 269, and 271. Local port 241 is coupled to remote port 261 through an individual physical link or channel. Connected ports are also referred to herein as peer ports. Local port 243 is coupled to remote port 263 and local port 245 is coupled to remote port 265. The two resulting physical links are aggregated to form port channel 235. Local ports 247, 249, and 251 are coupled to remote ports 267, 269, and 271 respectively. The three resulting physical links are aggregated to form port channel 237.”) Ghosh at [0030] (“According to various embodiments, local fibre channel switch 201 and remote fibre channel switch both have associated identifiers. In some examples, the identifiers are globally unique identifiers such as a global switch world wide names (WWNs). Each local port 241, 243, 245, 247, 249, and 251 and each remote port 261,263,265,267, 269, and 271 can also be associated with identifiers. In some examples, the identifiers are port WWNs. The port WWNs are typically used for debugging or identifying the peer port in alert or warning messages. However, according to various embodiments, the techniques of the present invention use WWNs as globally unique identifiers to aggregate ports instead of using compatibility keys which are only locally unique. Compatibility keys are mechanisms typically used by other protocols such as Ethernet for aggregation.”)

No.	’740 Patent Claim 7	Wiher ’530
		Ghosh at [0033] (“A variety of parameters can be used to aggregate physical ports. FIG. 3 is a flow process diagram showing one technique for aggregating physical ports into a logical port. And 301, it is determined if auto create functionality is enabled. According to various embodiments, auto create functionality allows automatic configuration and detection of compatible physical ports as well as aggregation into one or more logical ports. Auto creation does not require user intervention. In other examples, administrators can manu-ally arrange ports for aggregation.”) Ghosh at [0037] (“FIG. 4 is an exchange diagram showing one example of a bring up procedure used for a port creating a new port channel. A local switch 401 is coupled to a remote switch 403. The local switch 401 includes a physical port Al coupled to physical port Bl included in remote switch 403. When two peer ports Al and Bl are being aggregated into a port channel, the peer switches 401 and 403 typically already know the world wide names of the individual physical peer ports. However, the peer switches only know the world wide name of their own logical port or port channel. That is, both switches have the individual physical link configured, but the link is not yet part of a port channel. At 421, a local switch 401 sends a synchronize (sync) message 411 to the remote switch 403 to begin the process of creating a port channel including ports Al and Bl.”) Ghosh at [0038] (“In some examples, the sync message 411 includes a local port channel identifier and a remote port channel identifier. In one particular example, the local port channel identifier is set to the world wide name of the local port channel assigned by the local switch 401. The remote port channel identifier is left blank to indicate that the port Al is being aggregated as part of a new port channel. The sync message 411 can also include other parameters such as channel status, channel model, or channel intent.”) Ghosh at [0042] (“When two peer ports A2 and B2 are aggregated into a port channel Cl, the peer switches 501 and 503 typically already know the world wide names of the indi-vidual physical peer ports A2 and B2 as well as the world wide name information of the port channel Cl. Conse-quently, the port channel is already successfully established. According to various embodiments, local switch 501 and remote switch 503 perform parameter checking to ensure that the new physical port A2 and B2 can be safely added to the existing port channel Cl. At

No.	’740 Patent Claim 7	Wiher ’530
		521, a local switch can check configuration parameters to ensure that physical ports Al and A2 at the local switch 501 are compatible. The compatibility checking can be performed anytime. In some examples, compatibility checking is checked before a local switch 501 sends a synchronize (sync) message 511 to the remote switch 503 to begin the process of aggregating ports A2 and B2 into the port channel.) Ghosh at [0043] (“In some examples, the sync message 511 includes local port channel identifier and a remote port channel identifier. In one particular example, the local port channel identifier is set to the world wide name of the local port channel assigned by the local switch 501. The remote port channel identifier is filled with the existing port channel identifier to indicate that the port A2 is being aggregated into existing port channel C2. The sync message 511 can also include other parameters such as channel status, channel model, or channel intent.”) Ghosh at [0044] (“At 531, remote switch 503 uses the information received from the local switch 501 to verify port B2 is compatible with other port in port channel C2. In one example, configuration parameters associated with B2 are checked against configuration parameters associated with Bl. The remote switch 503 can also check if the port B2 is already assigned to a different port channel. If the port B2 is compatible with port Bl, the remote switch 503 can proceed and send a sync accept message 513 in response to the sync message 511 to indicate that the port B2 can be aggregated into the port channel. The sync accept message indicates that a port channel can now be modified. At 523, local switch 501 uses the information to update its own port channel database. However, the port channel may not yet be fully operational until the hardware configuration is completed. The local switch 501 continues hardware configuration such as line card configuration to make the port A2 part of the port channel Cl. An acknowledgment 527 is sent and received by remote switch 503 at 529. In some examples, the local switch 501 sends a commit signal 515 when hardware configuration is complete.”) Ghosh at [0045] (“The remote switch 503 receives the commit signal at 533 and begins its own hardware configuration. On completion of its hardware configuration, remote switch 503 sends out a commit accept signal 517 to indicate to local switch 501 that hardware

No.	’740 Patent Claim 7	Wiher ’530
		configuration is completed. According to various embodiments, local switch 501 receives the commit accept signal 517 and notifies relevant applications that the port channel is now fully operational at 525 and that port A2 has been aggregated into port channel Cl. The local switch 501 can also send out an acknowledge message 519. When the remote switch 503 receives the acknowledge, it notifies relevant applications that the port channel is operational at 535 and that port B2 has been aggregated into port channel Cl. In one embodiments, the techniques of the present invention contemplate using a two phase SYNC and COMMIT mechanism similar to the mechanism used in EPP.”) Ghosh at [0046] (“FIGS. 4 and 5 show examples of ports being aggregated into a port channel. At a particular switch, ports can be selected for aggregation into a port channel in a variety of manners. FIG. 6 is an exchange diagram showing automatic selection of ports at a switch for aggregation into a port channel. A local switch 601 is coupled to a remote switch 603. In one example, the local switch 601 includes physical ports Al, A2, A3, and A4 while remote switch 603 includes physical ports Bl, B2, B3, and B4. No port chan-nels have been formed.”) Ghosh at [0049] (“At 631, remote switch 603 uses the information received from the local switch 601 to verify that the peer ports of Al, A2, and A4 are compatible. That is, ports Bl, B2, and B4 are checked for compatibility. In one example, only ports Bl and B2 may be compatible, and consequently only ports Al, A2, Bl, and B2 can be included in the port channel. In another example, ports Bl, B2, and B4 are compatible, so ports Al, A2, A4, Bl, B2, and B4 can be aggregated into port channel Cl. According to various embodiments, if the port B2 is compatible with port Bl, the remote switch 603 can proceed and send a sync accept message 613 in response to the sync message 611 to indicate that the port B2 can be aggregated into the port channel. It should be noted that remote switch 603 can send a list indicating that ports B2 and B4 are compatible with Bl. However, the remote switch 603 sends only one compatible port B2 back for several reasons, and in the process of selection compatible port channels get priority over com-patible individual ports.”) Ghosh at [0050] (“One reason is that aggregation mechanisms and techniques can be implemented more elegantly by handling ports on an individual basis. Any individual port will either start a new port channel, be added to an existing port channel, or operate stand alone.

No.	’740 Patent Claim 7	Wiher ’530
		There is no need to keep track of groups of ports to be aggregated. Another reason is that fewer ports need to be locked if only a single port is being aggregated at any one time. The sync accept message indicates that a port channel can now be modified. At 623, local switch 601 receives the information and recognizes that Al and A2 can now be aggregated into port channel Cl. However, the port channel may not yet be fully operational until the hardware configuration is completed. An acknowl-edgment 627 is sent and received by remote switch 603 at 629. In some examples, the local switch 601 sends a commit signal 615 when hardware configuration is complete.”) Ghosh at [0051] (“The remote switch 603 receives the commit signal at 633 to create port channel Cl including ports Bl and B2. Hardware configuration can now be performed. On comple-tion of its hardware configuration, remote switch 603 sends out a commit accept signal 617 to indicate to local switch 601 that hardware configuration is completed. According to various embodiments, local switch 601 receives the commit accept signal 617 and notifies relevant applications that the port channel is now fully operational at 625 and that ports Al and A2 have been aggregated into port channel Cl. The local switch 601 can also send out an acknowledge message 619. When the remote switch 603 receives the acknowledge, it notifies relevant applications that the port channel is fully operational at 635 and that ports Bl and B2 have been aggregated into port channel Cl.”) Ghosh at [0053] (“FIG. 7 is a diagrammatic representation showing synchronous aggregation of ports into a port channel. A local switch 701 is coupled to a remote switch 703 through links 721, 723, 725, and 727. According to various embodiments, the links are being aggregated into port channel 711 at the local switch 701 and port channel 713 at the remote switch 703 in a synchronous manner. That is the peer ports corre-sponding to each link are brought up in the same order at both the local switch 701 and the remote switch 703.”) As another example, Hilla discloses selecting ports and connectors involving balancing data flow queues among the ports and connectors based on use and availability. Hilla at [0017] (“Even if the mapping of data flows to ports is uniform, underutilization of a bundle can occur. For example, consider the situation in which two packet flows include

No.	’740 Patent Claim 7	Wiher ’530
		10,000 data packets and 10 other packet flows each include 100 packets, all packets of the same size. If there are three communication links in the bundle, then the fixed-mapping process is likely to send four packet flows to each of the three ports connected to corresponding communica-tion links in the bundle. Two of the three ports might carry 10,300 data packets, the third will carry 400 data packets. The first two ports might become overused even while the third port is underused, leading to a reduction in the rate at which the 10,300 data packets are sent over the first and second communication links. The bundle as a whole per-forms at a rate less than its advertised capability. The situation could be even worse if both large data flows are sent over the same port; then as many as 20,200 packets are sent over the first port while the second and third ports carry only 400 packets each. It would likely be preferable to send one flow of 10,000 data packets over each of the first two ports and ten flows totaling 1,000 data packets over the third port.”) Hilla at [0033] (“A method and apparatus are described for dynamic balancing of data packet traffic loads over a link bundle in a network. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present inven-tion. It will be apparent, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.”) Hilla at [0041] (“The egress line card 230 includes a switching application specific integrated circuit (ASIC) 232, a traffic manager block 234, a bank 236 of physical ports, at least some of which are connected to the communication links in the link bundle. In the illustrated embodiment, a dynamic load balancing (DLB) block 240 is included in the traffic manager 234. In some embodiments, the switching ASIC 232, traffic manager 234 and ports bank 236 are standard components of conventional egress cards for link bundles, and the DLB block 240 is external to the traffic manager 234.”) Hilla at [0045] (“According to various embodiments, dynamic load balancing (DLB) block 240 selects the next packet from the data flow queues and directs the next data packet to one of the ports 238 in ports bank 236 based on the state of one or more ports 238 in the ports bank

No.	’740 Patent Claim 7	Wiher ’530
		236, or time gaps between packets of the same flow, or both. In the illustrated embodi-ment, DLB block 240 includes logic for a port selection process 242 (also called an arbiter process), logic for a port status update process 243, logic for an aging process 244, and logic for a queue selection process 248, all of which are described in greater detail in a later section.”) Hilla at [0047] (“In the illustrated embodiment, the data flow queues 235 include a mapping of blocked data flow queues called a blocked queue map 245. The blocked queue map holds data that indicates whether data packets from a particular queue are directed to a port with a status unsuitable for rapidly placing that packet on the corresponding physical link. The blocked queue map 245 is manipulated by the dynamic load balancing block 240, as described in more detail below. Thus, in the illustrated embodiment, the data flow queues 235 are depicted in the DLB block 240. In other embodi-ments, the blocked queue map 245 is separate from the data flow queues 235, and the data flow queues are in the traffic manager 234 but external to the DLB block 240.”) Hilla at [0065] (“FIG. 4A is a flow diagram that illustrates a method 400 for dynamically balancing data packet traffic load on a link bundle, according to an embodiment. Although steps are shown in FIG. 4A and subsequent flow diagrams 4B and 5 in a particular order for purposes of illustration, in other embodiments one or more steps are performed in a different order or overlapping in time or are omitted, or changed in some combination of ways. For example, in a preferred embodiment shown in FIG. 4B, step 490 to advance age counters is performed in a separate aging process 244 on DLB block 240, and thus overlaps in time any of the other steps depicted in FIG. 4A.”) Hilla at [0091] (“In some embodiments, step 450 determines that the current port is underused if the port is indicated to be the port with the smallest value in the commit depth field 347. In some of these embodiments, a status update process 243 continually cycles through the records of the flow control status table 344 and updates the contents of the fill level field 346, and determines the one port with the smallest value in the commit depth field 347 at any one time. The port# with the smallest commit depth is reported in a particular register where it is accessed during step 450. For example, using the information in Table 1, port #4 is the underused port to which the next data flow is directed. As various queues are assigned to

No.	’740 Patent Claim 7	Wiher ’530
		various ports, the values in the commit depth fields 347 change, and the port with the smallest commit depth changes. The commit depth value for a port buffer is increased whenever a flow queue is assigned to a port. That port's commit depth is the sum of all the packets assigned to that particular port. In the some embodiments, the commit depth is reduced as packets are played out of the flow queues an onto the port buffer. Thus, as the flow queue depth decreases, the commit depth value follows and also decreases. The update of the commit depth occurs when the flow queue update occurs; so decreases ripple to the commit depth from the flow queue. If the flow queue ages out, then the commit depth also gets adjusted, as described below in more detail with reference to step 499 in FIG. 4B. In some embodiments, the value in the commit depth field 347 is decremented as data packets are moved out of the port buffer onto the corresponding port.”) Hilla at [0108] (“As described above, the port selection process directs a data packet to a particular port. The port selection process does not determine the data packet that it receives; but reacts to the data packet sent. The selection of the next data packet is a function of the traffic manager. In some embodiments, the DLB block includes a queue selection process 248 in the traffic manager 234 to determine which data flow queue should be the source of the next data packet directed to a port.”) Hilla at [0109] (“In some embodiments, the traffic manager 234 determines not to send data packets from data flow queues that are directed to ports that are not suited to accepting more data at the current time. For example, in some embodiments DLB block 240 is included in traffic manager 234, so that traffic manager 234 has access to flow control status table 344. A queue selection process 248 in the traffic manager determines not to send any further data packets from flows directed to satisfied port buffers, until those status bits change to reflect a hungry or starving port buffer. This process is described further in FIG. 5.”) Hilla at [0110] (“FIG. 5 is a flow diagram that illustrates a queue selection process 500 in a traffic manager of an egress line card, according to an embodiment. Prior to step 510, data is received for a packet flow, and placed in the data packets field 338 of a data flow queue 332 of the data flow queues 235, as in a standard traffic manager. In the illustrated embodiment, a data queue exists in queues 235 for each flow ID. In some embodiments, the flow ID is one of

No.	’740 Patent Claim 7	Wiher ’530
		a particular number of hash values, as described above. The data packet is placed in the queue associated with the flow ID of the packet. Processing continues at step 510.”)

No.	’740 Patent Claim 8	Wiher ’530
8	The method according to claim 1, wherein selecting the first and second physical links comprises applying a mapping function to the at least one of the frame attributes.	Wiher ’530 discloses the method according to claim 1, wherein selecting the first and second physical links comprises applying a mapping function to the at least one of the frame attributes. For example, Wiher ’530 discloses header translation circuitry to process ATM cells then direct them to different destinations, including system physical links, based on the ATM cell header. *See supra* Claim 1. Wiher ’530 at 8:53-9:9 (“Cells arriving at the interface control circuitry 520 may include operations, administration, maintenance, and provisioning (OAMP) data or may contain user-data. OAMP cells may be identified by the payload type indicator (PTI) field in the ATM cell header. The multiplexer circuitry 521 extracts OAMP cells and send them to the processor 527 while user-data cells are sent to header translation circuitry 522. Header translation circuitry 522 performs VPI/VCI header field translation and other ATM cell header manipulation functions. Header translation circuitry 522 may deter-mine appropriate header manipulations based on, for example, programs and translation tables stored in RAM and ROM memory 526. Memory 526 may include header manipulation programs and translation tables that are stored by processor 527. After processing by header translation circuitry 522, ATM cells are directed to master line shelf adapter (MLA) interface circuitry 525. MLA interface cir-cuitry 525 controls and buffers cells flowing from the address translation circuitry 522 to a MLA and controls, for example, ATM cell flow over a SO NET OC-3c interface 530 between the LSM 500 and a MLA. Interface circuitry 525 may also insert OAMP cells from the processor 527 for transmission to a MLA and extract OAMP cells received from a MLA.”)

No.	’740 Patent Claim 8	Wiher ’530
		Wiher ’530 at 9:10-27 (“The LSM's MLA interface circuitry 525 may also receive ATM cells from a MLA. The MLA interface circuitry 525 may extract OAMP cells arriving from a MLA and send them to the processor 527. Data cells arriving at the interface 525 and destined to a line card are sent to header translation circuitry 524 which may perform ATM cell header manipu-lations. Header translation circuitry 524 performs VPINCI header field translation and other ATM cell header manipu-lation functions. Header translation circuitry 524 may deter-mine appropriate header manipulations based on, for example, programs and translation tables stored in RAM and ROM memory 526. Following header translation, data cells may be sent to de-multiplexer circuitry 523. The de-multiplexer circuitry controls the flow of ATM cells to interface control circuitry 520. Processor 527 may also send OAMP cells to the de-multiplexer 523 for transfer to line cards. Interface control circuitry 520 then transmits the ATM cell to a line card.”) Wiher ’530 at 11:20-39 (“Data transfers from a LSM to a line card include line card port identification information (a "port address"). A port address is a fixed value associated with a particular line card transceiver or subscriber loop connection. For example, a line card supporting two subscriber loops has port addresses "Pl" and "P2" that are associated, respectively, with the first and second subscriber loop at the line card. Each subscriber loop at a particular line card has an associated port address that is unique with respect to the port addresses of other subscriber loops at that line card. However, port addresses at one line card need not be unique with respect to port addresses at another line card. In a data transfer between a LSM and a line card, a line card port address may be identified by, for example, additional data bytes added to the ATM cell or by information in a modified (non-standard) cell header. Unlike VPINCI addresses which are dynami-cally associated with a transceiver, port address are perma-nently assigned (that is, they are static). Thus, the use of port addresses can simplify cell routing through a line card by simplifying the processing and storage of cell routing data.”) Wiher ’530 at 12:64-13:26 (“Referring to FIGS. 6 and 8, a line card's main LSM interface 610 includes control link signal lines 613. Line card operations, administration, maintenance and provision-ing (OAMP) functions can be controlled by data sent over the control link signal lines 613 between the line card and the main LSM. The control link signal lines 613 include a

No.	’740 Patent Claim 8	Wiher ’530
		clock signal line, a data receive signal line, and a data transmit signal line. The line card sends data to the LSM in serial fashion by modulating a signal over a data transmit signal line and receives modulated data from the LSM over a data receive signal line. Signals exchanged over the data receive and data transmit signal lines are, for example, asserted or de-asserted on the falling edge of a 64 kilohertz (KHz) clock pulse received on the clock receive signal line and are sampled on the rising edge of a received clock pulse. The format for the data exchanged on the control link signal lines 613 may conform to the Open Systems Interconnection (OSI) High-level Data Link Control (HDLC) protocol. The HDLC protocol is described in ISO/IEC 3309:1991 (E), Information Technology-Telecommunications and Information Exchange Between Systems-High-level Data Link Control (HDLC) procedures-Frame Structure, Interna-tional Organization for Standardization, Fourth Edition, 1991-06-01. A line card's main LSM interface 610 may also include clock signal lines 614. The clock signal lines include, for example, a 12.5 MHz clock signal line and a 8 KHz telephone network reference timing signal line received from the main LSM. Signals exchanged over clock signal lines 614 may be used to time data transmission over signal lines 611 and 612. Wiher ’530 at Table 1

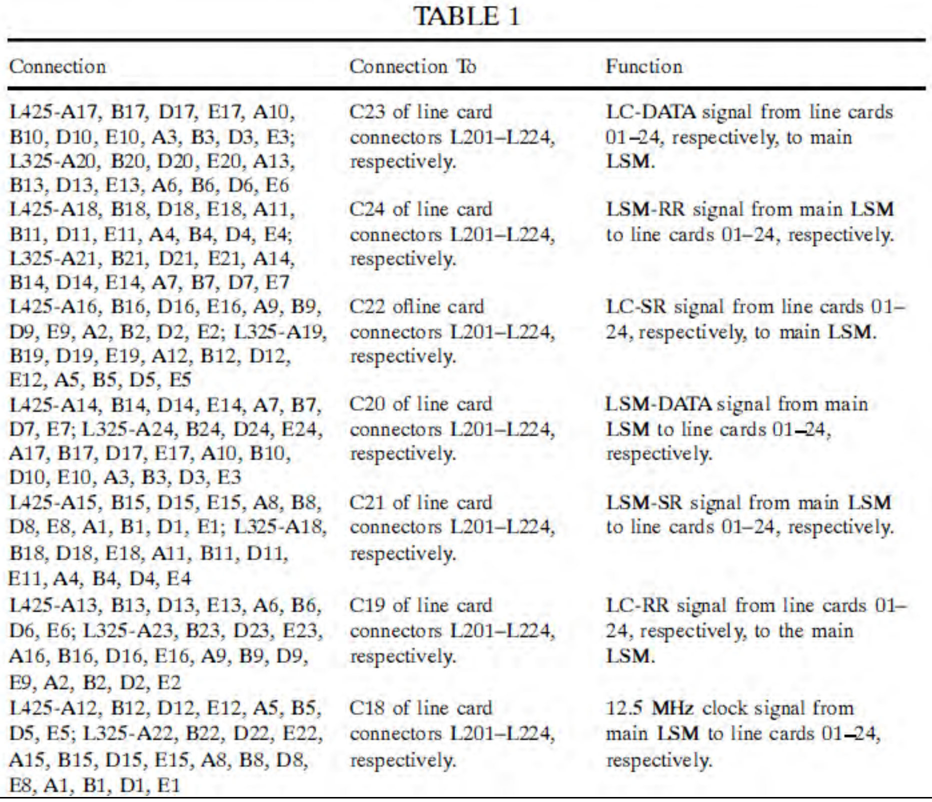
No.	’740 Patent Claim 8	Wiher ’530
		

TABLE 1

Connection	Connection To	Function
L425-A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3; L325-A20, B20, D20, E20, A13, B13, D13, E13, A6, B6, D6, E6	C23 of line card connectors L201–L224, respectively.	LC-DATA signal from line cards 01–24, respectively, to main LSM.
L425-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4; L325-A21, B21, D21, E21, A14, B14, D14, E14, A7, B7, D7, E7	C24 of line card connectors L201–L224, respectively.	LSM-RR signal from main LSM to line cards 01–24, respectively.
L425-A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2; L325-A19, B19, D19, E19, A12, B12, D12, E12, A5, B5, D5, E5	C22 ofline card connectors L201–L224, respectively.	LC-SR signal from line cards 01–24, respectively, to main LSM.
L425-A14, B14, D14, E14, A7, B7, D7, E7; L325-A24, B24, D24, E24, A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3	C20 of line card connectors L201–L224, respectively.	LSM-DATA signal from main LSM to line cards 01–24, respectively.
L425-A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1; L325-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4	C21 of line card connectors L201–L224, respectively.	LSM-SR signal from main LSM to line cards 01–24, respectively.
L425-A13, B13, D13, E13, A6, B6, D6, E6; L325-A23, B23, D23, E23, A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2	C19 of line card connectors L201–L224, respectively.	LC-RR signal from line cards 01–24, respectively, to the main LSM.
L425-A12, B12, D12, E12, A5, B5, D5, E5; L325-A22, B22, D22, E22, A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1	C18 of line card connectors L201–L224, respectively.	12.5 MHz clock signal from main LSM to line cards 01–24, respectively.

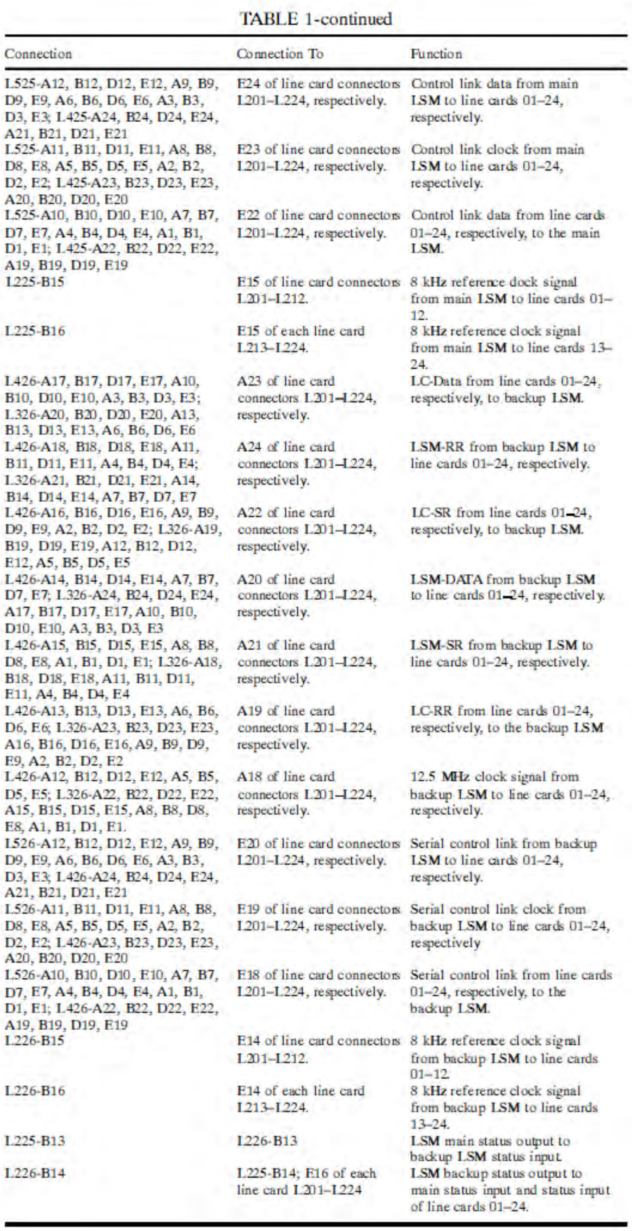


TABLE 1-continued

Connection	Connection To	Function
L525-A12, B12, D12, E12, A9, B9, D9, E9, A6, B6, D6, E6, A3, B3, D3, E3; L425-A24, B24, D24, E24, A21, B21, D21, E21	E24 of line card connectors L201–L224, respectively.	Control link data from main **LSM** to line cards 01–24, respectively.
L525-A11, B11, D11, E11, A8, B8, D8, E8, A5, B5, D5, E5, A2, B2, D2, E2; L425-A23, B23, D23, E23, A20, B20, D20, E20	E23 of line card connectors L201–L224, respectively.	Control link clock from main **LSM** to line cards 01–24, respectively.
L525-A10, B10, D10, E10, A7, B7, D7, E7, A4, B4, D4, E4, A1, B1, D1, E1; L425-A22, B22, D22, E22, A19, B19, D19, E19	E22 of line card connectors L201–L224, respectively.	Control link data from line cards 01–24, respectively, to the main LSM.
L225-B15	E15 of line card connectors L201–L212.	8 kHz reference dock signal from main **LSM** to line cards 01–12.
L225-B16	E15 of each line card L213–L224.	8 kHz reference clock signal from main **LSM** to line cards 13–24.
L426-A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3; L326-A20, B20, D20, E20, A13, B13, D13, E13, A6, B6, D6, E6	A23 of line card connectors L201–L224, respectively.	LC-Data from line cards 01–24, respectively, to backup **LSM**.
L426-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4; L326-A21, B21, D21, E21, A14, B14, D14, E14, A7, B7, D7, E7	A24 of line card connectors L201–L224, respectively.	LSM-RR from backup **LSM** to line cards 01–24, respectively.
L426-A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2; L326-A19, B19, D19, E19, A12, B12, D12, E12, A5, B5, D5, E5	A22 of line card connectors L201–L224, respectively.	LC-SR from line cards 01–24, respectively, to backup **LSM**.
L426-A14, B14, D14, E14, A7, B7, D7, E7; L326-A24, B24, D24, E24, A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3	A20 of line card connectors L201–L224, respectively.	LSM-DATA from backup **LSM** to line cards 01–24, respectively.
L426-A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1; L326-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4	A21 of line card connectors L201–L224, respectively.	LSM-SR from backup **LSM** to line cards 01–24, respectively.
L426-A13, B13, D13, E13, A6, B6, D6, E6; L326-A23, B23, D23, E23, A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2	A19 of line card connectors L201–L224, respectively.	LC-RR from line cards 01–24, respectively, to the backup **LSM**
L426-A12, B12, D12, E12, A5, B5, D5, E5; L326-A22, B22, D22, E22, A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1.	A18 of line card connectors L201–L224, respectively.	12.5 **MHz** clock signal from backup **LSM** to line cards 01–24, respectively.
L526-A12, B12, D12, E12, A9, B9, D9, E9, A6, B6, D6, E6, A3, B3, D3, E3; L426-A24, B24, D24, E24, A21, B21, D21, E21	E20 of line card connectors L201–L224, respectively.	Serial control link from backup **LSM** to line cards 01–24, respectively.
L526-A11, B11, D11, E11, A8, B8, D8, E8, A5, B5, D5, E5, A2, B2, D2, E2; L426-A23, B23, D23, E23, A20, B20, D20, E20	E19 of line card connectors L201–L224, respectively.	Serial control link clock from backup **LSM** to line cards 01–24, respectively
L526-A10, B10, D10, E10, A7, B7, D7, E7, A4, B4, D4, E4, A1, B1, D1, E1; L426-A22, B22, D22, E22, A19, B19, D19, E19	E18 of line card connectors L201–L224, respectively.	Serial control link from line cards 01–24, respectively, to the backup **LSM**.
L226-B15	E14 of line card connectors L201–L212.	8 kHz reference clock signal from backup **LSM** to line cards 01–12.
L226-B16	E14 of each line card L213–L224.	8 kHz reference clock signal from backup **LSM** to line cards 13–24.
L225-B13	L226-B13	**LSM** main status output to backup **LSM** status input.
L226-B14	L225-B14; E16 of each line card L201–L224	**LSM** backup status output to main status input and status input of line cards 01–24.

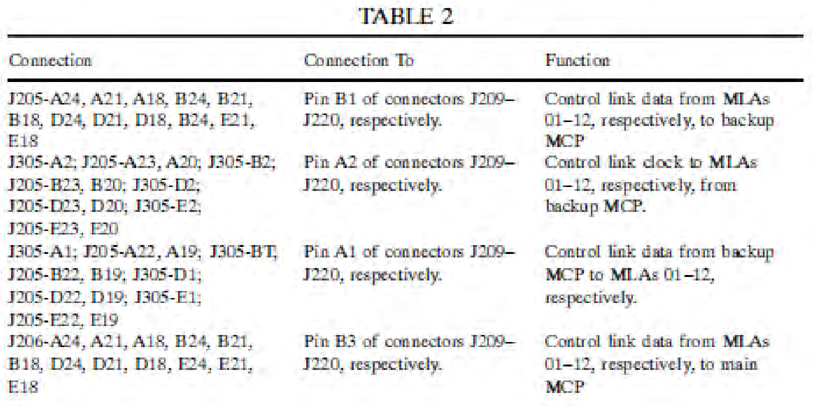
No.	’740 Patent Claim 8	Wiher ’530
		Wiher ’530 at 20:35-57 (“Trunk card interface control circuitry 1301 process ATM cell transfer signals exchanged over the main trunk card interface 1310 and backup trunk card interface 1315. Inter-face control circuitry 1301 provides, for example, ATM cell buffering and control of ATM cell exchange over interfaces 1310 and 1315. Additionally, interface circuitry 1301 may extract or insert OAMP cells to be exchanged between the processor 1305 and a trunk card. The interface control circuitry may also send ATM cells to and receive ATM cells from header translation circuitry 1303. Header translation circuitry 1303 performs VPINCI header field translation and other ATM cell header manipulation functions. Header translation circuitry 1303 may determine appropriate header manipulations based on, for example, programs and trans-lation tables stored in RAM and ROM memory 1304. Memory 1304 may include programs and data stored by processor 1305. Additionally, header translation circuitry 1303 interfaces with LSM interface control circuitry 1302. LSM interface control circuitry 1302 may extract or insert OAMP cells to be exchanged between the processor 1305 and a LSM. Additionally LSM interface control circuitry 1302 regulates ATM cell transport on, for example, a fiber optic SONET OC-3c interface 1330 to a LSM.”) Wiher ’530 at Table 2 

TABLE 2

Connection	Connection To	Function
J205-A24, A21, A18, B24, B21, B18, D24, D21, D18, B24, E21, E18	Pin B1 of connectors J209–J220, respectively.	Control link data from MLAs 01–12, respectively, to backup MCP
J305-A2; J205-A23, A20; J305-B2; J205-B23, B20; J305-D2; J205-D23, D20; J305-E2; J205-E23, E20	Pin A2 of connectors J209–J220, respectively.	Control link clock to MLAs 01–12, respectively, from backup MCP.
J305-A1; J205-A22, A19; J305-BT, J205-B22, B19; J305-D1; J205-D22, D19; J305-E1; J205-E22, E19	Pin A1 of connectors J209–J220, respectively.	Control link data from backup MCP to MLAs 01–12, respectively.
J206-A24, A21, A18, B24, B21, B18, D24, D21, D18, E24, E21, E18	Pin B3 of connectors J209–J220, respectively.	Control link data from MLAs 01–12, respectively, to main MCP

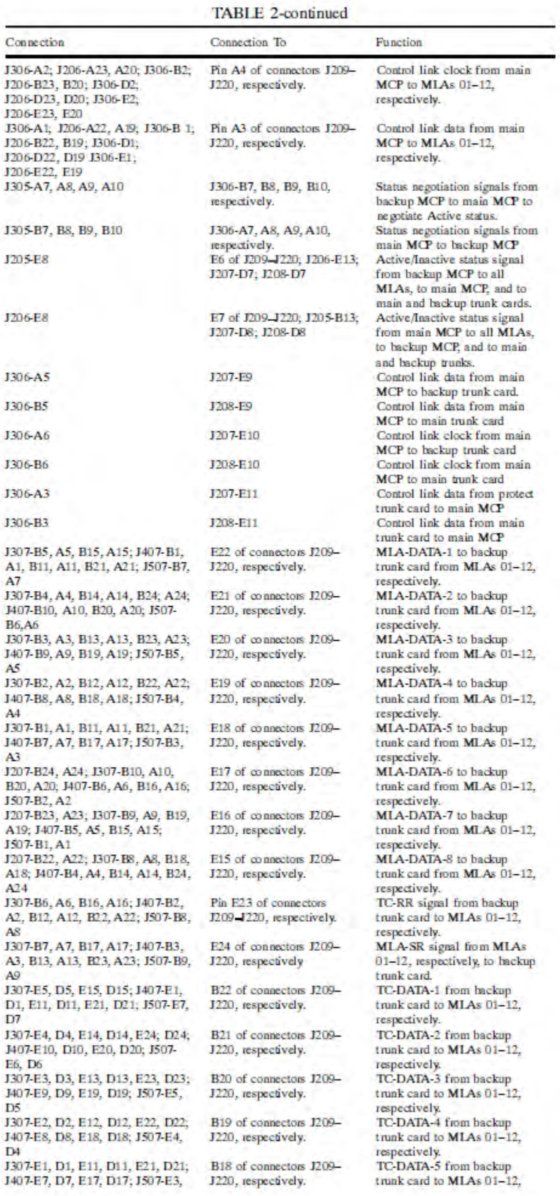


TABLE 2-continued

Connection	Connection To	Function
J306-A2; J206-A23, A20; J306-B2; J206-B23, B20; J306-D2; J206-D23, D20; J306-E2; J206-E23, E20	Pin A4 of connectors J209–J220, respectively.	Control link clock from main MCP to MLAs 01–12, respectively.
J306-A1; J206-A22, A19; J306-B 1; J206-B22, B19; J306-D1; J206-D22, D19 J306-E1; J206-E22, E19	Pin A3 of connectors J209–J220, respectively.	Control link data from main MCP to MLAs 01–12, respectively.
J305-A7, A8, A9, A10	J306-B7, B8, B9, B10, respectively.	Status negotiation signals from backup MCP to main MCP to negotiate Active status.
J305-B7, B8, B9, B10	J306-A7, A8, A9, A10, respectively.	Status negotiation signals from main MCP to backup MCP
J205-E8	E6 of J209–J220; J206-E13; J207-D7; J208-D7	Active/Inactive status signal from backup MCP to all MLAs, to main MCP, and to main and backup trunk cards.
J206-E8	E7 of J209–J220; J205-B13; J207-D8; J208-D8	Active/Inactive status signal from main MCP to all MLAs, to backup MCP, and to main and backup trunks.
J306-A5	J207-E9	Control link data from main MCP to backup trunk card.
J306-B5	J208-E9	Control link data from main MCP to main trunk card
J306-A6	J207-E10	Control link clock from main MCP to backup trunk card
J306-B6	J208-E10	Control link clock from main MCP to main trunk card
J306-A3	J207-E11	Control link data from protect trunk card to main MCP
J306-B3	J208-E11	Control link data from main trunk card to main MCP
J307-B5, A5, B15, A15; J407-B1, A1, B11, A11, B21, A21; J507-B7, A7	E22 of connectors J209–J220, respectively.	MLA-DATA-1 to backup trunk card from MLAs 01–12, respectively.
J307-B4, A4, B14, A14, B24; A24; J407-B10, A10, B20, A20; J507-B6,A6	E21 of connectors J209–J220, respectively.	MLA-DATA-2 to backup trunk card from MLAs 01–12, respectively.
J307-B3, A3, B13, A13, B23, A23; J407-B9, A9, B19, A19; J507-B5, A5	E20 of connectors J209–J220, respectively.	MLA-DATA-3 to backup trunk card from MLAs 01–12, respectively.
J307-B2, A2, B12, A12, B22, A22; J407-B8, A8, B18, A18; J507-B4, A4	E19 of connectors J209–J220, respectively.	MLA-DATA-4 to backup trunk card from MLAs 01–12, respectively.
J307-B1, A1, B11, A11, B21, A21; J407-B7, A7, B17, A17; J507-B3, A3	E18 of connectors J209–J220, respectively.	MLA-DATA-5 to backup trunk card from MLAs 01–12, respectively.
J207-B24, A24; J307-B10, A10, B20, A20; J407-B6, A6, B16, A16; J507-B2, A2	E17 of connectors J209–J220, respectively.	MLA-DATA-6 to backup trunk card from MLAs 01–12, respectively.
J207-B23, A23; J307-B9, A9, B19, A19; J407-B5, A5, B15, A15; J507-B1, A1	E16 of connectors J209–J220, respectively.	MLA-DATA-7 to backup trunk card from MLAs 01–12, respectively.
J207-B22, A22; J307-B8, A8, B18, A18; J407-B4, A4, B14, A14, B24, A24	E15 of connectors J209–J220, respectively.	MLA-DATA-8 to backup trunk card from MLAs 01–12, respectively.
J307-B6, A6, B16, A16; J407-B2, A2, B12, A12, B22, A22; J507-B8, A8	Pin E23 of connectors J209–J220, respectively.	TC-RR signal from backup trunk card to MLAs 01–12, respectively.
J307-B7, A7, B17, A17; J407-B3, A3, B13, A13, B23, A23; J507-B9, A9	E24 of connectors J209–J220, respectively	MLA-SR signal from MLAs 01–12, respectively, to backup trunk card.
J307-E5, D5, E15, D15; J407-E1, D1, E11, D11, E21, D21; J507-E7, D7	B22 of connectors J209–J220, respectively.	TC-DATA-1 from backup trunk card to MLAs 01–12, respectively.
J307-E4, D4, E14, D14, E24; D24; J407-E10, D10, E20, D20; J507-E6, D6	B21 of connectors J209–J220, respectively.	TC-DATA-2 from backup trunk card to MLAs 01–12, respectively.
J307-E3, D3, E13, D13, E23, D23; J407-E9, D9, E19, D19; J507-E5, D5	B20 of connectors J209–J220, respectively.	TC-DATA-3 from backup trunk card to MLAs 01–12, respectively.
J307-E2, D2, E12, D12, E22, D22; J407-E8, D8, E18, D18; J507-E4, D4	B19 of connectors J209–J220, respectively.	TC-DATA-4 from backup trunk card to MLAs 01–12, respectively.
J307-E1, D1, E11, D11, E21, D21; J407-E7, D7, E17, D17; J507-E3,	B18 of connectors J209–J220, respectively.	TC-DATA-5 from backup trunk card to MLAs 01–12,

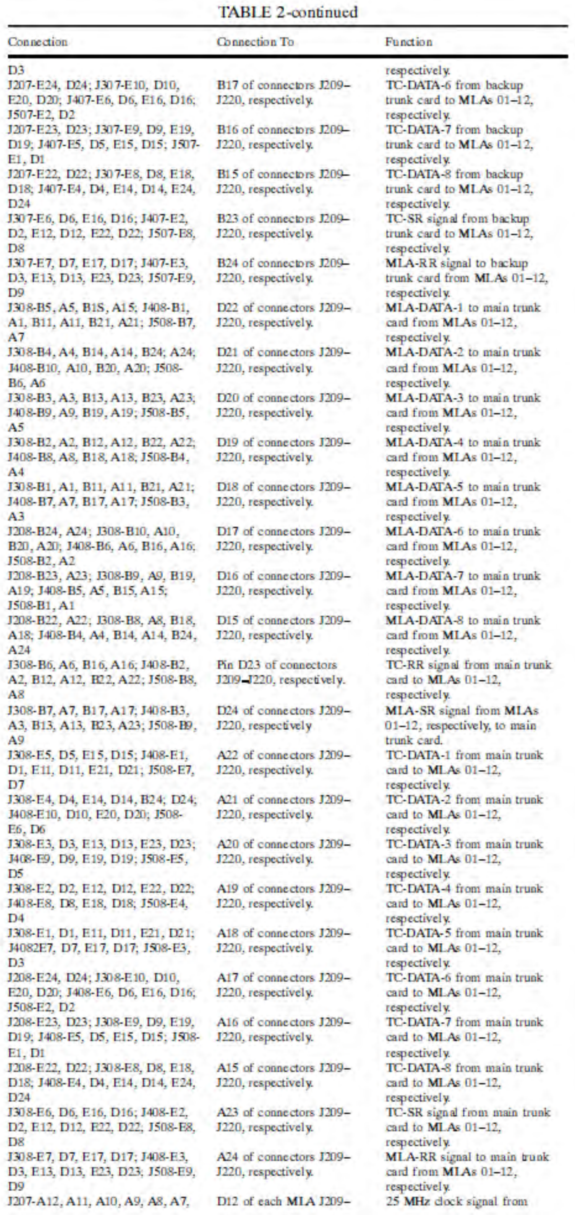


TABLE 2-continued

Connection	Connection To	Function
D3		respectively.
J207-E24, D24; J307-E10, D10, E20, D20; J407-E6, D6, E16, D16; J507-E2, D2	B17 of connectors J209–J220, respectively.	TC-DATA-6 from backup trunk card to MLAs 01–12, respectively.
J207-E23, D23; J307-E9, D9, E19, D19; J407-E5, D5, E15, D15; J507-E1, D1	B16 of connectors J209–J220, respectively.	TC-DATA-7 from backup trunk card to MLAs 01–12, respectively.
J207-E22, D22; J307-E8, D8, E18, D18; J407-E4, D4, E14, D14, E24, D24	B15 of connectors J209–J220, respectively.	TC-DATA-8 from backup trunk card to MLAs 01–12, respectively.
J307-E6, D6, E16, D16; J407-E2, D2, E12, D12, E22, D22; J507-E8, D8	B23 of connectors J209–J220, respectively.	TC-SR signal from backup trunk card to MLAs 01–12, respectively.
J307-E7, D7, E17, D17; J407-E3, D3, E13, D13, E23, D23; J507-E9, D9	B24 of connectors J209–J220, respectively.	MLA-RR signal to backup trunk card from MLAs 01–12, respectively.
J308-B5, A5, B1S, A15; J408-B1, A1, B11, A11, B21, A21; J508-B7, A7	D22 of connectors J209–J220, respectively.	MLA-DATA-1 to main trunk card from MLAs 01–12, respectively.
J308-B4, A4, B14, A14, B24; A24; J408-B10, A10, B20, A20; J508-B6, A6	D21 of connectors J209–J220, respectively.	MLA-DATA-2 to main trunk card from MLAs 01–12, respectively.
J308-B3, A3, B13, A13, B23, A23; J408-B9, A9, B19, A19; J508-B5, A5	D20 of connectors J209–J220, respectively.	MLA-DATA-3 to main trunk card from MLAs 01–12, respectively.
J308-B2, A2, B12, A12, B22, A22; J408-B8, A8, B18, A18; J508-B4, A4	D19 of connectors J209–J220, respectively.	MLA-DATA-4 to main trunk card from MLAs 01–12, respectively.
J308-B1, A1, B11, A11, B21, A21; J408-B7, A7, B17, A17; J508-B3, A3	D18 of connectors J209–J220, respectively.	MLA-DATA-5 to main trunk card from MLAs 01–12, respectively.
J208-B24, A24; J308-B10, A10, B20, A20; J408-B6, A6, B16, A16; J508-B2, A2	D17 of connectors J209–J220, respectively.	MLA-DATA-6 to main trunk card from MLAs 01–12, respectively.
J208-B23, A23; J308-B9, A9, B19, A19; J408-B5, A5, B15, A15; J508-B1, A1	D16 of connectors J209–J220, respectively.	MLA-DATA-7 to main trunk card from MLAs 01–12, respectively.
J208-B22, A22; J308-B8, A8, B18, A18; J408-B4, A4, B14, A14, B24, A24	D15 of connectors J209–J220, respectively.	MLA-DATA-8 to main trunk card from MLAs 01–12, respectively.
J308-B6, A6, B16, A16; J408-B2, A2, B12, A12, B22, A22; J508-B8, A8	Pin D23 of connectors J209–J220, respectively.	TC-RR signal from main trunk card to MLAs 01–12, respectively.
J308-B7, A7, B17, A17; J408-B3, A3, B13, A13, B23, A23; J508-B9, A9	D24 of connectors J209–J220, respectively	MLA-SR signal from MLAs 01–12, respectively, to main trunk card.
J308-E5, D5, E15, D15; J408-E1, D1, E11, D11, E21, D21; J508-E7, D7	A22 of connectors J209–J220, respectively.	TC-DATA-1 from main trunk card to MLAs 01–12, respectively.
J308-E4, D4, E14, D14, B24; D24; J408-E10, D10, E20, D20; J508-E6, D6	A21 of connectors J209–J220, respectively.	TC-DATA-2 from main trunk card to MLAs 01–12, respectively.
J308-E3, D3, E13, D13, E23, D23; J408-E9, D9, E19, D19; J508-E5, D5	A20 of connectors J209–J220, respectively.	TC-DATA-3 from main trunk card to MLAs 01–12, respectively.
J308-E2, D2, E12, D12, E22, D22; J408-E8, D8, E18, D18; J508-E4, D4	A19 of connectors J209–J220, respectively.	TC-DATA-4 from main trunk card to MLAs 01–12, respectively.
J308-E1, D1, E11, D11, E21, D21; J4082E7, D7, E17, D17; J508-E3, D3	A18 of connectors J209–J220, respectively.	TC-DATA-5 from main trunk card to MLAs 01–12, respectively.
J208-E24, D24; J308-E10, D10, E20, D20; J408-E6, D6, E16, D16; J508-E2, D2	A17 of connectors J209–J220, respectively.	TC-DATA-6 from main trunk card to MLAs 01–12, respectively.
J208-E23, D23; J308-E9, D9, E19, D19; J408-E5, D5, E15, D15; J508-E1, D1	A16 of connectors J209–J220, respectively.	TC-DATA-7 from main trunk card to MLAs 01–12, respectively.
J208-E22, D22; J308-E8, D8, E18, D18; J408-E4, D4, E14, D14, E24, D24	A15 of connectors J209–J220, respectively.	TC-DATA-8 from main trunk card to MLAs 01–12, respectively.
J308-E6, D6, E16, D16; J408-E2, D2, E12, D12, E22, D22; J508-E8, D8	A23 of connectors J209–J220, respectively.	TC-SR signal from main trunk card to MLAs 01–12, respectively.
J308-E7, D7, E17, D17; J408-E3, D3, E13, D13, E23, D23; J508-E9, D9	A24 of connectors J209–J220, respectively.	MLA-RR signal to main trunk card from MLAs 01–12, respectively.
J207-A12, A11, A10, A9, A8, A7,	D12 of each MLA J209–	25 MHz clock signal from

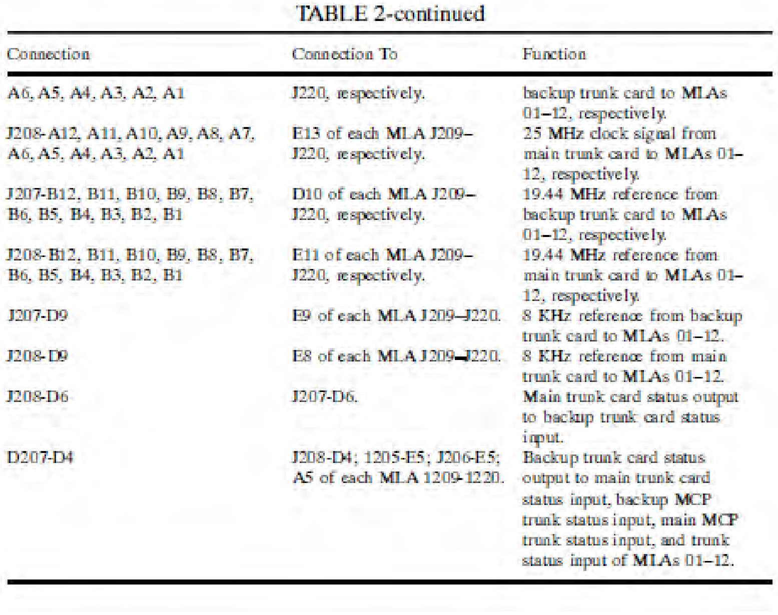
No.	’740 Patent Claim 8	Wiher ’530
		

TABLE 2-continued

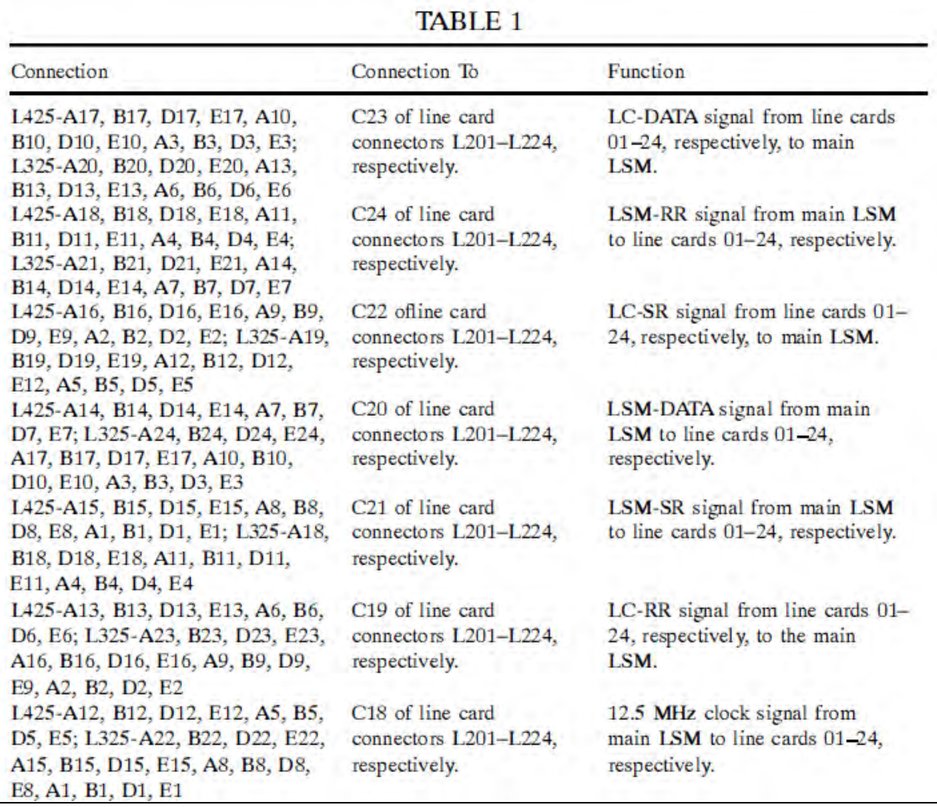
Connection	Connection To	Function
A6, A5, A4, A3, A2, A1	J220, respectively.	backup trunk card to MLAs 01–12, respectively.
J208-A12, A11, A10, A9, A8, A7, A6, A5, A4, A3, A2, A1	E13 of each MLA J209–J220, respectively.	25 MHz clock signal from main trunk card to MLAs 01–12, respectively.
J207-B12, B11, B10, B9, B8, B7, B6, B5, B4, B3, B2, B1	D10 of each MLA J209–J220, respectively.	19.44 MHz reference from backup trunk card to MLAs 01–12, respectively.
J208-B12, B11, B10, B9, B8, B7, B6, B5, B4, B3, B2, B1	E11 of each MLA J209–J220, respectively.	19.44 MHz reference from main trunk card to MLAs 01–12, respectively.
J207-D9	E9 of each MLA J209–J220.	8 KHz reference from backup trunk card to MLAs 01–12.
J208-D9	E8 of each MLA J209–J220.	8 KHz reference from main trunk card to MLAs 01–12.
J208-D6	J207-D6.	Main trunk card status output to backup trunk card status input.
D207-D4	J208-D4; 1205-E5; J206-E5; A5 of each MLA 1209-1220.	Backup trunk card status output to main trunk card status input, backup MCP trunk status input, main MCP trunk status input, and trunk status input of MLAs 01–12.

No.	’740 Patent Claim 9	Wiher ’530
9	The method according to claim 8, wherein applying the mapping function comprises applying a hashing function.	Wiher ’530 discloses the method according to claim 8, wherein applying the mapping function comprises applying a hashing function. For example, Wiher ’530 discloses ATM cell header manipulation during processing by header translation circuitry in order to direct the ATM cell to the selected links. A person of ordinary skill in the art would understand that selecting links to transmit the ATM cell involves a hash function. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Wiher ’530 is found to not meet this limitation, wherein applying the mapping function comprises applying a hashing function would have been obvious to a person having ordinary skill in the art, as explained below.

No.	’740 Patent Claim 9	Wiher ’530
		See supra Claim 8. Wiher ’530 at 8:53-9:9 (“Cells arriving at the interface control circuitry 520 may include operations, administration, maintenance, and provisioning (OAMP) data or may contain user-data. OAMP cells may be identified by the payload type indicator (PTI) field in the ATM cell header. The multiplexer circuitry 521 extracts OAMP cells and send them to the processor 527 while user-data cells are sent to header translation circuitry 522. Header translation circuitry 522 performs VPI/VCI header field translation and other ATM cell header manipulation functions. Header translation circuitry 522 may deter-mine appropriate header manipulations based on, for example, programs and translation tables stored in RAM and ROM memory 526. Memory 526 may include header manipulation programs and translation tables that are stored by processor 527. After processing by header translation circuitry 522, ATM cells are directed to master line shelf adapter (MLA) interface circuitry 525. MLA interface cir-cuitry 525 controls and buffers cells flowing from the address translation circuitry 522 to a MLA and controls, for example, ATM cell flow over a SO NET OC-3c interface 530 between the LSM 500 and a MLA. Interface circuitry 525 may also insert OAMP cells from the processor 527 for transmission to a MLA and extract OAMP cells received from a MLA.”) Wiher ’530 at 9:10-27 (“The LSM's MLA interface circuitry 525 may also receive ATM cells from a MLA. The MLA interface circuitry 525 may extract OAMP cells arriving from a MLA and send them to the processor 527. Data cells arriving at the interface 525 and destined to a line card are sent to header translation circuitry 524 which may perform ATM cell header manipu-lations. Header translation circuitry 524 performs VPI/VCI header field translation and other ATM cell header manipu-lation functions. Header translation circuitry 524 may deter-mine appropriate header manipulations based on, for example, programs and translation tables stored in RAM and ROM memory 526. Following header translation, data cells may be sent to de-multiplexer circuitry 523. The de-multiplexer circuitry controls the flow of ATM cells to interface control circuitry 520. Processor 527 may also send OAMP cells to the de-multiplexer 523 for transfer to line cards. Interface control circuitry 520 then transmits the ATM cell to a line card.”)

No.	’740 Patent Claim 9	Wiher ’530
		Wiher ’530 at 11:20-39 (“Data transfers from a LSM to a line card include line card port identification information (a "port address"). A port address is a fixed value associated with a particular line card transceiver or subscriber loop connection. For example, a line card supporting two subscriber loops has port addresses "Pl" and "P2" that are associated, respectively, with the first and second subscriber loop at the line card. Each subscriber loop at a particular line card has an associated port address that is unique with respect to the port addresses of other subscriber loops at that line card. However, port addresses at one line card need not be unique with respect to port addresses at another line card. In a data transfer between a LSM and a line card, a line card port address may be identified by, for example, additional data bytes added to the ATM cell or by information in a modified (non-standard) cell header. Unlike VPI/VCI addresses which are dynami-cally associated with a transceiver, port address are perma-nently assigned (that is, they are static). Thus, the use of port addresses can simplify cell routing through a line card by simplifying the processing and storage of cell routing data.”) Wiher ’530 at 12:64-13:26 (“Referring to FIGS. 6 and 8, a line card's main LSM interface 610 includes control link signal lines 613. Line card operations, administration, maintenance and provision-ing (OAMP) functions can be controlled by data sent over the control link signal lines 613 between the line card and the main LSM. The control link signal lines 613 include a clock signal line, a data receive signal line, and a data transmit signal line. The line card sends data to the LSM in serial fashion by modulating a signal over a data transmit signal line and receives modulated data from the LSM over a data receive signal line. Signals exchanged over the data receive and data transmit signal lines are, for example, asserted or de-asserted on the falling edge of a 64 kilohertz (KHz) clock pulse received on the clock receive signal line and are sampled on the rising edge of a received clock pulse. The format for the data exchanged on the control link signal lines 613 may conform to the Open Systems Interconnection (OSI) High-level Data Link Control (HDLC) protocol. The HDLC protocol is described in ISO/IEC 3309:1991 (E), Information Technology-Telecommunications and Information Exchange Between Systems-High-level Data Link Control (HDLC) procedures-Frame Structure, Interna-tional Organization for Standardization, Fourth Edition, 1991-06-01. A line card's main LSM interface 610 may also include clock signal lines 614. The clock signal lines include, for example, a 12.5 MHz clock signal line and a 8 KHz telephone network reference timing signal

<table>
<tr><th>No.</th><th>’740 Patent Claim 9</th><th>Wiher ’530</th></tr>
<tr><td></td><td></td><td>line received from the main LSM. Signals exchanged over clock signal lines 614 may be used to time data transmission over signal lines 611 and 612.

Wiher ’530 at Table 1

<table>
<tr><th colspan="3">TABLE 1</th></tr>
<tr><th>Connection</th><th>Connection To</th><th>Function</th></tr>
<tr><td>L425-A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3; L325-A20, B20, D20, E20, A13, B13, D13, E13, A6, B6, D6, E6</td><td>C23 of line card connectors L201–L224, respectively.</td><td>LC-DATA signal from line cards 01–24, respectively, to main LSM.</td></tr>
<tr><td>L425-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4; L325-A21, B21, D21, E21, A14, B14, D14, E14, A7, B7, D7, E7</td><td>C24 of line card connectors L201–L224, respectively.</td><td>LSM-RR signal from main LSM to line cards 01–24, respectively.</td></tr>
<tr><td>L425-A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2; L325-A19, B19, D19, E19, A12, B12, D12, E12, A5, B5, D5, E5</td><td>C22 ofline card connectors L201–L224, respectively.</td><td>LC-SR signal from line cards 01–24, respectively, to main LSM.</td></tr>
<tr><td>L425-A14, B14, D14, E14, A7, B7, D7, E7; L325-A24, B24, D24, E24, A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3</td><td>C20 of line card connectors L201–L224, respectively.</td><td>LSM-DATA signal from main LSM to line cards 01–24, respectively.</td></tr>
<tr><td>L425-A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1; L325-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4</td><td>C21 of line card connectors L201–L224, respectively.</td><td>LSM-SR signal from main LSM to line cards 01–24, respectively.</td></tr>
<tr><td>L425-A13, B13, D13, E13, A6, B6, D6, E6; L325-A23, B23, D23, E23, A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2</td><td>C19 of line card connectors L201–L224, respectively.</td><td>LC-RR signal from line cards 01–24, respectively, to the main LSM.</td></tr>
<tr><td>L425-A12, B12, D12, E12, A5, B5, D5, E5; L325-A22, B22, D22, E22, A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1</td><td>C18 of line card connectors L201–L224, respectively.</td><td>12.5 MHz clock signal from main LSM to line cards 01–24, respectively.</td></tr>
</table>
</td></tr>
</table>

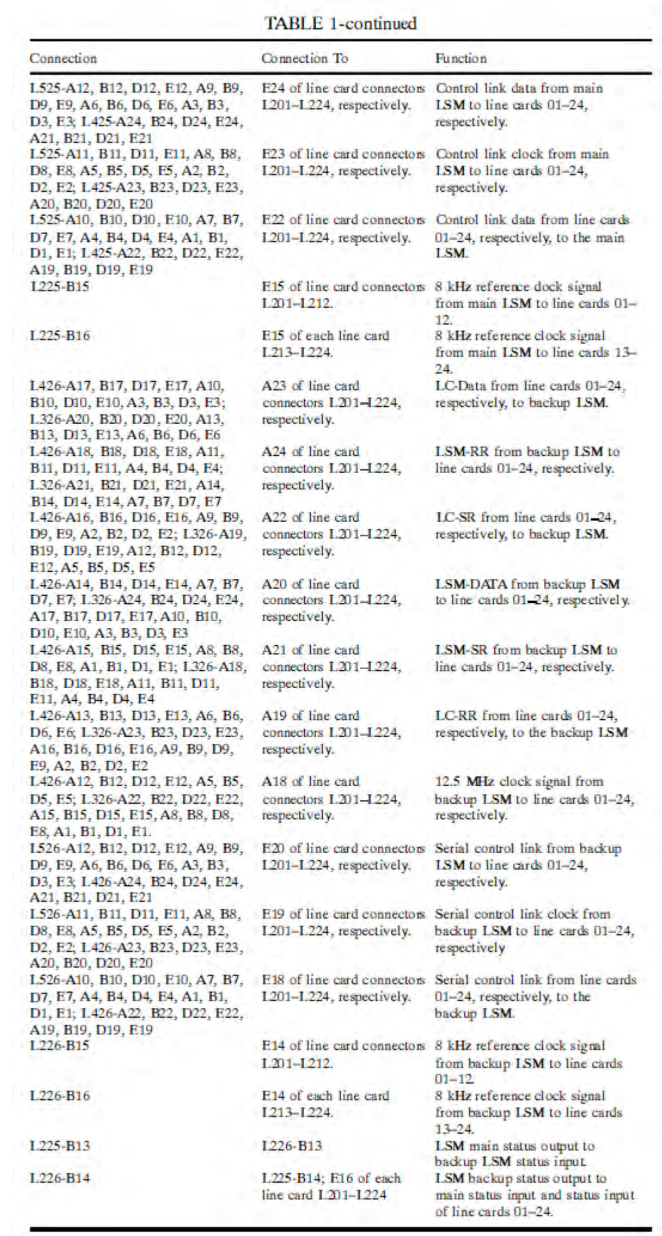


TABLE 1-continued

Connection	Connection To	Function
L525-A12, B12, D12, E12, A9, B9, D9, E9, A6, B6, D6, E6, A3, B3, D3, E3; L425-A24, B24, D24, E24, A21, B21, D21, E21	E24 of line card connectors L201–L224, respectively.	Control link data from main LSM to line cards 01–24, respectively.
L525-A11, B11, D11, E11, A8, B8, D8, E8, A5, B5, D5, E5, A2, B2, D2, E2; L425-A23, B23, D23, E23, A20, B20, D20, E20	E23 of line card connectors L201–L224, respectively.	Control link clock from main LSM to line cards 01–24, respectively.
L525-A10, B10, D10, E10, A7, B7, D7, E7, A4, B4, D4, E4, A1, B1, D1, E1; L425-A22, B22, D22, E22, A19, B19, D19, E19	E22 of line card connectors L201–L224, respectively.	Control link data from line cards 01–24, respectively, to the main LSM.
L225-B15	E15 of line card connectors L201–L212.	8 kHz reference dock signal from main LSM to line cards 01–12.
L225-B16	E15 of each line card L213–L224.	8 kHz reference clock signal from main LSM to line cards 13–24.
L426-A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3; L326-A20, B20, D20, E20, A13, B13, D13, E13, A6, B6, D6, E6	A23 of line card connectors L201–L224, respectively.	LC-Data from line cards 01–24, respectively, to backup LSM.
L426-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4; L326-A21, B21, D21, E21, A14, B14, D14, E14, A7, B7, D7, E7	A24 of line card connectors L201–L224, respectively.	LSM-RR from backup LSM to line cards 01–24, respectively.
L426-A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2; L326-A19, B19, D19, E19, A12, B12, D12, E12, A5, B5, D5, E5	A22 of line card connectors L201–L224, respectively.	LC-SR from line cards 01–24, respectively, to backup LSM.
L426-A14, B14, D14, E14, A7, B7, D7, E7; L326-A24, B24, D24, E24, A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3	A20 of line card connectors L201–L224, respectively.	LSM-DATA from backup LSM to line cards 01–24, respectively.
L426-A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1; L326-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4	A21 of line card connectors L201–L224, respectively.	LSM-SR from backup LSM to line cards 01–24, respectively.
L426-A13, B13, D13, E13, A6, B6, D6, E6; L326-A23, B23, D23, E23, A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2	A19 of line card connectors L201–L224, respectively.	LC-RR from line cards 01–24, respectively, to the backup LSM
L426-A12, B12, D12, E12, A5, B5, D5, E5; L326-A22, B22, D22, E22, A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1.	A18 of line card connectors L201–L224, respectively.	12.5 MHz clock signal from backup LSM to line cards 01–24, respectively.
L526-A12, B12, D12, E12, A9, B9, D9, E9, A6, B6, D6, E6, A3, B3, D3, E3; L426-A24, B24, D24, E24, A21, B21, D21, E21	E20 of line card connectors L201–L224, respectively.	Serial control link from backup LSM to line cards 01–24, respectively.
L526-A11, B11, D11, E11, A8, B8, D8, E8, A5, B5, D5, E5, A2, B2, D2, E2; L426-A23, B23, D23, E23, A20, B20, D20, E20	E19 of line card connectors L201–L224, respectively.	Serial control link clock from backup LSM to line cards 01–24, respectively
L526-A10, B10, D10, E10, A7, B7, D7, E7, A4, B4, D4, E4, A1, B1, D1, E1; L426-A22, B22, D22, E22, A19, B19, D19, E19	E18 of line card connectors L201–L224, respectively.	Serial control link from line cards 01–24, respectively, to the backup LSM.
L226-B15	E14 of line card connectors L201–L212.	8 kHz reference clock signal from backup LSM to line cards 01–12.
L226-B16	E14 of each line card L213–L224.	8 kHz reference clock signal from backup LSM to line cards 13–24.
L225-B13	L226-B13	LSM main status output to backup LSM status input.
L226-B14	L225-B14; E16 of each line card L201–L224	LSM backup status output to main status input and status input of line cards 01–24.

<table>
<tr><th>No.</th><th>’740 Patent Claim 9</th><th>Wiher ’530</th></tr>
<tr><td></td><td></td><td>
Wiher ’530 at 20:35-57 (“Trunk card interface control circuitry 1301 process ATM cell transfer signals exchanged over the main trunk card interface 1310 and backup trunk card interface 1315. Inter-face control circuitry 1301 provides, for example, ATM cell buffering and control of ATM cell exchange over interfaces 1310 and 1315. Additionally, interface circuitry 1301 may extract or insert OAMP cells to be exchanged between the processor 1305 and a trunk card. The interface control circuitry may also send ATM cells to and receive ATM cells from header translation circuitry 1303. Header translation circuitry 1303 performs VPINCI header field translation and other ATM cell header manipulation functions. Header translation circuitry 1303 may determine appropriate header manipulations based on, for example, programs and trans-lation tables stored in RAM and ROM memory 1304. Memory 1304 may include programs and data stored by processor 1305. Additionally, header translation circuitry 1303 interfaces with LSM interface control circuitry 1302. LSM interface control circuitry 1302 may extract or insert OAMP cells to be exchanged between the processor 1305 and a LSM. Additionally LSM interface control circuitry 1302 regulates ATM cell transport on, for example, a fiber optic SONET OC-3c interface 1330 to a LSM.”)

Wiher ’530 at Table 2

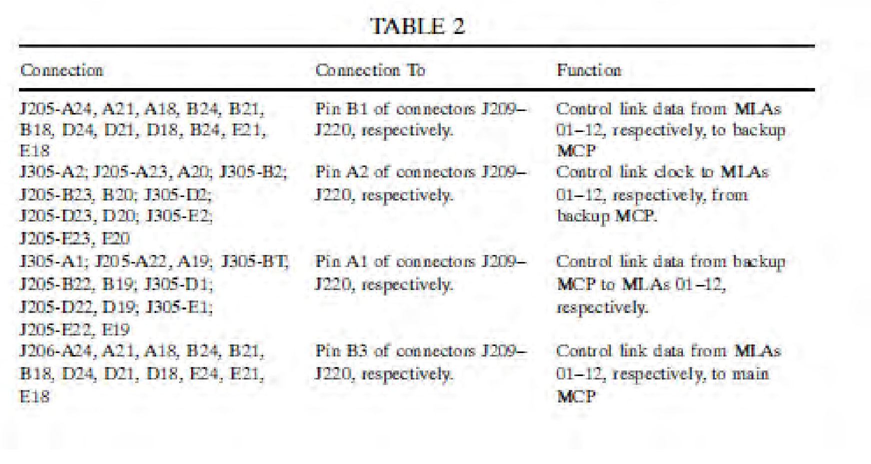


TABLE 2
<table>
<tr><th>Connection</th><th>Connection To</th><th>Function</th></tr>
<tr><td>J205-A24, A21, A18, B24, B21, B18, D24, D21, D18, B24, E21, E18</td><td>Pin B1 of connectors J209–J220, respectively.</td><td>Control link data from MLAs 01–12, respectively, to backup MCP</td></tr>
<tr><td>J305-A2; J205-A23, A20; J305-B2; J205-B23, B20; J305-D2; J205-D23, D20; J305-E2; J205-E23, E20</td><td>Pin A2 of connectors J209–J220, respectively.</td><td>Control link clock to MLAs 01–12, respectively, from backup MCP.</td></tr>
<tr><td>J305-A1; J205-A22, A19; J305-BT, J205-B22, B19; J305-D1; J205-D22, D19; J305-E1; J205-E22, E19</td><td>Pin A1 of connectors J209–J220, respectively.</td><td>Control link data from backup MCP to MLAs 01–12, respectively.</td></tr>
<tr><td>J206-A24, A21, A18, B24, B21, B18, D24, D21, D18, E24, E21, E18</td><td>Pin B3 of connectors J209–J220, respectively.</td><td>Control link data from MLAs 01–12, respectively, to main MCP</td></tr>
</table>
</td></tr>
</table>

No.	’740 Patent Claim 9	Wiher ’530

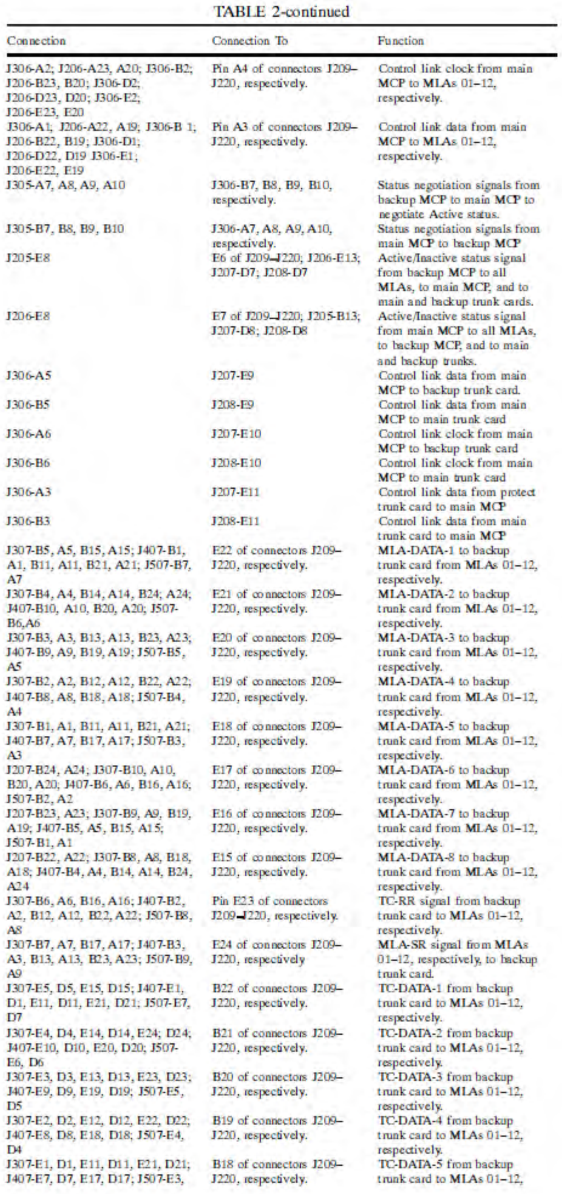


TABLE 2-continued

Connection	Connection To	Function
J306-A2; J206-A23, A20; J306-B2; J206-B23, B20; J306-D2; J206-D23, D20; J306-E2; J206-E23, E20	Pin A4 of connectors J209–J220, respectively.	Control link clock from main MCP to MLAs 01–12, respectively.
J306-A1; J206-A22, A19; J306-B 1; J206-B22, B19; J306-D1; J206-D22, D19 J306-E1; J206-E22, E19	Pin A3 of connectors J209–J220, respectively.	Control link data from main MCP to MLAs 01–12, respectively.
J305-A7, A8, A9, A10	J306-B7, B8, B9, B10, respectively.	Status negotiation signals from backup MCP to main MCP to negotiate Active status.
J305-B7, B8, B9, B10	J306-A7, A8, A9, A10, respectively.	Status negotiation signals from main MCP to backup MCP
J205-E8	E6 of J209–J220; J206-E13; J207-D7; J208-D7	Active/Inactive status signal from backup MCP to all MLAs, to main MCP, and to main and backup trunk cards.
J206-E8	E7 of J209–J220; J205-B13; J207-D8; J208-D8	Active/Inactive status signal from main MCP to all MLAs, to backup MCP, and to main and backup trunks.
J306-A5	J207-E9	Control link data from main MCP to backup trunk card.
J306-B5	J208-E9	Control link data from main MCP to main trunk card
J306-A6	J207-E10	Control link clock from main MCP to backup trunk card
J306-B6	J208-E10	Control link clock from main MCP to main trunk card
J306-A3	J207-E11	Control link data from protect trunk card to main MCP
J306-B3	J208-E11	Control link data from main trunk card to main MCP
J307-B5, A5, B15, A15; J407-B1, A1, B11, A11, B21, A21; J507-B7, A7	E22 of connectors J209–J220, respectively.	MLA-DATA-1 to backup trunk card from MLAs 01–12, respectively.
J307-B4, A4, B14, A14, B24; A24; J407-B10, A10, B20, A20; J507-B6,A6	E21 of connectors J209–J220, respectively.	MLA-DATA-2 to backup trunk card from MLAs 01–12, respectively.
J307-B3, A3, B13, A13, B23, A23; J407-B9, A9, B19, A19; J507-B5, A5	E20 of connectors J209–J220, respectively.	MLA-DATA-3 to backup trunk card from MLAs 01–12, respectively.
J307-B2, A2, B12, A12, B22, A22; J407-B8, A8, B18, A18; J507-B4, A4	E19 of connectors J209–J220, respectively.	MLA-DATA-4 to backup trunk card from MLAs 01–12, respectively.
J307-B1, A1, B11, A11, B21, A21; J407-B7, A7, B17, A17; J507-B3, A3	E18 of connectors J209–J220, respectively.	MLA-DATA-5 to backup trunk card from MLAs 01–12, respectively.
J207-B24, A24; J307-B10, A10, B20, A20; J407-B6, A6, B16, A16; J507-B2, A2	E17 of connectors J209–J220, respectively.	MLA-DATA-6 to backup trunk card from MLAs 01–12, respectively.
J207-B23, A23; J307-B9, A9, B19, A19; J407-B5, A5, B15, A15; J507-B1, A1	E16 of connectors J209–J220, respectively.	MLA-DATA-7 to backup trunk card from MLAs 01–12, respectively.
J207-B22, A22; J307-B8, A8, B18, A18; J407-B4, A4, B14, A14, B24, A24	E15 of connectors J209–J220, respectively.	MLA-DATA-8 to backup trunk card from MLAs 01–12, respectively.
J307-B6, A6, B16, A16; J407-B2, A2, B12, A12, B22, A22; J507-B8, A8	Pin E23 of connectors J209–J220, respectively.	TC-RR signal from backup trunk card to MLAs 01–12, respectively.
J307-B7, A7, B17, A17; J407-B3, A3, B13, A13, B23, A23; J507-B9, A9	E24 of connectors J209–J220, respectively	MLA-SR signal from MLAs 01–12, respectively, to backup trunk card.
J307-E5, D5, E15, D15; J407-E1, D1, E11, D11, E21, D21; J507-E7, D7	B22 of connectors J209–J220, respectively.	TC-DATA-1 from backup trunk card to MLAs 01–12, respectively.
J307-E4, D4, E14, D14, E24; D24; J407-E10, D10, E20, D20; J507-E6, D6	B21 of connectors J209–J220, respectively.	TC-DATA-2 from backup trunk card to MLAs 01–12, respectively.
J307-E3, D3, E13, D13, E23, D23; J407-E9, D9, E19, D19; J507-E5, D5	B20 of connectors J209–J220, respectively.	TC-DATA-3 from backup trunk card to MLAs 01–12, respectively.
J307-E2, D2, E12, D12, E22, D22; J407-E8, D8, E18, D18; J507-E4, D4	B19 of connectors J209–J220, respectively.	TC-DATA-4 from backup trunk card to MLAs 01–12, respectively.
J307-E1, D1, E11, D11, E21, D21; J407-E7, D7, E17, D17; J507-E3,	B18 of connectors J209–J220, respectively.	TC-DATA-5 from backup trunk card to MLAs 01–12,

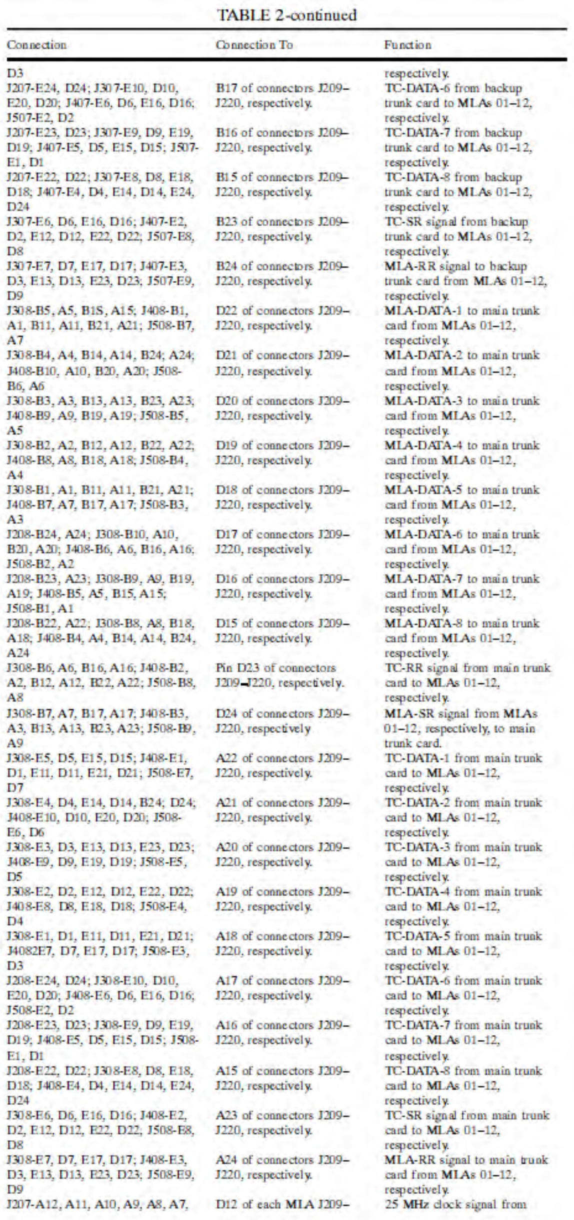


TABLE 2-continued

Connection	Connection To	Function
D3		respectively.
J207-E24, D24; J307-E10, D10, E20, D20; J407-E6, D6, E16, D16; J507-E2, D2	B17 of connectors J209–J220, respectively.	TC-DATA-6 from backup trunk card to MLAs 01–12, respectively.
J207-E23, D23; J307-E9, D9, E19, D19; J407-E5, D5, E15, D15; J507-E1, D1	B16 of connectors J209–J220, respectively.	TC-DATA-7 from backup trunk card to MLAs 01–12, respectively.
J207-E22, D22; J307-E8, D8, E18, D18; J407-E4, D4, E14, D14, E24, D24	B15 of connectors J209–J220, respectively.	TC-DATA-8 from backup trunk card to MLAs 01–12, respectively.
J307-E6, D6, E16, D16; J407-E2, D2, E12, D12, E22, D22; J507-E8, D8	B23 of connectors J209–J220, respectively.	TC-SR signal from backup trunk card to MLAs 01–12, respectively.
J307-E7, D7, E17, D17; J407-E3, D3, E13, D13, E23, D23; J507-E9, D9	B24 of connectors J209–J220, respectively.	MLA-RR signal to backup trunk card from MLAs 01–12, respectively.
J308-B5, A5, B1S, A15; J408-B1, A1, B11, A11, B21, A21; J508-B7, A7	D22 of connectors J209–J220, respectively.	MLA-DATA-1 to main trunk card from MLAs 01–12, respectively.
J308-B4, A4, B14, A14, B24; A24; J408-B10, A10, B20, A20; J508-B6, A6	D21 of connectors J209–J220, respectively.	MLA-DATA-2 to main trunk card from MLAs 01–12, respectively.
J308-B3, A3, B13, A13, B23, A23; J408-B9, A9, B19, A19; J508-B5, A5	D20 of connectors J209–J220, respectively.	MLA-DATA-3 to main trunk card from MLAs 01–12, respectively.
J308-B2, A2, B12, A12, B22, A22; J408-B8, A8, B18, A18; J508-B4, A4	D19 of connectors J209–J220, respectively.	MLA-DATA-4 to main trunk card from MLAs 01–12, respectively.
J308-B1, A1, B11, A11, B21, A21; J408-B7, A7, B17, A17; J508-B3, A3	D18 of connectors J209–J220, respectively.	MLA-DATA-5 to main trunk card from MLAs 01–12, respectively.
J208-B24, A24; J308-B10, A10, B20, A20; J408-B6, A6, B16, A16; J508-B2, A2	D17 of connectors J209–J220, respectively.	MLA-DATA-6 to main trunk card from MLAs 01–12, respectively.
J208-B23, A23; J308-B9, A9, B19, A19; J408-B5, A5, B15, A15; J508-B1, A1	D16 of connectors J209–J220, respectively.	MLA-DATA-7 to main trunk card from MLAs 01–12, respectively.
J208-B22, A22; J308-B8, A8, B18, A18; J408-B4, A4, B14, A14, B24, A24	D15 of connectors J209–J220, respectively.	MLA-DATA-8 to main trunk card from MLAs 01–12, respectively.
J308-B6, A6, B16, A16; J408-B2, A2, B12, A12, B22, A22; J508-B8, A8	Pin D23 of connectors J209–J220, respectively.	TC-RR signal from main trunk card to MLAs 01–12, respectively.
J308-B7, A7, B17, A17; J408-B3, A3, B13, A13, B23, A23; J508-B9, A9	D24 of connectors J209–J220, respectively	MLA-SR signal from MLAs 01–12, respectively, to main trunk card.
J308-E5, D5, E15, D15; J408-E1, D1, E11, D11, E21, D21; J508-E7, D7	A22 of connectors J209–J220, respectively.	TC-DATA-1 from main trunk card to MLAs 01–12, respectively.
J308-E4, D4, E14, D14, B24; D24; J408-E10, D10, E20, D20; J508-E6, D6	A21 of connectors J209–J220, respectively.	TC-DATA-2 from main trunk card to MLAs 01–12, respectively.
J308-E3, D3, E13, D13, E23, D23; J408-E9, D9, E19, D19; J508-E5, D5	A20 of connectors J209–J220, respectively.	TC-DATA-3 from main trunk card to MLAs 01–12, respectively.
J308-E2, D2, E12, D12, E22, D22; J408-E8, D8, E18, D18; J508-E4, D4	A19 of connectors J209–J220, respectively.	TC-DATA-4 from main trunk card to MLAs 01–12, respectively.
J308-E1, D1, E11, D11, E21, D21; J4082E7, D7, E17, D17; J508-E3, D3	A18 of connectors J209–J220, respectively.	TC-DATA-5 from main trunk card to MLAs 01–12, respectively.
J208-E24, D24; J308-E10, D10, E20, D20; J408-E6, D6, E16, D16; J508-E2, D2	A17 of connectors J209–J220, respectively.	TC-DATA-6 from main trunk card to MLAs 01–12, respectively.
J208-E23, D23; J308-E9, D9, E19, D19; J408-E5, D5, E15, D15; J508-E1, D1	A16 of connectors J209–J220, respectively.	TC-DATA-7 from main trunk card to MLAs 01–12, respectively.
J208-E22, D22; J308-E8, D8, E18, D18; J408-E4, D4, E14, D14, E24, D24	A15 of connectors J209–J220, respectively.	TC-DATA-8 from main trunk card to MLAs 01–12, respectively.
J308-E6, D6, E16, D16; J408-E2, D2, E12, D12, E22, D22; J508-E8, D8	A23 of connectors J209–J220, respectively.	TC-SR signal from main trunk card to MLAs 01–12, respectively.
J308-E7, D7, E17, D17; J408-E3, D3, E13, D13, E23, D23; J508-E9, D9	A24 of connectors J209–J220, respectively.	MLA-RR signal to main trunk card from MLAs 01–12, respectively.
J207-A12, A11, A10, A9, A8, A7,	D12 of each MLA J209–	25 MHz clock signal from

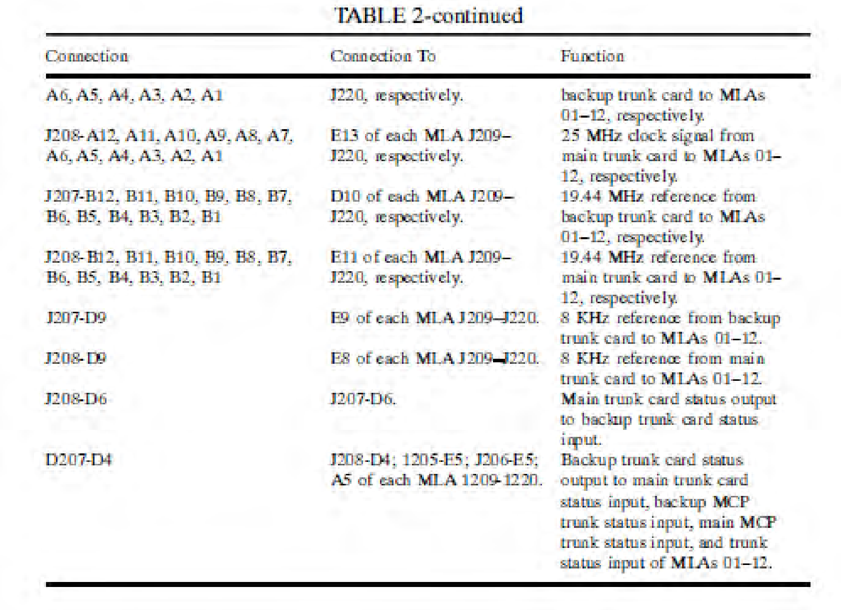
<table>
<tr><th>No.</th><th>’740 Patent Claim 9</th><th>Wiher ’530</th></tr>
<tr><td></td><td></td><td>


TABLE 2-continued

<table>
<tr><th>Connection</th><th>Connection To</th><th>Function</th></tr>
<tr><td>A6, A5, A4, A3, A2, A1</td><td>J220, respectively.</td><td>backup trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J208-A12, A11, A10, A9, A8, A7, A6, A5, A4, A3, A2, A1</td><td>E13 of each MLA J209–J220, respectively.</td><td>25 MHz clock signal from main trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J207-B12, B11, B10, B9, B8, B7, B6, B5, B4, B3, B2, B1</td><td>D10 of each MLA J209–J220, respectively.</td><td>19.44 MHz reference from backup trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J208-B12, B11, B10, B9, B8, B7, B6, B5, B4, B3, B2, B1</td><td>E11 of each MLA J209–J220, respectively.</td><td>19.44 MHz reference from main trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J207-D9</td><td>E9 of each MLA J209–J220.</td><td>8 KHz reference from backup trunk card to MLAs 01–12.</td></tr>
<tr><td>J208-D9</td><td>E8 of each MLA J209–J220.</td><td>8 KHz reference from main trunk card to MLAs 01–12.</td></tr>
<tr><td>J208-D6</td><td>J207-D6.</td><td>Main trunk card status output to backup trunk card status input.</td></tr>
<tr><td>D207-D4</td><td>J208-D4; 1205-E5; J206-E5; A5 of each MLA 1209-1220.</td><td>Backup trunk card status output to main trunk card status input, backup MCP trunk status input, main MCP trunk status input, and trunk status input of MLAs 01–12.</td></tr>
</table>

Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Wiher ’530 in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 9 of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references.

For example, Bruckman discloses applying a distributor hash function.

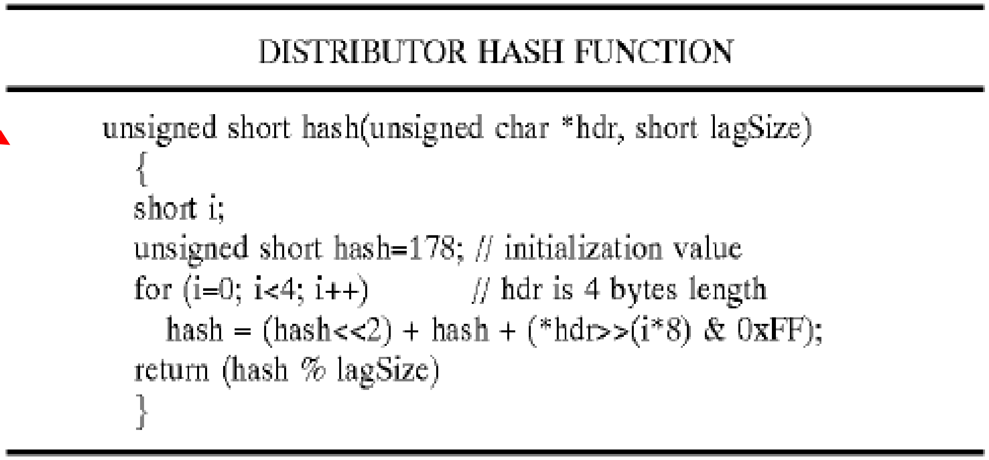
Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its
</td></tr>
</table>

No.	’740 Patent Claim 9	Wiher ’530
		reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card

No.	’740 Patent Claim 9	Wiher ’530
		32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”) Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I: Bruckman at Table 1 (annotated)

<table>
<tr><th>No.</th><th>’740 Patent Claim 9</th><th>Wiher ’530</th></tr>
<tr><td></td><td></td><td>

hashing function *“mapping function”*



DISTRIBUTOR HASH FUNCTION

```
unsigned short hash(unsigned char *hdr, short lagSize)
  {
  short i;
  unsigned short hash=178; // initialization value
  for (i=0; i<4; i++)          // hdr is 4 bytes length
     hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
  return (hash % lagSize)
  }
```

Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)

As another example, Solomon discloses applying a mapping function which comprises a hashing function performed by the mapper, i.e., wherein applying the mapping function comprises applying a hashing function.

Solomon at [0024] (“In another embodiment, switching the data packets includes mapping the data packets to the selected port responsively to the label. Additionally or alternatively, map-ping the data packets includes applying a hashing function to the label so as to determine a number of the selected port, and choosing the label includes applying an inverse of the hashing function to the number of the selected port.”)

</td></tr>
</table>

No.	’740 Patent Claim 9	Wiher ’530
		Solomon at [0048] (“The mapping function typically uses MPLS label 52 for mapping, since the MPLS label uniquely identifies MPLS tunnel 28, and it is required that all MPLS packets belonging to the same tunnel be switched through the same physical port 24. Additionally or alternatively, the mapping function uses a "PW" label (pseudo wire label, formerly known as a virtual connection, or VC label), which is optionally added to MPLS header 50. The PW label com-prises information that the egress node requires for deliver-ing the packet to its destination, and is optionally added during the encapsulation of MPLS packets. Additional details regarding the VC label can be found in an IETF draft by Martini et al. entitled "Encapsulation Methods for Trans-port of Ethernet Frames Over IP/MPLS Networks" (IETF draft-ietf-pwe3-ethernet-encap-07.txt, May, 2004), which is incorporated herein by reference. In some embodiments, mapper 34 applies a hashing function to the MPLS and/or PW label, as will be described below.”) Solomon at [0059] (“In this method, the mapping function used by mapper 34 of switch A is a hashing function. Various hashing functions are known in the art, and any suitable hashing function may be used in mapper 34. Since the hashing operation is performed for each packet, it is desirable to have a hashing function that is computationally simple.”) Solomon at [0060] (“As mentioned above, the hashing function typically hashes the value of MPLS label 52 to determine the selected physical port, as the MPLS label uniquely identifies tunnel 28. For example, the following hashing function may be used by mapper 34: Selected port number=l+((MPLS label) mod N), wherein N denotes the number of physical Ethernet ports in LAG group 25, and "mod" denotes the modulus operator. Assuming the values of MPLS labels are distrib-uted uniformly over a certain range, this function achieves a uniform distribution of port allocations for the different MPLS labels. It can also be seen that all packets carrying the same MPLS label (in other words-belonging to the same MPLS tunnel) will be mapped to the same physical port.”) Solomon at [0065] (“Mapper 34 of switch A maps each received packet to the selected physical port of LAG group 25 using the hashing function, at a hashing step 90. Mapper 34 extracts the MPLS label from each received packet and uses the hashing function to calculate the serial number of the selected physical port, which was selected by the CAC processor at step 82. Following the

No.	’740 Patent Claim 9	Wiher ’530
		numerical example given above, the mapper extracts MPLS label=65647 from the packet. Substituting this value and N=3 into the hashing function gives: Selected port number=1+(65647 mod 3)=2, which is indeed the port number selected in the example above.”)

No.	’740 Patent Claim 10	Wiher ’530
10[a]	The method according to claim 9, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second physical links,	Wiher ’530 discloses the method according to claim 9, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second physical links. For example, Wiher ’530 discloses header translation circuitry to process ATM cells then direct them to different destinations, including system physical links, based on the ATM cell header. A person of ordinary skill in the art would understand that applying a hash function includes determining parameters responsive to the system and packet features, and generating a result. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Wiher ’530 is found to not meet this limitation, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second physical links would have been obvious to a person having ordinary skill in the art, as explained below. *See supra* Claim 9 Wiher ’530 at 8:53-9:9 (“Cells arriving at the interface control circuitry 520 may include operations, administration, maintenance, and provisioning (OAMP) data or may contain user-data. OAMP cells may be identified by the payload type indicator (PTI) field in the ATM cell header. The multiplexer circuitry 521 extracts OAMP cells and send them to the processor 527 while user-data cells are sent to header translation circuitry 522. Header translation circuitry 522 performs VPI/VCI header field translation and other ATM cell header manipulation functions. Header

No.	’740 Patent Claim 10	Wiher ’530
		translation circuitry 522 may deter-mine appropriate header manipulations based on, for example, programs and translation tables stored in RAM and ROM memory 526. Memory 526 may include header manipulation programs and translation tables that are stored by processor 527. After processing by header translation circuitry 522, ATM cells are directed to master line shelf adapter (MLA) interface circuitry 525. MLA interface cir-cuitry 525 controls and buffers cells flowing from the address translation circuitry 522 to a MLA and controls, for example, ATM cell flow over a SO NET OC-3c interface 530 between the LSM 500 and a MLA. Interface circuitry 525 may also insert OAMP cells from the processor 527 for transmission to a MLA and extract OAMP cells received from a MLA.”) Wiher ’530 at 9:10-27 (“The LSM's MLA interface circuitry 525 may also receive ATM cells from a MLA. The MLA interface circuitry 525 may extract OAMP cells arriving from a MLA and send them to the processor 527. Data cells arriving at the interface 525 and destined to a line card are sent to header translation circuitry 524 which may perform ATM cell header manipu-lations. Header translation circuitry 524 performs VPINCI header field translation and other ATM cell header manipu-lation functions. Header translation circuitry 524 may deter-mine appropriate header manipulations based on, for example, programs and translation tables stored in RAM and ROM memory 526. Following header translation, data cells may be sent to de-multiplexer circuitry 523. The de-multiplexer circuitry controls the flow of ATM cells to interface control circuitry 520. Processor 527 may also send OAMP cells to the de-multiplexer 523 for transfer to line cards. Interface control circuitry 520 then transmits the ATM cell to a line card.”) Wiher ’530 at 11:20-39 (“Data transfers from a LSM to a line card include line card port identification information (a "port address"). A port address is a fixed value associated with a particular line card transceiver or subscriber loop connection. For example, a line card supporting two subscriber loops has port addresses "Pl" and "P2" that are associated, respectively, with the first and second subscriber loop at the line card. Each subscriber loop at a particular line card has an associated port address that is unique with respect to the port addresses of other subscriber loops at that line card. However, port addresses at one line card need not be unique with respect to port addresses at another line card. In a data transfer between a LSM and a line card, a line card port address may be identified by, for example, additional data bytes added to the ATM cell or by

No.	’740 Patent Claim 10	Wiher ’530
		information in a modified (non-standard) cell header. Unlike VPINCI addresses which are dynami-cally associated with a transceiver, port address are perma-nently assigned (that is, they are static). Thus, the use of port addresses can simplify cell routing through a line card by simplifying the processing and storage of cell routing data.”) Wiher ’530 at 12:64-13:26 (“Referring to FIGS. 6 and 8, a line card's main LSM interface 610 includes control link signal lines 613. Line card operations, administration, maintenance and provision-ing (OAMP) functions can be controlled by data sent over the control link signal lines 613 between the line card and the main LSM. The control link signal lines 613 include a clock signal line, a data receive signal line, and a data transmit signal line. The line card sends data to the LSM in serial fashion by modulating a signal over a data transmit signal line and receives modulated data from the LSM over a data receive signal line. Signals exchanged over the data receive and data transmit signal lines are, for example, asserted or de-asserted on the falling edge of a 64 kilohertz (KHz) clock pulse received on the clock receive signal line and are sampled on the rising edge of a received clock pulse. The format for the data exchanged on the control link signal lines 613 may conform to the Open Systems Interconnection (OSI) High-level Data Link Control (HDLC) protocol. The HDLC protocol is described in ISO/IEC 3309:1991 (E), Information Technology-Telecommunications and Information Exchange Between Systems-High-level Data Link Control (HDLC) procedures-Frame Structure, Interna-tional Organization for Standardization, Fourth Edition, 1991-06-01. A line card's main LSM interface 610 may also include clock signal lines 614. The clock signal lines include, for example, a 12.5 MHz clock signal line and a 8 KHz telephone network reference timing signal line received from the main LSM. Signals exchanged over clock signal lines 614 may be used to time data transmission over signal lines 611 and 612. Wiher ’530 at Table 1

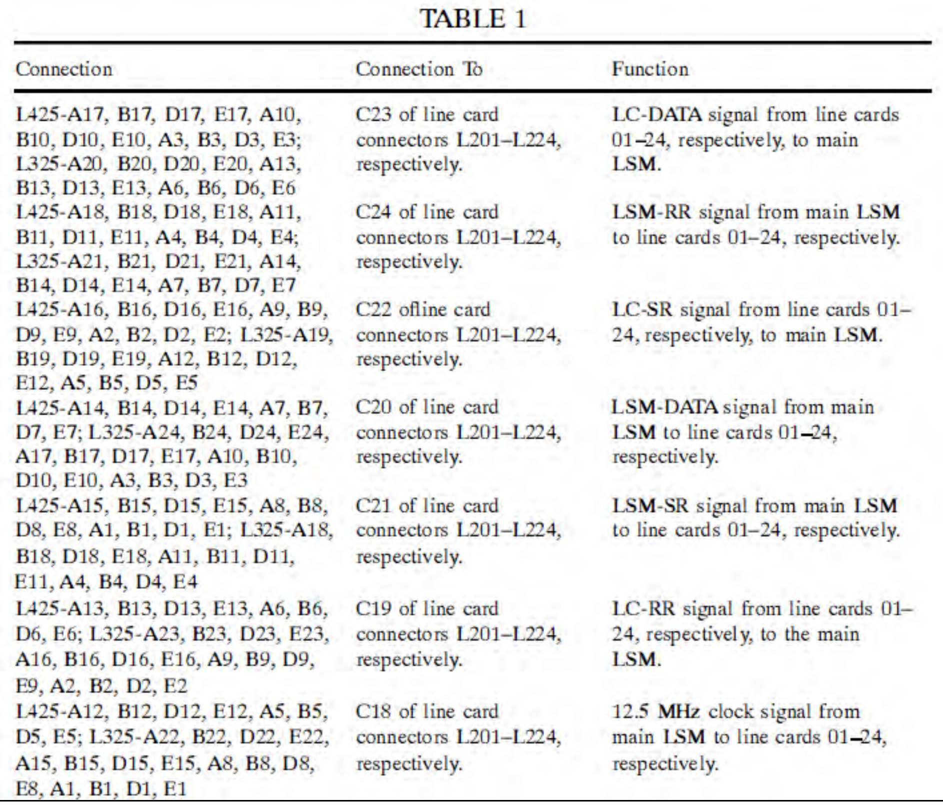
No.	’740 Patent Claim 10	Wiher ’530
		

TABLE 1

Connection	Connection To	Function
L425-A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3; L325-A20, B20, D20, E20, A13, B13, D13, E13, A6, B6, D6, E6	C23 of line card connectors L201–L224, respectively.	LC-DATA signal from line cards 01–24, respectively, to main LSM.
L425-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4; L325-A21, B21, D21, E21, A14, B14, D14, E14, A7, B7, D7, E7	C24 of line card connectors L201–L224, respectively.	LSM-RR signal from main LSM to line cards 01–24, respectively.
L425-A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2; L325-A19, B19, D19, E19, A12, B12, D12, E12, A5, B5, D5, E5	C22 ofline card connectors L201–L224, respectively.	LC-SR signal from line cards 01–24, respectively, to main LSM.
L425-A14, B14, D14, E14, A7, B7, D7, E7; L325-A24, B24, D24, E24, A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3	C20 of line card connectors L201–L224, respectively.	LSM-DATA signal from main LSM to line cards 01–24, respectively.
L425-A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1; L325-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4	C21 of line card connectors L201–L224, respectively.	LSM-SR signal from main LSM to line cards 01–24, respectively.
L425-A13, B13, D13, E13, A6, B6, D6, E6; L325-A23, B23, D23, E23, A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2	C19 of line card connectors L201–L224, respectively.	LC-RR signal from line cards 01–24, respectively, to the main LSM.
L425-A12, B12, D12, E12, A5, B5, D5, E5; L325-A22, B22, D22, E22, A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1	C18 of line card connectors L201–L224, respectively.	12.5 MHz clock signal from main LSM to line cards 01–24, respectively.

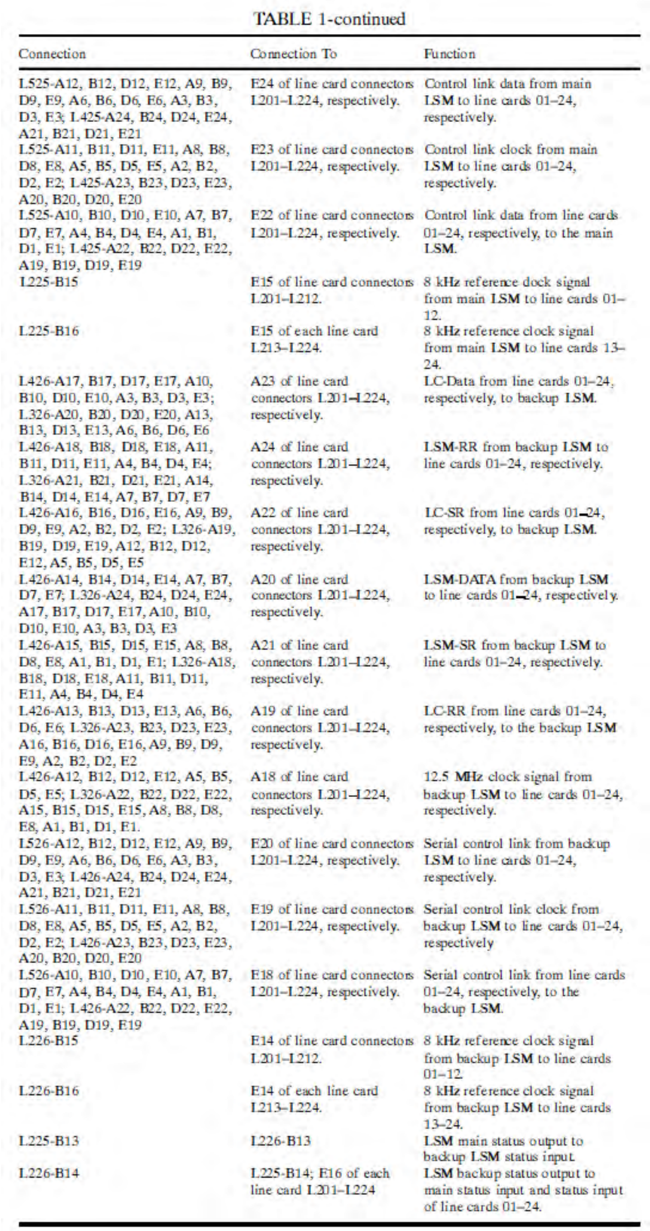


TABLE 1-continued

Connection	Connection To	Function
L525-A12, B12, D12, E12, A9, B9, D9, E9, A6, B6, D6, E6, A3, B3, D3, E3; L425-A24, B24, D24, E24, A21, B21, D21, E21	E24 of line card connectors L201–L224, respectively.	Control link data from main **LSM** to line cards 01–24, respectively.
L525-A11, B11, D11, E11, A8, B8, D8, E8, A5, B5, D5, E5, A2, B2, D2, E2; L425-A23, B23, D23, E23, A20, B20, D20, E20	E23 of line card connectors L201–L224, respectively.	Control link clock from main **LSM** to line cards 01–24, respectively.
L525-A10, B10, D10, E10, A7, B7, D7, E7, A4, B4, D4, E4, A1, B1, D1, E1; L425-A22, B22, D22, E22, A19, B19, D19, E19	E22 of line card connectors L201–L224, respectively.	Control link data from line cards 01–24, respectively, to the main **LSM**.
L225-B15	E15 of line card connectors L201–L212.	8 kHz reference dock signal from main **LSM** to line cards 01–12.
L225-B16	E15 of each line card L213–L224.	8 kHz reference clock signal from main **LSM** to line cards 13–24.
L426-A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3; L326-A20, B20, D20, E20, A13, B13, D13, E13, A6, B6, D6, E6	A23 of line card connectors L201–L224, respectively.	LC-Data from line cards 01–24, respectively, to backup **LSM**.
L426-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4; L326-A21, B21, D21, E21, A14, B14, D14, E14, A7, B7, D7, E7	A24 of line card connectors L201–L224, respectively.	**LSM**-RR from backup **LSM** to line cards 01–24, respectively.
L426-A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2; L326-A19, B19, D19, E19, A12, B12, D12, E12, A5, B5, D5, E5	A22 of line card connectors L201–L224, respectively.	LC-SR from line cards 01–24, respectively, to backup **LSM**.
L426-A14, B14, D14, E14, A7, B7, D7, E7; L326-A24, B24, D24, E24, A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3	A20 of line card connectors L201–L224, respectively.	**LSM**-DATA from backup **LSM** to line cards 01–24, respectively.
L426-A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1; L326-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4	A21 of line card connectors L201–L224, respectively.	**LSM**-SR from backup **LSM** to line cards 01–24, respectively.
L426-A13, B13, D13, E13, A6, B6, D6, E6; L326-A23, B23, D23, E23, A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2	A19 of line card connectors L201–L224, respectively.	LC-RR from line cards 01–24, respectively, to the backup **LSM**
L426-A12, B12, D12, E12, A5, B5, D5, E5; L326-A22, B22, D22, E22, A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1.	A18 of line card connectors L201–L224, respectively.	12.5 **MHz** clock signal from backup **LSM** to line cards 01–24, respectively.
L526-A12, B12, D12, E12, A9, B9, D9, E9, A6, B6, D6, E6, A3, B3, D3, E3; L426-A24, B24, D24, E24, A21, B21, D21, E21	E20 of line card connectors L201–L224, respectively.	Serial control link from backup **LSM** to line cards 01–24, respectively.
L526-A11, B11, D11, E11, A8, B8, D8, E8, A5, B5, D5, E5, A2, B2, D2, E2; L426-A23, B23, D23, E23, A20, B20, D20, E20	E19 of line card connectors L201–L224, respectively.	Serial control link clock from backup **LSM** to line cards 01–24, respectively
L526-A10, B10, D10, E10, A7, B7, D7, E7, A4, B4, D4, E4, A1, B1, D1, E1; L426-A22, B22, D22, E22, A19, B19, D19, E19	E18 of line card connectors L201–L224, respectively.	Serial control link from line cards 01–24, respectively, to the backup **LSM**.
L226-B15	E14 of line card connectors L201–L212.	8 kHz reference clock signal from backup **LSM** to line cards 01–12.
L226-B16	E14 of each line card L213–L224.	8 kHz reference clock signal from backup **LSM** to line cards 13–24.
L225-B13	L226-B13	**LSM** main status output to backup **LSM** status input.
L226-B14	L225-B14; E16 of each line card L201–L224	**LSM** backup status output to main status input and status input of line cards 01–24.

<table>
<tr><th>No.</th><th>’740 Patent Claim 10</th><th>Wiher ’530</th></tr>
<tr><td></td><td></td><td>

Wiher ’530 at 20:35-57 (“Trunk card interface control circuitry 1301 process ATM cell transfer signals exchanged over the main trunk card interface 1310 and backup trunk card interface 1315. Inter-face control circuitry 1301 provides, for example, ATM cell buffering and control of ATM cell exchange over interfaces 1310 and 1315. Additionally, interface circuitry 1301 may extract or insert OAMP cells to be exchanged between the processor 1305 and a trunk card. The interface control circuitry may also send ATM cells to and receive ATM cells from header translation circuitry 1303. Header translation circuitry 1303 performs VPINCI header field translation and other ATM cell header manipulation functions. Header translation circuitry 1303 may determine appropriate header manipulations based on, for example, programs and trans-lation tables stored in RAM and ROM memory 1304. Memory 1304 may include programs and data stored by processor 1305. Additionally, header translation circuitry 1303 interfaces with LSM interface control circuitry 1302. LSM interface control circuitry 1302 may extract or insert OAMP cells to be exchanged between the processor 1305 and a LSM. Additionally LSM interface control circuitry 1302 regulates ATM cell transport on, for example, a fiber optic SONET OC-3c interface 1330 to a LSM.”)

Wiher ’530 at Table 2

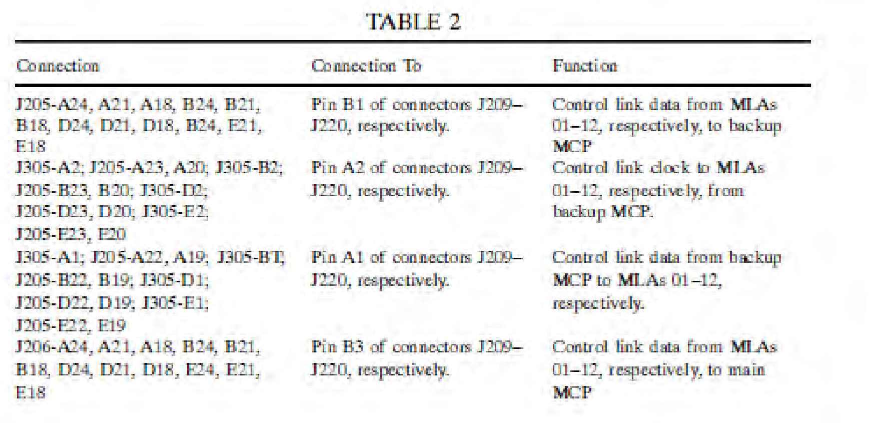


TABLE 2

<table>
<tr><th>Connection</th><th>Connection To</th><th>Function</th></tr>
<tr><td>J205-A24, A21, A18, B24, B21, B18, D24, D21, D18, B24, E21, E18</td><td>Pin B1 of connectors J209–J220, respectively.</td><td>Control link data from MLAs 01–12, respectively, to backup MCP</td></tr>
<tr><td>J305-A2; J205-A23, A20; J305-B2; J205-B23, B20; J305-D2; J205-D23, D20; J305-E2; J205-E23, E20</td><td>Pin A2 of connectors J209–J220, respectively.</td><td>Control link clock to MLAs 01–12, respectively, from backup MCP.</td></tr>
<tr><td>J305-A1; J205-A22, A19; J305-BT, J205-B22, B19; J305-D1; J205-D22, D19; J305-E1; J205-E22, E19</td><td>Pin A1 of connectors J209–J220, respectively.</td><td>Control link data from backup MCP to MLAs 01–12, respectively.</td></tr>
<tr><td>J206-A24, A21, A18, B24, B21, B18, D24, D21, D18, E24, E21, E18</td><td>Pin B3 of connectors J209–J220, respectively.</td><td>Control link data from MLAs 01–12, respectively, to main MCP</td></tr>
</table>

</td></tr>
</table>

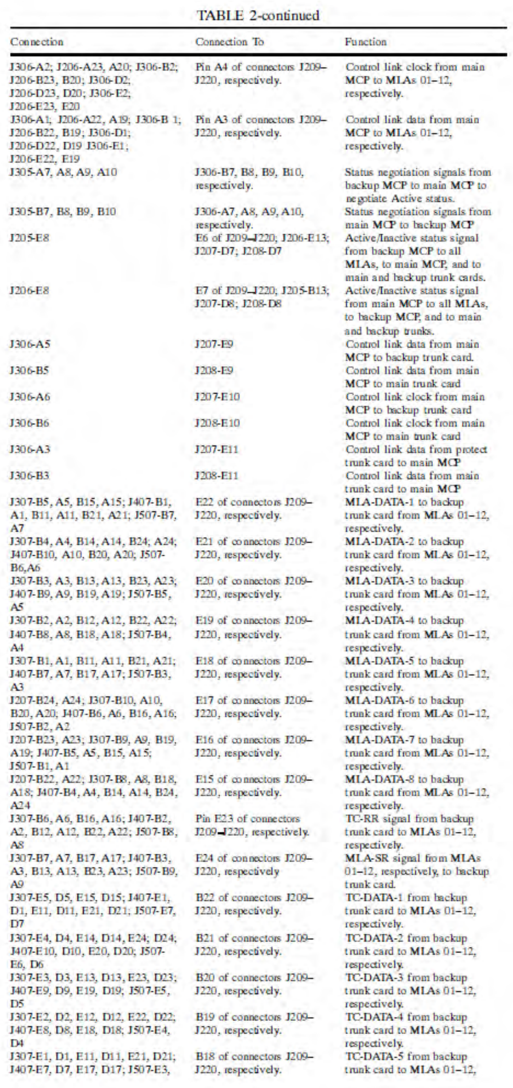
No.	’740 Patent Claim 10	Wiher ’530
		

TABLE 2-continued

Connection	Connection To	Function
J306-A2; J206-A23, A20; J306-B2; J206-B23, B20; J306-D2; J206-D23, D20; J306-E2; J206-E23, E20	Pin A4 of connectors J209–J220, respectively.	Control link clock from main MCP to MLAs 01–12, respectively.
J306-A1; J206-A22, A19; J306-B 1; J206-B22, B19; J306-D1; J206-D22, D19 J306-E1; J206-E22, E19	Pin A3 of connectors J209–J220, respectively.	Control link data from main MCP to MLAs 01–12, respectively.
J305-A7, A8, A9, A10	J306-B7, B8, B9, B10, respectively.	Status negotiation signals from backup MCP to main MCP to negotiate Active status.
J305-B7, B8, B9, B10	J306-A7, A8, A9, A10, respectively.	Status negotiation signals from main MCP to backup MCP
J205-E8	E6 of J209–J220; J206-E13; J207-D7; J208-D7	Active/Inactive status signal from backup MCP to all MLAs, to main MCP, and to main and backup trunk cards.
J206-E8	E7 of J209–J220; J205-B13; J207-D8; J208-D8	Active/Inactive status signal from main MCP to all MLAs, to backup MCP, and to main and backup trunks.
J306-A5	J207-E9	Control link data from main MCP to backup trunk card.
J306-B5	J208-E9	Control link data from main MCP to main trunk card
J306-A6	J207-E10	Control link clock from main MCP to backup trunk card
J306-B6	J208-E10	Control link clock from main MCP to main trunk card
J306-A3	J207-E11	Control link data from protect trunk card to main MCP
J306-B3	J208-E11	Control link data from main trunk card to main MCP
J307-B5, A5, B15, A15; J407-B1, A1, B11, A11, B21, A21; J507-B7, A7	E22 of connectors J209–J220, respectively.	MLA-DATA-1 to backup trunk card from MLAs 01–12, respectively.
J307-B4, A4, B14, A14, B24; A24; J407-B10, A10, B20, A20; J507-B6,A6	E21 of connectors J209–J220, respectively.	MLA-DATA-2 to backup trunk card from MLAs 01–12, respectively.
J307-B3, A3, B13, A13, B23, A23; J407-B9, A9, B19, A19; J507-B5, A5	E20 of connectors J209–J220, respectively.	MLA-DATA-3 to backup trunk card from MLAs 01–12, respectively.
J307-B2, A2, B12, A12, B22, A22; J407-B8, A8, B18, A18; J507-B4, A4	E19 of connectors J209–J220, respectively.	MLA-DATA-4 to backup trunk card from MLAs 01–12, respectively.
J307-B1, A1, B11, A11, B21, A21; J407-B7, A7, B17, A17; J507-B3, A3	E18 of connectors J209–J220, respectively.	MLA-DATA-5 to backup trunk card from MLAs 01–12, respectively.
J207-B24, A24; J307-B10, A10, B20, A20; J407-B6, A6, B16, A16; J507-B2, A2	E17 of connectors J209–J220, respectively.	MLA-DATA-6 to backup trunk card from MLAs 01–12, respectively.
J207-B23, A23; J307-B9, A9, B19, A19; J407-B5, A5, B15, A15; J507-B1, A1	E16 of connectors J209–J220, respectively.	MLA-DATA-7 to backup trunk card from MLAs 01–12, respectively.
J207-B22, A22; J307-B8, A8, B18, A18; J407-B4, A4, B14, A14, B24, A24	E15 of connectors J209–J220, respectively.	MLA-DATA-8 to backup trunk card from MLAs 01–12, respectively.
J307-B6, A6, B16, A16; J407-B2, A2, B12, A12, B22, A22; J507-B8, A8	Pin E23 of connectors J209–J220, respectively.	TC-RR signal from backup trunk card to MLAs 01–12, respectively.
J307-B7, A7, B17, A17; J407-B3, A3, B13, A13, B23, A23; J507-B9, A9	E24 of connectors J209–J220, respectively	MLA-SR signal from MLAs 01–12, respectively, to backup trunk card.
J307-E5, D5, E15, D15; J407-E1, D1, E11, D11, E21, D21; J507-E7, D7	B22 of connectors J209–J220, respectively.	TC-DATA-1 from backup trunk card to MLAs 01–12, respectively.
J307-E4, D4, E14, D14, E24; D24; J407-E10, D10, E20, D20; J507-E6, D6	B21 of connectors J209–J220, respectively.	TC-DATA-2 from backup trunk card to MLAs 01–12, respectively.
J307-E3, D3, E13, D13, E23, D23; J407-E9, D9, E19, D19; J507-E5, D5	B20 of connectors J209–J220, respectively.	TC-DATA-3 from backup trunk card to MLAs 01–12, respectively.
J307-E2, D2, E12, D12, E22, D22; J407-E8, D8, E18, D18; J507-E4, D4	B19 of connectors J209–J220, respectively.	TC-DATA-4 from backup trunk card to MLAs 01–12, respectively.
J307-E1, D1, E11, D11, E21, D21; J407-E7, D7, E17, D17; J507-E3,	B18 of connectors J209–J220, respectively.	TC-DATA-5 from backup trunk card to MLAs 01–12,

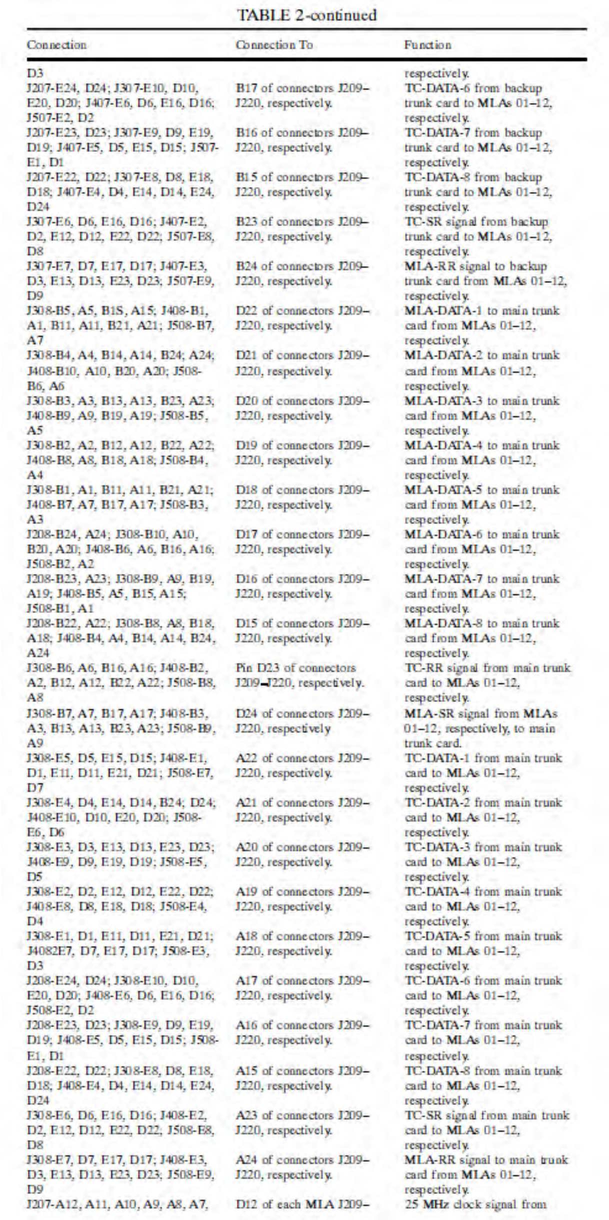


TABLE 2-continued

Connection	Connection To	Function
D3		respectively.
J207-E24, D24; J307-E10, D10, E20, D20; J407-E6, D6, E16, D16; J507-E2, D2	B17 of connectors J209–J220, respectively.	TC-DATA-6 from backup trunk card to MLAs 01–12, respectively.
J207-E23, D23; J307-E9, D9, E19, D19; J407-E5, D5, E15, D15; J507-E1, D1	B16 of connectors J209–J220, respectively.	TC-DATA-7 from backup trunk card to MLAs 01–12, respectively.
J207-E22, D22; J307-E8, D8, E18, D18; J407-E4, D4, E14, D14, E24, D24	B15 of connectors J209–J220, respectively.	TC-DATA-8 from backup trunk card to MLAs 01–12, respectively.
J307-E6, D6, E16, D16; J407-E2, D2, E12, D12, E22, D22; J507-E8, D8	B23 of connectors J209–J220, respectively.	TC-SR signal from backup trunk card to MLAs 01–12, respectively.
J307-E7, D7, E17, D17; J407-E3, D3, E13, D13, E23, D23; J507-E9, D9	B24 of connectors J209–J220, respectively.	MLA-RR signal to backup trunk card from MLAs 01–12, respectively.
J308-B5, A5, B1S, A15; J408-B1, A1, B11, A11, B21, A21; J508-B7, A7	D22 of connectors J209–J220, respectively.	MLA-DATA-1 to main trunk card from MLAs 01–12, respectively.
J308-B4, A4, B14, A14, B24; A24; J408-B10, A10, B20, A20; J508-B6, A6	D21 of connectors J209–J220, respectively.	MLA-DATA-2 to main trunk card from MLAs 01–12, respectively.
J308-B3, A3, B13, A13, B23, A23; J408-B9, A9, B19, A19; J508-B5, A5	D20 of connectors J209–J220, respectively.	MLA-DATA-3 to main trunk card from MLAs 01–12, respectively.
J308-B2, A2, B12, A12, B22, A22; J408-B8, A8, B18, A18; J508-B4, A4	D19 of connectors J209–J220, respectively.	MLA-DATA-4 to main trunk card from MLAs 01–12, respectively.
J308-B1, A1, B11, A11, B21, A21; J408-B7, A7, B17, A17; J508-B3, A3	D18 of connectors J209–J220, respectively.	MLA-DATA-5 to main trunk card from MLAs 01–12, respectively.
J208-B24, A24; J308-B10, A10, B20, A20; J408-B6, A6, B16, A16; J508-B2, A2	D17 of connectors J209–J220, respectively.	MLA-DATA-6 to main trunk card from MLAs 01–12, respectively.
J208-B23, A23; J308-B9, A9, B19, A19; J408-B5, A5, B15, A15; J508-B1, A1	D16 of connectors J209–J220, respectively.	MLA-DATA-7 to main trunk card from MLAs 01–12, respectively.
J208-B22, A22; J308-B8, A8, B18, A18; J408-B4, A4, B14, A14, B24, A24	D15 of connectors J209–J220, respectively.	MLA-DATA-8 to main trunk card from MLAs 01–12, respectively.
J308-B6, A6, B16, A16; J408-B2, A2, B12, A12, B22, A22; J508-B8, A8	Pin D23 of connectors J209–J220, respectively.	TC-RR signal from main trunk card to MLAs 01–12, respectively.
J308-B7, A7, B17, A17; J408-B3, A3, B13, A13, B23, A23; J508-B9, A9	D24 of connectors J209–J220, respectively	MLA-SR signal from MLAs 01–12; respectively, to main trunk card.
J308-E5, D5, E15, D15; J408-E1, D1, E11, D11, E21, D21; J508-E7, D7	A22 of connectors J209–J220, respectively.	TC-DATA-1 from main trunk card to MLAs 01–12, respectively.
J308-E4, D4, E14, D14, B24; D24; J408-E10, D10, E20, D20; J508-E6, D6	A21 of connectors J209–J220, respectively.	TC-DATA-2 from main trunk card to MLAs 01–12, respectively.
J308-E3, D3, E13, D13, E23, D23; J408-E9, D9, E19, D19; J508-E5, D5	A20 of connectors J209–J220, respectively.	TC-DATA-3 from main trunk card to MLAs 01–12, respectively.
J308-E2, D2, E12, D12, E22, D22; J408-E8, D8, E18, D18; J508-E4, D4	A19 of connectors J209–J220, respectively.	TC-DATA-4 from main trunk card to MLAs 01–12, respectively.
J308-E1, D1, E11, D11, E21, D21; J4082E7, D7, E17, D17; J508-E3, D3	A18 of connectors J209–J220, respectively.	TC-DATA-5 from main trunk card to MLAs 01–12, respectively.
J208-E24, D24; J308-E10, D10, E20, D20; J408-E6, D6, E16, D16; J508-E2, D2	A17 of connectors J209–J220, respectively.	TC-DATA-6 from main trunk card to MLAs 01–12, respectively.
J208-E23, D23; J308-E9, D9, E19, D19; J408-E5, D5, E15, D15; J508-E1, D1	A16 of connectors J209–J220, respectively.	TC-DATA-7 from main trunk card to MLAs 01–12, respectively.
J208-E22, D22; J308-E8, D8, E18, D18; J408-E4, D4, E14, D14, E24, D24	A15 of connectors J209–J220, respectively.	TC-DATA-8 from main trunk card to MLAs 01–12, respectively.
J308-E6, D6, E16, D16; J408-E2, D2, E12, D12, E22, D22; J508-E8, D8	A23 of connectors J209–J220, respectively.	TC-SR signal from main trunk card to MLAs 01–12, respectively.
J308-E7, D7, E17, D17; J408-E3, D3, E13, D13, E23, D23; J508-E9, D9	A24 of connectors J209–J220, respectively.	MLA-RR signal to main trunk card from MLAs 01–12, respectively.
J207-A12, A11, A10, A9, A8, A7,	D12 of each MLA J209–	25 MHz clock signal from

<table>
<tr><th>No.</th><th>’740 Patent Claim 10</th><th>Wiher ’530</th></tr>
<tr><td></td><td></td><td>

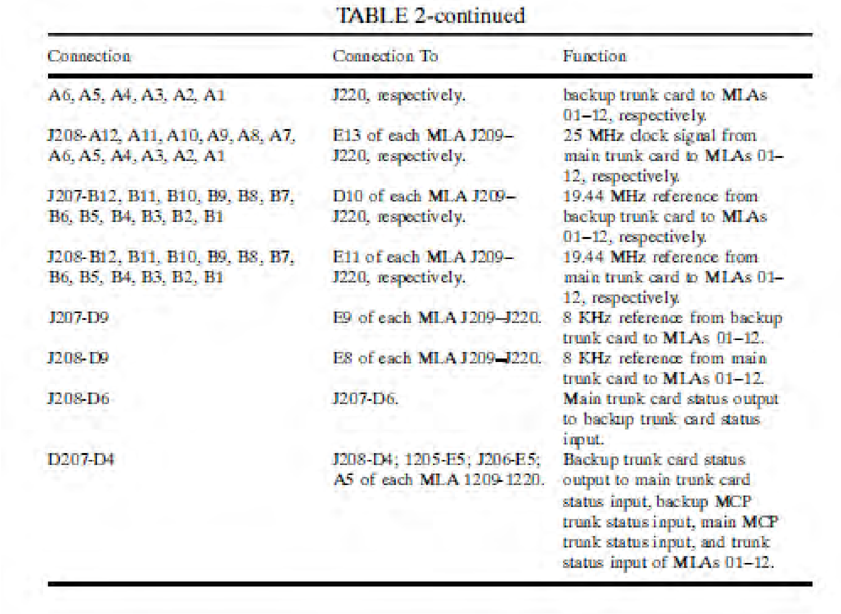



TABLE 2-continued
<table>
<tr><th>Connection</th><th>Connection To</th><th>Function</th></tr>
<tr><td>A6, A5, A4, A3, A2, A1</td><td>J220, respectively.</td><td>backup trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J208-A12, A11, A10, A9, A8, A7, A6, A5, A4, A3, A2, A1</td><td>E13 of each MLA J209–J220, respectively.</td><td>25 MHz clock signal from main trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J207-B12, B11, B10, B9, B8, B7, B6, B5, B4, B3, B2, B1</td><td>D10 of each MLA J209–J220, respectively.</td><td>19.44 MHz reference from backup trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J208-B12, B11, B10, B9, B8, B7, B6, B5, B4, B3, B2, B1</td><td>E11 of each MLA J209–J220, respectively.</td><td>19.44 MHz reference from main trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J207-D9</td><td>E9 of each MLA J209–J220.</td><td>8 KHz reference from backup trunk card to MLAs 01–12.</td></tr>
<tr><td>J208-D9</td><td>E8 of each MLA J209–J220.</td><td>8 KHz reference from main trunk card to MLAs 01–12.</td></tr>
<tr><td>J208-D6</td><td>J207-D6.</td><td>Main trunk card status output to backup trunk card status input.</td></tr>
<tr><td>D207-D4</td><td>J208-D4; 1205-E5; J206-E5; A5 of each MLA 1209-1220.</td><td>Backup trunk card status output to main trunk card status input, backup MCP trunk status input, main MCP trunk status input, and trunk status input of MLAs 01–12.</td></tr>
</table>

Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Wiher ’530 in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[a] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references.

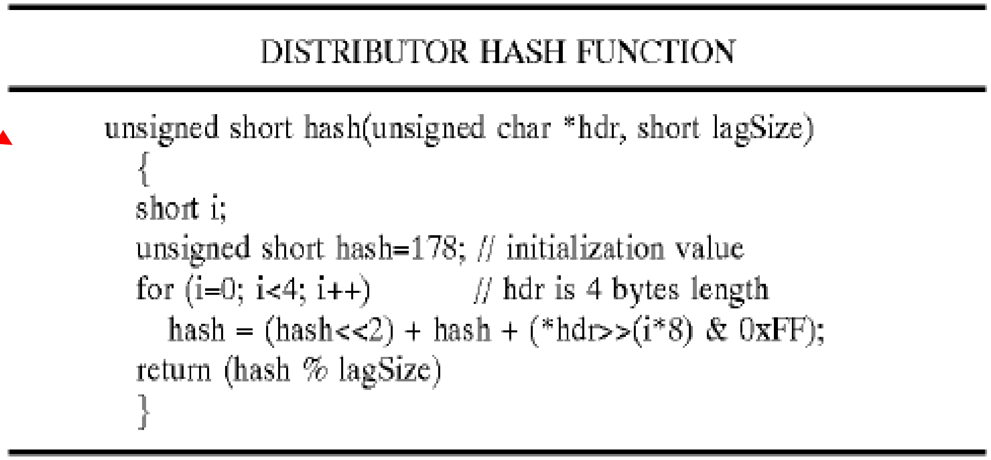
For example, Bruckman discloses applying a distributor hash function to the frame information which includes determining a number of the plurality of physical links.

Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in
</td></tr>
</table>

No.	’740 Patent Claim 10	Wiher ’530
		order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards,

No.	’740 Patent Claim 10	Wiher ’530
		the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”) Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I: Bruckman at Table 1 (annotated)

No.	’740 Patent Claim 10	Wiher ’530

		hashing function *“mapping function”*
		
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		For example, Solomon discloses applying a distributor hash function to the frame information which includes determining a number of the plurality of physical links.
		Solomon at [0024] (“In another embodiment, switching the data packets includes mapping the data packets to the selected port responsively to the label. Additionally or alternatively, map-ping the data packets includes applying a hashing function to the label so as to determine a number of the selected port, and choosing the label includes applying an inverse of the hashing function to the number of the selected port.”)

No.	’740 Patent Claim 10	Wiher ’530
		Solomon at [0048] (“The mapping function typically uses MPLS label 52 for mapping, since the MPLS label uniquely identifies MPLS tunnel 28, and it is required that all MPLS packets belonging to the same tunnel be switched through the same physical port 24. Additionally or alternatively, the mapping function uses a "PW" label (pseudo wire label, formerly known as a virtual connection, or VC label), which is optionally added to MPLS header 50. The PW label com-prises information that the egress node requires for deliver-ing the packet to its destination, and is optionally added during the encapsulation of MPLS packets. Additional details regarding the VC label can be found in an IETF draft by Martini et al. entitled "Encapsulation Methods for Trans-port of Ethernet Frames Over IP/MPLS Networks" (IETF draft-ietf-pwe3-ethernet-encap-07.txt, May, 2004), which is incorporated herein by reference. In some embodiments, mapper 34 applies a hashing function to the MPLS and/or PW label, as will be described below.”) Solomon at [0059] (“In this method, the mapping function used by mapper 34 of switch A is a hashing function. Various hashing functions are known in the art, and any suitable hashing function may be used in mapper 34. Since the hashing operation is performed for each packet, it is desirable to have a hashing function that is computationally simple.”) Solomon at [0060] (“As mentioned above, the hashing function typically hashes the value of MPLS label 52 to determine the selected physical port, as the MPLS label uniquely identifies tunnel 28. For example, the following hashing function may be used by mapper 34: Selected port number=l+((MPLS label) mod N), wherein N denotes the number of physical Ethernet ports in LAG group 25, and "mod" denotes the modulus operator. Assuming the values of MPLS labels are distrib-uted uniformly over a certain range, this function achieves a uniform distribution of port allocations for the different MPLS labels. It can also be seen that all packets carrying the same MPLS label (in other words-belonging to the same MPLS tunnel) will be mapped to the same physical port.”) Solomon at [0065] (“Mapper 34 of switch A maps each received packet to the selected physical port of LAG group 25 using the hashing function, at a hashing step 90. Mapper 34 extracts the MPLS label from each received packet and uses the hashing function to calculate the serial number of the selected physical port, which was selected by the CAC processor at step 82. Following the

No.	’740 Patent Claim 10	Wiher ’530
		numerical example given above, the mapper extracts MPLS label=65647 from the packet. Substituting this value and N=3 into the hashing function gives: Selected port number=1+(65647 mod 3)=2, which is indeed the port number selected in the example above.”)
10[b]	applying the hashing function to the at least one of the frame attributes to produce a hashing key,	Wiher ’530 discloses applying the hashing function to the at least one of the frame attributes to produce a hashing key. For example, Wiher ’530 discloses header translation circuitry to process ATM cells then direct them to different destinations, including system physical links, based on the ATM cell header. A person of ordinary skill in the art would understand that applying a hash function includes determining parameters responsive to the system and packet features, and generating a result. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Wiher ’530 is found to not meet this limitation, applying the hashing function to the at least one of the frame attributes to produce a hashing key would have been obvious to a person having ordinary skill in the art, as explained below. Wiher ’530 at 8:53-9:9 (“Cells arriving at the interface control circuitry 520 may include operations, administration, maintenance, and provisioning (OAMP) data or may contain user-data. OAMP cells may be identified by the payload type indicator (PTI) field in the ATM cell header. The multiplexer circuitry 521 extracts OAMP cells and send them to the processor 527 while user-data cells are sent to header translation circuitry 522. Header translation circuitry 522 performs VPI/VCI header field translation and other ATM cell header manipulation functions. Header translation circuitry 522 may deter-mine appropriate header manipulations based on, for example, programs and translation tables stored in RAM and ROM memory 526. Memory 526 may include header manipulation programs and translation tables that are stored by processor 527. After processing by header translation circuitry 522, ATM cells are directed to master line shelf adapter (MLA) interface circuitry 525. MLA interface cir-cuitry 525 controls and buffers cells flowing from the address translation circuitry 522 to a MLA and controls, for example, ATM cell flow over a SO NET OC-3c interface 530 between the LSM 500 and a MLA. Interface circuitry 525

No.	’740 Patent Claim 10	Wiher ’530
		may also insert OAMP cells from the processor 527 for transmission to a MLA and extract OAMP cells received from a MLA.”) Wiher ’530 at 9:10-27 (“The LSM's MLA interface circuitry 525 may also receive ATM cells from a MLA. The MLA interface circuitry 525 may extract OAMP cells arriving from a MLA and send them to the processor 527. Data cells arriving at the interface 525 and destined to a line card are sent to header translation circuitry 524 which may perform ATM cell header manipu-lations. Header translation circuitry 524 performs VPINCI header field translation and other ATM cell header manipu-lation functions. Header translation circuitry 524 may deter-mine appropriate header manipulations based on, for example, programs and translation tables stored in RAM and ROM memory 526. Following header translation, data cells may be sent to de-multiplexer circuitry 523. The de-multiplexer circuitry controls the flow of ATM cells to interface control circuitry 520. Processor 527 may also send OAMP cells to the de-multiplexer 523 for transfer to line cards. Interface control circuitry 520 then transmits the ATM cell to a line card.”) Wiher ’530 at 11:20-39 (“Data transfers from a LSM to a line card include line card port identification information (a "port address"). A port address is a fixed value associated with a particular line card transceiver or subscriber loop connection. For example, a line card supporting two subscriber loops has port addresses "Pl" and "P2" that are associated, respectively, with the first and second subscriber loop at the line card. Each subscriber loop at a particular line card has an associated port address that is unique with respect to the port addresses of other subscriber loops at that line card. However, port addresses at one line card need not be unique with respect to port addresses at another line card. In a data transfer between a LSM and a line card, a line card port address may be identified by, for example, additional data bytes added to the ATM cell or by information in a modified (non-standard) cell header. Unlike VPINCI addresses which are dynami-cally associated with a transceiver, port address are perma-nently assigned (that is, they are static). Thus, the use of port addresses can simplify cell routing through a line card by simplifying the processing and storage of cell routing data.”) Wiher ’530 at 12:64-13:26 (“Referring to FIGS. 6 and 8, a line card's main LSM interface 610 includes control link signal lines 613. Line card operations, administration, maintenance and

No.	’740 Patent Claim 10	Wiher ’530
		provision-ing (OAMP) functions can be controlled by data sent over the control link signal lines 613 between the line card and the main LSM. The control link signal lines 613 include a clock signal line, a data receive signal line, and a data transmit signal line. The line card sends data to the LSM in serial fashion by modulating a signal over a data transmit signal line and receives modulated data from the LSM over a data receive signal line. Signals exchanged over the data receive and data transmit signal lines are, for example, asserted or de-asserted on the falling edge of a 64 kilohertz (KHz) clock pulse received on the clock receive signal line and are sampled on the rising edge of a received clock pulse. The format for the data exchanged on the control link signal lines 613 may conform to the Open Systems Interconnection (OSI) High-level Data Link Control (HDLC) protocol. The HDLC protocol is described in ISO/IEC 3309:1991 (E), Information Technology-Telecommunications and Information Exchange Between Systems-High-level Data Link Control (HDLC) procedures-Frame Structure, Interna-tional Organization for Standardization, Fourth Edition, 1991-06-01. A line card's main LSM interface 610 may also include clock signal lines 614. The clock signal lines include, for example, a 12.5 MHz clock signal line and a 8 KHz telephone network reference timing signal line received from the main LSM. Signals exchanged over clock signal lines 614 may be used to time data transmission over signal lines 611 and 612. Wiher ’530 at Table 1

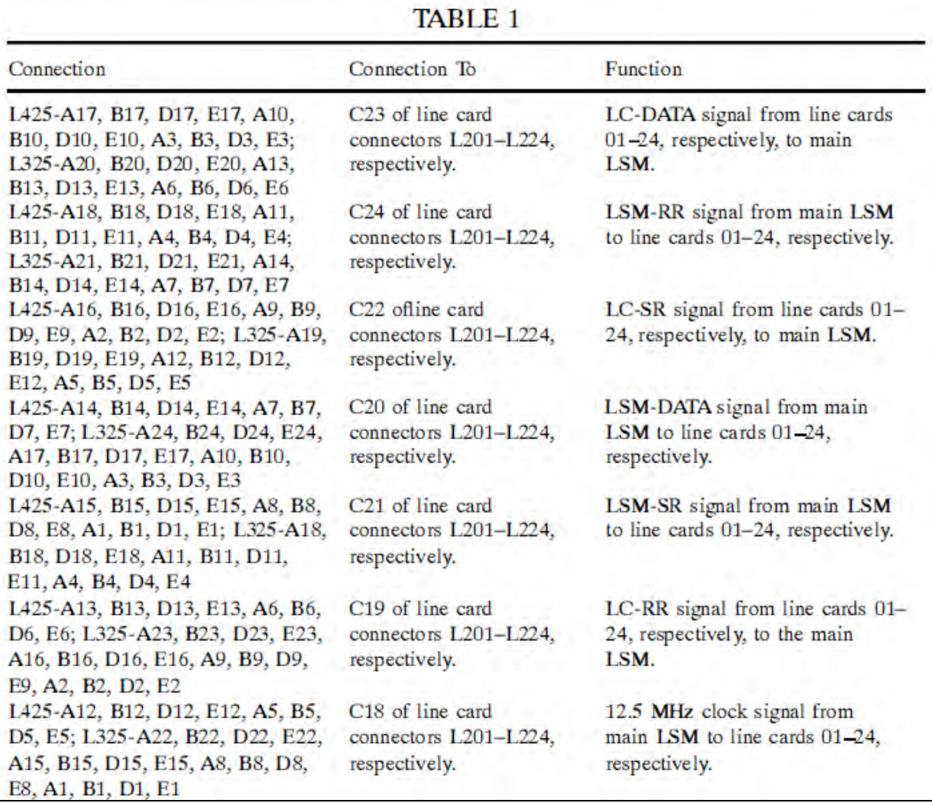
No.	’740 Patent Claim 10	Wiher ’530
		

TABLE 1

Connection	Connection To	Function
L425-A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3; L325-A20, B20, D20, E20, A13, B13, D13, E13, A6, B6, D6, E6	C23 of line card connectors L201–L224, respectively.	LC-DATA signal from line cards 01–24, respectively, to main LSM.
L425-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4; L325-A21, B21, D21, E21, A14, B14, D14, E14, A7, B7, D7, E7	C24 of line card connectors L201–L224, respectively.	LSM-RR signal from main LSM to line cards 01–24, respectively.
L425-A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2; L325-A19, B19, D19, E19, A12, B12, D12, E12, A5, B5, D5, E5	C22 ofline card connectors L201–L224, respectively.	LC-SR signal from line cards 01–24, respectively, to main LSM.
L425-A14, B14, D14, E14, A7, B7, D7, E7; L325-A24, B24, D24, E24, A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3	C20 of line card connectors L201–L224, respectively.	LSM-DATA signal from main LSM to line cards 01–24, respectively.
L425-A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1; L325-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4	C21 of line card connectors L201–L224, respectively.	LSM-SR signal from main LSM to line cards 01–24, respectively.
L425-A13, B13, D13, E13, A6, B6, D6, E6; L325-A23, B23, D23, E23, A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2	C19 of line card connectors L201–L224, respectively.	LC-RR signal from line cards 01–24, respectively, to the main LSM.
L425-A12, B12, D12, E12, A5, B5, D5, E5; L325-A22, B22, D22, E22, A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1	C18 of line card connectors L201–L224, respectively.	12.5 MHz clock signal from main LSM to line cards 01–24, respectively.

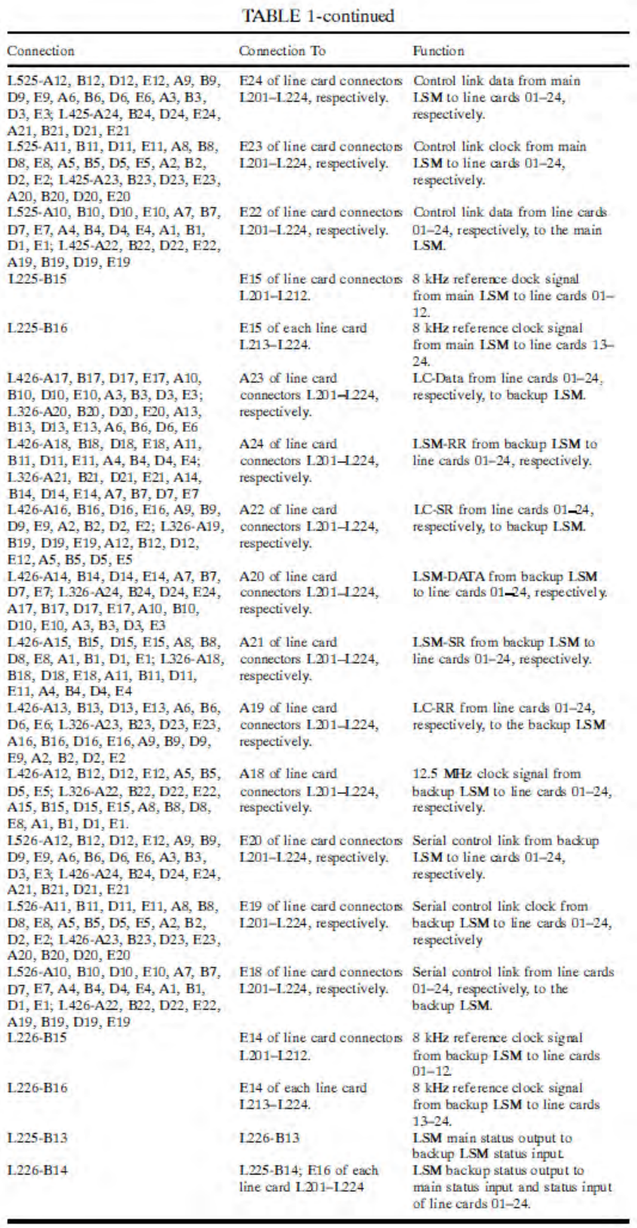
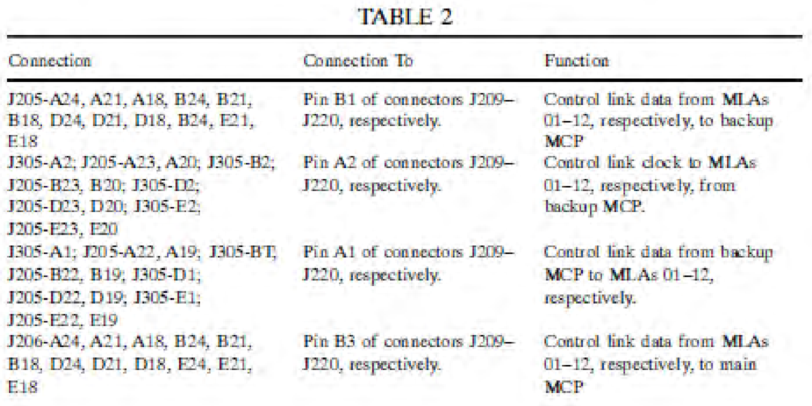


TABLE 1-continued

Connection	Connection To	Function
L525-A12, B12, D12, E12, A9, B9, D9, E9, A6, B6, D6, E6, A3, B3, D3, E3; L425-A24, B24, D24, E24, A21, B21, D21, E21	E24 of line card connectors L201–L224, respectively.	Control link data from main LSM to line cards 01–24, respectively.
L525-A11, B11, D11, E11, A8, B8, D8, E8, A5, B5, D5, E5, A2, B2, D2, E2; L425-A23, B23, D23, E23, A20, B20, D20, E20	E23 of line card connectors L201–L224, respectively.	Control link clock from main LSM to line cards 01–24, respectively.
L525-A10, B10, D10, E10, A7, B7, D7, E7, A4, B4, D4, E4, A1, B1, D1, E1; L425-A22, B22, D22, E22, A19, B19, D19, E19	E22 of line card connectors L201–L224, respectively.	Control link data from line cards 01–24, respectively, to the main LSM.
L225-B15	E15 of line card connectors L201–L212.	8 kHz reference dock signal from main LSM to line cards 01–12.
L225-B16	E15 of each line card L213–L224.	8 kHz reference clock signal from main LSM to line cards 13–24.
L426-A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3; L326-A20, B20, D20, E20, A13, B13, D13, E13, A6, B6, D6, E6	A23 of line card connectors L201–L224, respectively.	LC-Data from line cards 01–24, respectively, to backup LSM.
L426-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4; L326-A21, B21, D21, E21, A14, B14, D14, E14, A7, B7, D7, E7	A24 of line card connectors L201–L224, respectively.	LSM-RR from backup LSM to line cards 01–24, respectively.
L426-A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2; L326-A19, B19, D19, E19, A12, B12, D12, E12, A5, B5, D5, E5	A22 of line card connectors L201–L224, respectively.	LC-SR from line cards 01–24, respectively, to backup LSM.
L426-A14, B14, D14, E14, A7, B7, D7, E7; L326-A24, B24, D24, E24, A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3	A20 of line card connectors L201–L224, respectively.	LSM-DATA from backup LSM to line cards 01–24, respectively.
L426-A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1; L326-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4	A21 of line card connectors L201–L224, respectively.	LSM-SR from backup LSM to line cards 01–24, respectively.
L426-A13, B13, D13, E13, A6, B6, D6, E6; L326-A23, B23, D23, E23, A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2	A19 of line card connectors L201–L224, respectively.	LC-RR from line cards 01–24, respectively, to the backup LSM
L426-A12, B12, D12, E12, A5, B5, D5, E5; L326-A22, B22, D22, E22, A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1.	A18 of line card connectors L201–L224, respectively.	12.5 MHz clock signal from backup LSM to line cards 01–24, respectively.
L526-A12, B12, D12, E12, A9, B9, D9, E9, A6, B6, D6, E6, A3, B3, D3, E3; L426-A24, B24, D24, E24, A21, B21, D21, E21	E20 of line card connectors L201–L224, respectively.	Serial control link from backup LSM to line cards 01–24, respectively.
L526-A11, B11, D11, E11, A8, B8, D8, E8, A5, B5, D5, E5, A2, B2, D2, E2; L426-A23, B23, D23, E23, A20, B20, D20, E20	E19 of line card connectors L201–L224, respectively.	Serial control link clock from backup LSM to line cards 01–24, respectively
L526-A10, B10, D10, E10, A7, B7, D7, E7, A4, B4, D4, E4, A1, B1, D1, E1; L426-A22, B22, D22, E22, A19, B19, D19, E19	E18 of line card connectors L201–L224, respectively.	Serial control link from line cards 01–24, respectively, to the backup LSM.
L226-B15	E14 of line card connectors L201–L212.	8 kHz reference clock signal from backup LSM to line cards 01–12.
L226-B16	E14 of each line card L213–L224.	8 kHz reference clock signal from backup LSM to line cards 13–24.
L225-B13	L226-B13	LSM main status output to backup LSM status input.
L226-B14	L225-B14; E16 of each line card L201–L224	LSM backup status output to main status input and status input of line cards 01–24.

<table>
<tr><th>No.</th><th>’740 Patent Claim 10</th><th>Wiher ’530</th></tr>
<tr><td></td><td></td><td>
<p>Wiher ’530 at 20:35-57 (“Trunk card interface control circuitry 1301 process ATM cell transfer signals exchanged over the main trunk card interface 1310 and backup trunk card interface 1315. Inter-face control circuitry 1301 provides, for example, ATM cell buffering and control of ATM cell exchange over interfaces 1310 and 1315. Additionally, interface circuitry 1301 may extract or insert OAMP cells to be exchanged between the processor 1305 and a trunk card. The interface control circuitry may also send ATM cells to and receive ATM cells from header translation circuitry 1303. Header translation circuitry 1303 performs VPINCI header field translation and other ATM cell header manipulation functions. Header translation circuitry 1303 may determine appropriate header manipulations based on, for example, programs and trans-lation tables stored in RAM and ROM memory 1304. Memory 1304 may include programs and data stored by processor 1305. Additionally, header translation circuitry 1303 interfaces with LSM interface control circuitry 1302. LSM interface control circuitry 1302 may extract or insert OAMP cells to be exchanged between the processor 1305 and a LSM. Additionally LSM interface control circuitry 1302 regulates ATM cell transport on, for example, a fiber optic SONET OC-3c interface 1330 to a LSM.”)</p>
<p>Wiher ’530 at Table 2</p>

<p>TABLE 2</p>
<table>
<tr><th>Connection</th><th>Connection To</th><th>Function</th></tr>
<tr><td>J205-A24, A21, A18, B24, B21, B18, D24, D21, D18, B24, E21, E18</td><td>Pin B1 of connectors J209–J220, respectively.</td><td>Control link data from MLAs 01–12, respectively, to backup MCP</td></tr>
<tr><td>J305-A2; J205-A23, A20; J305-B2; J205-B23, B20; J305-D2; J205-D23, D20; J305-E2; J205-E23, E20</td><td>Pin A2 of connectors J209–J220, respectively.</td><td>Control link clock to MLAs 01–12, respectively, from backup MCP.</td></tr>
<tr><td>J305-A1; J205-A22, A19; J305-BT, J205-B22, B19; J305-D1; J205-D22, D19; J305-E1; J205-E22, E19</td><td>Pin A1 of connectors J209–J220, respectively.</td><td>Control link data from backup MCP to MLAs 01–12, respectively.</td></tr>
<tr><td>J206-A24, A21, A18, B24, B21, B18, D24, D21, D18, E24, E21, E18</td><td>Pin B3 of connectors J209–J220, respectively.</td><td>Control link data from MLAs 01–12, respectively, to main MCP</td></tr>
</table>
</td></tr>
</table>

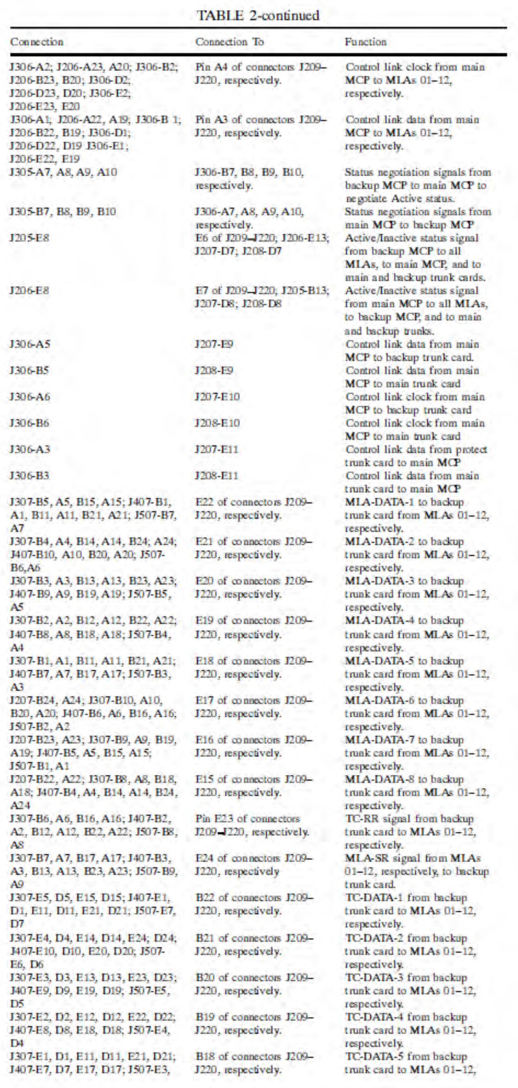
No.	’740 Patent Claim 10	Wiher ’530
		

TABLE 2-continued

Connection	Connection To	Function
J306-A2; J206-A23, A20; J306-B2; J206-B23, B20; J306-D2; J206-D23, D20; J306-E2; J206-E23, E20	Pin A4 of connectors J209–J220, respectively.	Control link clock from main MCP to MLAs 01–12, respectively.
J306-A1; J206-A22, A19; J306-B 1; J206-B22, B19; J306-D1; J206-D22, D19 J306-E1; J206-E22, E19	Pin A3 of connectors J209–J220, respectively.	Control link data from main MCP to MLAs 01–12, respectively.
J305-A7, A8, A9, A10	J306-B7, B8, B9, B10, respectively.	Status negotiation signals from backup MCP to main MCP to negotiate Active status.
J305-B7, B8, B9, B10	J306-A7, A8, A9, A10, respectively.	Status negotiation signals from main MCP to backup MCP
J205-E8	E6 of J209–J220; J206-E13; J207-D7; J208-D7	Active/Inactive status signal from backup MCP to all MLAs, to main MCP, and to main and backup trunk cards.
J206-E8	E7 of J209–J220; J205-B13; J207-D8; J208-D8	Active/Inactive status signal from main MCP to all MLAs, to backup MCP, and to main and backup trunks.
J306-A5	J207-E9	Control link data from main MCP to backup trunk card.
J306-B5	J208-E9	Control link data from main MCP to main trunk card
J306-A6	J207-E10	Control link clock from main MCP to backup trunk card
J306-B6	J208-E10	Control link clock from main MCP to main trunk card
J306-A3	J207-E11	Control link data from protect trunk card to main MCP
J306-B3	J208-E11	Control link data from main trunk card to main MCP
J307-B5, A5, B15, A15; J407-B1, A1, B11, A11, B21, A21; J507-B7, A7	E22 of connectors J209–J220, respectively.	MLA-DATA-1 to backup trunk card from MLAs 01–12, respectively.
J307-B4, A4, B14, A14, B24; A24; J407-B10, A10, B20, A20; J507-B6,A6	E21 of connectors J209–J220, respectively.	MLA-DATA-2 to backup trunk card from MLAs 01–12, respectively.
J307-B3, A3, B13, A13, B23, A23; J407-B9, A9, B19, A19; J507-B5, A5	E20 of connectors J209–J220, respectively.	MLA-DATA-3 to backup trunk card from MLAs 01–12, respectively.
J307-B2, A2, B12, A12, B22, A22; J407-B8, A8, B18, A18; J507-B4, A4	E19 of connectors J209–J220, respectively.	MLA-DATA-4 to backup trunk card from MLAs 01–12, respectively.
J307-B1, A1, B11, A11, B21, A21; J407-B7, A7, B17, A17; J507-B3, A3	E18 of connectors J209–J220, respectively.	MLA-DATA-5 to backup trunk card from MLAs 01–12, respectively.
J207-B24, A24; J307-B10, A10, B20, A20; J407-B6, A6, B16, A16; J507-B2, A2	E17 of connectors J209–J220, respectively.	MLA-DATA-6 to backup trunk card from MLAs 01–12, respectively.
J207-B23, A23; J307-B9, A9, B19, A19; J407-B5, A5, B15, A15; J507-B1, A1	E16 of connectors J209–J220, respectively.	MLA-DATA-7 to backup trunk card from MLAs 01–12, respectively.
J207-B22, A22; J307-B8, A8, B18, A18; J407-B4, A4, B14, A14, B24, A24	E15 of connectors J209–J220, respectively.	MLA-DATA-8 to backup trunk card from MLAs 01–12, respectively.
J307-B6, A6, B16, A16; J407-B2, A2, B12, A12, B22, A22; J507-B8, A8	Pin E23 of connectors J209–J220, respectively.	TC-RR signal from backup trunk card to MLAs 01–12, respectively.
J307-B7, A7, B17, A17; J407-B3, A3, B13, A13, B23, A23; J507-B9, A9	E24 of connectors J209–J220, respectively	MLA-SR signal from MLAs 01–12, respectively, to backup trunk card.
J307-E5, D5, E15, D15; J407-E1, D1, E11, D11, E21, D21; J507-E7, D7	B22 of connectors J209–J220, respectively.	TC-DATA-1 from backup trunk card to MLAs 01–12, respectively.
J307-E4, D4, E14, D14, E24; D24; J407-E10, D10, E20, D20; J507-E6, D6	B21 of connectors J209–J220, respectively.	TC-DATA-2 from backup trunk card to MLAs 01–12, respectively.
J307-E3, D3, E13, D13, E23, D23; J407-E9, D9, E19, D19; J507-E5, D5	B20 of connectors J209–J220, respectively.	TC-DATA-3 from backup trunk card to MLAs 01–12, respectively.
J307-E2, D2, E12, D12, E22, D22; J407-E8, D8, E18, D18; J507-E4, D4	B19 of connectors J209–J220, respectively.	TC-DATA-4 from backup trunk card to MLAs 01–12, respectively.
J307-E1, D1, E11, D11, E21, D21; J407-E7, D7, E17, D17; J507-E3,	B18 of connectors J209–J220, respectively.	TC-DATA-5 from backup trunk card to MLAs 01–12,

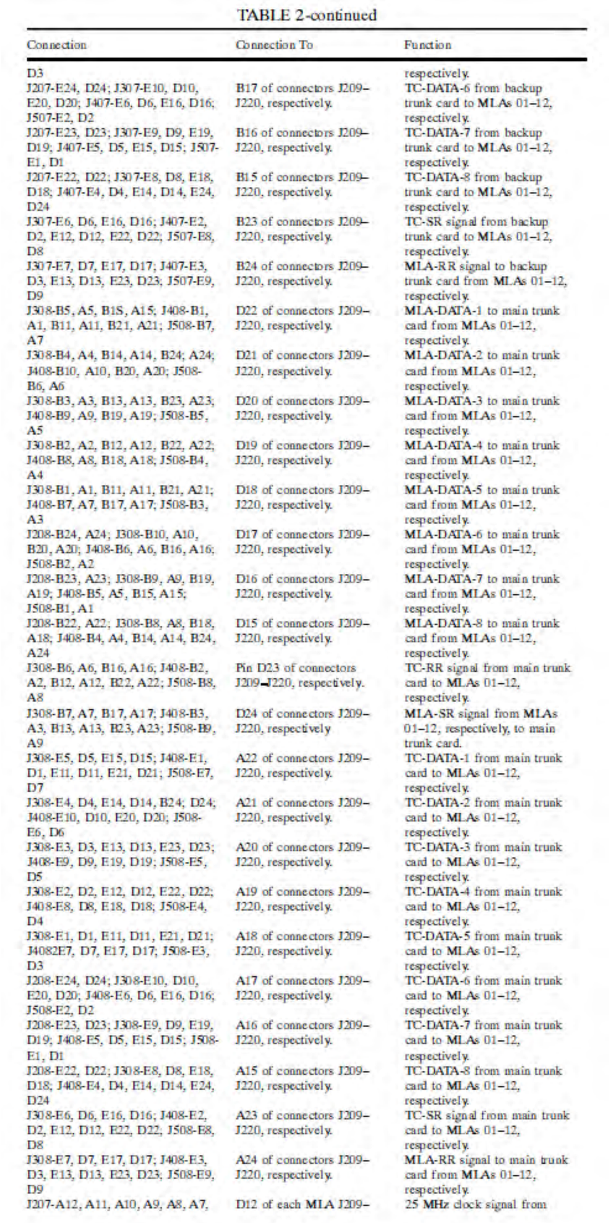


TABLE 2-continued

Connection	Connection To	Function
D3		respectively.
J207-E24, D24; J307-E10, D10, E20, D20; J407-E6, D6, E16, D16; J507-E2, D2	B17 of connectors J209–J220, respectively.	TC-DATA-6 from backup trunk card to MLAs 01–12, respectively.
J207-E23, D23; J307-E9, D9, E19, D19; J407-E5, D5, E15, D15; J507-E1, D1	B16 of connectors J209–J220, respectively.	TC-DATA-7 from backup trunk card to MLAs 01–12, respectively.
J207-E22, D22; J307-E8, D8, E18, D18; J407-E4, D4, E14, D14, E24, D24	B15 of connectors J209–J220, respectively.	TC-DATA-8 from backup trunk card to MLAs 01–12, respectively.
J307-E6, D6, E16, D16; J407-E2, D2, E12, D12, E22, D22; J507-E8, D8	B23 of connectors J209–J220, respectively.	TC-SR signal from backup trunk card to MLAs 01–12, respectively.
J307-E7, D7, E17, D17; J407-E3, D3, E13, D13, E23, D23; J507-E9, D9	B24 of connectors J209–J220, respectively.	MLA-RR signal to backup trunk card from MLAs 01–12, respectively.
J308-B5, A5, B1S, A15; J408-B1, A1, B11, A11, B21, A21; J508-B7, A7	D22 of connectors J209–J220, respectively.	MLA-DATA-1 to main trunk card from MLAs 01–12, respectively.
J308-B4, A4, B14, A14, B24; A24; J408-B10, A10, B20, A20; J508-B6, A6	D21 of connectors J209–J220, respectively.	MLA-DATA-2 to main trunk card from MLAs 01–12, respectively.
J308-B3, A3, B13, A13, B23, A23; J408-B9, A9, B19, A19; J508-B5, A5	D20 of connectors J209–J220, respectively.	MLA-DATA-3 to main trunk card from MLAs 01–12, respectively.
J308-B2, A2, B12, A12, B22, A22; J408-B8, A8, B18, A18; J508-B4, A4	D19 of connectors J209–J220, respectively.	MLA-DATA-4 to main trunk card from MLAs 01–12, respectively.
J308-B1, A1, B11, A11, B21, A21; J408-B7, A7, B17, A17; J508-B3, A3	D18 of connectors J209–J220, respectively.	MLA-DATA-5 to main trunk card from MLAs 01–12, respectively.
J208-B24, A24; J308-B10, A10, B20, A20; J408-B6, A6, B16, A16; J508-B2, A2	D17 of connectors J209–J220, respectively.	MLA-DATA-6 to main trunk card from MLAs 01–12, respectively.
J208-B23, A23; J308-B9, A9, B19, A19; J408-B5, A5, B15, A15; J508-B1, A1	D16 of connectors J209–J220, respectively.	MLA-DATA-7 to main trunk card from MLAs 01–12, respectively.
J208-B22, A22; J308-B8, A8, B18, A18; J408-B4, A4, B14, A14, B24, A24	D15 of connectors J209–J220, respectively.	MLA-DATA-8 to main trunk card from MLAs 01–12, respectively.
J308-B6, A6, B16, A16; J408-B2, A2, B12, A12, B22, A22; J508-B8, A8	Pin D23 of connectors J209–J220, respectively.	TC-RR signal from main trunk card to MLAs 01–12, respectively.
J308-B7, A7, B17, A17; J408-B3, A3, B13, A13, B23, A23; J508-B9, A9	D24 of connectors J209–J220, respectively	MLA-SR signal from MLAs 01–12, respectively, to main trunk card.
J308-E5, D5, E15, D15; J408-E1, D1, E11, D11, E21, D21; J508-E7, D7	A22 of connectors J209–J220, respectively.	TC-DATA-1 from main trunk card to MLAs 01–12, respectively.
J308-E4, D4, E14, D14, B24; D24; J408-E10, D10, E20, D20; J508-E6, D6	A21 of connectors J209–J220, respectively.	TC-DATA-2 from main trunk card to MLAs 01–12, respectively.
J308-E3, D3, E13, D13, E23, D23; J408-E9, D9, E19, D19; J508-E5, D5	A20 of connectors J209–J220, respectively.	TC-DATA-3 from main trunk card to MLAs 01–12, respectively.
J308-E2, D2, E12, D12, E22, D22; J408-E8, D8, E18, D18; J508-E4, D4	A19 of connectors J209–J220, respectively.	TC-DATA-4 from main trunk card to MLAs 01–12, respectively.
J308-E1, D1, E11, D11, E21, D21; J4082E7, D7, E17, D17; J508-E3, D3	A18 of connectors J209–J220, respectively.	TC-DATA-5 from main trunk card to MLAs 01–12, respectively.
J208-E24, D24; J308-E10, D10, E20, D20; J408-E6, D6, E16, D16; J508-E2, D2	A17 of connectors J209–J220, respectively.	TC-DATA-6 from main trunk card to MLAs 01–12, respectively.
J208-E23, D23; J308-E9, D9, E19, D19; J408-E5, D5, E15, D15; J508-E1, D1	A16 of connectors J209–J220, respectively.	TC-DATA-7 from main trunk card to MLAs 01–12, respectively.
J208-E22, D22; J308-E8, D8, E18, D18; J408-E4, D4, E14, D14, E24, D24	A15 of connectors J209–J220, respectively.	TC-DATA-8 from main trunk card to MLAs 01–12, respectively.
J308-E6, D6, E16, D16; J408-E2, D2, E12, D12, E22, D22; J508-E8, D8	A23 of connectors J209–J220, respectively.	TC-SR signal from main trunk card to MLAs 01–12, respectively.
J308-E7, D7, E17, D17; J408-E3, D3, E13, D13, E23, D23; J508-E9, D9	A24 of connectors J209–J220, respectively.	MLA-RR signal to main trunk card from MLAs 01–12, respectively.
J207-A12, A11, A10, A9, A8, A7,	D12 of each MLA J209–	25 MHz clock signal from

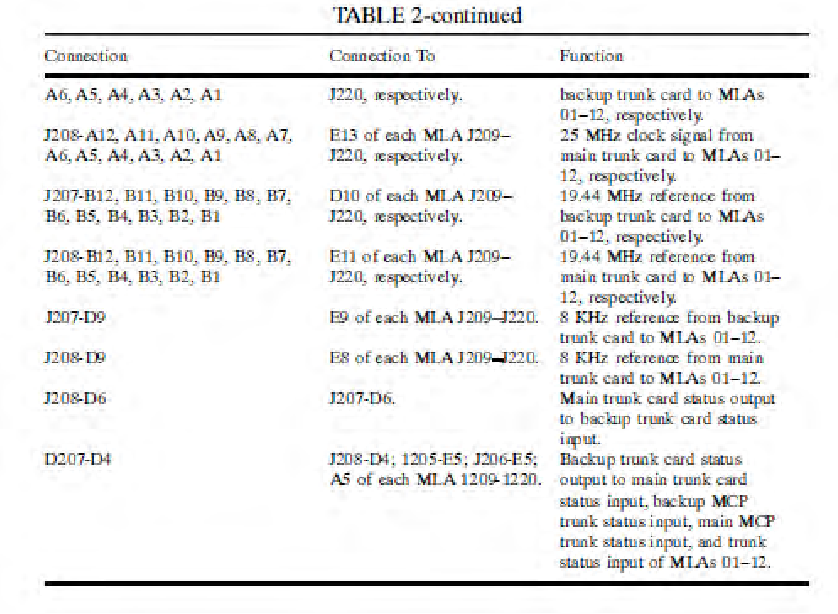
<table>
<tr><th>No.</th><th>’740 Patent Claim 10</th><th>Wiher ’530</th></tr>
<tr><td></td><td></td><td>


TABLE 2-continued

<table>
<tr><th>Connection</th><th>Connection To</th><th>Function</th></tr>
<tr><td>A6, A5, A4, A3, A2, A1</td><td>J220, respectively.</td><td>backup trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J208-A12, A11, A10, A9, A8, A7, A6, A5, A4, A3, A2, A1</td><td>E13 of each MLA J209–J220, respectively.</td><td>25 MHz clock signal from main trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J207-B12, B11, B10, B9, B8, B7, B6, B5, B4, B3, B2, B1</td><td>D10 of each MLA J209–J220, respectively.</td><td>19.44 MHz reference from backup trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J208-B12, B11, B10, B9, B8, B7, B6, B5, B4, B3, B2, B1</td><td>E11 of each MLA J209–J220, respectively.</td><td>19.44 MHz reference from main trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J207-D9</td><td>E9 of each MLA J209–J220.</td><td>8 KHz reference from backup trunk card to MLAs 01–12.</td></tr>
<tr><td>J208-D9</td><td>E8 of each MLA J209–J220.</td><td>8 KHz reference from main trunk card to MLAs 01–12.</td></tr>
<tr><td>J208-D6</td><td>J207-D6.</td><td>Main trunk card status output to backup trunk card status input.</td></tr>
<tr><td>D207-D4</td><td>J208-D4; 1205-E5; J206-E5; A5 of each MLA 1209-1220.</td><td>Backup trunk card status output to main trunk card status input, backup MCP trunk status input, main MCP trunk status input, and trunk status input of MLAs 01–12.</td></tr>
</table>

Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Wiher ’530 in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[b] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references.

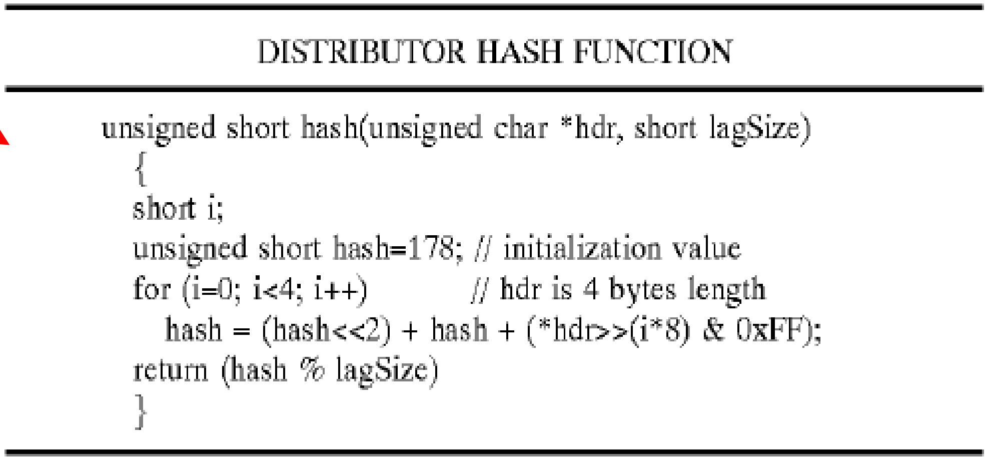
For example, Bruckman discloses applying a distributor hash function to the frame information which includes determining a number of the plurality of physical links.

Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in
</td></tr>
</table>

No.	’740 Patent Claim 10	Wiher ’530
		order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”) Bruckman at [0057] (“An aggregator 54 controls the link aggregation functions performed by equipment 22. A similar aggregator resides on node 24 (System B). Aggregator 54, too, may be a software process running on controller 42 or, alternatively, on a different embedded processor. Further alternatively or additionally, at least some of the functions of the aggregator may be carried out by hard-wired logic or by a program-mable logic component, such as a gate array. In the example shown in FIG. 2, aggregation group 36 comprises links Ll and L2, which are connected to LCl, and links L3 and L4, which are connected to LC2. This arrangement is advanta-geous in that it ensures that group 36 can continue to operate in the event not only of a facility failure (i.e., failure of one of links 30 in the group), but also of an equipment failure (i.e., a failure in one of the line cards). As a result of spreading group 36 over two (or more) line cards,

No.	’740 Patent Claim 10	Wiher ’530
		the link aggregation function applies not only to links 30 in group 36 but also to traces 52 that connect to multiplexers 50 that serve these links. Therefore, aggregator 54 resides on main card 32. Alternatively, if all the links in an aggregation group connect to the same multiplexer, the link aggregation func-tion may reside on line card 34.”) Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria: [0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link. [0060] The hash function uses all the specified input data from the frame headers. [0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values [0062] The hash function generates very different hash values for similar data.”) Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I: Bruckman at Table 1 (annotated)

No.	’740 Patent Claim 10	Wiher ’530

		hashing function *“mapping function”*
		
		DISTRIBUTOR HASH FUNCTION
		`unsigned short hash(unsigned char *hdr, short lagSize)`
		`{`
		`short i;`
		`unsigned short hash=178; // initialization value`
		`for (i=0; i<4; i++)          // hdr is 4 bytes length`
		`   hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);`
		`return (hash % lagSize)`
		`}`
		
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		
		For example, Alexander discloses applying a distributor hash function to the frame information which includes determining a hash key based on packet information.
		
		Alexander at 3:1-40 (“The hash function is preferably selected such that suc-cessive application of the hash function to all source and destination addresses expected to be seen by the Ethernet switch will produce a lowest value hash key, a highest value hash key, and a group of hash keys having intermediate values distributed evenly between the lowest and highest values.

No.	’740 Patent Claim 10	Wiher ’530
		The distribution table contains a separate port identifier look-up table for each aggregated grouping of outgoing ports. Advantageously, the hash key is an N bit hash key; and, each port identifier look-up table contains 2^N entries occupying 2^N consecutive locations, with each entry being an identifier of a particular one of the physical outgoing ports. Identifiers for particular outgoing ports are retrieved from the distribution table by extracting first and second N bit hash keys which form part of the retrieved destination and source address contexts respectively. The hash keys are combined to form an N bit connection identifier. The port identifier look-up table corresponding to the aggregated grouping represented by the retrieved destination address is selected, and the entry at the table location corresponding to the value of the N bit connection identifier is retrieved. If the address look-up table does not contain a destination address corresponding to the extracted destination address then first and second hash keys are produced by applying a hash function to the extracted source and destination addresses respectively. The hash keys are combined to form an N bit connection identifier. The incoming port on which the packet containing the extracted source address was received is identified. All of the aggregated groupings are scanned to identify all outgoing ports to which packets may be directed from the incoming port on which the packet was received. For each one of those outgoing ports, the port identifier look-up table corresponding to the aggregated grouping containing that outgoing port is selected, the entry at the table location corresponding to the value of the N bit connection identifier is retrieved, and the received packet is queued for outgoing transmission on the outgoing port corresponding to the retrieved entry.”) Alexander at 5:10-35 (“If a packet arrives bearing a source Ethernet MAC address that was not found in look-up table 12 by address resolution unit 10, learning function 16 is invoked to update look-up table 12 with the new address (i.e. processing branches along the "No" exit from FIG. 2, block 36). Learning function 16 first computes a hash function on the source Ethernet MAC address, generating an N-bit hash key ("partial connection identifier") from the 48-bit MAC address, where N is some small integer in the range of 3 to 8 (FIG. 2, block 38). The physical port on which the packet arrived is then determined. If the physical port is found to be associated with an aggregate group (i.e., it is one of a set of ports that have been bound into a single logical port), then the logical identifier assigned to the aggregate group is also determined. The hash key is then

No.	’740 Patent Claim 10	Wiher ’530
		stored into address look-up table 12 in conjunction with the actual Ethernet MAC address and the port identifier (FIG. 2, block 40). The physical port identifier is used if the port is not part of an aggregate group (i.e. if processing branched along the "No" exit from block 30 and through block 32), while the logical identifier is used for ports that have been aggregated (i.e. if processing branched along the "Yes" exit from block 30 and through block 34). The hash key and port identifier are considered to form the "context" for the given MAC address.”) Alexander at 5:36-46 (“The hash function should be selected to ensure an even distribution of hash key values over the range of MAC addresses that are expected to be seen by the Ethernet switch. As a specific example, the EXACT™ Ethernet switch system employs an exclusive-OR based hash function, wherein the 48-bit MAC address is divided into 16-bit blocks, which are then exclusive-ORed together to form a single 16-bit number; the 3 least significant bits (LSBs) of this number are taken to produce a 3-bit hash key. Other schemes such as CRC-based or checksum-based hashes may also be used.”) Alexander at 6:49-65 (“If the context information for the destination address indicates, however, that the target is an aggregate group (i.e. if processing branches along the "Yes" exit from FIG. 2, block 42) then the logical identifier assigned to the aggregate group is retrieved and is used to select the proper look-up table contained within the distribution table data structure. The hash keys (partial connection identifiers) stored into the contexts for the source and destination MAC addresses are obtained from address resolution unit 10 and combined to generate a "connection identifier" with the same number of bits (FIG. 2, block 44). (In the EXACT™ Ethernet switch, a Boolean exclusive-OR operation is used to combine the hash keys without increasing the number of bits.) This connection identifier is then used to index into the selected look-up table, and finally retrieve an actual physical port index on which the packet must be transmitted (FIG. 2, block 46).”)
10[c]	calculating a modulo of a division operation	Wiher ’530 discloses calculating a modulo of a division operation of the hashing key by the hashing size.

No.	’740 Patent Claim 10	Wiher ’530
	of the hashing key by the hashing size, and	For example, Wiher ’530 discloses header translation circuitry to process ATM cells then direct them to different destinations, including system physical links, based on the ATM cell header. A person of ordinary skill in the art would understand that applying a hash function includes determining parameters responsive to the system and packet features, and generating a result. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Wiher ’530 is found to not meet this limitation, calculating a modulo of a division operation of the hashing key by the hashing size would have been obvious to a person having ordinary skill in the art, as explained below. *See supra* Claim 9 Wiher ’530 at 8:53-9:9 (“Cells arriving at the interface control circuitry 520 may include operations, administration, maintenance, and provisioning (OAMP) data or may contain user-data. OAMP cells may be identified by the payload type indicator (PTI) field in the ATM cell header. The multiplexer circuitry 521 extracts OAMP cells and send them to the processor 527 while user-data cells are sent to header translation circuitry 522. Header translation circuitry 522 performs VPI/VCI header field translation and other ATM cell header manipulation functions. Header translation circuitry 522 may deter-mine appropriate header manipulations based on, for example, programs and translation tables stored in RAM and ROM memory 526. Memory 526 may include header manipulation programs and translation tables that are stored by processor 527. After processing by header translation circuitry 522, ATM cells are directed to master line shelf adapter (MLA) interface circuitry 525. MLA interface cir-cuitry 525 controls and buffers cells flowing from the address translation circuitry 522 to a MLA and controls, for example, ATM cell flow over a SO NET OC-3c interface 530 between the LSM 500 and a MLA. Interface circuitry 525 may also insert OAMP cells from the processor 527 for transmission to a MLA and extract OAMP cells received from a MLA.”) Wiher ’530 at 9:10-27 (“The LSM's MLA interface circuitry 525 may also receive ATM cells from a MLA. The MLA interface circuitry 525 may extract OAMP cells arriving from a MLA and send them to the processor 527. Data cells arriving at the interface 525 and destined to a line

No.	’740 Patent Claim 10	Wiher ’530
		card are sent to header translation circuitry 524 which may perform ATM cell header manipu-lations. Header translation circuitry 524 performs VPINCI header field translation and other ATM cell header manipu-lation functions. Header translation circuitry 524 may deter-mine appropriate header manipulations based on, for example, programs and translation tables stored in RAM and ROM memory 526. Following header translation, data cells may be sent to de-multiplexer circuitry 523. The de-multiplexer circuitry controls the flow of ATM cells to interface control circuitry 520. Processor 527 may also send OAMP cells to the de-multiplexer 523 for transfer to line cards. Interface control circuitry 520 then transmits the ATM cell to a line card.”) Wiher ’530 at 11:20-39 (“Data transfers from a LSM to a line card include line card port identification information (a "port address"). A port address is a fixed value associated with a particular line card transceiver or subscriber loop connection. For example, a line card supporting two subscriber loops has port addresses "Pl" and "P2" that are associated, respectively, with the first and second subscriber loop at the line card. Each subscriber loop at a particular line card has an associated port address that is unique with respect to the port addresses of other subscriber loops at that line card. However, port addresses at one line card need not be unique with respect to port addresses at another line card. In a data transfer between a LSM and a line card, a line card port address may be identified by, for example, additional data bytes added to the ATM cell or by information in a modified (non-standard) cell header. Unlike VPINCI addresses which are dynami-cally associated with a transceiver, port address are perma-nently assigned (that is, they are static). Thus, the use of port addresses can simplify cell routing through a line card by simplifying the processing and storage of cell routing data.”) Wiher ’530 at 12:64-13:26 (“Referring to FIGS. 6 and 8, a line card's main LSM interface 610 includes control link signal lines 613. Line card operations, administration, maintenance and provision-ing (OAMP) functions can be controlled by data sent over the control link signal lines 613 between the line card and the main LSM. The control link signal lines 613 include a clock signal line, a data receive signal line, and a data transmit signal line. The line card sends data to the LSM in serial fashion by modulating a signal over a data transmit signal line and receives modulated data from the LSM over a data receive signal line. Signals exchanged over the data receive and data transmit signal lines are, for example, asserted or de-asserted on the falling edge

No.	’740 Patent Claim 10	Wiher ’530
		of a 64 kilohertz (KHz) clock pulse received on the clock receive signal line and are sampled on the rising edge of a received clock pulse. The format for the data exchanged on the control link signal lines 613 may conform to the Open Systems Interconnection (OSI) High-level Data Link Control (HDLC) protocol. The HDLC protocol is described in ISO/IEC 3309:1991 (E), Information Technology-Telecommunications and Information Exchange Between Systems-High-level Data Link Control (HDLC) procedures-Frame Structure, Interna-tional Organization for Standardization, Fourth Edition, 1991-06-01. A line card's main LSM interface 610 may also include clock signal lines 614. The clock signal lines include, for example, a 12.5 MHz clock signal line and a 8 KHz telephone network reference timing signal line received from the main LSM. Signals exchanged over clock signal lines 614 may be used to time data transmission over signal lines 611 and 612. Wiher ’530 at Table 1

<table>
<tr><th>No.</th><th>’740 Patent Claim 10</th><th>Wiher ’530</th></tr>
<tr><td></td><td></td><td>

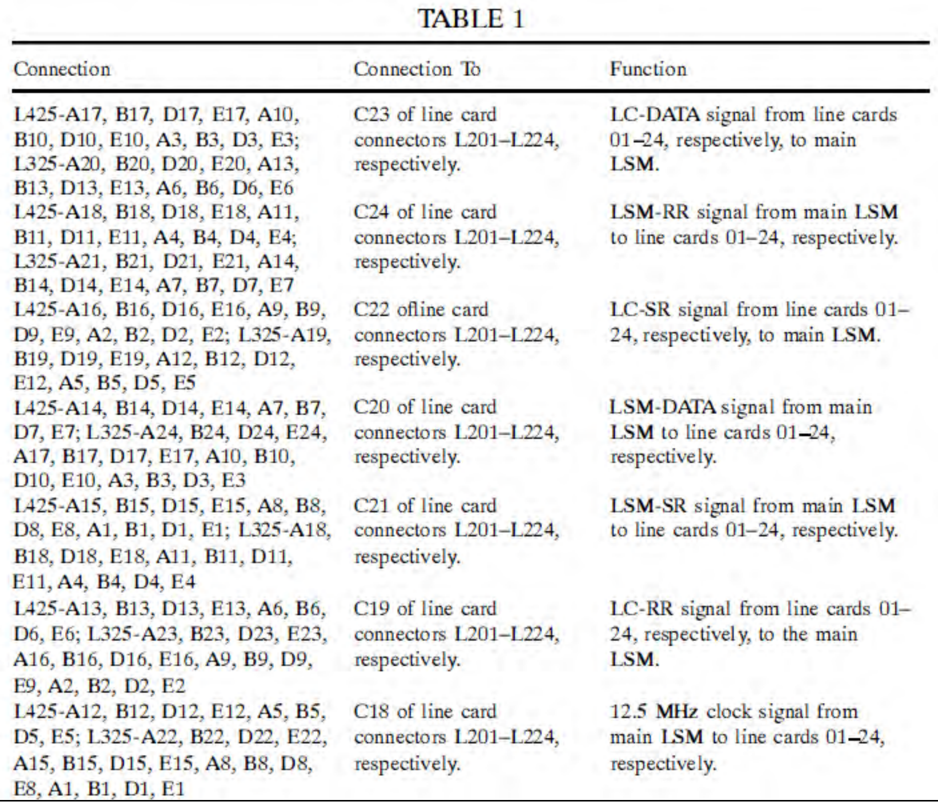



TABLE 1
<table>
<tr><th>Connection</th><th>Connection To</th><th>Function</th></tr>
<tr><td>L425-A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3; L325-A20, B20, D20, E20, A13, B13, D13, E13, A6, B6, D6, E6</td><td>C23 of line card connectors L201–L224, respectively.</td><td>LC-DATA signal from line cards 01–24, respectively, to main LSM.</td></tr>
<tr><td>L425-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4; L325-A21, B21, D21, E21, A14, B14, D14, E14, A7, B7, D7, E7</td><td>C24 of line card connectors L201–L224, respectively.</td><td>LSM-RR signal from main LSM to line cards 01–24, respectively.</td></tr>
<tr><td>L425-A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2; L325-A19, B19, D19, E19, A12, B12, D12, E12, A5, B5, D5, E5</td><td>C22 ofline card connectors L201–L224, respectively.</td><td>LC-SR signal from line cards 01–24, respectively, to main LSM.</td></tr>
<tr><td>L425-A14, B14, D14, E14, A7, B7, D7, E7; L325-A24, B24, D24, E24, A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3</td><td>C20 of line card connectors L201–L224, respectively.</td><td>LSM-DATA signal from main LSM to line cards 01–24, respectively.</td></tr>
<tr><td>L425-A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1; L325-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4</td><td>C21 of line card connectors L201–L224, respectively.</td><td>LSM-SR signal from main LSM to line cards 01–24, respectively.</td></tr>
<tr><td>L425-A13, B13, D13, E13, A6, B6, D6, E6; L325-A23, B23, D23, E23, A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2</td><td>C19 of line card connectors L201–L224, respectively.</td><td>LC-RR signal from line cards 01–24, respectively, to the main LSM.</td></tr>
<tr><td>L425-A12, B12, D12, E12, A5, B5, D5, E5; L325-A22, B22, D22, E22, A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1</td><td>C18 of line card connectors L201–L224, respectively.</td><td>12.5 MHz clock signal from main LSM to line cards 01–24, respectively.</td></tr>
</table>
</td></tr>
</table>

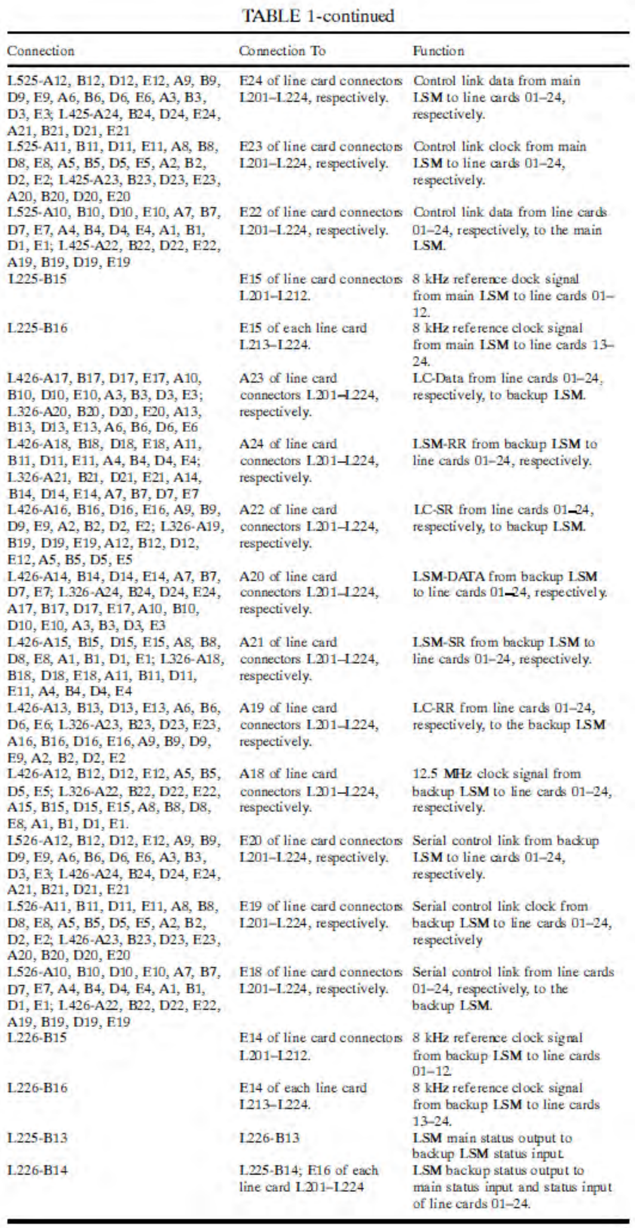


TABLE 1-continued

Connection	Connection To	Function
L525-A12, B12, D12, E12, A9, B9, D9, E9, A6, B6, D6, E6, A3, B3, D3, E3; L425-A24, B24, D24, E24, A21, B21, D21, E21	E24 of line card connectors L201–L224, respectively.	Control link data from main LSM to line cards 01–24, respectively.
L525-A11, B11, D11, E11, A8, B8, D8, E8, A5, B5, D5, E5, A2, B2, D2, E2; L425-A23, B23, D23, E23, A20, B20, D20, E20	E23 of line card connectors L201–L224, respectively.	Control link clock from main LSM to line cards 01–24, respectively.
L525-A10, B10, D10, E10, A7, B7, D7, E7, A4, B4, D4, E4, A1, B1, D1, E1; L425-A22, B22, D22, E22, A19, B19, D19, E19	E22 of line card connectors L201–L224, respectively.	Control link data from line cards 01–24, respectively, to the main LSM.
L225-B15	E15 of line card connectors L201–L212.	8 kHz reference dock signal from main LSM to line cards 01–12.
L225-B16	E15 of each line card L213–L224.	8 kHz reference clock signal from main LSM to line cards 13–24.
L426-A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3; L326-A20, B20, D20, E20, A13, B13, D13, E13, A6, B6, D6, E6	A23 of line card connectors L201–L224, respectively.	LC-Data from line cards 01–24, respectively, to backup LSM.
L426-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4; L326-A21, B21, D21, E21, A14, B14, D14, E14, A7, B7, D7, E7	A24 of line card connectors L201–L224, respectively.	LSM-RR from backup LSM to line cards 01–24, respectively.
L426-A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2; L326-A19, B19, D19, E19, A12, B12, D12, E12, A5, B5, D5, E5	A22 of line card connectors L201–L224, respectively.	LC-SR from line cards 01–24, respectively, to backup LSM.
L426-A14, B14, D14, E14, A7, B7, D7, E7; L326-A24, B24, D24, E24, A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3	A20 of line card connectors L201–L224, respectively.	LSM-DATA from backup LSM to line cards 01–24, respectively.
L426-A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1; L326-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4	A21 of line card connectors L201–L224, respectively.	LSM-SR from backup LSM to line cards 01–24, respectively.
L426-A13, B13, D13, E13, A6, B6, D6, E6; L326-A23, B23, D23, E23, A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2	A19 of line card connectors L201–L224, respectively.	LC-RR from line cards 01–24, respectively, to the backup LSM
L426-A12, B12, D12, E12, A5, B5, D5, E5; L326-A22, B22, D22, E22, A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1.	A18 of line card connectors L201–L224, respectively.	12.5 MHz clock signal from backup LSM to line cards 01–24, respectively.
L526-A12, B12, D12, E12, A9, B9, D9, E9, A6, B6, D6, E6, A3, B3, D3, E3; L426-A24, B24, D24, E24, A21, B21, D21, E21	E20 of line card connectors L201–L224, respectively.	Serial control link from backup LSM to line cards 01–24, respectively.
L526-A11, B11, D11, E11, A8, B8, D8, E8, A5, B5, D5, E5, A2, B2, D2, E2; L426-A23, B23, D23, E23, A20, B20, D20, E20	E19 of line card connectors L201–L224, respectively.	Serial control link clock from backup LSM to line cards 01–24, respectively
L526-A10, B10, D10, E10, A7, B7, D7, E7, A4, B4, D4, E4, A1, B1, D1, E1; L426-A22, B22, D22, E22, A19, B19, D19, E19	E18 of line card connectors L201–L224, respectively.	Serial control link from line cards 01–24, respectively, to the backup LSM.
L226-B15	E14 of line card connectors L201–L212.	8 kHz reference clock signal from backup LSM to line cards 01–12.
L226-B16	E14 of each line card L213–L224.	8 kHz reference clock signal from backup LSM to line cards 13–24.
L225-B13	L226-B13	LSM main status output to backup LSM status input.
L226-B14	L225-B14; E16 of each line card L201–L224	LSM backup status output to main status input and status input of line cards 01–24.

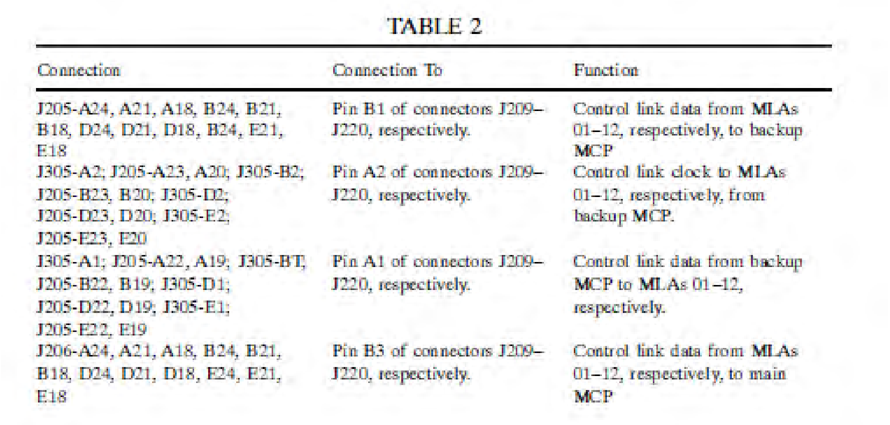
No.	’740 Patent Claim 10	Wiher ’530
		Wiher ’530 at 20:35-57 (“Trunk card interface control circuitry 1301 process ATM cell transfer signals exchanged over the main trunk card interface 1310 and backup trunk card interface 1315. Inter-face control circuitry 1301 provides, for example, ATM cell buffering and control of ATM cell exchange over interfaces 1310 and 1315. Additionally, interface circuitry 1301 may extract or insert OAMP cells to be exchanged between the processor 1305 and a trunk card. The interface control circuitry may also send ATM cells to and receive ATM cells from header translation circuitry 1303. Header translation circuitry 1303 performs VPINCI header field translation and other ATM cell header manipulation functions. Header translation circuitry 1303 may determine appropriate header manipulations based on, for example, programs and trans-lation tables stored in RAM and ROM memory 1304. Memory 1304 may include programs and data stored by processor 1305. Additionally, header translation circuitry 1303 interfaces with LSM interface control circuitry 1302. LSM interface control circuitry 1302 may extract or insert OAMP cells to be exchanged between the processor 1305 and a LSM. Additionally LSM interface control circuitry 1302 regulates ATM cell transport on, for example, a fiber optic SONET OC-3c interface 1330 to a LSM.”) Wiher ’530 at Table 2 

TABLE 2

Connection	Connection To	Function
J205-A24, A21, A18, B24, B21, B18, D24, D21, D18, B24, E21, E18	Pin B1 of connectors J209–J220, respectively.	Control link data from MLAs 01–12, respectively, to backup MCP
J305-A2; J205-A23, A20; J305-B2; J205-B23, B20; J305-D2; J205-D23, D20; J305-E2; J205-E23, E20	Pin A2 of connectors J209–J220, respectively.	Control link clock to MLAs 01–12, respectively, from backup MCP.
J305-A1; J205-A22, A19; J305-BT, J205-B22, B19; J305-D1; J205-D22, D19; J305-E1; J205-E22, E19	Pin A1 of connectors J209–J220, respectively.	Control link data from backup MCP to MLAs 01–12, respectively.
J206-A24, A21, A18, B24, B21, B18, D24, D21, D18, E24, E21, E18	Pin B3 of connectors J209–J220, respectively.	Control link data from MLAs 01–12, respectively, to main MCP

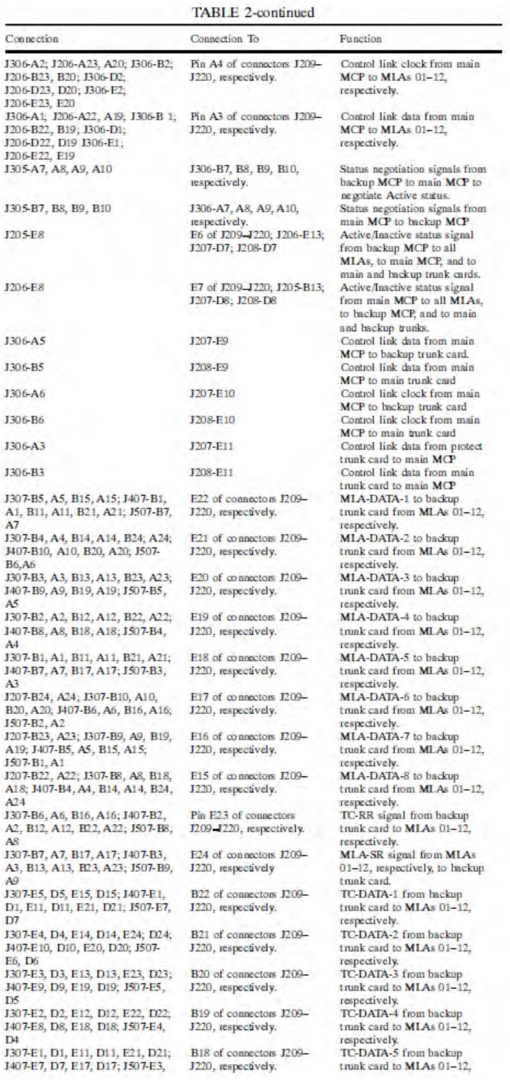
No.	’740 Patent Claim 10	Wiher ’530
		

TABLE 2-continued

Connection	Connection To	Function
J306-A2; J206-A23, A20; J306-B2; J206-B23, B20; J306-D2; J206-D23, D20; J306-E2; J206-E23, E20	Pin A4 of connectors J209–J220, respectively.	Control link clock from main MCP to MLAs 01–12, respectively.
J306-A1; J206-A22, A19; J306-B 1; J206-B22, B19; J306-D1; J206-D22, D19 J306-E1; J206-E22, E19	Pin A3 of connectors J209–J220, respectively.	Control link data from main MCP to MLAs 01–12, respectively.
J305-A7, A8, A9, A10	J306-B7, B8, B9, B10, respectively.	Status negotiation signals from backup MCP to main MCP to negotiate Active status.
J305-B7, B8, B9, B10	J306-A7, A8, A9, A10, respectively.	Status negotiation signals from main MCP to backup MCP
J205-E8	E6 of J209–J220; J206-E13; J207-D7; J208-D7	Active/Inactive status signal from backup MCP to all MLAs, to main MCP, and to main and backup trunk cards.
J206-E8	E7 of J209–J220; J205-B13; J207-D8; J208-D8	Active/Inactive status signal from main MCP to all MLAs, to backup MCP, and to main and backup trunks.
J306-A5	J207-E9	Control link data from main MCP to backup trunk card.
J306-B5	J208-E9	Control link data from main MCP to main trunk card
J306-A6	J207-E10	Control link clock from main MCP to backup trunk card
J306-B6	J208-E10	Control link clock from main MCP to main trunk card
J306-A3	J207-E11	Control link data from protect trunk card to main MCP
J306-B3	J208-E11	Control link data from main trunk card to main MCP
J307-B5, A5, B15, A15; J407-B1, A1, B11, A11, B21, A21; J507-B7, A7	E22 of connectors J209–J220, respectively.	MLA-DATA-1 to backup trunk card from MLAs 01–12, respectively.
J307-B4, A4, B14, A14, B24; A24; J407-B10, A10, B20, A20; J507-B6,A6	E21 of connectors J209–J220, respectively.	MLA-DATA-2 to backup trunk card from MLAs 01–12, respectively.
J307-B3, A3, B13, A13, B23, A23; J407-B9, A9, B19, A19; J507-B5, A5	E20 of connectors J209–J220, respectively.	MLA-DATA-3 to backup trunk card from MLAs 01–12, respectively.
J307-B2, A2, B12, A12, B22, A22; J407-B8, A8, B18, A18; J507-B4, A4	E19 of connectors J209–J220, respectively.	MLA-DATA-4 to backup trunk card from MLAs 01–12, respectively.
J307-B1, A1, B11, A11, B21, A21; J407-B7, A7, B17, A17; J507-B3, A3	E18 of connectors J209–J220, respectively.	MLA-DATA-5 to backup trunk card from MLAs 01–12, respectively.
J207-B24, A24; J307-B10, A10, B20, A20; J407-B6, A6, B16, A16; J507-B2, A2	E17 of connectors J209–J220, respectively.	MLA-DATA-6 to backup trunk card from MLAs 01–12, respectively.
J207-B23, A23; J307-B9, A9, B19, A19; J407-B5, A5, B15, A15; J507-B1, A1	E16 of connectors J209–J220, respectively.	MLA-DATA-7 to backup trunk card from MLAs 01–12, respectively.
J207-B22, A22; J307-B8, A8, B18, A18; J407-B4, A4, B14, A14, B24, A24	E15 of connectors J209–J220, respectively.	MLA-DATA-8 to backup trunk card from MLAs 01–12, respectively.
J307-B6, A6, B16, A16; J407-B2, A2, B12, A12, B22, A22; J507-B8, A8	Pin E23 of connectors J209–J220, respectively.	TC-RR signal from backup trunk card to MLAs 01–12, respectively.
J307-B7, A7, B17, A17; J407-B3, A3, B13, A13, B23, A23; J507-B9, A9	E24 of connectors J209–J220, respectively	MLA-SR signal from MLAs 01–12, respectively, to backup trunk card.
J307-E5, D5, E15, D15; J407-E1, D1, E11, D11, E21, D21; J507-E7, D7	B22 of connectors J209–J220, respectively.	TC-DATA-1 from backup trunk card to MLAs 01–12, respectively.
J307-E4, D4, E14, D14, E24; D24; J407-E10, D10, E20, D20; J507-E6, D6	B21 of connectors J209–J220, respectively.	TC-DATA-2 from backup trunk card to MLAs 01–12, respectively.
J307-E3, D3, E13, D13, E23, D23; J407-E9, D9, E19, D19; J507-E5, D5	B20 of connectors J209–J220, respectively.	TC-DATA-3 from backup trunk card to MLAs 01–12, respectively.
J307-E2, D2, E12, D12, E22, D22; J407-E8, D8, E18, D18; J507-E4, D4	B19 of connectors J209–J220, respectively.	TC-DATA-4 from backup trunk card to MLAs 01–12, respectively.
J307-E1, D1, E11, D11, E21, D21; J407-E7, D7, E17, D17; J507-E3,	B18 of connectors J209–J220, respectively.	TC-DATA-5 from backup trunk card to MLAs 01–12,

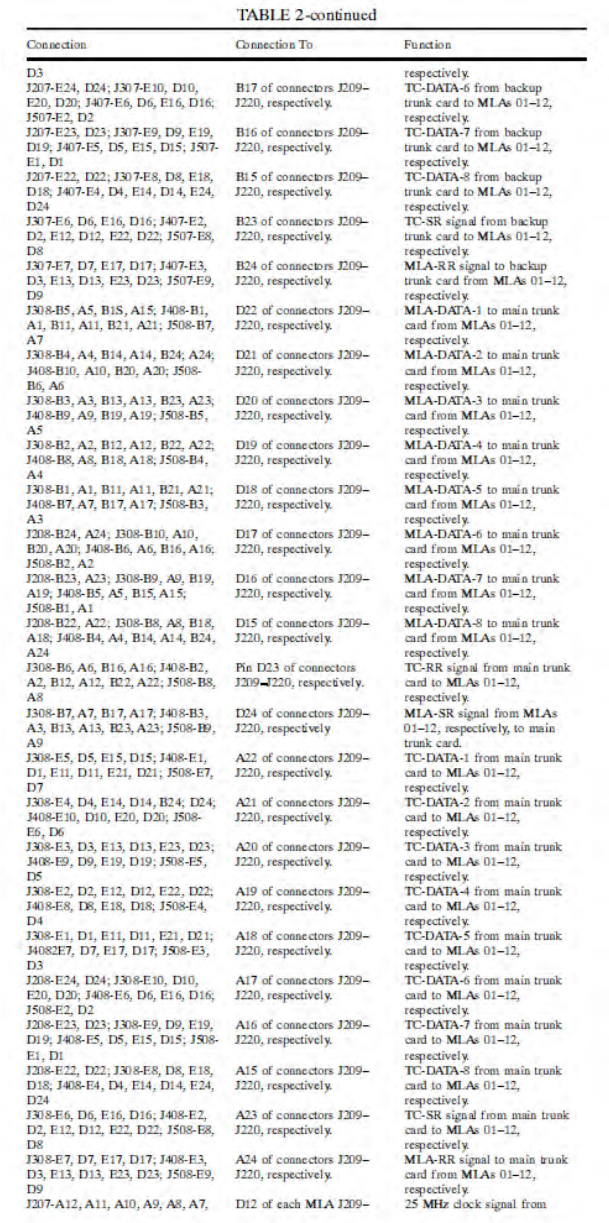
		

TABLE 2-continued

Connection	Connection To	Function
D3		respectively.
J207-E24, D24; J307-E10, D10, E20, D20; J407-E6, D6, E16, D16; J507-E2, D2	B17 of connectors J209–J220, respectively.	TC-DATA-6 from backup trunk card to MLAs 01–12, respectively.
J207-E23, D23; J307-E9, D9, E19, D19; J407-E5, D5, E15, D15; J507-E1, D1	B16 of connectors J209–J220, respectively.	TC-DATA-7 from backup trunk card to MLAs 01–12, respectively.
J207-E22, D22; J307-E8, D8, E18, D18; J407-E4, D4, E14, D14, E24, D24	B15 of connectors J209–J220, respectively.	TC-DATA-8 from backup trunk card to MLAs 01–12, respectively.
J307-E6, D6, E16, D16; J407-E2, D2, E12, D12, E22, D22; J507-E8, D8	B23 of connectors J209–J220, respectively.	TC-SR signal from backup trunk card to MLAs 01–12, respectively.
J307-E7, D7, E17, D17; J407-E3, D3, E13, D13, E23, D23; J507-E9, D9	B24 of connectors J209–J220, respectively.	MLA-RR signal to backup trunk card from MLAs 01–12, respectively.
J308-B5, A5, B1S, A15; J408-B1, A1, B11, A11, B21, A21; J508-B7, A7	D22 of connectors J209–J220, respectively.	MLA-DATA-1 to main trunk card from MLAs 01–12, respectively.
J308-B4, A4, B14, A14, B24; A24; J408-B10, A10, B20, A20; J508-B6, A6	D21 of connectors J209–J220, respectively.	MLA-DATA-2 to main trunk card from MLAs 01–12, respectively.
J308-B3, A3, B13, A13, B23, A23; J408-B9, A9, B19, A19; J508-B5, A5	D20 of connectors J209–J220, respectively.	MLA-DATA-3 to main trunk card from MLAs 01–12, respectively.
J308-B2, A2, B12, A12, B22, A22; J408-B8, A8, B18, A18; J508-B4, A4	D19 of connectors J209–J220, respectively.	MLA-DATA-4 to main trunk card from MLAs 01–12, respectively.
J308-B1, A1, B11, A11, B21, A21; J408-B7, A7, B17, A17; J508-B3, A3	D18 of connectors J209–J220, respectively.	MLA-DATA-5 to main trunk card from MLAs 01–12, respectively.
J208-B24, A24; J308-B10, A10, B20, A20; J408-B6, A6, B16, A16; J508-B2, A2	D17 of connectors J209–J220, respectively.	MLA-DATA-6 to main trunk card from MLAs 01–12, respectively.
J208-B23, A23; J308-B9, A9, B19, A19; J408-B5, A5, B15, A15; J508-B1, A1	D16 of connectors J209–J220, respectively.	MLA-DATA-7 to main trunk card from MLAs 01–12, respectively.
J208-B22, A22; J308-B8, A8, B18, A18; J408-B4, A4, B14, A14, B24, A24	D15 of connectors J209–J220, respectively.	MLA-DATA-8 to main trunk card from MLAs 01–12, respectively.
J308-B6, A6, B16, A16; J408-B2, A2, B12, A12, B22, A22; J508-B8, A8	Pin D23 of connectors J209–J220, respectively.	TC-RR signal from main trunk card to MLAs 01–12, respectively.
J308-B7, A7, B17, A17; J408-B3, A3, B13, A13, B23, A23; J508-B9, A9	D24 of connectors J209–J220, respectively	MLA-SR signal from MLAs 01–12, respectively, to main trunk card.
J308-E5, D5, E15, D15; J408-E1, D1, E11, D11, E21, D21; J508-E7, D7	A22 of connectors J209–J220, respectively.	TC-DATA-1 from main trunk card to MLAs 01–12, respectively.
J308-E4, D4, E14, D14, B24; D24; J408-E10, D10, E20, D20; J508-E6, D6	A21 of connectors J209–J220, respectively.	TC-DATA-2 from main trunk card to MLAs 01–12, respectively.
J308-E3, D3, E13, D13, E23, D23; J408-E9, D9, E19, D19; J508-E5, D5	A20 of connectors J209–J220, respectively.	TC-DATA-3 from main trunk card to MLAs 01–12, respectively.
J308-E2, D2, E12, D12, E22, D22; J408-E8, D8, E18, D18; J508-E4, D4	A19 of connectors J209–J220, respectively.	TC-DATA-4 from main trunk card to MLAs 01–12, respectively.
J308-E1, D1, E11, D11, E21, D21; J4082E7, D7, E17, D17; J508-E3, D3	A18 of connectors J209–J220, respectively.	TC-DATA-5 from main trunk card to MLAs 01–12, respectively.
J208-E24, D24; J308-E10, D10, E20, D20; J408-E6, D6, E16, D16; J508-E2, D2	A17 of connectors J209–J220, respectively.	TC-DATA-6 from main trunk card to MLAs 01–12, respectively.
J208-E23, D23; J308-E9, D9, E19, D19; J408-E5, D5, E15, D15; J508-E1, D1	A16 of connectors J209–J220, respectively.	TC-DATA-7 from main trunk card to MLAs 01–12, respectively.
J208-E22, D22; J308-E8, D8, E18, D18; J408-E4, D4, E14, D14, E24, D24	A15 of connectors J209–J220, respectively.	TC-DATA-8 from main trunk card to MLAs 01–12, respectively.
J308-E6, D6, E16, D16; J408-E2, D2, E12, D12, E22, D22; J508-E8, D8	A23 of connectors J209–J220, respectively.	TC-SR signal from main trunk card to MLAs 01–12, respectively.
J308-E7, D7, E17, D17; J408-E3, D3, E13, D13, E23, D23; J508-E9, D9	A24 of connectors J209–J220, respectively.	MLA-RR signal to main trunk card from MLAs 01–12, respectively.
J207-A12, A11, A10, A9, A8, A7,	D12 of each MLA J209–	25 MHz clock signal from

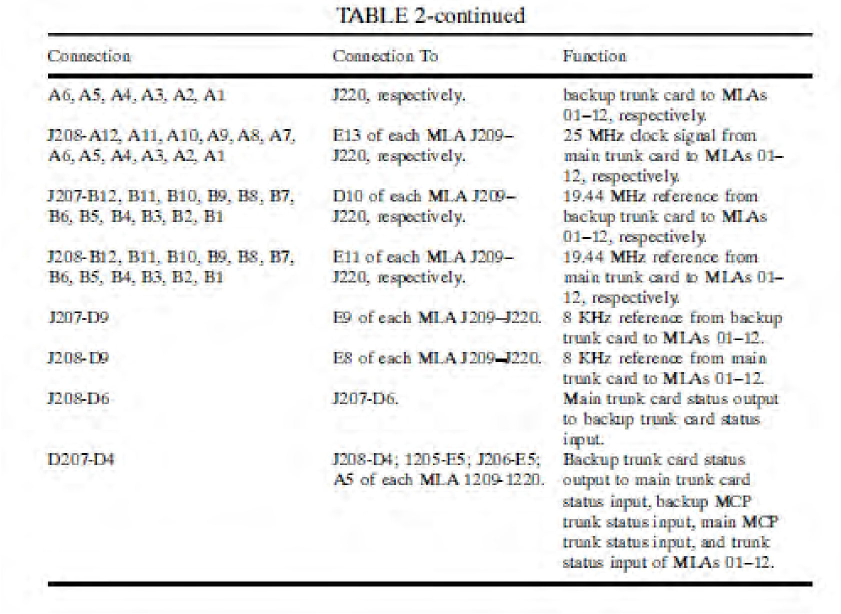
<table>
<tr><th>No.</th><th>’740 Patent Claim 10</th><th>Wiher ’530</th></tr>
<tr><td></td><td></td><td>


TABLE 2-continued

<table>
<tr><th>Connection</th><th>Connection To</th><th>Function</th></tr>
<tr><td>A6, A5, A4, A3, A2, A1</td><td>J220, respectively.</td><td>backup trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J208-A12, A11, A10, A9, A8, A7, A6, A5, A4, A3, A2, A1</td><td>E13 of each MLA J209–J220, respectively.</td><td>25 MHz clock signal from main trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J207-B12, B11, B10, B9, B8, B7, B6, B5, B4, B3, B2, B1</td><td>D10 of each MLA J209–J220, respectively.</td><td>19.44 MHz reference from backup trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J208-B12, B11, B10, B9, B8, B7, B6, B5, B4, B3, B2, B1</td><td>E11 of each MLA J209–J220, respectively.</td><td>19.44 MHz reference from main trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J207-D9</td><td>E9 of each MLA J209–J220.</td><td>8 KHz reference from backup trunk card to MLAs 01–12.</td></tr>
<tr><td>J208-D9</td><td>E8 of each MLA J209–J220.</td><td>8 KHz reference from main trunk card to MLAs 01–12.</td></tr>
<tr><td>J208-D6</td><td>J207-D6.</td><td>Main trunk card status output to backup trunk card status input.</td></tr>
<tr><td>D207-D4</td><td>J208-D4; 1205-E5; J206-E5; A5 of each MLA 1209-1220.</td><td>Backup trunk card status output to main trunk card status input, backup MCP trunk status input, main MCP trunk status input, and trunk status input of MLAs 01–12.</td></tr>
</table>

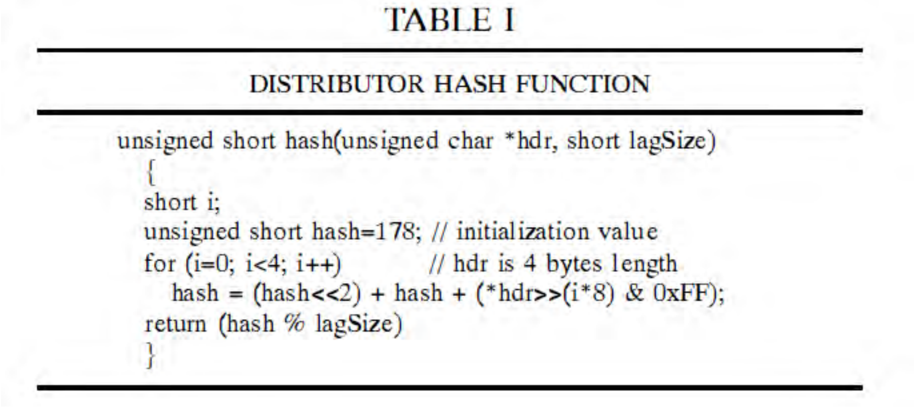
Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Wiher ’530 in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[c] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references.

For example, Bruckman discloses distributing data frames over physical links and traces based on a hash function involving a division operation (%), i.e., calculating a modulo of a division operation of the hashing key by the hashing size.

Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is
</td></tr>
</table>

No.	’740 Patent Claim 10	Wiher ’530
		difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0025] (“Typically, setting the protection policy includes determining a maximum number of the physical links that may fail while the logical link continues to provide at least the guaranteed bandwidth for the connection. In one embodiment, the guaranteed bandwidth is a bandwidth B, and the plurality of physical links consists of N links, and the maximum number is an integer P, and the link bandwidth allocated to each of the links is no less than B/(N-P). Conveying the data may further include managing the transmission of the data responsively to an actual number X of the physical links that have failed so that the guaranteed bandwidth on each of the links is limited to B/(N-X), X P, and an excess bandwidth on the physical links over the guaranteed bandwidth is available for other connections.”) Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”) Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:

No.	’740 Patent Claim 10	Wiher ’530

		
		TABLE I
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		”)
		
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		
		Bruckman at [0067] (“A similar problem may arise if there is a failure in a link in an aggregation group or in one of a number of line cards serving the aggregation group. In this case, to maintain the bandwidth allocation B made by CAC 44, each of the remaining links in the group must now carry, on average, B/(N-M) traffic, wherein M is the number of links in the group that are out of service. If only BIN has been allocated to each link, the remaining active links may not have sufficient bandwidth to continue to provide the bandwidth that has been guaranteed to the connections that they are required to carry. A similar problem arises with respect to loading of traces 52. For example, if there is a failure in LC2 or in one of links 30 in group 36 that connect to LC2, the trace connecting the multiplexer 50 in LCl will have to carry a substantially larger share of the bandwidth, or even all of the bandwidth, that is allocated to the connection in ques-tion.”)

No.	’740 Patent Claim 10	Wiher ’530
		Bruckman at [0068] (“FIG. 3 is a flow chart that schematically illustrates a method for dealing with these problems of fluctuating bandwidth requirements, in accordance with an embodiment of the present invention. In order to provide sufficient bandwidth for failure protection, CAC 44 uses a safety margin based on a protection parameter P, which is assigned at a protection setting step 60. P represents the maximum number of links in the group that can be out of service while still permitting the aggregation group to provide a given connection with the bandwidth that has been guaranteed to the connection. CAC 44 will then allocate at least B/(N-P) bandwidth to each link in the group, so that if P links fail, the group still provides total bandwidth of (N-P)*B/(N-P)= B. Setting P=l is equivalent to 1:N protection, so that the group will be unaffected by failure of a single link. In the example of group 36, shown in FIG. 2, setting P=2 will give both facility and equipment protection, i.e., the group will be unaffected not only by failure of a link, but also by failure of one of line cards 34. In the extreme case, in which P=N-1, CAC 44 will allocate the full bandwidth B on each link in the group.”) As another example, Singh discloses determining a ratio between the number of ingress and egress links and the number of links carrying data to the backplane and using a modulo to correspond to the channel’s link number. Singh at 9:30-43 (“The ratio between the number of line ingress links and the number of links carrying data to the backplane gives the backplane speedup for the system. In this example, there are 10 ingress links into the MS and 20 links (2 backplane channels) carrying that data to the backplane. This gives a backplane speedup of 2x. As another example, with 8 ingress links and 12 backplane links, there is a speedup of 1.5x. It should be noted that in addition to the backplane speedup, there is also an ingress/egress speedup. With 10 ingress links capable of carrying 2 Gbps each of raw data, this presents a 20 Gbps interface to the MS. An OC-192 only has approximately 10 Gbps worth of data. Taking into account cell overhead and cell quantization inefficiencies, there still remains excess capacity in the links.”) Singh at 11:29-38 (“FIG. 9 is a diagram illustrating link to channel assignments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of

No.	’740 Patent Claim 10	Wiher ’530
		the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a channel, whether it is a backplane channel or an ingress/egress channel. There is no logical relationship between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.”) Singh at 15:15-39 (“The number of crossbars that are required in a system is dependent on how many links are being used to create the backplane channels. There should be an even number of crossbars and they would be divided evenly across the switch cards. The following equation, for most cases, pro-vides the correct number of crossbars: # of Crossbars=(# links per ingress channelx# of ingress channels per portx# of port cardsx speedup)/32. For the 8x8 configuration, the # of crossbars should be multiplied by (4x# of iMS)/(# backplane channels per port card). The number of port cards should be rounded up to the nearest supported configuration, i.e. 8, 16, or 32. The speedup in the case of crossbars should be the fractional speedup that is desired. Example to determine the number of arbiters and cross-bars for the following system: 4 channel port cards (40 Gbps) 8 links per channel 16 port cards Speedup=l.5 # of arbiters=(4x2x2)/2=8 # of crossbars=(8x4x16xl.5)/32=24. This would give 3crossbars per arbiter.”) Singh at 16:28-44 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 16x16 and 32x32 is the organization of the switchplane. The port card remains the same. Backplane channels 1 and 2 are used for the backplane connectivity. Ingress and egress links 30-39 on the MS would not be used

No.	’740 Patent Claim 10	Wiher ’530
		and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 16, correspond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”) Singh at 17:31-49 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 8x8 and 16x16 is the organi-zation of the switchplane. The port card remains the same. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Links 0-7 and 24-31 on the arbiters would not be used and would be powered off. Links 0-7 and 24-31 on the crossbars would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Backplane channels 1 and 2 are used for the backplane connectivity. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 8, corre-spond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”)
10[d]	selecting the first and second physical links responsively to the modulo.	Wiher ’530 discloses selecting the first and second physical links responsively to the modulo. For example, Wiher ’530 discloses header translation circuitry to process ATM cells then direct them to different destinations, including system physical links, based on the ATM cell header. A person of ordinary skill in the art would understand that applying a hash function includes determining parameters responsive to the system and packet features, and generating a result. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Wiher ’530 is found to not meet this limitation, selecting the first and second physical links responsively to the modulo would have been obvious to a person having ordinary skill in the art, as explained below.

No.	’740 Patent Claim 10	Wiher ’530
		See supra Claim 9 Wiher ’530 at 8:53-9:9 (“Cells arriving at the interface control circuitry 520 may include operations, administration, maintenance, and provisioning (OAMP) data or may contain user-data. OAMP cells may be identified by the payload type indicator (PTI) field in the ATM cell header. The multiplexer circuitry 521 extracts OAMP cells and send them to the processor 527 while user-data cells are sent to header translation circuitry 522. Header translation circuitry 522 performs VPI/VCI header field translation and other ATM cell header manipulation functions. Header translation circuitry 522 may deter-mine appropriate header manipulations based on, for example, programs and translation tables stored in RAM and ROM memory 526. Memory 526 may include header manipulation programs and translation tables that are stored by processor 527. After processing by header translation circuitry 522, ATM cells are directed to master line shelf adapter (MLA) interface circuitry 525. MLA interface cir-cuitry 525 controls and buffers cells flowing from the address translation circuitry 522 to a MLA and controls, for example, ATM cell flow over a SO NET OC-3c interface 530 between the LSM 500 and a MLA. Interface circuitry 525 may also insert OAMP cells from the processor 527 for transmission to a MLA and extract OAMP cells received from a MLA.”) Wiher ’530 at 9:10-27 (“The LSM's MLA interface circuitry 525 may also receive ATM cells from a MLA. The MLA interface circuitry 525 may extract OAMP cells arriving from a MLA and send them to the processor 527. Data cells arriving at the interface 525 and destined to a line card are sent to header translation circuitry 524 which may perform ATM cell header manipu-lations. Header translation circuitry 524 performs VPINCI header field translation and other ATM cell header manipu-lation functions. Header translation circuitry 524 may deter-mine appropriate header manipulations based on, for example, programs and translation tables stored in RAM and ROM memory 526. Following header translation, data cells may be sent to de-multiplexer circuitry 523. The de-multiplexer circuitry controls the flow of ATM cells to interface control circuitry 520. Processor 527 may also send OAMP cells to the de-multiplexer 523 for transfer to line cards. Interface control circuitry 520 then transmits the ATM cell to a line card.”)

No.	’740 Patent Claim 10	Wiher ’530
		Wiher ’530 at 11:20-39 (“Data transfers from a LSM to a line card include line card port identification information (a "port address"). A port address is a fixed value associated with a particular line card transceiver or subscriber loop connection. For example, a line card supporting two subscriber loops has port addresses "Pl" and "P2" that are associated, respectively, with the first and second subscriber loop at the line card. Each subscriber loop at a particular line card has an associated port address that is unique with respect to the port addresses of other subscriber loops at that line card. However, port addresses at one line card need not be unique with respect to port addresses at another line card. In a data transfer between a LSM and a line card, a line card port address may be identified by, for example, additional data bytes added to the ATM cell or by information in a modified (non-standard) cell header. Unlike VPINCI addresses which are dynami-cally associated with a transceiver, port address are perma-nently assigned (that is, they are static). Thus, the use of port addresses can simplify cell routing through a line card by simplifying the processing and storage of cell routing data.”) Wiher ’530 at 12:64-13:26 (“Referring to FIGS. 6 and 8, a line card's main LSM interface 610 includes control link signal lines 613. Line card operations, administration, maintenance and provision-ing (OAMP) functions can be controlled by data sent over the control link signal lines 613 between the line card and the main LSM. The control link signal lines 613 include a clock signal line, a data receive signal line, and a data transmit signal line. The line card sends data to the LSM in serial fashion by modulating a signal over a data transmit signal line and receives modulated data from the LSM over a data receive signal line. Signals exchanged over the data receive and data transmit signal lines are, for example, asserted or de-asserted on the falling edge of a 64 kilohertz (KHz) clock pulse received on the clock receive signal line and are sampled on the rising edge of a received clock pulse. The format for the data exchanged on the control link signal lines 613 may conform to the Open Systems Interconnection (OSI) High-level Data Link Control (HDLC) protocol. The HDLC protocol is described in ISO/IEC 3309:1991 (E), Information Technology-Telecommunications and Information Exchange Between Systems-High-level Data Link Control (HDLC) procedures-Frame Structure, Interna-tional Organization for Standardization, Fourth Edition, 1991-06-01. A line card's main LSM interface 610 may also include clock signal lines 614. The clock signal lines include, for example, a 12.5 MHz clock signal line and a 8 KHz telephone network reference timing signal

<table>
<tr><th>No.</th><th>’740 Patent Claim 10</th><th>Wiher ’530</th></tr>
<tr><td></td><td></td><td>line received from the main LSM. Signals exchanged over clock signal lines 614 may be used to time data transmission over signal lines 611 and 612.
Wiher ’530 at Table 1

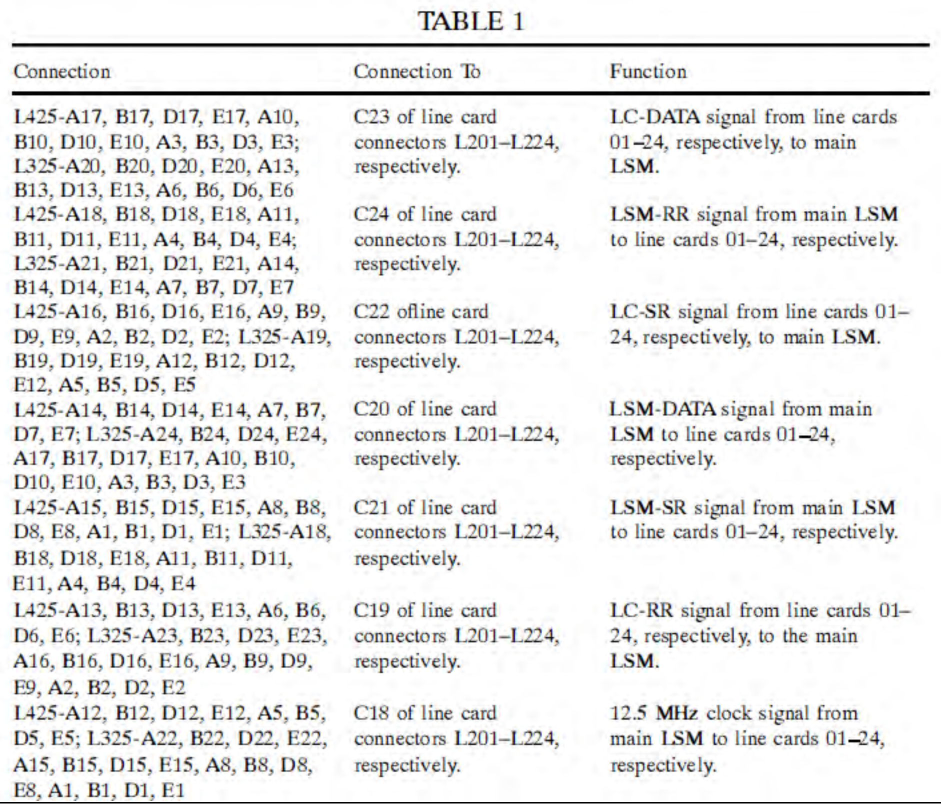



TABLE 1
<table>
<tr><th>Connection</th><th>Connection To</th><th>Function</th></tr>
<tr><td>L425-A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3; L325-A20, B20, D20, E20, A13, B13, D13, E13, A6, B6, D6, E6</td><td>C23 of line card connectors L201–L224, respectively.</td><td>LC-DATA signal from line cards 01–24, respectively, to main LSM.</td></tr>
<tr><td>L425-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4; L325-A21, B21, D21, E21, A14, B14, D14, E14, A7, B7, D7, E7</td><td>C24 of line card connectors L201–L224, respectively.</td><td>LSM-RR signal from main LSM to line cards 01–24, respectively.</td></tr>
<tr><td>L425-A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2; L325-A19, B19, D19, E19, A12, B12, D12, E12, A5, B5, D5, E5</td><td>C22 ofline card connectors L201–L224, respectively.</td><td>LC-SR signal from line cards 01–24, respectively, to main LSM.</td></tr>
<tr><td>L425-A14, B14, D14, E14, A7, B7, D7, E7; L325-A24, B24, D24, E24, A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3</td><td>C20 of line card connectors L201–L224, respectively.</td><td>LSM-DATA signal from main LSM to line cards 01–24, respectively.</td></tr>
<tr><td>L425-A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1; L325-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4</td><td>C21 of line card connectors L201–L224, respectively.</td><td>LSM-SR signal from main LSM to line cards 01–24, respectively.</td></tr>
<tr><td>L425-A13, B13, D13, E13, A6, B6, D6, E6; L325-A23, B23, D23, E23, A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2</td><td>C19 of line card connectors L201–L224, respectively.</td><td>LC-RR signal from line cards 01–24, respectively, to the main LSM.</td></tr>
<tr><td>L425-A12, B12, D12, E12, A5, B5, D5, E5; L325-A22, B22, D22, E22, A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1</td><td>C18 of line card connectors L201–L224, respectively.</td><td>12.5 MHz clock signal from main LSM to line cards 01–24, respectively.</td></tr>
</table></td></tr>
</table>

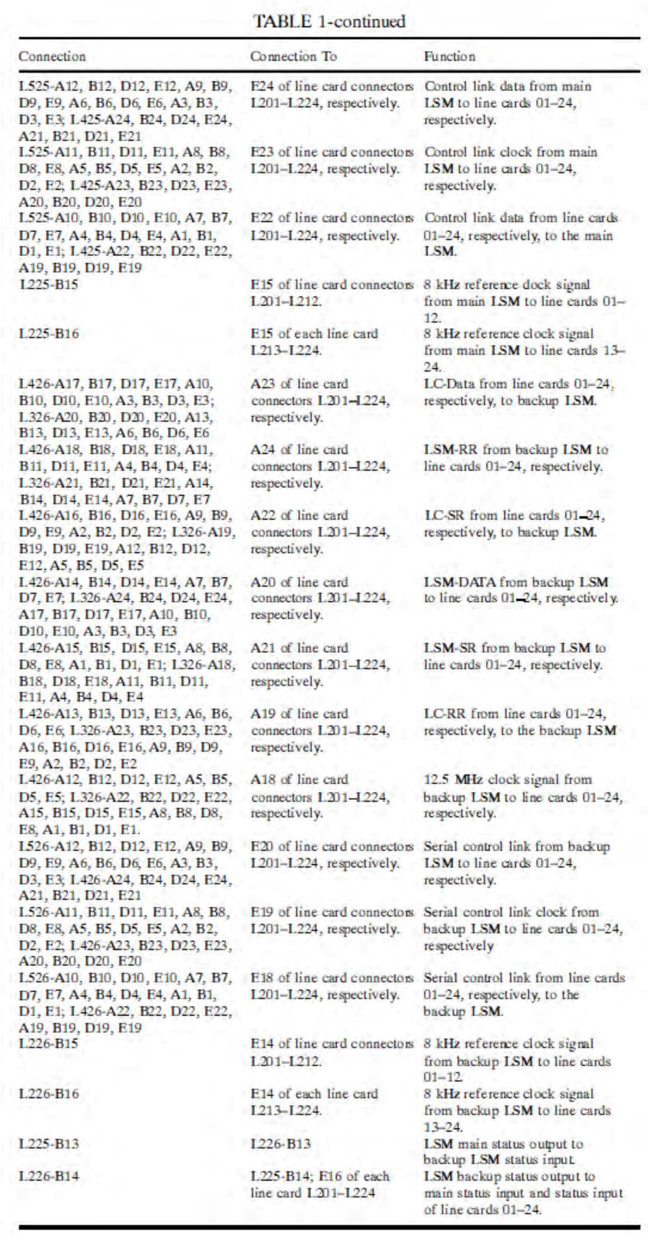


TABLE 1-continued

Connection	Connection To	Function
L525-A12, B12, D12, E12, A9, B9, D9, E9, A6, B6, D6, E6, A3, B3, D3, E3; L425-A24, B24, D24, E24, A21, B21, D21, E21	E24 of line card connectors L201–L224, respectively.	Control link data from main **LSM** to line cards 01–24, respectively.
L525-A11, B11, D11, E11, A8, B8, D8, E8, A5, B5, D5, E5, A2, B2, D2, E2; L425-A23, B23, D23, E23, A20, B20, D20, E20	E23 of line card connectors L201–L224, respectively.	Control link clock from main **LSM** to line cards 01–24, respectively.
L525-A10, B10, D10, E10, A7, B7, D7, E7, A4, B4, D4, E4, A1, B1, D1, E1; L425-A22, B22, D22, E22, A19, B19, D19, E19	E22 of line card connectors L201–L224, respectively.	Control link data from line cards 01–24, respectively, to the main LSM.
L225-B15	E15 of line card connectors L201–L212.	8 kHz reference dock signal from main **LSM** to line cards 01–12.
L225-B16	E15 of each line card L213–L224.	8 kHz reference clock signal from main **LSM** to line cards 13–24.
L426-A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3; L326-A20, B20, D20, E20, A13, B13, D13, E13, A6, B6, D6, E6	A23 of line card connectors L201–L224, respectively.	LC-Data from line cards 01–24, respectively, to backup **LSM**.
L426-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4; L326-A21, B21, D21, E21, A14, B14, D14, E14, A7, B7, D7, E7	A24 of line card connectors L201–L224, respectively.	LSM-RR from backup **LSM** to line cards 01–24, respectively.
L426-A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2; L326-A19, B19, D19, E19, A12, B12, D12, E12, A5, B5, D5, E5	A22 of line card connectors L201–L224, respectively.	LC-SR from line cards 01–24, respectively, to backup **LSM**.
L426-A14, B14, D14, E14, A7, B7, D7, E7; L326-A24, B24, D24, E24, A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3	A20 of line card connectors L201–L224, respectively.	LSM-DATA from backup **LSM** to line cards 01–24, respectively.
L426-A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1; L326-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4	A21 of line card connectors L201–L224, respectively.	LSM-SR from backup **LSM** to line cards 01–24, respectively.
L426-A13, B13, D13, E13, A6, B6, D6, E6; L326-A23, B23, D23, E23, A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2	A19 of line card connectors L201–L224, respectively.	LC-RR from line cards 01–24, respectively, to the backup **LSM**
L426-A12, B12, D12, E12, A5, B5, D5, E5; L326-A22, B22, D22, E22, A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1.	A18 of line card connectors L201–L224, respectively.	12.5 MHz clock signal from backup **LSM** to line cards 01–24, respectively.
L526-A12, B12, D12, E12, A9, B9, D9, E9, A6, B6, D6, E6, A3, B3, D3, E3; L426-A24, B24, D24, E24, A21, B21, D21, E21	E20 of line card connectors L201–L224, respectively.	Serial control link from backup **LSM** to line cards 01–24, respectively.
L526-A11, B11, D11, E11, A8, B8, D8, E8, A5, B5, D5, E5, A2, B2, D2, E2; L426-A23, B23, D23, E23, A20, B20, D20, E20	E19 of line card connectors L201–L224, respectively.	Serial control link clock from backup **LSM** to line cards 01–24, respectively
L526-A10, B10, D10, E10, A7, B7, D7, E7, A4, B4, D4, E4, A1, B1, D1, E1; L426-A22, B22, D22, E22, A19, B19, D19, E19	E18 of line card connectors L201–L224, respectively.	Serial control link from line cards 01–24, respectively, to the backup LSM.
L226-B15	E14 of line card connectors L201–L212.	8 kHz reference clock signal from backup **LSM** to line cards 01–12.
L226-B16	E14 of each line card L213–L224.	8 kHz reference clock signal from backup **LSM** to line cards 13–24.
L225-B13	L226-B13	**LSM** main status output to backup **LSM** status input.
L226-B14	L225-B14; E16 of each line card L201–L224	**LSM** backup status output to main status input and status input of line cards 01–24.

<table>
<tr><th>No.</th><th>’740 Patent Claim 10</th><th>Wiher ’530</th></tr>
<tr><td></td><td></td><td>

Wiher ’530 at 20:35-57 (“Trunk card interface control circuitry 1301 process ATM cell transfer signals exchanged over the main trunk card interface 1310 and backup trunk card interface 1315. Inter-face control circuitry 1301 provides, for example, ATM cell buffering and control of ATM cell exchange over interfaces 1310 and 1315. Additionally, interface circuitry 1301 may extract or insert OAMP cells to be exchanged between the processor 1305 and a trunk card. The interface control circuitry may also send ATM cells to and receive ATM cells from header translation circuitry 1303. Header translation circuitry 1303 performs VPINCI header field translation and other ATM cell header manipulation functions. Header translation circuitry 1303 may determine appropriate header manipulations based on, for example, programs and trans-lation tables stored in RAM and ROM memory 1304. Memory 1304 may include programs and data stored by processor 1305. Additionally, header translation circuitry 1303 interfaces with LSM interface control circuitry 1302. LSM interface control circuitry 1302 may extract or insert OAMP cells to be exchanged between the processor 1305 and a LSM. Additionally LSM interface control circuitry 1302 regulates ATM cell transport on, for example, a fiber optic SONET OC-3c interface 1330 to a LSM.”)

Wiher ’530 at Table 2

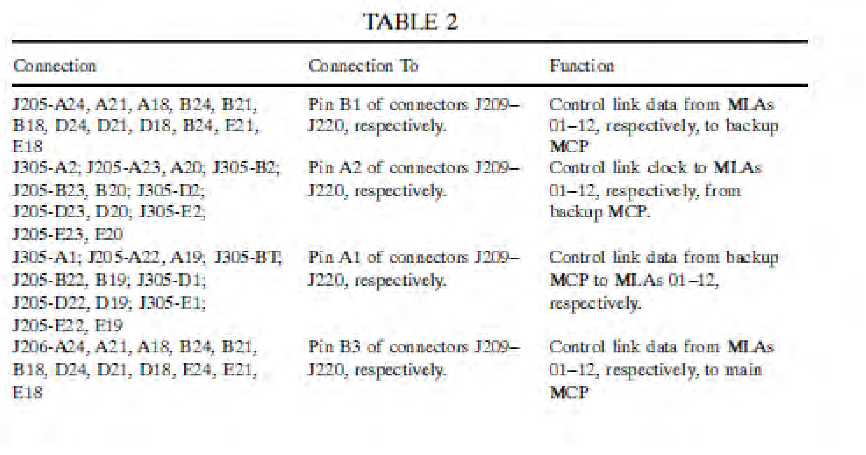


TABLE 2

<table>
<tr><th>Connection</th><th>Connection To</th><th>Function</th></tr>
<tr><td>J205-A24, A21, A18, B24, B21, B18, D24, D21, D18, B24, E21, E18</td><td>Pin B1 of connectors J209–J220, respectively.</td><td>Control link data from MLAs 01–12, respectively, to backup MCP</td></tr>
<tr><td>J305-A2; J205-A23, A20; J305-B2; J205-B23, B20; J305-D2; J205-D23, D20; J305-E2; J205-E23, E20</td><td>Pin A2 of connectors J209–J220, respectively.</td><td>Control link clock to MLAs 01–12, respectively, from backup MCP.</td></tr>
<tr><td>J305-A1; J205-A22, A19; J305-BT, J205-B22, B19; J305-D1; J205-D22, D19; J305-E1; J205-E22, E19</td><td>Pin A1 of connectors J209–J220, respectively.</td><td>Control link data from backup MCP to MLAs 01–12, respectively.</td></tr>
<tr><td>J206-A24, A21, A18, B24, B21, B18, D24, D21, D18, E24, E21, E18</td><td>Pin B3 of connectors J209–J220, respectively.</td><td>Control link data from MLAs 01–12, respectively, to main MCP</td></tr>
</table>

</td></tr>
</table>

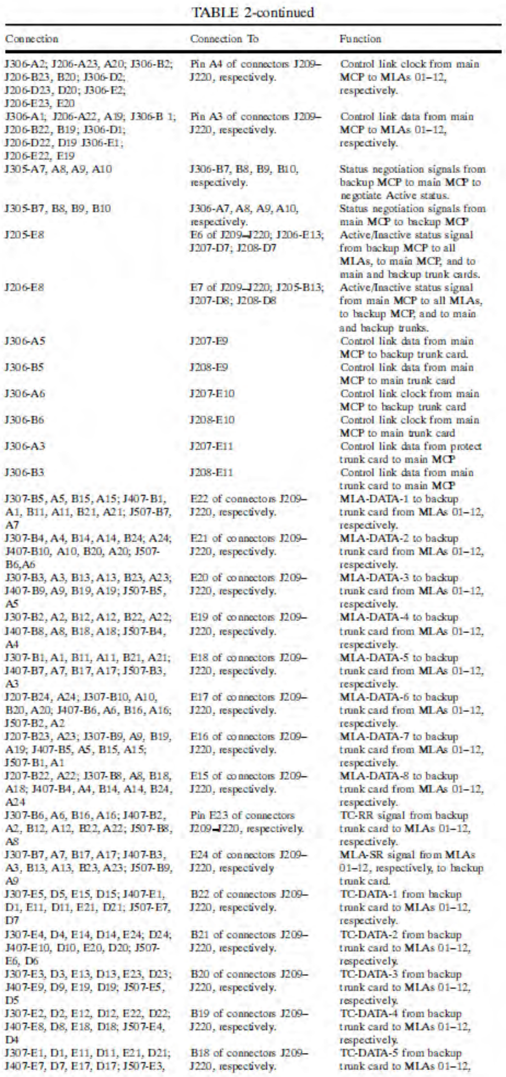
<table>
<tr><th>No.</th><th>’740 Patent Claim 10</th><th>Wiher ’530</th></tr>
<tr><td></td><td></td><td>

TABLE 2-continued

<table>
<tr><th>Connection</th><th>Connection To</th><th>Function</th></tr>
<tr><td>J306-A2; J206-A23, A20; J306-B2; J206-B23, B20; J306-D2; J206-D23, D20; J306-E2; J206-E23, E20</td><td>Pin A4 of connectors J209–J220, respectively.</td><td>Control link clock from main MCP to MLAs 01–12, respectively.</td></tr>
<tr><td>J306-A1; J206-A22, A19; J306-B 1; J206-B22, B19; J306-D1; J206-D22, D19 J306-E1; J206-E22, E19</td><td>Pin A3 of connectors J209–J220, respectively.</td><td>Control link data from main MCP to MLAs 01–12, respectively.</td></tr>
<tr><td>J305-A7, A8, A9, A10</td><td>J306-B7, B8, B9, B10, respectively.</td><td>Status negotiation signals from backup MCP to main MCP to negotiate Active status.</td></tr>
<tr><td>J305-B7, B8, B9, B10</td><td>J306-A7, A8, A9, A10, respectively.</td><td>Status negotiation signals from main MCP to backup MCP</td></tr>
<tr><td>J205-E8</td><td>E6 of J209–J220; J206-E13; J207-D7; J208-D7</td><td>Active/Inactive status signal from backup MCP to all MLAs, to main MCP, and to main and backup trunk cards.</td></tr>
<tr><td>J206-E8</td><td>E7 of J209–J220; J205-B13; J207-D8; J208-D8</td><td>Active/Inactive status signal from main MCP to all MLAs, to backup MCP, and to main and backup trunks.</td></tr>
<tr><td>J306-A5</td><td>J207-E9</td><td>Control link data from main MCP to backup trunk card.</td></tr>
<tr><td>J306-B5</td><td>J208-E9</td><td>Control link data from main MCP to main trunk card</td></tr>
<tr><td>J306-A6</td><td>J207-E10</td><td>Control link clock from main MCP to backup trunk card</td></tr>
<tr><td>J306-B6</td><td>J208-E10</td><td>Control link clock from main MCP to main trunk card</td></tr>
<tr><td>J306-A3</td><td>J207-E11</td><td>Control link data from protect trunk card to main MCP</td></tr>
<tr><td>J306-B3</td><td>J208-E11</td><td>Control link data from main trunk card to main MCP</td></tr>
<tr><td>J307-B5, A5, B15, A15; J407-B1, A1, B11, A11, B21, A21; J507-B7, A7</td><td>E22 of connectors J209–J220, respectively.</td><td>MLA-DATA-1 to backup trunk card from MLAs 01–12, respectively.</td></tr>
<tr><td>J307-B4, A4, B14, A14, B24; A24; J407-B10, A10, B20, A20; J507-B6,A6</td><td>E21 of connectors J209–J220, respectively.</td><td>MLA-DATA-2 to backup trunk card from MLAs 01–12, respectively.</td></tr>
<tr><td>J307-B3, A3, B13, A13, B23, A23; J407-B9, A9, B19, A19; J507-B5, A5</td><td>E20 of connectors J209–J220, respectively.</td><td>MLA-DATA-3 to backup trunk card from MLAs 01–12, respectively.</td></tr>
<tr><td>J307-B2, A2, B12, A12, B22, A22; J407-B8, A8, B18, A18; J507-B4, A4</td><td>E19 of connectors J209–J220, respectively.</td><td>MLA-DATA-4 to backup trunk card from MLAs 01–12, respectively.</td></tr>
<tr><td>J307-B1, A1, B11, A11, B21, A21; J407-B7, A7, B17, A17; J507-B3, A3</td><td>E18 of connectors J209–J220, respectively.</td><td>MLA-DATA-5 to backup trunk card from MLAs 01–12, respectively.</td></tr>
<tr><td>J207-B24, A24; J307-B10, A10, B20, A20; J407-B6, A6, B16, A16; J507-B2, A2</td><td>E17 of connectors J209–J220, respectively.</td><td>MLA-DATA-6 to backup trunk card from MLAs 01–12, respectively.</td></tr>
<tr><td>J207-B23, A23; J307-B9, A9, B19, A19; J407-B5, A5, B15, A15; J507-B1, A1</td><td>E16 of connectors J209–J220, respectively.</td><td>MLA-DATA-7 to backup trunk card from MLAs 01–12, respectively.</td></tr>
<tr><td>J207-B22, A22; J307-B8, A8, B18, A18; J407-B4, A4, B14, A14, B24, A24</td><td>E15 of connectors J209–J220, respectively.</td><td>MLA-DATA-8 to backup trunk card from MLAs 01–12, respectively.</td></tr>
<tr><td>J307-B6, A6, B16, A16; J407-B2, A2, B12, A12, B22, A22; J507-B8, A8</td><td>Pin E23 of connectors J209–J220, respectively.</td><td>TC-RR signal from backup trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J307-B7, A7, B17, A17; J407-B3, A3, B13, A13, B23, A23; J507-B9, A9</td><td>E24 of connectors J209–J220, respectively</td><td>MLA-SR signal from MLAs 01–12, respectively, to backup trunk card.</td></tr>
<tr><td>J307-E5, D5, E15, D15; J407-E1, D1, E11, D11, E21, D21; J507-E7, D7</td><td>B22 of connectors J209–J220, respectively.</td><td>TC-DATA-1 from backup trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J307-E4, D4, E14, D14, E24; D24; J407-E10, D10, E20, D20; J507-E6, D6</td><td>B21 of connectors J209–J220, respectively.</td><td>TC-DATA-2 from backup trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J307-E3, D3, E13, D13, E23, D23; J407-E9, D9, E19, D19; J507-E5, D5</td><td>B20 of connectors J209–J220, respectively.</td><td>TC-DATA-3 from backup trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J307-E2, D2, E12, D12, E22, D22; J407-E8, D8, E18, D18; J507-E4, D4</td><td>B19 of connectors J209–J220, respectively.</td><td>TC-DATA-4 from backup trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J307-E1, D1, E11, D11, E21, D21; J407-E7, D7, E17, D17; J507-E3,</td><td>B18 of connectors J209–J220, respectively.</td><td>TC-DATA-5 from backup trunk card to MLAs 01–12,</td></tr>
</table>
</td></tr>
</table>

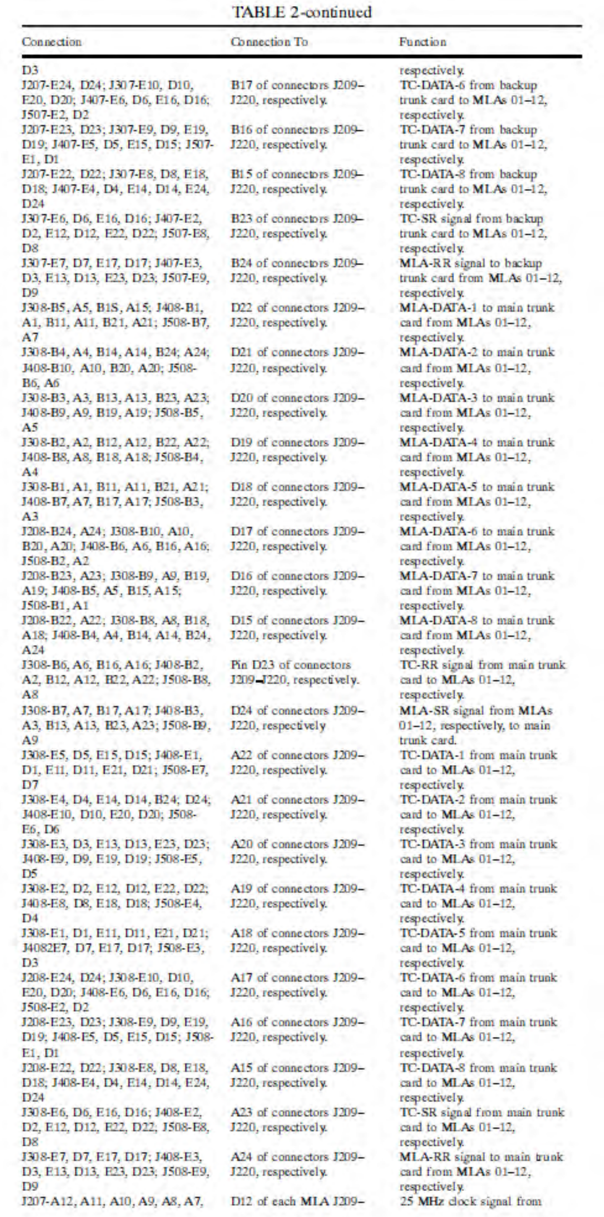


TABLE 2-continued

Connection	Connection To	Function
D3		respectively.
J207-E24, D24; J307-E10, D10, E20, D20; J407-E6, D6, E16, D16; J507-E2, D2	B17 of connectors J209–J220, respectively.	TC-DATA-6 from backup trunk card to MLAs 01–12, respectively.
J207-E23, D23; J307-E9, D9, E19, D19; J407-E5, D5, E15, D15; J507-E1, D1	B16 of connectors J209–J220, respectively.	TC-DATA-7 from backup trunk card to MLAs 01–12, respectively.
J207-E22, D22; J307-E8, D8, E18, D18; J407-E4, D4, E14, D14, E24, D24	B15 of connectors J209–J220, respectively.	TC-DATA-8 from backup trunk card to MLAs 01–12, respectively.
J307-E6, D6, E16, D16; J407-E2, D2, E12, D12, E22, D22; J507-E8, D8	B23 of connectors J209–J220, respectively.	TC-SR signal from backup trunk card to MLAs 01–12, respectively.
J307-E7, D7, E17, D17; J407-E3, D3, E13, D13, E23, D23; J507-E9, D9	B24 of connectors J209–J220, respectively.	MLA-RR signal to backup trunk card from MLAs 01–12, respectively.
J308-B5, A5, B1S, A15; J408-B1, A1, B11, A11, B21, A21; J508-B7, A7	D22 of connectors J209–J220, respectively.	MLA-DATA-1 to main trunk card from MLAs 01–12, respectively.
J308-B4, A4, B14, A14, B24; A24; J408-B10, A10, B20, A20; J508-B6, A6	D21 of connectors J209–J220, respectively.	MLA-DATA-2 to main trunk card from MLAs 01–12, respectively.
J308-B3, A3, B13, A13, B23, A23; J408-B9, A9, B19, A19; J508-B5, A5	D20 of connectors J209–J220, respectively.	MLA-DATA-3 to main trunk card from MLAs 01–12, respectively.
J308-B2, A2, B12, A12, B22, A22; J408-B8, A8, B18, A18; J508-B4, A4	D19 of connectors J209–J220, respectively.	MLA-DATA-4 to main trunk card from MLAs 01–12, respectively.
J308-B1, A1, B11, A11, B21, A21; J408-B7, A7, B17, A17; J508-B3, A3	D18 of connectors J209–J220, respectively.	MLA-DATA-5 to main trunk card from MLAs 01–12, respectively.
J208-B24, A24; J308-B10, A10, B20, A20; J408-B6, A6, B16, A16; J508-B2, A2	D17 of connectors J209–J220, respectively.	MLA-DATA-6 to main trunk card from MLAs 01–12, respectively.
J208-B23, A23; J308-B9, A9, B19, A19; J408-B5, A5, B15, A15; J508-B1, A1	D16 of connectors J209–J220, respectively.	MLA-DATA-7 to main trunk card from MLAs 01–12, respectively.
J208-B22, A22; J308-B8, A8, B18, A18; J408-B4, A4, B14, A14, B24, A24	D15 of connectors J209–J220, respectively.	MLA-DATA-8 to main trunk card from MLAs 01–12, respectively.
J308-B6, A6, B16, A16; J408-B2, A2, B12, A12, B22, A22; J508-B8, A8	Pin D23 of connectors J209–J220, respectively.	TC-RR signal from main trunk card to MLAs 01–12, respectively.
J308-B7, A7, B17, A17; J408-B3, A3, B13, A13, B23, A23; J508-B9, A9	D24 of connectors J209–J220, respectively	MLA-SR signal from MLAs 01–12, respectively, to main trunk card.
J308-E5, D5, E15, D15; J408-E1, D1, E11, D11, E21, D21; J508-E7, D7	A22 of connectors J209–J220, respectively.	TC-DATA-1 from main trunk card to MLAs 01–12, respectively.
J308-E4, D4, E14, D14, B24; D24; J408-E10, D10, E20, D20; J508-E6, D6	A21 of connectors J209–J220, respectively.	TC-DATA-2 from main trunk card to MLAs 01–12, respectively.
J308-E3, D3, E13, D13, E23, D23; J408-E9, D9, E19, D19; J508-E5, D5	A20 of connectors J209–J220, respectively.	TC-DATA-3 from main trunk card to MLAs 01–12, respectively.
J308-E2, D2, E12, D12, E22, D22; J408-E8, D8, E18, D18; J508-E4, D4	A19 of connectors J209–J220, respectively.	TC-DATA-4 from main trunk card to MLAs 01–12, respectively.
J308-E1, D1, E11, D11, E21, D21; J4082E7, D7, E17, D17; J508-E3, D3	A18 of connectors J209–J220, respectively.	TC-DATA-5 from main trunk card to MLAs 01–12, respectively.
J208-E24, D24; J308-E10, D10, E20, D20; J408-E6, D6, E16, D16; J508-E2, D2	A17 of connectors J209–J220, respectively.	TC-DATA-6 from main trunk card to MLAs 01–12, respectively.
J208-E23, D23; J308-E9, D9, E19, D19; J408-E5, D5, E15, D15; J508-E1, D1	A16 of connectors J209–J220, respectively.	TC-DATA-7 from main trunk card to MLAs 01–12, respectively.
J208-E22, D22; J308-E8, D8, E18, D18; J408-E4, D4, E14, D14, E24, D24	A15 of connectors J209–J220, respectively.	TC-DATA-8 from main trunk card to MLAs 01–12, respectively.
J308-E6, D6, E16, D16; J408-E2, D2, E12, D12, E22, D22; J508-E8, D8	A23 of connectors J209–J220, respectively.	TC-SR signal from main trunk card to MLAs 01–12, respectively.
J308-E7, D7, E17, D17; J408-E3, D3, E13, D13, E23, D23; J508-E9, D9	A24 of connectors J209–J220, respectively.	MLA-RR signal to main trunk card from MLAs 01–12, respectively.
J207-A12, A11, A10, A9, A8, A7,	D12 of each MLA J209–	25 MHz clock signal from

<table>
<tr><th>No.</th><th>’740 Patent Claim 10</th><th>Wiher ’530</th></tr>
<tr><td></td><td></td><td>

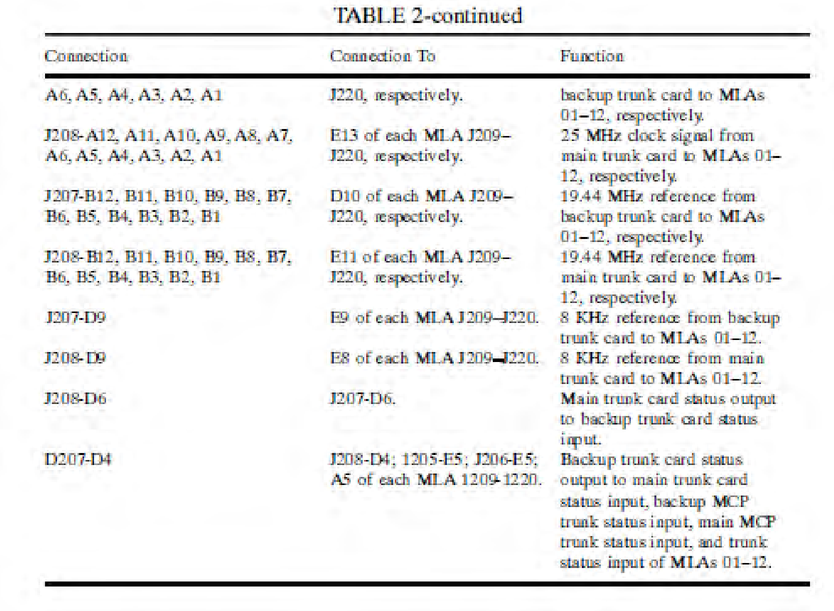



TABLE 2-continued
<table>
<tr><th>Connection</th><th>Connection To</th><th>Function</th></tr>
<tr><td>A6, A5, A4, A3, A2, A1</td><td>J220, respectively.</td><td>backup trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J208-A12, A11, A10, A9, A8, A7, A6, A5, A4, A3, A2, A1</td><td>E13 of each MLA J209–J220, respectively.</td><td>25 MHz clock signal from main trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J207-B12, B11, B10, B9, B8, B7, B6, B5, B4, B3, B2, B1</td><td>D10 of each MLA J209–J220, respectively.</td><td>19.44 MHz reference from backup trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J208-B12, B11, B10, B9, B8, B7, B6, B5, B4, B3, B2, B1</td><td>E11 of each MLA J209–J220, respectively.</td><td>19.44 MHz reference from main trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J207-D9</td><td>E9 of each MLA J209–J220.</td><td>8 KHz reference from backup trunk card to MLAs 01–12.</td></tr>
<tr><td>J208-D9</td><td>E8 of each MLA J209–J220.</td><td>8 KHz reference from main trunk card to MLAs 01–12.</td></tr>
<tr><td>J208-D6</td><td>J207-D6.</td><td>Main trunk card status output to backup trunk card status input.</td></tr>
<tr><td>D207-D4</td><td>J208-D4; 1205-E5; J206-E5; A5 of each MLA 1209-1220.</td><td>Backup trunk card status output to main trunk card status input, backup MCP trunk status input, main MCP trunk status input, and trunk status input of MLAs 01–12.</td></tr>
</table>

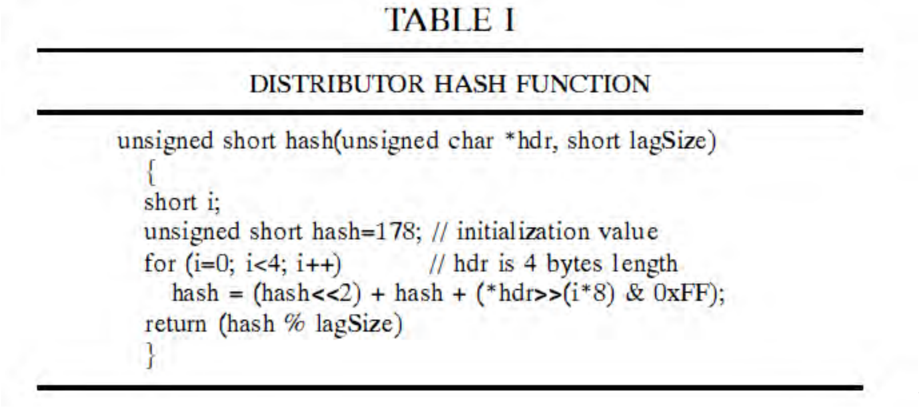
Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Wiher ’530 in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 10[d] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references.

For example, Bruckman discloses distributing data frames over physical links and traces based on a hash function involving a division operation (%).

Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”)
</td></tr>
</table>

No.	’740 Patent Claim 10	Wiher ’530
		Bruckman at [0025] (“Typically, setting the protection policy includes determining a maximum number of the physical links that may fail while the logical link continues to provide at least the guaranteed bandwidth for the connection. In one embodiment, the guaranteed bandwidth is a bandwidth B, and the plurality of physical links consists of N links, and the maximum number is an integer P, and the link bandwidth allocated to each of the links is no less than B/(N-P). Conveying the data may further include managing the transmission of the data responsively to an actual number X of the physical links that have failed so that the guaranteed bandwidth on each of the links is limited to B/(N-X), X P, and an excess bandwidth on the physical links over the guaranteed bandwidth is available for other connections.”) Bruckman at [0038] (“In some embodiments, the data transmission cir-cuitry includes a main card and a plurality of line cards, which are connected to the main card by respective traces, the line cards having ports connecting to the physical links and including concentrators for multiplexing the data between the physical links and the traces in accordance with the distribution determined by the distributor. In one embodiment, the plurality of line cards includes at least first and second line cards, and the physical links included in the logical link include at least first and second physical links, which are connected respectively to the first and second line cards. Typically, the safety margin is selected to be sufficient so that the guaranteed bandwidth is provided by the logical link subject to one or more of a facility failure of a predetermined number of the physical links and an equip-ment failure of one of the first and second line cards.”) Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:

No.	’740 Patent Claim 10	Wiher ’530

		
		TABLE I
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		”)
		
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		
		Bruckman at [0067] (“A similar problem may arise if there is a failure in a link in an aggregation group or in one of a number of line cards serving the aggregation group. In this case, to maintain the bandwidth allocation B made by CAC 44, each of the remaining links in the group must now carry, on average, B/(N-M) traffic, wherein M is the number of links in the group that are out of service. If only BIN has been allocated to each link, the remaining active links may not have sufficient bandwidth to continue to provide the bandwidth that has been guaranteed to the connections that they are required to carry. A similar problem arises with respect to loading of traces 52. For example, if there is a failure in LC2 or in one of links 30 in group 36 that connect to LC2, the trace connecting the multiplexer 50 in LCl will have to carry a substantially larger share of the bandwidth, or even all of the bandwidth, that is allocated to the connection in ques-tion.”)

No.	’740 Patent Claim 10	Wiher ’530
		Bruckman at [0068] (“FIG. 3 is a flow chart that schematically illustrates a method for dealing with these problems of fluctuating bandwidth requirements, in accordance with an embodiment of the present invention. In order to provide sufficient bandwidth for failure protection, CAC 44 uses a safety margin based on a protection parameter P, which is assigned at a protection setting step 60. P represents the maximum number of links in the group that can be out of service while still permitting the aggregation group to provide a given connection with the bandwidth that has been guaranteed to the connection. CAC 44 will then allocate at least B/(N-P) bandwidth to each link in the group, so that if P links fail, the group still provides total bandwidth of (N-P)*B/(N-P)= B. Setting P=l is equivalent to 1:N protection, so that the group will be unaffected by failure of a single link. In the example of group 36, shown in FIG. 2, setting P=2 will give both facility and equipment protection, i.e., the group will be unaffected not only by failure of a link, but also by failure of one of line cards 34. In the extreme case, in which P=N-1, CAC 44 will allocate the full bandwidth B on each link in the group.”) As another example, Singh discloses determining a ratio between the number of ingress and egress links and the number of links carrying data to the backplane and using a modulo to correspond to the channel’s link number. Singh at 9:30-43 (“The ratio between the number of line ingress links and the number of links carrying data to the backplane gives the backplane speedup for the system. In this example, there are 10 ingress links into the MS and 20 links (2 backplane channels) carrying that data to the backplane. This gives a backplane speedup of 2x. As another example, with 8 ingress links and 12 backplane links, there is a speedup of 1.5x. It should be noted that in addition to the backplane speedup, there is also an ingress/egress speedup. With 10 ingress links capable of carrying 2 Gbps each of raw data, this presents a 20 Gbps interface to the MS. An OC-192 only has approximately 10 Gbps worth of data. Taking into account cell overhead and cell quantization inefficiencies, there still remains excess capacity in the links.”) Singh at 11:29-38 (“FIG. 9 is a diagram illustrating link to channel assignments. The MS provides the interface between the line side and the fabric. As mentioned previously, the ratio between the number of backplane links used and the number of ingress/egress links used sets the speedup of

No.	’740 Patent Claim 10	Wiher ’530
		the fabric. Each MS has 40 input/output data links which can be used. Every 10 links create a channel, whether it is a backplane channel or an ingress/egress channel. There is no logical relationship between backplane and ingress/egress channels. A packet that arrives on one link can, in general, leave on any other link.”) Singh at 15:15-39 (“The number of crossbars that are required in a system is dependent on how many links are being used to create the backplane channels. There should be an even number of crossbars and they would be divided evenly across the switch cards. The following equation, for most cases, pro-vides the correct number of crossbars: # of Crossbars=(# links per ingress channelx# of ingress channels per portx# of port cardsx speedup)/32. For the 8x8 configuration, the # of crossbars should be multiplied by (4x# of iMS)/(# backplane channels per port card). The number of port cards should be rounded up to the nearest supported configuration, i.e. 8, 16, or 32. The speedup in the case of crossbars should be the fractional speedup that is desired. Example to determine the number of arbiters and cross-bars for the following system: 4 channel port cards (40 Gbps) 8 links per channel 16 port cards Speedup=l.5 # of arbiters=(4x2x2)/2=8 # of crossbars=(8x4x16xl.5)/32=24. This would give 3crossbars per arbiter.”) Singh at 16:28-44 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 16x16 and 32x32 is the organization of the switchplane. The port card remains the same. Backplane channels 1 and 2 are used for the backplane connectivity. Ingress and egress links 30-39 on the MS would not be used

No.	’740 Patent Claim 10	Wiher ’530
		and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 16, correspond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”) Singh at 17:31-49 (“In the single channel configuration, the egress MS is the same as the ingress MS. As far as the port card is concerned, the only difference between 8x8 and 16x16 is the organi-zation of the switchplane. The port card remains the same. Ingress and egress links 30-39 on the MS would not be used and would be powered off. Links 0-7 and 24-31 on the arbiters would not be used and would be powered off. Links 0-7 and 24-31 on the crossbars would not be used and would be powered off. Arbiter interfaces O.A, O.B, 3.A and 3.B on the PQ are unused and would be powered off. MS links 0-7 are used for both the ingress and egress to the traffic manager. Backplane channels 1 and 2 are used for the backplane connectivity. Each crossbar always handles the same numbered link within a backplane channel from each port card. Link numbers on the crossbars, modulo 8, corre-spond to the port card numbers. Link numbers on the MSs to the backplane, modulo 10, correspond to the backplane channel's link number. If it were desired to run IO-links per channel, a 5th crossbar would be added to each switch card.”)

No.	’740 Patent Claim 11	Wiher ’530
11	The method according to claim 10, wherein selecting the first and second physical links responsively to the modulo	Wiher ’530 discloses the method according to claim 10, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo. For example, Wiher ’530 discloses header translation circuitry to process ATM cells then direct them to different destinations, including system physical links, based on the ATM cell header. A person of ordinary skill in the art would understand that applying a hash function includes

No.	’740 Patent Claim 11	Wiher ’530
	comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo.	determining parameters responsive to the system and packet features, and generating a result. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Wiher ’530 is found to not meet this limitation, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo would have been obvious to a person having ordinary skill in the art, as explained below. *See supra* Claim 10 Wiher ’530 at 8:53-9:9 (“Cells arriving at the interface control circuitry 520 may include operations, administration, maintenance, and provisioning (OAMP) data or may contain user-data. OAMP cells may be identified by the payload type indicator (PTI) field in the ATM cell header. The multiplexer circuitry 521 extracts OAMP cells and send them to the processor 527 while user-data cells are sent to header translation circuitry 522. Header translation circuitry 522 performs VPI/VCI header field translation and other ATM cell header manipulation functions. Header translation circuitry 522 may deter-mine appropriate header manipulations based on, for example, programs and translation tables stored in RAM and ROM memory 526. Memory 526 may include header manipulation programs and translation tables that are stored by processor 527. After processing by header translation circuitry 522, ATM cells are directed to master line shelf adapter (MLA) interface circuitry 525. MLA interface cir-cuitry 525 controls and buffers cells flowing from the address translation circuitry 522 to a MLA and controls, for example, ATM cell flow over a SO NET OC-3c interface 530 between the LSM 500 and a MLA. Interface circuitry 525 may also insert OAMP cells from the processor 527 for transmission to a MLA and extract OAMP cells received from a MLA.”) Wiher ’530 at 9:10-27 (“The LSM's MLA interface circuitry 525 may also receive ATM cells from a MLA. The MLA interface circuitry 525 may extract OAMP cells arriving from a MLA and send them to the processor 527. Data cells arriving at the interface 525 and destined to a line card are sent to header translation circuitry 524 which may perform ATM cell header manipu-lations. Header translation circuitry 524 performs VPINCI header field translation and

No.	’740 Patent Claim 11	Wiher ’530
		other ATM cell header manipu-lation functions. Header translation circuitry 524 may deter-mine appropriate header manipulations based on, for example, programs and translation tables stored in RAM and ROM memory 526. Following header translation, data cells may be sent to de-multiplexer circuitry 523. The de-multiplexer circuitry controls the flow of ATM cells to interface control circuitry 520. Processor 527 may also send OAMP cells to the de-multiplexer 523 for transfer to line cards. Interface control circuitry 520 then transmits the ATM cell to a line card.”) Wiher ’530 at 11:20-39 (“Data transfers from a LSM to a line card include line card port identification information (a "port address"). A port address is a fixed value associated with a particular line card transceiver or subscriber loop connection. For example, a line card supporting two subscriber loops has port addresses "Pl" and "P2" that are associated, respectively, with the first and second subscriber loop at the line card. Each subscriber loop at a particular line card has an associated port address that is unique with respect to the port addresses of other subscriber loops at that line card. However, port addresses at one line card need not be unique with respect to port addresses at another line card. In a data transfer between a LSM and a line card, a line card port address may be identified by, for example, additional data bytes added to the ATM cell or by information in a modified (non-standard) cell header. Unlike VPINCI addresses which are dynami-cally associated with a transceiver, port address are perma-nently assigned (that is, they are static). Thus, the use of port addresses can simplify cell routing through a line card by simplifying the processing and storage of cell routing data.”) Wiher ’530 at 12:64-13:26 (“Referring to FIGS. 6 and 8, a line card's main LSM interface 610 includes control link signal lines 613. Line card operations, administration, maintenance and provision-ing (OAMP) functions can be controlled by data sent over the control link signal lines 613 between the line card and the main LSM. The control link signal lines 613 include a clock signal line, a data receive signal line, and a data transmit signal line. The line card sends data to the LSM in serial fashion by modulating a signal over a data transmit signal line and receives modulated data from the LSM over a data receive signal line. Signals exchanged over the data receive and data transmit signal lines are, for example, asserted or de-asserted on the falling edge of a 64 kilohertz (KHz) clock pulse received on the clock receive signal line and are sampled on

No.	’740 Patent Claim 11	Wiher ’530
		the rising edge of a received clock pulse. The format for the data exchanged on the control link signal lines 613 may conform to the Open Systems Interconnection (OSI) High-level Data Link Control (HDLC) protocol. The HDLC protocol is described in ISO/IEC 3309:1991 (E), Information Technology-Telecommunications and Information Exchange Between Systems-High-level Data Link Control (HDLC) procedures-Frame Structure, Interna-tional Organization for Standardization, Fourth Edition, 1991-06-01. A line card's main LSM interface 610 may also include clock signal lines 614. The clock signal lines include, for example, a 12.5 MHz clock signal line and a 8 KHz telephone network reference timing signal line received from the main LSM. Signals exchanged over clock signal lines 614 may be used to time data transmission over signal lines 611 and 612. Wiher ’530 at Table 1

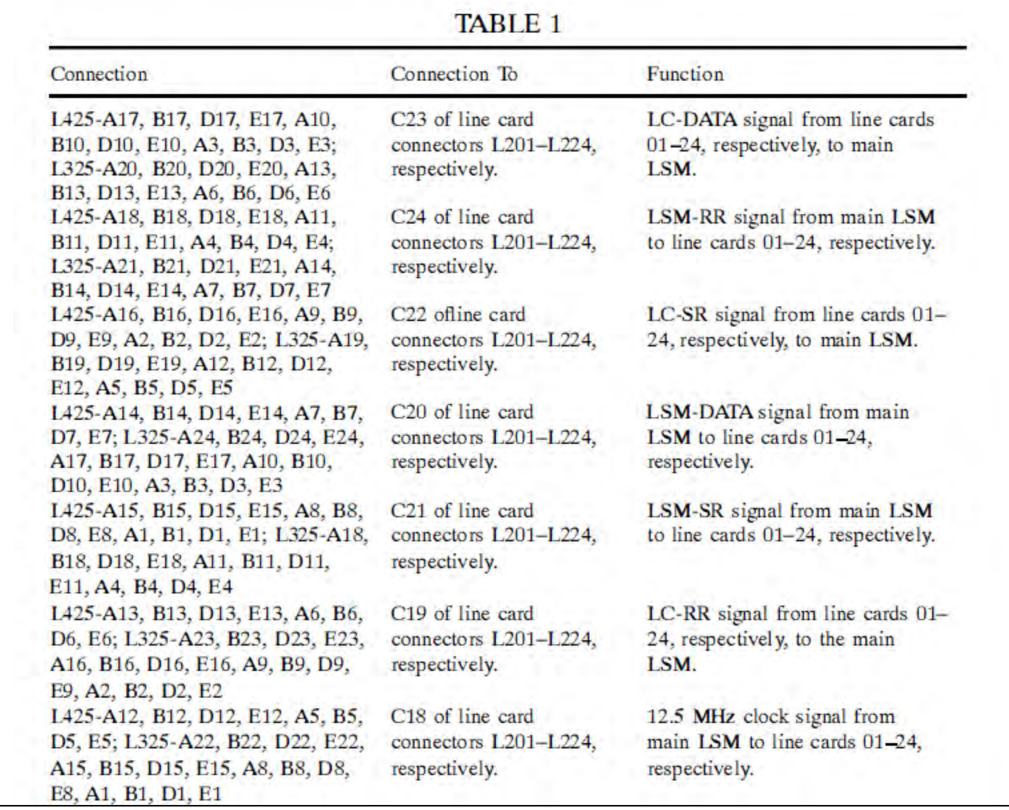
No.	’740 Patent Claim 11	Wiher ’530
		

TABLE 1

Connection	Connection To	Function
L425-A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3; L325-A20, B20, D20, E20, A13, B13, D13, E13, A6, B6, D6, E6	C23 of line card connectors L201–L224, respectively.	LC-DATA signal from line cards 01–24, respectively, to main LSM.
L425-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4; L325-A21, B21, D21, E21, A14, B14, D14, E14, A7, B7, D7, E7	C24 of line card connectors L201–L224, respectively.	LSM-RR signal from main LSM to line cards 01–24, respectively.
L425-A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2; L325-A19, B19, D19, E19, A12, B12, D12, E12, A5, B5, D5, E5	C22 ofline card connectors L201–L224, respectively.	LC-SR signal from line cards 01–24, respectively, to main LSM.
L425-A14, B14, D14, E14, A7, B7, D7, E7; L325-A24, B24, D24, E24, A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3	C20 of line card connectors L201–L224, respectively.	LSM-DATA signal from main LSM to line cards 01–24, respectively.
L425-A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1; L325-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4	C21 of line card connectors L201–L224, respectively.	LSM-SR signal from main LSM to line cards 01–24, respectively.
L425-A13, B13, D13, E13, A6, B6, D6, E6; L325-A23, B23, D23, E23, A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2	C19 of line card connectors L201–L224, respectively.	LC-RR signal from line cards 01–24, respectively, to the main LSM.
L425-A12, B12, D12, E12, A5, B5, D5, E5; L325-A22, B22, D22, E22, A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1	C18 of line card connectors L201–L224, respectively.	12.5 MHz clock signal from main LSM to line cards 01–24, respectively.

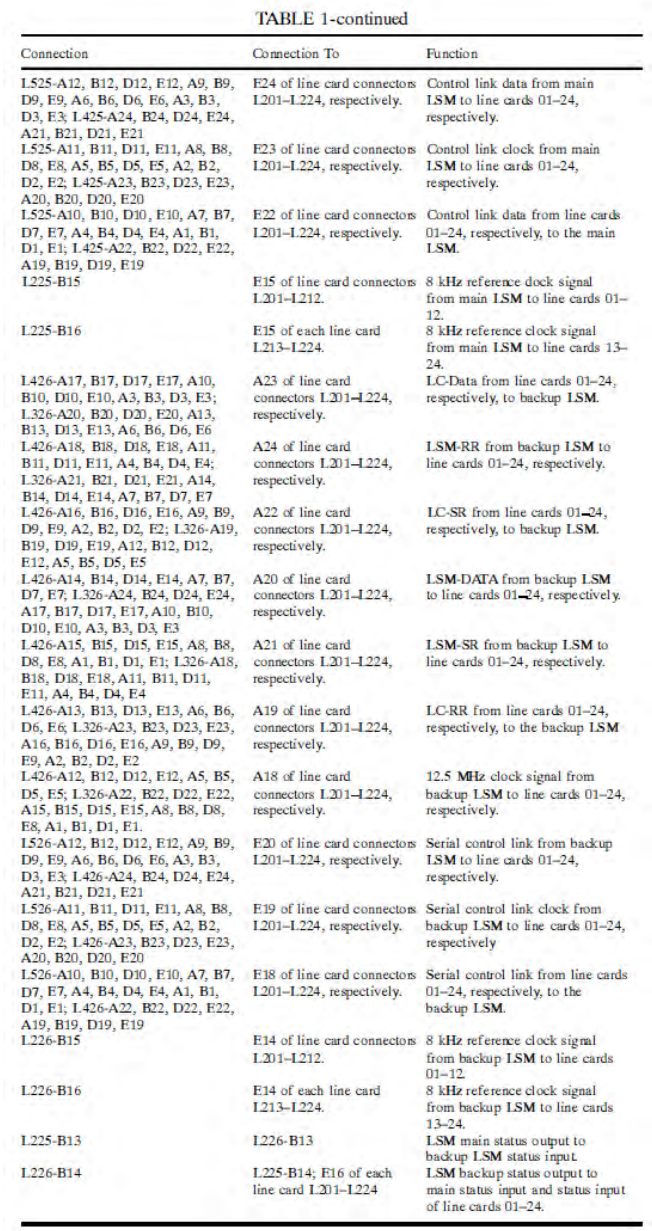
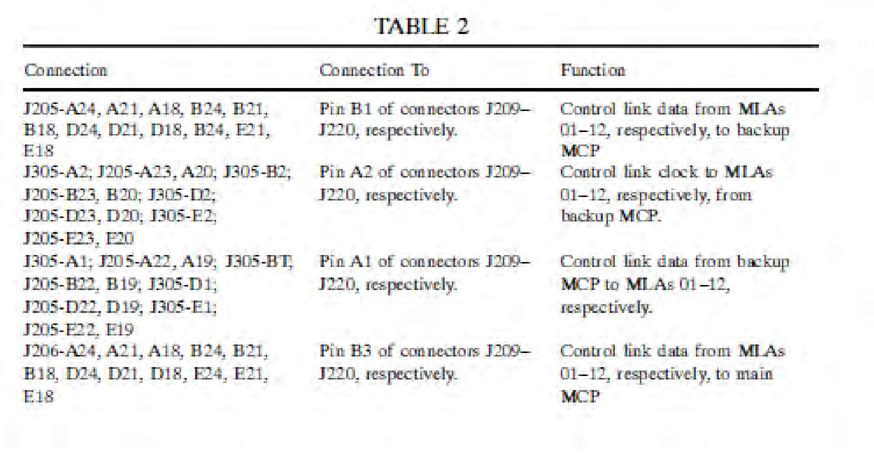


TABLE 1-continued

Connection	Connection To	Function
L525-A12, B12, D12, E12, A9, B9, D9, E9, A6, B6, D6, E6, A3, B3, D3, E3; L425-A24, B24, D24, E24, A21, B21, D21, E21	E24 of line card connectors L201–L224, respectively.	Control link data from main **LSM** to line cards 01–24, respectively.
L525-A11, B11, D11, E11, A8, B8, D8, E8, A5, B5, D5, E5, A2, B2, D2, E2; L425-A23, B23, D23, E23, A20, B20, D20, E20	E23 of line card connectors L201–L224, respectively.	Control link clock from main **LSM** to line cards 01–24, respectively.
L525-A10, B10, D10, E10, A7, B7, D7, E7, A4, B4, D4, E4, A1, B1, D1, E1; L425-A22, B22, D22, E22, A19, B19, D19, E19	E22 of line card connectors L201–L224, respectively.	Control link data from line cards 01–24, respectively, to the main **LSM**.
L225-B15	E15 of line card connectors L201–L212.	8 kHz reference dock signal from main **LSM** to line cards 01–12.
L225-B16	E15 of each line card L213–L224.	8 kHz reference clock signal from main **LSM** to line cards 13–24.
L426-A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3; L326-A20, B20, D20, E20, A13, B13, D13, E13, A6, B6, D6, E6	A23 of line card connectors L201–L224, respectively.	LC-Data from line cards 01–24, respectively, to backup **LSM**.
L426-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4; L326-A21, B21, D21, E21, A14, B14, D14, E14, A7, B7, D7, E7	A24 of line card connectors L201–L224, respectively.	**LSM**-RR from backup **LSM** to line cards 01–24, respectively.
L426-A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2; L326-A19, B19, D19, E19, A12, B12, D12, E12, A5, B5, D5, E5	A22 of line card connectors L201–L224, respectively.	LC-SR from line cards 01–24, respectively, to backup **LSM**.
L426-A14, B14, D14, E14, A7, B7, D7, E7; L326-A24, B24, D24, E24, A17, B17, D17, E17, A10, B10, D10, E10, A3, B3, D3, E3	A20 of line card connectors L201–L224, respectively.	**LSM**-DATA from backup **LSM** to line cards 01–24, respectively.
L426-A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1; L326-A18, B18, D18, E18, A11, B11, D11, E11, A4, B4, D4, E4	A21 of line card connectors L201–L224, respectively.	**LSM**-SR from backup **LSM** to line cards 01–24, respectively.
L426-A13, B13, D13, E13, A6, B6, D6, E6; L326-A23, B23, D23, E23, A16, B16, D16, E16, A9, B9, D9, E9, A2, B2, D2, E2	A19 of line card connectors L201–L224, respectively.	LC-RR from line cards 01–24, respectively, to the backup **LSM**
L426-A12, B12, D12, E12, A5, B5, D5, E5; L326-A22, B22, D22, E22, A15, B15, D15, E15, A8, B8, D8, E8, A1, B1, D1, E1.	A18 of line card connectors L201–L224, respectively.	12.5 **MHz** clock signal from backup **LSM** to line cards 01–24, respectively.
L526-A12, B12, D12, E12, A9, B9, D9, E9, A6, B6, D6, E6, A3, B3, D3, E3; L426-A24, B24, D24, E24, A21, B21, D21, E21	E20 of line card connectors L201–L224, respectively.	Serial control link from backup **LSM** to line cards 01–24, respectively.
L526-A11, B11, D11, E11, A8, B8, D8, E8, A5, B5, D5, E5, A2, B2, D2, E2; L426-A23, B23, D23, E23, A20, B20, D20, E20	E19 of line card connectors L201–L224, respectively.	Serial control link clock from backup **LSM** to line cards 01–24, respectively
L526-A10, B10, D10, E10, A7, B7, D7, E7, A4, B4, D4, E4, A1, B1, D1, E1; L426-A22, B22, D22, E22, A19, B19, D19, E19	E18 of line card connectors L201–L224, respectively.	Serial control link from line cards 01–24, respectively, to the backup **LSM**.
L226-B15	E14 of line card connectors L201–L212.	8 kHz reference clock signal from backup **LSM** to line cards 01–12.
L226-B16	E14 of each line card L213–L224.	8 kHz reference clock signal from backup **LSM** to line cards 13–24.
L225-B13	L226-B13	**LSM** main status output to backup **LSM** status input.
L226-B14	L225-B14; E16 of each line card L201–L224	**LSM** backup status output to main status input and status input of line cards 01–24.

<table>
<tr><th>No.</th><th>’740 Patent Claim 11</th><th>Wiher ’530</th></tr>
<tr><td></td><td></td><td>
<p>Wiher ’530 at 20:35-57 (“Trunk card interface control circuitry 1301 process ATM cell transfer signals exchanged over the main trunk card interface 1310 and backup trunk card interface 1315. Inter-face control circuitry 1301 provides, for example, ATM cell buffering and control of ATM cell exchange over interfaces 1310 and 1315. Additionally, interface circuitry 1301 may extract or insert OAMP cells to be exchanged between the processor 1305 and a trunk card. The interface control circuitry may also send ATM cells to and receive ATM cells from header translation circuitry 1303. Header translation circuitry 1303 performs VPINCI header field translation and other ATM cell header manipulation functions. Header translation circuitry 1303 may determine appropriate header manipulations based on, for example, programs and trans-lation tables stored in RAM and ROM memory 1304. Memory 1304 may include programs and data stored by processor 1305. Additionally, header translation circuitry 1303 interfaces with LSM interface control circuitry 1302. LSM interface control circuitry 1302 may extract or insert OAMP cells to be exchanged between the processor 1305 and a LSM. Additionally LSM interface control circuitry 1302 regulates ATM cell transport on, for example, a fiber optic SONET OC-3c interface 1330 to a LSM.”)</p>
<p>Wiher ’530 at Table 2</p>

<p>TABLE 2</p>
<table>
<tr><th>Connection</th><th>Connection To</th><th>Function</th></tr>
<tr><td>J205-A24, A21, A18, B24, B21, B18, D24, D21, D18, B24, E21, E18</td><td>Pin B1 of connectors J209–J220, respectively.</td><td>Control link data from MLAs 01–12, respectively, to backup MCP</td></tr>
<tr><td>J305-A2; J205-A23, A20; J305-B2; J205-B23, B20; J305-D2; J205-D23, D20; J305-E2; J205-E23, E20</td><td>Pin A2 of connectors J209–J220, respectively.</td><td>Control link clock to MLAs 01–12, respectively, from backup MCP.</td></tr>
<tr><td>J305-A1; J205-A22, A19; J305-BT, J205-B22, B19; J305-D1; J205-D22, D19; J305-E1; J205-E22, E19</td><td>Pin A1 of connectors J209–J220, respectively.</td><td>Control link data from backup MCP to MLAs 01–12, respectively.</td></tr>
<tr><td>J206-A24, A21, A18, B24, B21, B18, D24, D21, D18, E24, E21, E18</td><td>Pin B3 of connectors J209–J220, respectively.</td><td>Control link data from MLAs 01–12, respectively, to main MCP</td></tr>
</table>
</td></tr>
</table>

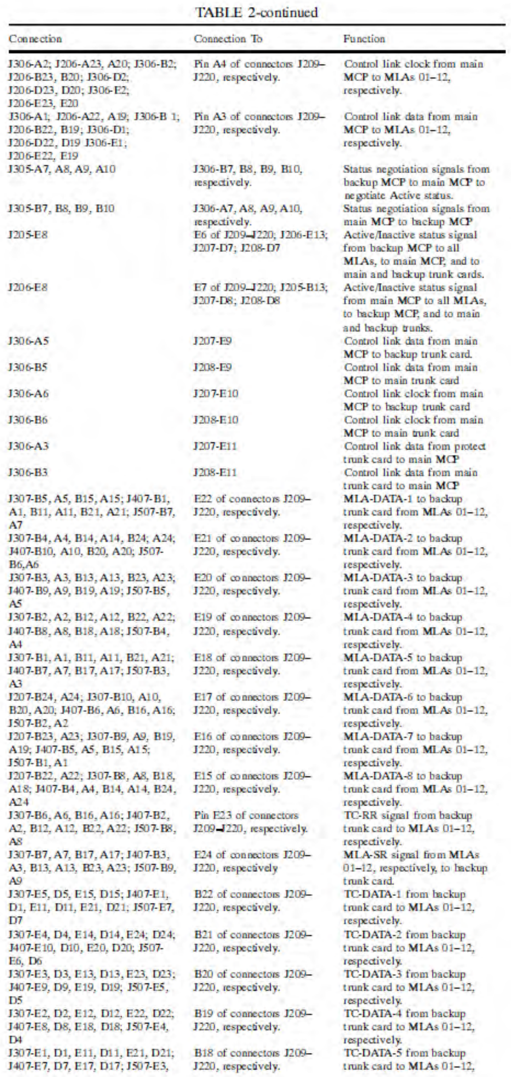
No.	'740 Patent Claim 11	Wiher '530
		

TABLE 2-continued

Connection	Connection To	Function
J306-A2; J206-A23, A20; J306-B2; J206-B23, B20; J306-D2; J206-D23, D20; J306-E2; J206-E23, E20	Pin A4 of connectors J209–J220, respectively.	Control link clock from main MCP to MLAs 01–12, respectively.
J306-A1; J206-A22, A19; J306-B 1; J206-B22, B19; J306-D1; J206-D22, D19 J306-E1; J206-E22, E19	Pin A3 of connectors J209–J220, respectively.	Control link data from main MCP to MLAs 01–12, respectively.
J305-A7, A8, A9, A10	J306-B7, B8, B9, B10, respectively.	Status negotiation signals from backup MCP to main MCP to negotiate Active status.
J305-B7, B8, B9, B10	J306-A7, A8, A9, A10, respectively.	Status negotiation signals from main MCP to backup MCP
J205-E8	E6 of J209–J220; J206-E13; J207-D7; J208-D7	Active/Inactive status signal from backup MCP to all MLAs, to main MCP, and to main and backup trunk cards.
J206-E8	E7 of J209–J220; J205-B13; J207-D8; J208-D8	Active/Inactive status signal from main MCP to all MLAs, to backup MCP, and to main and backup trunks.
J306-A5	J207-E9	Control link data from main MCP to backup trunk card.
J306-B5	J208-E9	Control link data from main MCP to main trunk card
J306-A6	J207-E10	Control link clock from main MCP to backup trunk card
J306-B6	J208-E10	Control link clock from main MCP to main trunk card
J306-A3	J207-E11	Control link data from protect trunk card to main MCP
J306-B3	J208-E11	Control link data from main trunk card to main MCP
J307-B5, A5, B15, A15; J407-B1, A1, B11, A11, B21, A21; J507-B7, A7	E22 of connectors J209–J220, respectively.	MLA-DATA-1 to backup trunk card from MLAs 01–12, respectively.
J307-B4, A4, B14, A14, B24; A24; J407-B10, A10, B20, A20; J507-B6,A6	E21 of connectors J209–J220, respectively.	MLA-DATA-2 to backup trunk card from MLAs 01–12, respectively.
J307-B3, A3, B13, A13, B23, A23; J407-B9, A9, B19, A19; J507-B5, A5	E20 of connectors J209–J220, respectively.	MLA-DATA-3 to backup trunk card from MLAs 01–12, respectively.
J307-B2, A2, B12, A12, B22, A22; J407-B8, A8, B18, A18; J507-B4, A4	E19 of connectors J209–J220, respectively.	MLA-DATA-4 to backup trunk card from MLAs 01–12, respectively.
J307-B1, A1, B11, A11, B21, A21; J407-B7, A7, B17, A17; J507-B3, A3	E18 of connectors J209–J220, respectively.	MLA-DATA-5 to backup trunk card from MLAs 01–12, respectively.
J207-B24, A24; J307-B10, A10, B20, A20; J407-B6, A6, B16, A16; J507-B2, A2	E17 of connectors J209–J220, respectively.	MLA-DATA-6 to backup trunk card from MLAs 01–12, respectively.
J207-B23, A23; J307-B9, A9, B19, A19; J407-B5, A5, B15, A15; J507-B1, A1	E16 of connectors J209–J220, respectively.	MLA-DATA-7 to backup trunk card from MLAs 01–12, respectively.
J207-B22, A22; J307-B8, A8, B18, A18; J407-B4, A4, B14, A14, B24, A24	E15 of connectors J209–J220, respectively.	MLA-DATA-8 to backup trunk card from MLAs 01–12, respectively.
J307-B6, A6, B16, A16; J407-B2, A2, B12, A12, B22, A22; J507-B8, A8	Pin E23 of connectors J209–J220, respectively.	TC-RR signal from backup trunk card to MLAs 01–12, respectively.
J307-B7, A7, B17, A17; J407-B3, A3, B13, A13, B23, A23; J507-B9, A9	E24 of connectors J209–J220, respectively	MLA-SR signal from MLAs 01–12, respectively, to backup trunk card.
J307-E5, D5, E15, D15; J407-E1, D1, E11, D11, E21, D21; J507-E7, D7	B22 of connectors J209–J220, respectively.	TC-DATA-1 from backup trunk card to MLAs 01–12, respectively.
J307-E4, D4, E14, D14, E24; D24; J407-E10, D10, E20, D20; J507-E6, D6	B21 of connectors J209–J220, respectively.	TC-DATA-2 from backup trunk card to MLAs 01–12, respectively.
J307-E3, D3, E13, D13, E23, D23; J407-E9, D9, E19, D19; J507-E5, D5	B20 of connectors J209–J220, respectively.	TC-DATA-3 from backup trunk card to MLAs 01–12, respectively.
J307-E2, D2, E12, D12, E22, D22; J407-E8, D8, E18, D18; J507-E4, D4	B19 of connectors J209–J220, respectively.	TC-DATA-4 from backup trunk card to MLAs 01–12, respectively.
J307-E1, D1, E11, D11, E21, D21; J407-E7, D7, E17, D17; J507-E3,	B18 of connectors J209–J220, respectively.	TC-DATA-5 from backup trunk card to MLAs 01–12,

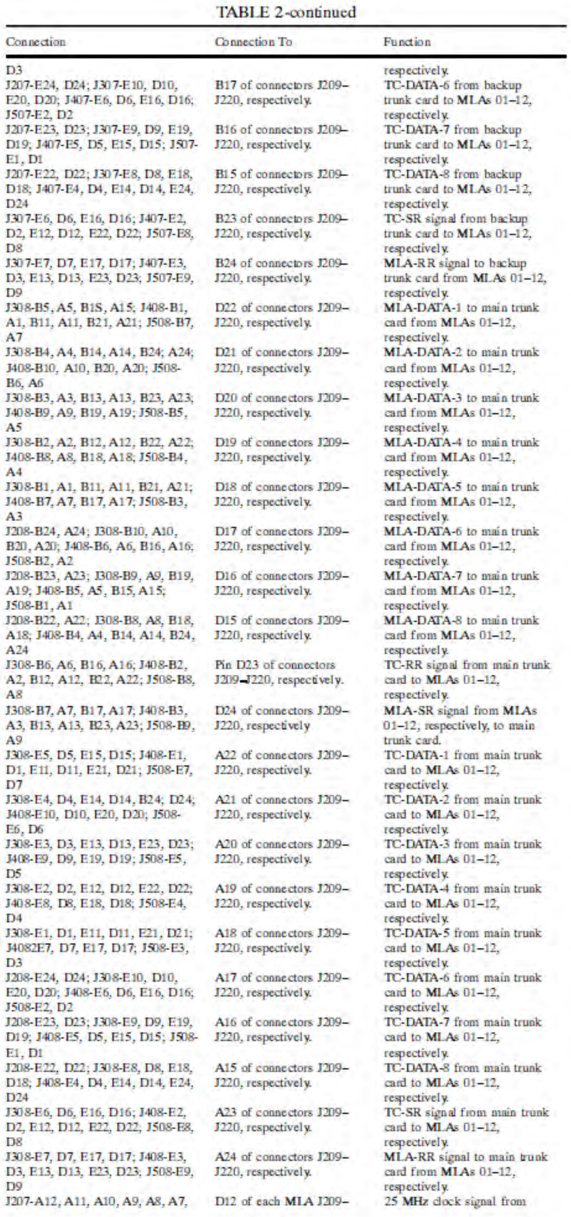


TABLE 2-continued

Connection	Connection To	Function
D3		respectively.
J207-E24, D24; J307-E10, D10, E20, D20; J407-E6, D6, E16, D16; J507-E2, D2	B17 of connectors J209–J220, respectively.	TC-DATA-6 from backup trunk card to MLAs 01–12, respectively.
J207-E23, D23; J307-E9, D9, E19, D19; J407-E5, D5, E15, D15; J507-E1, D1	B16 of connectors J209–J220, respectively.	TC-DATA-7 from backup trunk card to MLAs 01–12, respectively.
J207-E22, D22; J307-E8, D8, E18, D18; J407-E4, D4, E14, D14, E24, D24	B15 of connectors J209–J220, respectively.	TC-DATA-8 from backup trunk card to MLAs 01–12, respectively.
J307-E6, D6, E16, D16; J407-E2, D2, E12, D12, E22, D22; J507-E8, D8	B23 of connectors J209–J220, respectively.	TC-SR signal from backup trunk card to MLAs 01–12, respectively.
J307-E7, D7, E17, D17; J407-E3, D3, E13, D13, E23, D23; J507-E9, D9	B24 of connectors J209–J220, respectively.	MLA-RR signal to backup trunk card from MLAs 01–12, respectively.
J308-B5, A5, B1S, A15; J408-B1, A1, B11, A11, B21, A21; J508-B7, A7	D22 of connectors J209–J220, respectively.	MLA-DATA-1 to main trunk card from MLAs 01–12, respectively.
J308-B4, A4, B14, A14, B24; A24; J408-B10, A10, B20, A20; J508-B6, A6	D21 of connectors J209–J220, respectively.	MLA-DATA-2 to main trunk card from MLAs 01–12, respectively.
J308-B3, A3, B13, A13, B23, A23; J408-B9, A9, B19, A19; J508-B5, A5	D20 of connectors J209–J220, respectively.	MLA-DATA-3 to main trunk card from MLAs 01–12, respectively.
J308-B2, A2, B12, A12, B22, A22; J408-B8, A8, B18, A18; J508-B4, A4	D19 of connectors J209–J220, respectively.	MLA-DATA-4 to main trunk card from MLAs 01–12, respectively.
J308-B1, A1, B11, A11, B21, A21; J408-B7, A7, B17, A17; J508-B3, A3	D18 of connectors J209–J220, respectively.	MLA-DATA-5 to main trunk card from MLAs 01–12, respectively.
J208-B24, A24; J308-B10, A10, B20, A20; J408-B6, A6, B16, A16; J508-B2, A2	D17 of connectors J209–J220, respectively.	MLA-DATA-6 to main trunk card from MLAs 01–12, respectively.
J208-B23, A23; J308-B9, A9, B19, A19; J408-B5, A5, B15, A15; J508-B1, A1	D16 of connectors J209–J220, respectively.	MLA-DATA-7 to main trunk card from MLAs 01–12, respectively.
J208-B22, A22; J308-B8, A8, B18, A18; J408-B4, A4, B14, A14, B24, A24	D15 of connectors J209–J220, respectively.	MLA-DATA-8 to main trunk card from MLAs 01–12, respectively.
J308-B6, A6, B16, A16; J408-B2, A2, B12, A12, B22, A22; J508-B8, A8	Pin D23 of connectors J209–J220, respectively.	TC-RR signal from main trunk card to MLAs 01–12, respectively.
J308-B7, A7, B17, A17; J408-B3, A3, B13, A13, B23, A23; J508-B9, A9	D24 of connectors J209–J220, respectively	MLA-SR signal from MLAs 01–12, respectively, to main trunk card.
J308-E5, D5, E15, D15; J408-E1, D1, E11, D11, E21, D21; J508-E7, D7	A22 of connectors J209–J220, respectively.	TC-DATA-1 from main trunk card to MLAs 01–12, respectively.
J308-E4, D4, E14, D14, B24; D24; J408-E10, D10, E20, D20; J508-E6, D6	A21 of connectors J209–J220, respectively.	TC-DATA-2 from main trunk card to MLAs 01–12, respectively.
J308-E3, D3, E13, D13, E23, D23; J408-E9, D9, E19, D19; J508-E5, D5	A20 of connectors J209–J220, respectively.	TC-DATA-3 from main trunk card to MLAs 01–12, respectively.
J308-E2, D2, E12, D12, E22, D22; J408-E8, D8, E18, D18; J508-E4, D4	A19 of connectors J209–J220, respectively.	TC-DATA-4 from main trunk card to MLAs 01–12, respectively.
J308-E1, D1, E11, D11, E21, D21; J4082E7, D7, E17, D17; J508-E3, D3	A18 of connectors J209–J220, respectively.	TC-DATA-5 from main trunk card to MLAs 01–12, respectively.
J208-E24, D24; J308-E10, D10, E20, D20; J408-E6, D6, E16, D16; J508-E2, D2	A17 of connectors J209–J220, respectively.	TC-DATA-6 from main trunk card to MLAs 01–12, respectively.
J208-E23, D23; J308-E9, D9, E19, D19; J408-E5, D5, E15, D15; J508-E1, D1	A16 of connectors J209–J220, respectively.	TC-DATA-7 from main trunk card to MLAs 01–12, respectively.
J208-E22, D22; J308-E8, D8, E18, D18; J408-E4, D4, E14, D14, E24, D24	A15 of connectors J209–J220, respectively.	TC-DATA-8 from main trunk card to MLAs 01–12, respectively.
J308-E6, D6, E16, D16; J408-E2, D2, E12, D12, E22, D22; J508-E8, D8	A23 of connectors J209–J220, respectively.	TC-SR signal from main trunk card to MLAs 01–12, respectively.
J308-E7, D7, E17, D17; J408-E3, D3, E13, D13, E23, D23; J508-E9, D9	A24 of connectors J209–J220, respectively.	MLA-RR signal to main trunk card from MLAs 01–12, respectively.
J207-A12, A11, A10, A9, A8, A7,	D12 of each MLA J209–	25 MHz clock signal from

<table>
<tr><th>No.</th><th>’740 Patent Claim 11</th><th>Wiher ’530</th></tr>
<tr><td></td><td></td><td>

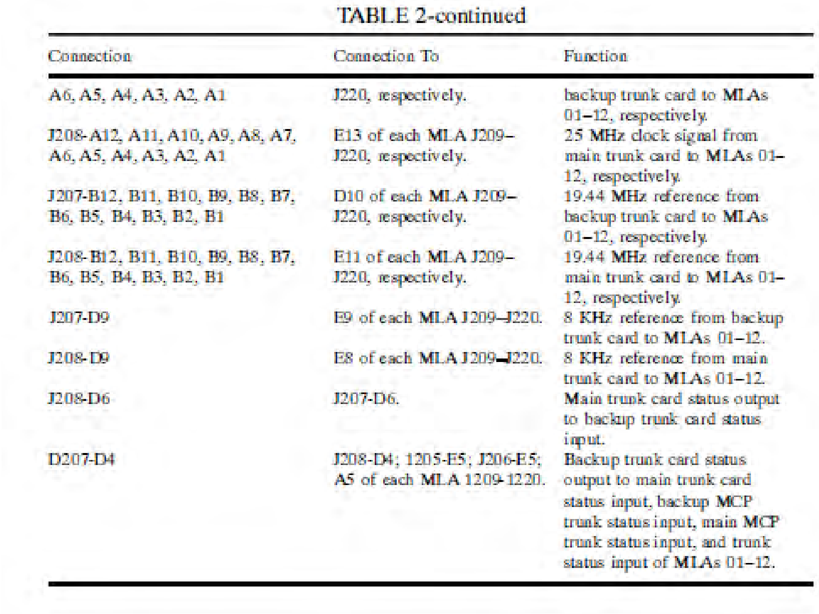
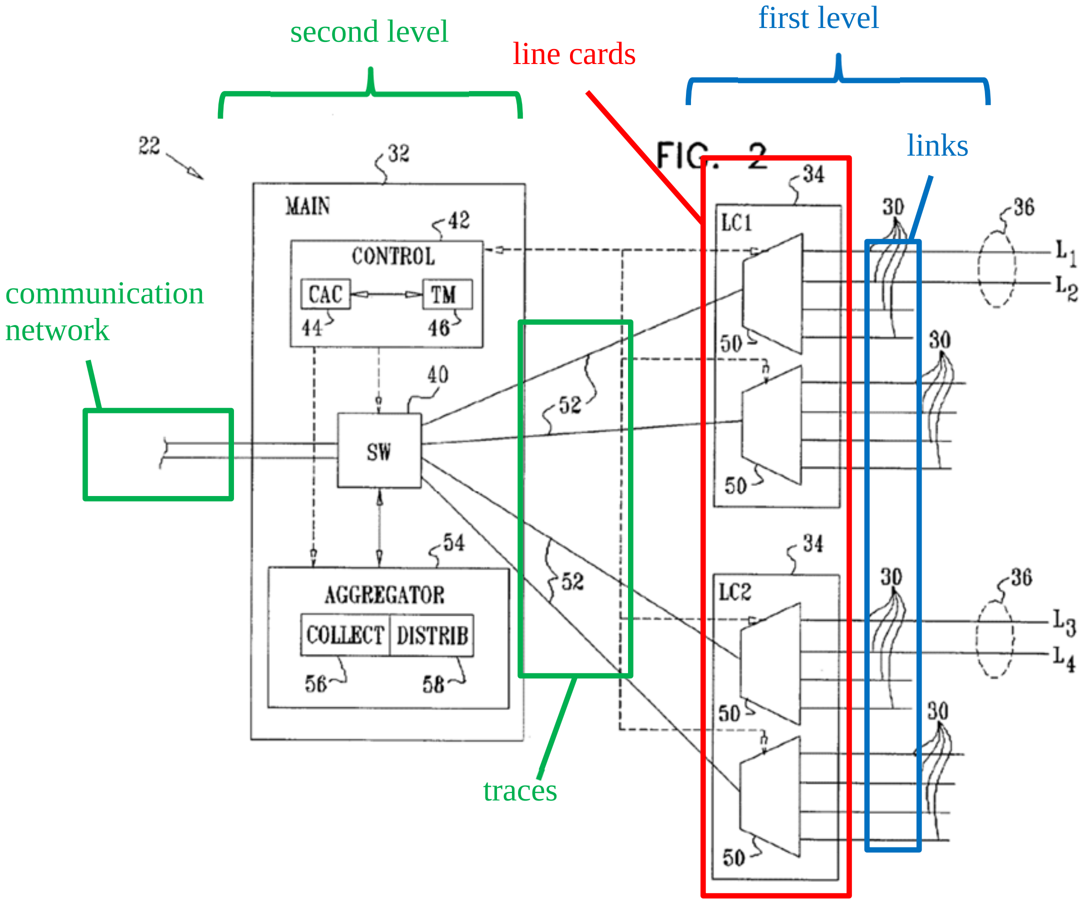



TABLE 2-continued
<table>
<tr><th>Connection</th><th>Connection To</th><th>Function</th></tr>
<tr><td>A6, A5, A4, A3, A2, A1</td><td>J220, respectively.</td><td>backup trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J208-A12, A11, A10, A9, A8, A7, A6, A5, A4, A3, A2, A1</td><td>E13 of each MLA J209–J220, respectively.</td><td>25 MHz clock signal from main trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J207-B12, B11, B10, B9, B8, B7, B6, B5, B4, B3, B2, B1</td><td>D10 of each MLA J209–J220, respectively.</td><td>19.44 MHz reference from backup trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J208-B12, B11, B10, B9, B8, B7, B6, B5, B4, B3, B2, B1</td><td>E11 of each MLA J209–J220, respectively.</td><td>19.44 MHz reference from main trunk card to MLAs 01–12, respectively.</td></tr>
<tr><td>J207-D9</td><td>E9 of each MLA J209–J220.</td><td>8 KHz reference from backup trunk card to MLAs 01–12.</td></tr>
<tr><td>J208-D9</td><td>E8 of each MLA J209–J220.</td><td>8 KHz reference from main trunk card to MLAs 01–12.</td></tr>
<tr><td>J208-D6</td><td>J207-D6.</td><td>Main trunk card status output to backup trunk card status input.</td></tr>
<tr><td>D207-D4</td><td>J208-D4; 1205-E5; J206-E5; A5 of each MLA 1209-1220.</td><td>Backup trunk card status output to main trunk card status input, backup MCP trunk status input, main MCP trunk status input, and trunk status input of MLAs 01–12.</td></tr>
</table>

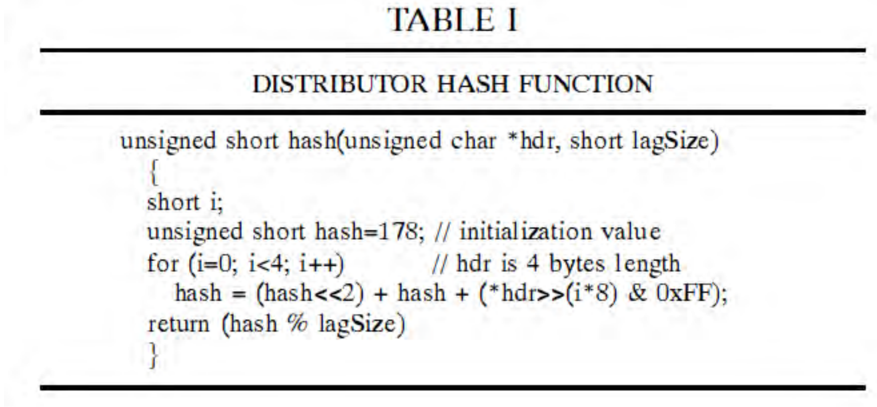
Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Wiher ’530 in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 11 of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references.

For example, Bruckman discloses distributing data frames over physical links and traces based on a division operation in the hash function involving specific byte lengths of the frame information.

Bruckman at Figure 2 (annotated)
</td></tr>
</table>

No.	’740 Patent Claim 11	Wiher ’530
		 FIG. 2 second level line cards first level links communication network traces 22 32 MAIN 42 CONTROL CAC TM 44 46 40 SW 54 AGGREGATOR COLLECT DISTRIB 56 58 52 52 34 LC1 50 50 34 LC2 50 50 30 30 30 30 36 36 L_1 L_2 L_3 L_4 Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:

No.	’740 Patent Claim 11	Wiher ’530

		
		TABLE I
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		  {
		  short i;
		  unsigned short hash=178; // initialization value
		  for (i=0; i<4; i++) // hdr is 4 bytes length
		    hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		  return (hash % lagSize)
		  }
		”)
		
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively, dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)
		
		As another example, Solomon discloses using a subset of bits to encode for the selected physical port involving specific byte lengths of the frame information.
		
		Solomon at [0054] (“Having selected a physical port, RSVP-TE proces-sor 30 of switch A now generates a suitable MPLS label, at a label generation step 64. The preceding node upstream of switch A will subsequently attach this MPLS label to all MPLS packets transmitted through tunnel 28 to switch A. The label is assigned, in conjunction with the mapping function of mapper 34, so as to ensure that all MPLS packets carrying this label are switched through the physical port that was selected for this tunnel at step 62. For this purpose, RSVP-TE processor 30 of switch A dedicates a sub-set of the bits of MPLS label 52 to encode the serial number of the selected physical port. For example, the four least-signifi-cant bits of MPLS label 52 may be used for encoding the selected port number. This configuration is suitable for representing LAG groups

No.	’740 Patent Claim 11	Wiher ’530
		having up to 16 physical ports (N<16). The remaining bits of MPLS label 52 may be chosen at random or using any suitable method known in the art.”) Solomon at [0056] (“Mapper 34 of switch A maps the received packets belonging to tunnel 28 to the selected physical Ethernet port at a mapping step 70. For this purpose, mapper 34 extracts the MPLS label from each received packet and decodes the selected physical port number from the dedicated sub-set of bits, such as the four LSB, as described in step 64 above. The decoded value is used for mapping the packet to the selected physical port, which was allocated by the CAC processor at step 62 above. In the four-bit example described above, the mapping function may be written explicitly as: Selected port number=((MPLS label) and (0x0000F)), wherein "and" denotes the "bitwise and" operator.”)

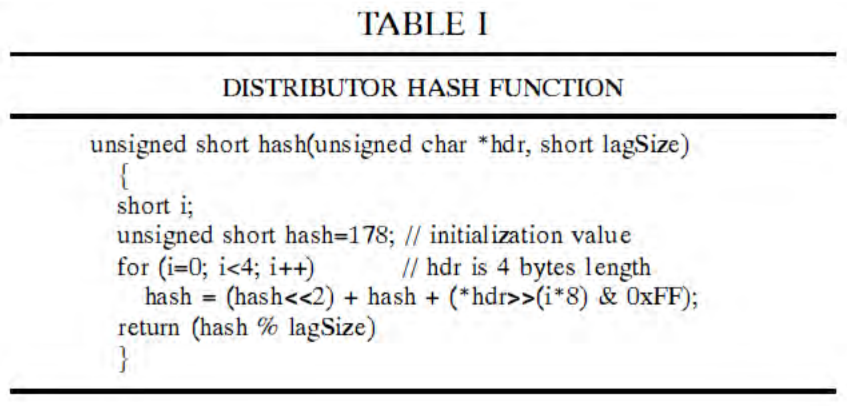
No.	’740 Patent Claim 12	Wiher ’530
12	The method according to claim 1, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC)	Wiher ’530 discloses the method according to claim 1, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port. For example, Wiher ’530 discloses data cells with cell header fields that contain cell information, including, for example, port address data. Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Wiher ’530 is found to not meet this limitation, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port would have been obvious to a person having ordinary skill in the art, as explained below.

No.	’740 Patent Claim 12	Wiher ’530
	address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port.	*See supra* Claim 1. Wiher ’530 at 2:44-58 (“In various aspects, implementations of the invention may include second transceiver circuitry to transmitting and receiving data cells over a second data link and to provide data cells to and receive data cells from the backplane interconnection circuitry. The first transceiver and second transceivers may each have associated unique port addresses. The backplane interconnection circuitry may receive data cells having port address data in the cell headers and provide data cells having the first port address to the first transceiver but not to the second transceiver, while data cells having the second port address are provided to the second transceiver but not to the first transceiver. Each cell may include a five byte header field and a forty-eight byte payload field. The fifth byte of the header field may include the port address identifier.”) Wiher ’530 at 5:26-43 (“FIG. lA illustrates an ATM cell having a 53-byte format as defined by the ITU. The ATM cell 100 includes a header field 101 and a payload field 102. An ITU-standard header field 101 may be either a user-network interface header or a network-network interface header. FIGS. lB and lC illustrate, respectively, a user-network interface header 125 and a network-network interface header 150. In general, ATM cells having a user-network interface header 125 are sent between ATM network access equipment that is located at an endpoint of an ATM connection and ATM network switching equipment ("nodes"). Cells having a network-network interface header 150 are sent between nodes in the ATM network, i.e., from non-endpoint to non-endpoint ATM cell switching equipment. User-network interface headers 125 and network-network interface headers 150 include multiple information sub-fields and differ in the information contained in the first byte ("Byte l") of cell header 125 and cell header 150.”) Wiher ’530 at 5:44-65 (“A user-network header 125 (FIG. lB) includes a four-bit generic flow control (GFC) field, an eight-bit virtual path identifier (VPI) field, a sixteen-bit virtual channel identifier (VCI) field, a three-bit payload type identifier (PTI) field, a one-bit cell loss priority

No.	’740 Patent Claim 12	Wiher ’530
		(CLP) field and an eight-bit header error control (HEC) field. The GFC field carries information to assist in controlling the flow of ATM cells over the user-network interface. The VPI field identifies a virtual path and the VCI field identifies the virtual channel for routing the ATM cell through a network. The PTI field identifies whether the cell contains user or network management related information. The CLP field indicates the cell loss priority. If the value of the CLP field is 1, the cell may be subject to discard, depending on network conditions such as a network overload. If the value of the CLP field is 0, the cell has high priority and therefore ATM nodes should allocate sufficient network resources to prevent cell discard and ensure transport of the cell. The HEC field contains header error control information to detect transmission errors in header 101 information. Additional information on these standard header fields can be found in ATM User-Network Interface Specification Version 3.1, ATM Forum, 1994.”) Wiher ’530 at 11:20-39 (“Data transfers from a LSM to a line card include line card port identification information (a "port address"). A port address is a fixed value associated with a particular line card transceiver or subscriber loop connection. For example, a line card supporting two subscriber loops has port addresses "Pl" and "P2" that are associated, respectively, with the first and second subscriber loop at the line card. Each subscriber loop at a particular line card has an associated port address that is unique with respect to the port addresses of other subscriber loops at that line card. However, port addresses at one line card need not be unique with respect to port addresses at another line card. In a data transfer between a LSM and a line card, a line card port address may be identified by, for example, additional data bytes added to the ATM cell or by information in a modified (non-standard) cell header. Unlike VPINCI addresses which are dynami-cally associated with a transceiver, port address are perma-nently assigned (that is, they are static). Thus, the use of port addresses can simplify cell routing through a line card by simplifying the processing and storage of cell routing data.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Wiher ’530 in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 12 of Exhibit E-3 renders the claim, including the present limitation, obvious. Below is one such example.

No.	’740 Patent Claim 12	Wiher ’530
		For example, Bruckman discloses frame information including header fields, source addresses, destination addresses, ports, etc. Bruckman specifically discloses Ethernet frame information including source MAC address, destination MAC address, reception port, type of destination address, Ethernet Length/Type value, and higher layer protocol information. Bruckman at [0005]-[0011] (“Annex 43A of the 802.3 standard, which is also incorporated herein by reference, describes possible distri-bution algorithms that meet the requirements of the stan-dard, while providing some measure of load balancing among the physical links in the aggregation group. The algorithm may make use of information carried in each Ethernet frame in order to make its decision as to the physical port to which the frame should be sent. The frame information may be combined with other information asso-ciated with the frame, such as its reception port in the case of a MAC bridge. The information used to assign conver-sations to ports could thus include one or more of the following pieces of information: [0006] a) Source MAC address [0007] b) Destination MAC address [0008] c) Reception port [0009] d) Type of destination address [0010] e) Ethernet Length/Type value [0011] t) Higher layer protocol information”) Bruckman at [0012] (“A hash function, for example may be applied to the selected information in order to generate a port number. Because conversations can vary greatly in length, however, it is difficult to select a hash function that will generate a uniform distribution of load across the set of ports for all traffic models.”) Bruckman at [0024] (“In a disclosed embodiment, the data include a sequence of data frames having respective headers, and distributing the data includes applying a hash function to the headers to select a respective one of the physical links over which to transmit each of the data frames.”)

No.	’740 Patent Claim 12	Wiher ’530

		Bruckman at [0058]-[0062] (“Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36. Typically, distributor 58 determines the link over which to send each frame based on information in the frame header, as described in the Background of the Invention. Preferably, distributor 58 applies a predetermined hash function to the header information, wherein the hash function satisfies the follow-ing criteria:
		[0059] The hash value output by the function is fully determined by the data being hashed, so that frames with the same header will always be distributed to the same link.
		[0060] The hash function uses all the specified input data from the frame headers.
		[0061] The hash function distributes traffic in an approximately uniform manner across the entire set of possible hash values
		[0062] The hash function generates very different hash values for similar data.”)
		
		Bruckman at [0063] (“For example, distributor 58 may implement the hash function shown below in Table I:
		
		TABLE I
		DISTRIBUTOR HASH FUNCTION
		unsigned short hash(unsigned char *hdr, short lagSize)
		{
		short i;
		unsigned short hash=178; // initialization value
		for (i=0; i<4; i++) // hdr is 4 bytes length
		hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
		return (hash % lagSize)
		}
		”)
		
		Bruckman at [0064] (“Here hdr is the header of the frame to be distrib-uted, and lagsize is the number of active ports (available links 30) in link aggregation group 46. Alternatively,

No.	’740 Patent Claim 12	Wiher ’530
		dis-tributor 58 may use other means, such as look-up tables, for determining the distribution of frames among links 30.”)

No.	’740 Patent Claim 13	Wiher ’530
13[preamble]	A method for communication, comprising:	Wiher ’530 discloses a method for communication. *See supra at* 1[preamble].
13[a]	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;	Wiher ’530 discloses coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel. *See supra at* 1[a].
13[b]	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;	Wiher ’530 discloses coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel. *See supra at* 1[c].
13[c]	receiving a data frame having frame attributes sent between the communication network and the network node:	Wiher ’530 discloses receiving a data frame having frame attributes sent between the communication network and the network node. *See supra at* 1[e].

No.	’740 Patent Claim 13	Wiher ’530
13[d]	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and	Wiher ’530 discloses selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group. *See supra at* 1[f].
13[e]	sending the data frame over the selected first and second physical links,	Wiher ’530 discloses sending the data frame over the selected first and second physical links. *See supra at* 1[g].
13[f]	coupling the network node to the one or more interface modules and	Wiher ’530 discloses coupling the network node to the one or more interface modules. *See supra at* 1[a].
13[g]	coupling each of the one or more interface modules to the communication network comprising	Wiher ’530 discloses coupling each of the one or more interface modules to the communication network. *See supra at* 1[c].
13[h]	specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess	Wiher ’530 discloses specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node. For example, Wiher ’530 discloses bandwidth attributes of the ATM system. Thus, a person of ordinary skill in the art would understand that the specified bandwidth attributes of the ATM system could include a committed information rate (CIR), a peak information rate (PIR) and

No.	’740 Patent Claim 13	Wiher ’530
	information rate (EIR) of a communication service provided by the communication network to the network node, and	an excess information rate (EIR). Thus, at least under the apparent claim scope alleged by Orckit’s Infringement Disclosures, this limitation is met. To the extent that the Wiher ’530 is found to not meet this limitation, coupling each of the one or more interface modules to the communication network comprising specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node would have been obvious to a person having ordinary skill in the art, as explained below. Wiher ’530 at 2:10-18 (“Implementations of the invention may also include one or more of the following features. The apparatus may be implemented on a card configured to be plugged into a backplane having backplane mating connectors correspond-ing to the separate terminal connectors. A fiber-optic data link interface may couple the transceiver circuitry to a synchronous optical network (SONET) data link. A high bit-rate digital subscriber line (HDSL) data link interface may be coupled to the transceiver circuitry.”) Wiher ’530 at 6:8-25 (“FIG. 2 is an exemplary ATM network. ATM cells can be used to establish a communication path between network access equipment 201-208. Network access equipment 201-208 forms the originating and terminating point in an ATM communication and may convert non-ATM data traffic into an ATM format. Conversion of non-ATM data traffic to ATM cells is provided by ATM adaptation layer (AAL) services. Standard AAL services are defined in Bellcore publication GR-1113-CO RE, Asynchronous Transfer Mode and ATM Adaptation Layer (AAL) Protocols, 1994. AAL services may be used, for example, to convert a 1.544 megabit per second continuous bit rate (CBR) circuit- oriented Tl connection to an ATM virtual circuit connection or to convert variable-length packet data traffic originating on a local area network (LAN) to ATM cells for transport on an ATM network. ATM cells are sent from network access equipment to the ATM network using a user-network interface header 125 (FIG. 1B).”) Wiher ’530 at 7:47-58 (“ATM cells are sent between an ATM network access unit 201-208 and a line card in a line card shelf 211-214 over a wire loop 261-268. FIG. 3 illustrates a line

No.	’740 Patent Claim 13	Wiher ’530
		card shelf 300 having, for example, twelve line cards 301-312. Each line card 301-312 terminates, for example, two subscriber loop connections to network access units 201-208. The line cards support, for example, high bit rate digital subscriber line (HDSL), asymmetric digital subscriber line (ADSL), or a rate adaptive digital subscriber line (RADSL) data transmission over the subscriber loops. A line card shelf 300 also includes a main line card shelf multiplexer (LSM) 330 and a backup LSM 340.”) Wiher ’530 at 12:32-63 (“ATM cells are transferred from the line card to the by the exchange of LC-SR, LSM-RR, and LC-DATA signals on the line card to LSM cell transfer signal lines 612. FIG. 7C is a line card to LSM cell transport interface signal timing diagram. The signal timing diagram 750 illustrates timing and modulation of LC-SR, LSM-RR, and LC-DATA signals exchanged during an ATM cell transfer from the line card to a LSM. At clock cycle 2, the line card indicates that it is ready to transfer a data cell to a LSM by asserting (low) the LC-SR signal. At clock cycle 3, the LSM indicates that it is ready to accept data from the line card by asserting (low) the LSM-RR signal. Note that the LSM need not assert the LSM-RR signal immediately after receipt of the LC-SR signal, but rather the LSM may delay assertion of LSM-RR until it is ready to receive the data transfer. Following the LSM's assertion of the LSM-RR signal, the line card waits for two clock cycles (clock 3 and clock 4) before the start of data transfer. This two clock cycle delay facilitates backplane signal timing and LSM to line card synchronization. In alternative implementations, this two clock cycle delay period may be reduced or increased depending on, for example, backplane signal propagation characteristics and required LSM and line card circuitry response times. Following the two clock cycle delay period, the line card begin a serial transfer of data by modulating the LC-DATA signal. For example, FIG. 7C shows the transfer of a '1' bit during clock cycles 5, 8, 9, 11,426,427, and 428 by asserting (high) the LC-DATA signal and the transfer of a '0' bit during clock cycles 6, 7, 10, and 12 by de-asserting the LC-DATA signal. In alternative embodiments, the LSM-SR may be de-asserted once a cell transfer has begun, for example, at clock cycle 5 of FIG. 7B.”) Wiher ’530 at 30:45-55 (“Clock frequencies at line card, LSM, MLA, trunk card, and MCP interfaces may differ from those described herein. For example, a MCS may include MLA control interfaces having a 256 KHz clock frequency allowing data transfer between the MCS

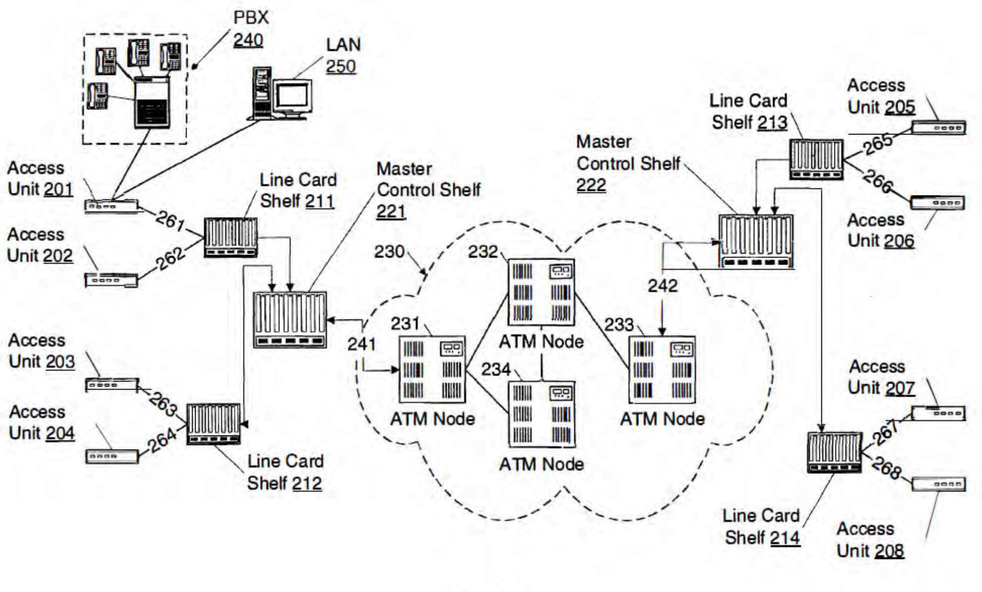
No.	’740 Patent Claim 13	Wiher ’530
		and MLA at 256 Kbits/second and trunk so cards may include cell transport interfaces having a clock frequency allowing data transfer between the trunk card and the MLA at 50 Mbytes/second. Clock frequencies may be varied depending on, for example, desired data transfer rates, signal propagation constraints, and circuitry response times.”) Under at least the apparent claim scope alleged by Orckit’s Infringement Disclosures, Wiher ’530 in combination with (1) the knowledge of a person of ordinary skill in the art, alone or in further combination with (2) each (individually, as well as one or more together) of the references identified in element 13[h] of Exhibit E-3 renders the claim, including the present limitation, obvious. Below are examples of two such references. For example, Bruckman discloses specifying certain committed, excess, and guaranteed bandwidths, including CIR, EIR, and PIR, respectively. Bruckman at [0013] (“Service level agreements between network service providers and customers commonly specify a certain com-mitted bandwidth, or committed information rate (CIR), which the service provider guarantees to provide to the customer at all times, regardless of bandwidth stress on the network. Additionally or alternatively, the agreement may specify an excess bandwidth, which is available to the customer when network traffic permits. The excess band-width is typically used by customers for lower-priority services, which do not require committed bandwidth. The network service provider may guarantee the customer a certain minimum excess bandwidth, or excess information rate (EIR), in order to avoid starvation of such services in case of bandwidth stress. In general, the bandwidth guaran-teed by a service provider, referred to as the peak informa-tion rate (PIR), may include either CIR, or EIR, or both CIR and EIR (in which case PIR=CIR+EIR). The term "guaran-teed bandwidth," as used in the context of the present patent application and in the claims, includes all these types of guaranteed bandwidth.”) As another example, Solomon discloses a service property of a guaranteed bandwidth, sometimes denoted as CIR-Committed Information Rate and PIR-Peak Information Rate.

No.	’740 Patent Claim 13	Wiher ’530
		Solomon at [0023] (“In another embodiment, establishing the path includes receiving an indication of a requested service property of the flow, and selecting the port includes assign-ing the port to the flow so as to comply with the requested service property. In a disclosed embodiment, the requested service property includes at least one of a guaranteed bandwidth, a peak bandwidth and a class-of-service. Addi-tionally or alternatively, assigning the port includes selecting the port having a maximum available bandwidth out of the plurality of aggregated ports. Further additionally or alter-natively, assigning the port includes selecting the port hav-ing a minimum available bandwidth out of the plurality of aggregated ports, which is still greater than or equal to the guaranteed bandwidth.”) Solomon at [0050] (“The method of FIG. 3 begins when the preceding node asks to establish a part of tunnel 28 (comprising one or more hops) for sending MPLS packets to MPLS/LAG switch 26 A. The preceding node requests and then receives the MPLS label, which it will subsequently attach to all packets that are sent to MPLS/LAG switch 26 labeledA. The preceding node sends downstream an RSVP-TE PATH mes-sage augmented with a LABEL_REQUEST object, as defined by RSVP-TE, to MPLS/LAG switch A, at a label requesting step 60. The PATH message typically comprises information regarding service properties that are requested for tunnel 28. The service properties may comprise a guar-anteed bandwidth (sometimes denoted CIR-Committed Information Rate) and a peak bandwidth (sometimes denoted PIR-Peak Information Rate), as well as a requested CoS (Class of Service-a measure of packet priority).”)
13[i]	allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements.	Wiher ’530 discloses allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements. For examples, Wiher ’530 discloses multi-wired communication service with additional signal couplings for alternative bandwidth requirements. Wiher ’530 at 10:5-24 (“In various implementations, transceivers 643 and 644 implement, for example, Digital Subscriber Line (DSL), Integrated Services Digital Network (ISDN), Rate Adaptive Digital Subscriber Line (RADSL), High Bit Rate Digital Subscriber Line (HDSL),

No.	’740 Patent Claim 13	Wiher ’530
		Asymmetric Digital Subscriber Line (ADSL) modulation, or other digital modulation technique. Line card circuitry 643-647 may be implemented in one or more integrated circuit chips, may include discrete circuit components, and may include additional functional-ity. In the line card 600, each transceiver 643 and 644 has a two-wire coupling to loop interface 632 over which both transmit and receive signals may be sent. In alternate implementations, transceivers 643 and 644 may have addi-tional signal couplings to provide, for example, a transmis-sion and reception of a four-wire service, or may provide a four-wire transceiver coupling for use with external hybrid circuitry that adapts the four-wire coupling to a two-wire loop interface. Still other transceiver-to-loop signal coupling arrangements may be used depending on the particular transceiver type and line card application.”)

No.	’740 Patent Claim 14	Wiher ’530
14[preamble]	A method for connecting user ports to a communication network, comprising:	Wiher ’530 discloses a method for connecting user ports to a communication network. For example, Wiher ’530 discloses a method in an ATM network in which ATM nodes are connected to line cards in a line card shelf via line card ports. Wiher ’530 at 6:26-42 (“Network access equipment 201-208 may combine data from multiple sources. For example, data from a LAN 250 and circuit oriented traffic, such as a Tl connection from a private branch exchange phone system (PBX) 240, may each be converted to ATM cells at network access equipment 201. ATM cells corresponding to LAN 250 and PBX 240 data are multiplexed together and sent by the network access equipment 201 over media 261 to a line card in a line card shelf 211. VPI and VCI information in transmitted ATM cells is used to uniquely identify data sources and destina-tions at, for example, network access equipment 201, line card shelf 211, master control shelf 221, and within the ATM network 230. For example, by assigning a unique VPI/VCI value to ATM cells transporting LAN 250 data and different VPI/VCI value to cells transporting PBX 240 data, inde-pendent routing and logical separation of the PBX 240 and LAN 250 data can be maintained.

No.	’740 Patent Claim 14	Wiher ’530
		Wiher ’530 at 6:43-64 (“ATM cells originating at network access equipment 201-208 are sent over transmission loops 261-268 between the network access equipment and line cards in line card shelves 211-214. A loop 261-268 may be, for example, a digital subscriber line operated over a twisted wire pair connection. A loop 261-268 terminates at a line card in a line card shelf 211-214. A line card shelf 211-214 may house multiple line cards. Each line card terminates one or more loop connections to network access equipment 201-208. A master control shelf 221, 222 is connected to one or more line card shelves. For example, line card shelves 211 and 212 are connected to master control shelf 221 and line card shelves 213 and 214 are connected to master control shelf 222. A master control shelf 221, 222 is a card shelf that controls and regulates the flow of data between line card shelves and a trunk interface 241, 242. A trunk interface 241, 242 provides a trunk connection 241, 242 between the master control shelf 221, 222 and the ATM network 230. Trunk interface 241, 242 is, for example, a 45 Mbit/second T-3 interface or a standard 155 Mbit/second fiber optic synchronous optical network optical carrier level 3 concatenated data (SONET OC-3c) interface.”) Wiher ’530 at 6:65-7:14 (“In an ATM network, a particular VPI/VCI value in a cell header is used to route a cell between the switching ports of two connected nodes, but the particular VPI/VCI value does not provide for routing through multiple nodes. To route a cell from one endpoint to another endpoint through multiple nodes in an ATM network, VPI/VCI information must be translated at each node. Thus, to route an ATM cell, a node performs the following steps: 1) an incoming cell's VPI/VCI information is read, 2) a node output port providing cell transport to a destination node is determined based on the VPI/VCI information in the incoming cell's header; 3) the node replaces the cell's VPINCI information with new VPI/VCI information for routing through the destination node, and 4) the node forwards the cell through the deter-mined output port to the destination node. The destination node repeats this process until the cell reaches its final destination.”) Wiher ’530 at 7:47-58 (“ATM cells are sent between an ATM network access unit 201-208 and a line card in a line card shelf 211-214 over a wire loop 261-268. FIG. 3 illustrates a line card shelf 300 having, for example, twelve line cards 301-312. Each line card 301-312

No.	’740 Patent Claim 14	Wiher ’530
		terminates, for example, two subscriber loop connections to network access units 201-208. The line cards support, for example, high bit rate digital subscriber line (HDSL), asymmetric digital subscriber line (ADSL), or a rate adaptive digital subscriber line (RADSL) data transmission over the subscriber loops. A line card shelf 300 also includes a main line card shelf multiplexer (LSM) 330 and a backup LSM 340.”) Wiher ’530 at Figure 2  PBX 240 LAN 250 Access Unit 201 Access Unit 202 Access Unit 203 Access Unit 204 261 262 263 264 Line Card Shelf 211 Line Card Shelf 212 Master Control Shelf 221 230 231 232 233 234 241 242 ATM Node ATM Node ATM Node ATM Node Master Control Shelf 222 Line Card Shelf 213 Access Unit 205 265 266 Access Unit 206 Access Unit 207 267 268 Line Card Shelf 214 Access Unit 208 Fig. 2 Wiher ’530 at 11:40-57 (“ATM cells are transferred from a LSM to a line card by the exchange of the LC-RR, LSM-SR, and LSM-DATA signals on the LSM to line card cell transport signal lines 611. FIGS. 7A and 7B illustrate timing and modulation of the LC-RR, LSM-SR, and LSM-DATA signals. The LC-RR signal is sent from the line card to the LSM to indicate line

No.	’740 Patent Claim 14	Wiher ’530
		card ports that are ready to receive ATM cell transfers. FIG. 7A is a LC-RR signal timing diagram for a line card supporting two line card ports. The LC-RR signal is modu-lated to periodically transmit a framing indicator and port status information from the line card to the LSM. The framing indicator is sent by asserting the LC-RR signal for one clock cycle at, for example, sixteen clock cycle inter-vals. During each clock cycle following the framing indicator, port status information may be sent from the line card to the LSM. Port status information is sent by asserting or de-asserting the LC-RR signal during a clock period that is unique for each port on the line card.”) Wiher ’530 at 12:4-12 (“A line card supporting more than two subscriber loops will typically have additional ports. For example, a line card supporting four subscriber loops may receive data from the LSM at four line card ports. A line card with more than two ports will convey additional port status information follow-ing the illustrated 'P2' indicator period in FIG. 7A. In an alternative line card implementation, VPI/VCI information rather than a line card port identifier, may be used to identify the particular destination subscriber loop.”)
14[a]	coupling the user ports to one or more user interface modules;	Wiher ’530 discloses coupling the user ports to one or more user interface modules. *See supra at* 1[a].
14[b]	coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel,	Wiher ’530 discloses coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel. *See supra at* 1[c], 3.
14[c]	at least one of said backplane traces being bi-directional	Wiher ’530 discloses at least one of said backplane traces being bi-directional and operative to communicate in both an upstream direction and a downstream direction.

No.	’740 Patent Claim 14	Wiher ’530
	and operative to communicate in both an upstream direction and a downstream direction;	*See supra at* 14[b], 1[d].
14[d]	receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes;	Wiher ’530 discloses receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes. *See supra at* 14[a], 1[e].
14[e]	for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces; and	Wiher ’530 discloses for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces. *See supra at* 14[b], 1[f].
14[f]	sending the data frame over the selected backplane trace;	Wiher ’530 discloses sending the data frame over the selected backplane trace. *See supra at* 14[e], 1[g].
14[g]	said sending comprising communicating along said at least one of said backplane traces.	Wiher ’530 discloses said sending comprising communicating along said at least one of said backplane traces. *See supra at* 14[f], 1[h].

No.	’740 Patent Claim 15	Wiher ’530
15[preamble]	A method for connecting user ports to a communication network, comprising:	Wiher ’530 discloses a method for connecting user ports to a communication network. *See supra at* 14[preamble].
15[a]	coupling the user ports to one or more user interface modules;	Wiher ’530 discloses coupling the user ports to one or more user interface modules. *See supra at* 14[a].
15[b]	coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel;	Wiher ’530 discloses coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel. *See supra at* 14[b].
15[c]	receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes;	Wiher ’530 discloses receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes. *See supra at* 14[d].
15[d]	for each data frame, selecting responsively to at least one of the respective frame attributes a backplane	Wiher ’530 discloses for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces. *See supra at* 14[e].

No.	’740 Patent Claim 15	Wiher ’530
	trace from the two or more backplane traces; and	
15[e]	sending the data frame over the selected backplane trace,	Wiher ’530 discloses sending the data frame over the selected backplane trace. *See supra at* 14[f].
15[f]	at least some of the backplane traces being aggregated into an Ethernet link aggregation (LAG) group.	Wiher ’530 discloses at least some of the backplane traces being aggregated into an Ethernet link aggregation (LAG) group. *See supra at* 15[e], 4[f], 3.

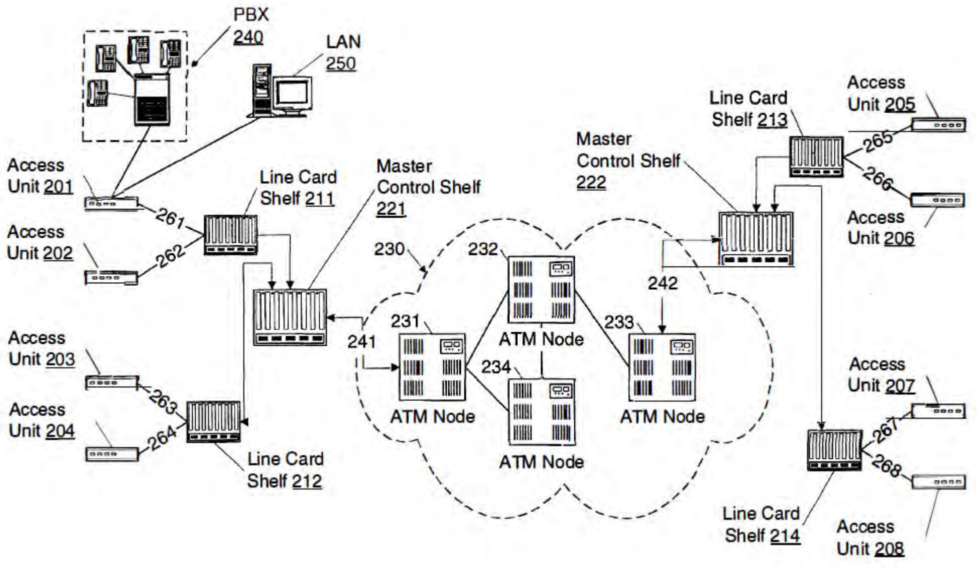
No.	’740 Patent Claim 16	Wiher ’530
16	The method according to claim 14, wherein selecting the backplane trace comprises applying a hashing function to the at least one of the frame attributes.	Wiher ’530 discloses the method according to claim 14, wherein selecting the backplane trace comprises applying a hashing function to the at least one of the frame attributes. *See supra* at 14, 9, 8.

No.	’740 Patent Claim 17	Wiher ’530
17[preamble]	Apparatus for connecting a network node with a	Wiher ’530 discloses apparatus for connecting a network node with a communication network. For example, Wiher ’530 discloses an system for connecting an ATM network with network access equipment.

No.	’740 Patent Claim 17	Wiher ’530
	communication network, comprising:	Wiher ’530 at Abstract (“A system and method of transmitting data cells are disclosed. The system includes a data transmitting and receiving unit including transceiver circuitry, a main backplane interface, and backplane interconnection circuitry. The transceiver circuitry transmits and receive data cells over the data link, the main backplane interface provides physical interconnection to the backplane, and the backplane inter-connection circuitry transmits and receives cells. The main backplane interface including at least one cell signal termi-nal and at least one operations data signal terminal. The operations data signal terminals are separate from the cell signal terminals. The operations data signal terminals and the cell signal terminals are configured to connect to mating connectors on a backplane. Backplane interconnection cir-cuitry couples the transceiver circuitry to the main back-plane interface. The interconnection circuitry receives data cells from the transceiver circuitry and transmit them over cell signal terminals, receives data cells from the cell signal terminals and provide them to the transceiver circuitry for transmission over the first data link, and transmit and receive operations data over the operations data signal terminals. The method includes asserting a signal indicating the pri-ority of a cell to be transmitted over the backplane, receiving on a second backplane signal line a signal that the apparatus may begin transmitting the data cell, and transmitting bits of the data cell on a third backplane signal line after receiving the signal that the apparatus may begin transmitting the data cell.”) Wiher ’530 at 1:4-9 (“The present invention relates to the transmission of asyn-chronous transfer mode (ATM) cells. Asynchronous transfer mode (ATM) data transfer is a communication technology in which fixed-size packets of data, known as "cells," are transferred between ATM switch-ing devices ("switches").”) Wiher ’530 at 3:25-34 (“In general, in another aspect, the invention features a method of sending a data cell over backplane signal lines. The method includes asserting on a first backplane signal line a signal indicating the priority of a cell to be transmitted over the backplane. Receiving on a second backplane signal line a signal that the apparatus may begin transmitting the data cell, and transmitting bits of the data cell on a third backplane signal line after receiving the signal that the apparatus may begin transmitting the data cell.”)

No.	’740 Patent Claim 17	Wiher ’530
		Wiher ’530 at 3:35-42 (“In general, in another aspect, the invention features a method of receiving a data cell. The method includes asserting on a first backplane signal line a signal identifying an addressable apparatus port that is ready to receive a data cell. Receiving on a second signal line a signal indicating that a data cell is being transferred to the apparatus, and receiving on a third signal line bits of the data cell being transferred to the apparatus.”) Wiher ’530 at 4:45-57 (“In general, in another aspect, the invention features a method of transmitting a data cell over a backplane. The method includes receiving a data cell over a data link interface. Examining header information in the received data cell. Selecting one of a plurality of backplane cell interfaces. Receiving on a first signal line of the selected interface a signal indicating that a data cell may be transmitted on the interface. Transmitting on a second signal line of the selected interface a signal indicating that transfer of the data cell is occurring, and transmitting bits of the data cell on a third signal line of the selected interface. Implementations of the invention may include selecting based on a port address in a data cell header.”)
17[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network,	Wiher ’530 discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. For example, Wiher ’530 discloses line cards in a line card shelf that are arranged to process ATM data cells with specific cell information transmitted between network access equipment and the ATM network. Wiher ’530 at 6:26-42 (“Network access equipment 201-208 may combine data from multiple sources. For example, data from a LAN 250 and circuit oriented traffic, such as a Tl connection from a private branch exchange phone system (PBX) 240, may each be converted to ATM cells at network access equipment 201. ATM cells corresponding to LAN 250 and PBX 240 data are multiplexed together and sent by the network access equipment 201 over media 261 to a line card in a line card shelf 211. VPI and VCI information in transmitted ATM cells is used to uniquely identify data sources and destina-tions at, for example, network access equipment

No.	’740 Patent Claim 17	Wiher ’530
		201, line card shelf 211, master control shelf 221, and within the ATM network 230. For example, by assigning a unique VPI/VCI value to ATM cells transporting LAN 250 data and different VPI/VCI value to cells transporting PBX 240 data, inde-pendent routing and logical separation of the PBX 240 and LAN 250 data can be maintained. Wiher ’530 at 6:43-64 (“ATM cells originating at network access equipment 201-208 are sent over transmission loops 261-268 between the network access equipment and line cards in line card shelves 211-214. A loop 261-268 may be, for example, a digital subscriber line operated over a twisted wire pair connection. A loop 261-268 terminates at a line card in a line card shelf 211-214. A line card shelf 211-214 may house multiple line cards. Each line card terminates one or more loop connections to network access equipment 201-208. A master control shelf 221, 222 is connected to one or more line card shelves. For example, line card shelves 211 and 212 are connected to master control shelf 221 and line card shelves 213 and 214 are connected to master control shelf 222. A master control shelf 221, 222 is a card shelf that controls and regulates the flow of data between line card shelves and a trunk interface 241, 242. A trunk interface 241, 242 provides a trunk connection 241, 242 between the master control shelf 221, 222 and the ATM network 230. Trunk interface 241, 242 is, for example, a 45 Mbit/second T-3 interface or a standard 155 Mbit/second fiber optic synchronous optical network optical carrier level 3 concatenated data (SONET OC-3c) interface.”) Wiher ’530 at 6:65-7:14 (“In an ATM network, a particular VPI/VCI value in a cell header is used to route a cell between the switching ports of two connected nodes, but the particular VPI/VCI value does not provide for routing through multiple nodes. To route a cell from one endpoint to another endpoint through multiple nodes in an ATM network, VPI/VCI information must be translated at each node. Thus, to route an ATM cell, a node performs the following steps: 1) an incoming cell's VPI/VCI information is read, 2) a node output port providing cell transport to a destination node is determined based on the VPI/VCI information in the incoming cell's header; 3) the node replaces the cell's VPINCI information with new VPI/VCI information for routing through the destination node, and 4) the node forwards the cell through the deter-mined output port to the destination node. The destination node repeats this process until the cell reaches its final destination.”)

No.	’740 Patent Claim 17	Wiher ’530
		Wiher ’530 at 7:47-58 (“ATM cells are sent between an ATM network access unit 201-208 and a line card in a line card shelf 211-214 over a wire loop 261-268. FIG. 3 illustrates a line card shelf 300 having, for example, twelve line cards 301-312. Each line card 301-312 terminates, for example, two subscriber loop connections to network access units 201-208. The line cards support, for example, high bit rate digital subscriber line (HDSL), asymmetric digital subscriber line (ADSL), or a rate adaptive digital subscriber line (RADSL) data transmission over the subscriber loops. A line card shelf 300 also includes a main line card shelf multiplexer (LSM) 330 and a backup LSM 340.”) Wiher ’530 at Figure 2  PBX 240 LAN 250 Access Unit 201 Access Unit 202 Access Unit 203 Access Unit 204 Access Unit 205 Access Unit 206 Access Unit 207 Access Unit 208 Line Card Shelf 211 Line Card Shelf 212 Line Card Shelf 213 Line Card Shelf 214 Master Control Shelf 221 Master Control Shelf 222 230 231 232 233 234 241 242 261 262 263 264 265 266 267 268 ATM Node ATM Node ATM Node ATM Node Fig. 2

No.	’740 Patent Claim 17	Wiher ’530
		Wiher ’530 at 11:40-57 (“ATM cells are transferred from a LSM to a line card by the exchange of the LC-RR, LSM-SR, and LSM-DATA signals on the LSM to line card cell transport signal lines 611. FIGS. 7A and 7B illustrate timing and modulation of the LC-RR, LSM-SR, and LSM-DATA signals. The LC-RR signal is sent from the line card to the LSM to indicate line card ports that are ready to receive ATM cell transfers. FIG. 7A is a LC-RR signal timing diagram for a line card supporting two line card ports. The LC-RR signal is modu-lated to periodically transmit a framing indicator and port status information from the line card to the LSM. The framing indicator is sent by asserting the LC-RR signal for one clock cycle at, for example, sixteen clock cycle inter-vals. During each clock cycle following the framing indicator, port status information may be sent from the line card to the LSM. Port status information is sent by asserting or de-asserting the LC-RR signal during a clock period that is unique for each port on the line card.”) Wiher ’530 at 12:4-12 (“A line card supporting more than two subscriber loops will typically have additional ports. For example, a line card supporting four subscriber loops may receive data from the LSM at four line card ports. A line card with more than two ports will convey additional port status information follow-ing the illustrated 'P2' indicator period in FIG. 7A. In an alternative line card implementation, VPI/VCI information rather than a line card port identifier, may be used to identify the particular destination subscriber loop.”)
17[b]	at least one of said interface modules being operative to communicate in both an upstream direction and a downstream direction;	Wiher ’530 discloses at least one of said interface modules being operative to communicate in both an upstream direction and a downstream direction. For example, Wiher ’530 discloses line cards in a line card shelf over which data cells may be transmitted or received in both directions. Wiher ’530 at 6:65-7:14 (“In an ATM network, a particular VPI/VCI value in a cell header is used to route a cell between the switching ports of two connected nodes, but the particular VPINCI value does not provide for routing through multiple nodes. To route a cell from one endpoint to another endpoint through multiple nodes in an ATM network, VPI/VCI information must be translated at each node. Thus, to route an ATM cell, a node performs the

No.	’740 Patent Claim 17	Wiher ’530
		following steps: 1) an incoming cell's VPINCI information is read, 2) a node output port providing cell transport to a destination node is determined based on the VPI/VCI information in the incoming cell's header; 3) the node replaces the cell's VPINCI information with new VPI/VCI information for routing through the destination node, and 4) the node forwards the cell through the deter-mined output port to the destination node. The destination node repeats this process until the cell reaches its final destination.”) Wiher ’530 at 10:5-24 (“In various implementations, transceivers 643 and 644 implement, for example, Digital Subscriber Line (DSL), Integrated Services Digital Network (ISDN), Rate Adaptive Digital Subscriber Line (RADSL), High Bit Rate Digital Subscriber Line (HDSL), Asymmetric Digital Subscriber Line (ADSL) modulation, or other digital modulation tech-nique. Line card circuitry 643-647 may be implemented in one or more integrated circuit chips, may include discrete circuit components, and may include additional functional-ity. In the line card 600, each transceiver 643 and 644 has a two-wire coupling to loop interface 632 over which both transmit and receive signals may be sent. In alternate implementations, transceivers 643 and 644 may have addi-tional signal couplings to provide, for example, a transmis-sion and reception of a four-wire service, or may provide a four-wire transceiver coupling for use with external hybrid circuitry that adapts the four-wire coupling to a two-wire loop interface. Still other transceiver-to-loop signal coupling arrangements may be used depending on the particular transceiver type and line card application.”) Wiher ’530 at 12:4-12 (“A line card supporting more than two subscriber loops will typically have additional ports. For example, a line card supporting four subscriber loops may receive data from the LSM at four line card ports. A line card with more than two ports will convey additional port status information follow-ing the illustrated 'P2' indicator period in FIG. 7A. In an alternative line card implementation, VPI/VCI information, rather than a line card port identifier, may be used to identify the particular destination subscriber loop.”) Wiher ’530 at 12:13-31 (“When a line card port is ready to receive an ATM cell, a cell may be transferred using the LSM-SR and LSM-DATA signals. FIG. 7B is an exemplary signal timing diagram illustrating states of the LSM-SR and LSM-DATA signals during an ATM cell

No.	’740 Patent Claim 17	Wiher ’530
		transfer from the LSM to the line card. When a LSM is ready to send a cell to a waiting line card port, the LSM asserts (low) the LSM-SR signal and simultaneously begins modulating data over the interface 611 using the LSM-DATA signal. For example, at clock 4, the LSM asserts the LSM-SR signal and begins a serial transfer of a data cells by modulating the LSM-DATA signal. To send a '1' bit value, the LSM-DATA signal is asserted (high) during a clock cycle, and to send a '0' bit value the LSM-DATA signal is de-asserted (low) during a clock cycle. To send a 53-byte (424 bit) ATM cell, the LSM-DATA signal is modulated for a 424 clock cycle period. After transmission of the ATM cell, the LSM-SR signal is de-asserted. When the LSM-SR signal is de-asserted, the LSM-DATA signal is not sampled.”)
17[c]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Wiher ’530 discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 1[a].
17[d]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Wiher ’530 discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 1[c].
17[e]	a control module, which is arranged to select for each data frame sent between the communication network and the	Wiher ’530 discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 1[f].

No.	’740 Patent Claim 17	Wiher ’530
	network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame;	
17[f]	at least one of said first physical links and at least one of said second links being bi-directional links operative to communicate in both said upstream direction and said downstream direction.	Wiher ’530 discloses at least one of said first physical links and at least one of said second links being bi-directional links operative to communicate in both said upstream direction and said downstream direction. *See supra at* 1[b], 1[d].

No.	’740 Patent Claim 18	Wiher ’530
18[a]	The apparatus according to claim 17, and comprising a backplane to which the one or more	Wiher ’530 discloses the apparatus according to claim 17, and comprising a backplane to which the one or more interface modules are coupled. *See supra at* 3, 17.

No.	’740 Patent Claim 18	Wiher ’530
	interface modules are coupled,	
18[b]	wherein the second physical links comprise backplane traces formed on the backplane.	Wiher ’530 discloses wherein the second physical links comprise backplane traces formed on the backplane. *See supra at* 3, 17.

No.	’740 Patent Claim 19	Wiher ’530
19[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Wiher ’530 discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
19[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	Wiher ’530 discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].
19[b]	a first group of first physical links arranged in parallel so as to couple the network node to the	Wiher ’530 discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].

No.	’740 Patent Claim 19	Wiher ’530
	one or more interface modules;	
19[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Wiher ’530 discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].
19[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,	Wiher ’530 discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].
19[e]	at least one of the first and second groups of physical links comprising an	Wiher ’530 discloses at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group. *See supra at* 4[f].

No.	’740 Patent Claim 19	Wiher ’530
	Ethernet link aggregation (LAG) group.	

No.	’740 Patent Claim 20	Wiher ’530
20[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Wiher ’530 discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
20[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	Wiher ’530 discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].
20[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Wiher ’530 discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].
20[c]	a second group of second physical links arranged in parallel so as to couple the	Wiher ’530 discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].

No.	’740 Patent Claim 20	Wiher ’530
	one or more interface modules to the communication network; and	
20[d]	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,	Wiher ’530 discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].
20[e]	two or more of the first physical links being aggregated into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node.	Wiher ’530 discloses two or more of the first physical links being aggregated into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node. *See supra at* 19[e], 5[f].

No.	’740 Patent Claim 21	Wiher ’530
21	The apparatus according to claim 17, and comprising a multiplexer, which is arranged to perform at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node.	Wiher ’530 discloses the apparatus according to claim 17, and comprising a multiplexer, which is arranged to perform at least one of multiplexing upstream data frames sent from the network node to the communication network, and demultiplexing downstream data frames sent from the communication network to the network node. *See supra at* 6, 17.

No.	’740 Patent Claim 22	Wiher ’530
22	The apparatus according to claim 17, wherein the control module is arranged to balance a frame data rate among at least some of the first and second physical links.	Wiher ’530 discloses the apparatus according to claim 17, wherein the control module is arranged to balance a frame data rate among at least some of the first and second physical links. *See supra at* 7, 17.

No.	’740 Patent Claim 23	Wiher ’530
23	The apparatus according to claim 17, wherein the control module is arranged to apply a mapping function to the at least one of the frame attributes so as to select the first and second physical links.	Wiher ’530 discloses the apparatus according to claim 17, wherein the control module is arranged to apply a mapping function to the at least one of the frame attributes so as to select the first and second physical links. *See supra at* 8, 17.

No.	’740 Patent Claim 24	Wiher ’530
24	The apparatus according to claim 23, wherein the mapping function comprises a hashing function.	Wiher ’530 discloses the apparatus according to claim 23, wherein the mapping function comprises a hashing function. *See supra at* 9, 23.

No.	’740 Patent Claim 25	Wiher ’530
25[a]	The apparatus according to claim 24, wherein the control module is arranged to determine a hashing size responsively to a	Wiher ’530 discloses the apparatus according to claim 24, wherein the control module is arranged to determine a hashing size responsively to a number of at least some of the first and second physical links. *See supra at* 10[a], 24.

No.	’740 Patent Claim 25	Wiher ’530
	number of at least some of the first and second physical links,	
25[b]	to apply the hashing function to the at least one of the frame attributes to produce a hashing key,	Wiher ’530 discloses to apply the hashing function to the at least one of the frame attributes to produce a hashing key. *See supra at* 10[b].
25[c]	to calculate a modulo of a division operation of the hashing key by the hashing size, and	Wiher ’530 discloses to calculate a modulo of a division operation of the hashing key by the hashing size. *See supra at* 10[c].
25[d]	to select the first and second physical links responsively to the modulo.	Wiher ’530 discloses to select the first and second physical links responsively to the modulo. *See supra at* 10[d].

No.	’740 Patent Claim 26	Wiher ’530
26	The apparatus according to claim 25, wherein the control module is arranged to select the first and second physical links responsively to respective first and	Wiher ’530 discloses the apparatus according to claim 25, wherein the control module is arranged to select the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo. *See supra at* 11, 25.

No.	’740 Patent Claim 26	Wiher ’530
	second subsets of bits in a binary representation of the modulo.	

No.	’740 Patent Claim 27	Wiher ’530
27	The apparatus according to claim 17, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port.	Wiher ’530 discloses the apparatus according to claim 17, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port. *See supra at* 12, 17.

No.	’740 Patent Claim 28	Wiher ’530
28[preamble]	Apparatus for connecting a network node with a communication network, comprising:	Wiher ’530 discloses apparatus for connecting a network node with a communication network. *See supra at* 17[preamble].
28[a]	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;	Wiher ’530 discloses one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network. *See supra at* 17[a].
28[b]	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;	Wiher ’530 discloses a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules. *See supra at* 17[c].
28[c]	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and	Wiher ’530 discloses a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network. *See supra at* 17[d].
28[d]	a control module, which is arranged to	Wiher ’530 discloses a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on

No.	’740 Patent Claim 28	Wiher ’530
	select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,	at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame. *See supra at* 17[e].
28[e]	the communication network being arranged to provide a communication service to the network node,	Wiher ’530 discloses the communication network being arranged to provide a communication service to the network node. *See supra at* 2[b].
28[f]	the service having specified bandwidth requirements comprising at least one of a committed information rate (CR), a peak information rate (PIR) and an excess	Wiher ’530 discloses the service having specified bandwidth requirements comprising at least one of a committed information rate (CR), a peak information rate (PIR) and an excess information rate (EIR). *See supra at* 13[i].

No.	’740 Patent Claim 28	Wiher ’530
	information rate (EIR), and	
28[g]	the first and second groups of physical links being dimensioned to provide an allocated bandwidth for the communication service responsively to the bandwidth requirements.	Wiher ’530 discloses the first and second groups of physical links being dimensioned to provide an allocated bandwidth for the communication service responsively to the bandwidth requirements. *See supra at* 13[j].

No.	’740 Patent Claim 29	Wiher ’530
29[preamble]	Apparatus for connecting user ports to a communication network, comprising:	Wiher ’530 discloses apparatus for connecting user ports to a communication network. *See supra at* 17[preamble], 14[preamble].
29[a]	one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network,	Wiher ’530 discloses one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network. *See supra at* 17[a], 14[a].

No.	’740 Patent Claim 29	Wiher ’530
29[b]	at least one of said user interface modules being bi-directional and operative to communicate in both an upstream direction and a downstream direction;	Wiher ’530 discloses at least one of said user interface modules being bi-directional and operative to communicate in both an upstream direction and a downstream direction. *See supra at* 17[b], 14[c].
29[c]	a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network,	Wiher ’530 discloses a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network. *See supra at* 14[b]-[e].
29[d]	at least one of said backplane traces being bi-directional and operative to communicate in both said upstream direction and said	Wiher ’530 discloses at least one of said backplane traces being bi-directional and operative to communicate in both said upstream direction and said downstream direction. *See supra at* 14[c], 17[b].

No.	’740 Patent Claim 29	Wiher ’530
	downstream direction; and	
29[e]	a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame.	Wiher ’530 discloses a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame. *See supra at* 14[e], 17[e].

No.	’740 Patent Claim 30	Wiher ’530
30[preamble]	Apparatus for connecting user ports to a communication network, comprising:	Wiher ’530 discloses apparatus for connecting user ports to a communication network. *See supra at* 29[preamble].
30[a]	one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network;	Wiher ’530 discloses one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network. *See supra at* 29[a].

No.	’740 Patent Claim 30	Wiher ’530
30[b]	a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network;	Wiher ’530 discloses a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network. *See supra at* 29[c].
30[c]	a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame;	Wiher ’530 discloses a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame. *See supra at* 29[e].
30[d]	at least some of the backplane traces are aggregated into an Ethernet link	Wiher ’530 discloses at least some of the backplane traces are aggregated into an Ethernet link aggregation (LAG) group. *See supra at* 4[f], 15[f].

No.	’740 Patent Claim 30	Wiher ’530
	aggregation (LAG) group.	

No.	’740 Patent Claim 31	Wiher ’530
31	The apparatus according to claim 29, wherein the control module is arranged to apply a hashing function to the at least one of the frame attributes so as to select the backplane trace.	Wiher ’530 discloses the apparatus according to claim 29, wherein the control module is arranged to apply a hashing function to the at least one of the frame attributes so as to select the backplane trace. *See supra at* 16, 29, 30[c].