11-02-00

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P. O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION A

ATTORNEY DOCKET NO. 10008102-1

JC956 U.S. PTO 10/31/00

IN THE U.S. PATENT AND TRADEMARK OFFICE
Patent Application Transmittal Letter

COMMISSIONER FOR PATENTS
Washington, D.C. 20231

JC825 U.S. PTO 09/703942 10/31/00

Sir:

Transmitted herewith for filing under 37 CFR 1.53(b) is a(n): **(X)** Utility ( ) Design

(X) original patent application,

( ) continuation-in-part application

INVENTOR(S): **Eric A Pulsipher et al**

TITLE: **Method And System For Identifying And Processing Changes To A Network Topology**

Enclosed are:

(X) The Declaration and Power of Attorney. (X) signed ( ) unsigned or partially signed

(X) 26 sheets of drawings (one set) ( ) Associate Power of Attorney

( ) Form PTO-1449 ( ) Information Disclosure Statement and Form PTO-1449

( ) Priority document(s) ( ) (Other) ____________ (fee $ ________)

CLAIMS AS FILED BY OTHER THAN A SMALL ENTITY				
(1) FOR	(2) NUMBER FILED	(3) NUMBER EXTRA	(4) RATE	(5) TOTALS
TOTAL CLAIMS	20 — 20	0	X $18	$ 0
INDEPENDENT CLAIMS	3 — 3	0	X $80	$ 0
ANY MULTIPLE DEPENDENT CLAIMS	0		$270	$ 0
BASIC FEE: Design ($320.00); Utility ($710.00)				$ 710
TOTAL FILING FEE				$ 710
OTHER FEES				$
TOTAL CHARGES TO DEPOSIT ACCOUNT				$ 710

Charge $ 710 to Deposit Account 08-2025. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16, 1.17,1.19, 1.20 and 1.21. A duplicate copy of this sheet is enclosed.

"Express Mail" label no. EL523338183US

Date of Deposit Oct. 31, 2000

I hereby certify that this is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to: Commissioner for Patents, Washington, D.C. 20231.

By Laura M Clark

Typed Name: Laura M. Clark

Respectfully submitted,

Eric A Pulsipher et al

By T. G. R.

T. Grant Ritz

Attorney/Agent for Applicant(s)

Reg. No. **39,819**

Date: **Oct. 31, 2000**

Telephone No.: **(970) 898-0697**

HP 2003
ServiceNow v. HP
IPR2015-00717

Rev 10/00 (TransNew)

- Attach as First Page to Transmitted Papers -

Title

Method and System for Identifying and Processing Changes to a Network Topology

Field of Invention

The present invention relates generally to computer networks. More particularly, it relates to a method and system for identifying changes to a network topology and for acting upon the network based on the changes.

Background

As communications networks, such as the Internet, carry more and more traffic, efficient use of the bandwidth available in the network becomes more and more important. Switching technology was developed in order to reduce congestion and associated competition for the available bandwidth. Switching technology works by restricting traffic. Instead of broadcasting a given data packet to all parts of the network, switches are used to control data flow such that the data packet is sent only along those network segments necessary to deliver it to the target node. The smaller volume of traffic on any given segment results in few packet collisions on that segment and, thus, the smoother and faster delivery of data. A choice between alternative paths is usually possible and is typically made based upon current traffic patterns.

The intelligent routing of data packets with resultant reduction in network congestion can only be effected if the network topology is known. The topology of a network is a description of the network which includes the location of and interconnections between nodes on the network. The word "topology" refers to either the physical or logical layout of the network, including devices, and their connections in relationship to one another. Information necessary to create the topology layout can be derived from tables stored in network devices such as hubs, bridges, and switches. The information in these tables is in a constant state of flux as new entries are being added and old entries time out. Many times there simply is not enough information to determine where to place a particular device.

Switches examine each data packet that they receive, read the source addresses, and log those addresses into tables along with the switch ports on which the packets were received. If a packet is received with a target address without an entry in the switches table, the switch receiving it

broadcasts that packet to each of its ports. When the switch receives a reply, it will have identified where the new node lies.

In a large network with multiple possible paths from the switch to the target node, this table can become quite large and may require a significant amount of the switch's resources to develop and maintain. As an additional complication, the physical layout of devices and their connections are typically in a state of constant change. Devices are continually being removed from, added to, and moved to new physical locations on the network. To be effectively managed, the topology of a network must be accurately and efficiently ascertained, as well as maintained.

Existing mapping methods have limitations that prevent them from accurately mapping topological relationships. Multiple connectivity problems are one sort of difficulty encountered by existing methods. For example, connectors such as routers, switches, and bridges may be interconnected devices in a network. Some existing methods assume that these devices have only a single connection between them. In newer devices, however, it is common for manufacturers to provide multiple connections between devices to improve network efficiency and to increase capacity of links between the devices. The multiple connectivity allows the devices to maintain connection in case one connection fails. Methods that do not consider multiple connectivity do not present a complete and accurate topological map of the network.

Another limitation of existing topology methods is the use of a single reference to identify a device. Existing methods use a reference interface or a reference address in a set of devices to orient all other devices in the same area. These methods assumed that every working device would be able to identify, or "hear," this reference and identify it with a particular port of the device. With newer devices, however, it is possible that the same address or reference may be heard out of multiple ports of the same device. It is also possible that the address or reference may not be heard from any ports, for example, if switching technology is used.

Still another limitation of existing mapping systems is that they require a complete copy of the topological database to be stored in memory. In larger networks, the database is so large that this really is not feasible, because it requires the computer to be very large and expensive.

Still another difficulty with existing systems is that they focus on the minutia without considering the larger mapping considerations. Whenever an individual change in the system is detected, existing methods immediately act on that change, rather than taking a broader view of the change in the context of other system changes. For example, a device may be removed from the network temporarily and replaced with its ports reversed. In existing systems, this swapped port scenario could require hundreds or thousands of changes because the reference addresses will have changed for all interconnected devices.

Still another disadvantage of existing methods is that they use a continuous polling paradigm. These methods continuously poll network addresses throughout the day and make decisions based on those continuous polling results. This creates traffic on the network that slows other processes.

Still another limitation of existing methods is the assumption that network parts of a particular layer would be physically separated from other parts. Network layer 1 may represent the physical cabling of the network, layer 2 may represent the device connectivity, and layer 3 may represent a higher level of abstraction, such as the groupings of devices into regions. Existing methods assume that all layer 3 region groupings are self-contained, running on the same unique physical networking. However, in an internet protocol (IP) network, multiple IP domains may co-exist on the same lower layer networking infrastructure. It has become common for a network to employ a virtual local area network (LAN) to improve security or to simplify network maintenance, for example. Using virtual LANs, a system may have any number of different IP domains sharing the same physical connectivity. As a result, existing methods create confusion with respect to topological mapping because networks with multiple IP addresses in different subnets for the infrastructure devices cannot be properly represented because they assume the physical separation of connectivity for separate IP domains. Still another limitation of existing methods is that they do not allow topological loops, such as port aggregation or trunking, and switch meshing.

Summary of Invention

A method and system are disclosed for mapping the topology of a network having interconnected nodes by identifying changes in the network and updating a stored network topology

based on the changes. The nodal connections are represented by data tuples that store information such as a host identifier, a connector interface, and a port specification for each connection. A topology database stores an existing topology of a network. A topology converter accesses the topology database and converts the existing topology into a list of current tuples. A connection calculator calculates tuples to represent connections in the new topology. The topology converter receives the new tuples, identifies changes to the topology, and updates the topology database using the new tuples. The topology converter identifies duplicate tuples that appear in both the new tuples and the existing tuples and marks the duplicate tuples to reflect that no change has occurred to these connections. The topology converter attempts to resolve swapped port conditions and searches for new singly-heard and multi-heard host link tuples in the list of existing tuples. The topology converter also searches for new conflict link tuples in the existing tuples. The topology converter updates the topology database with the new topology.

Summary of Drawings

Figure 1 is a drawing of a typical topological bus segment for representing the connectivity of nodes on a network.

Figure 2 is a drawing of a typical topological serial segment for representing the connectivity of nodes on a network.

Figure 3 is a drawing of a typical topological star segment for representing the connectivity of nodes on a network.

Figure 4 is a drawing of another typical topological star segment for representing the connectivity of nodes on a network.

Figure 5 is a drawing of the connectivity of an example network system.

Figure 6 is a drawing of the connectivity of another example network system.

Figure 7 is a block diagram of the system.

Figure 8 is a flow chart of the method of the system.

Figure 9 is a flow chart of the method used by the tuple manager.

Figure 10 is a flow chart of the method used by the connection calculator.

Figure 11 is a flow chart of the first weeding phase of the method used by the connection calculator.

Figures 12a-d are flow charts of an infrastructure-building phase of the method used by the connection calculator.

Figure 13 is a flow chart of a second weeding phase of the method used by the connection calculator.

Figure 14 is a flow chart of the noise reduction phase of the method used by the connection calculator.

Figure 15 is a flow chart of the look-for phase of the method used by the connection calculator.

Figures 16a-b are flow charts of the consolidation phase of the method used by the connection calculator.

Figure 17 is a flow chart of the method used by the topology converter.

Figures 18a-b are flow charts of the morph topo phase of the method used by the topology converter.

Figure 19 is a flow chart of the duplication discard phase of the method used by the topology converter.

Figures 20a-d are flow charts of the identify different tuples phase of the method used by the topology converter.

Detailed Description

The system provides an improved method for creating topological maps of communication networks based. Connectivity information is retrieved from the network nodes and stored as "tuples" to track specifically the desired information necessary to map the topology. These light weight data structures may store the host identifier, interface index, and a port. From this tuple information, the topology may be determined. A tuple may be a binary element insofar as it has two parts representing the two nodes on either end of a network link or segment. A "tuco" refers to a tuple component, such as half of a binary tuple.

As used herein, a node is any electronic component, such as a connector or a host, or combination of electronic components with their interconnections. A connector is any network device other than a host, including a switching device. A switching device is one type of connector and refers to any device that controls the flow of messages on a network. Switching devices include, but are not limited to, any of the following devices: repeaters, hubs, routers, bridges, and switches.

As used herein, the term "tuple" refers to any collection of assorted data. Tuples may be used to track information about network topology by storing data from network nodes. In one use, tuples may include a host identifier, interface information, and a port specification for each node. The port specification (also described as the group/port) may include a group number and a port number, or just a port number, depending upon the manufacturer's specifications. A binary tuple may include this information about two nodes as a means of showing the connectivity between them, whether the nodes are connected directly or indirectly through other nodes. A "conn-to-conn" tuple refers to a tuple that has connectivity data about connector nodes. A "conn-to-host" tuple refers to a tuple that has connectivity data about a connector node and a host node. In one use, tuples may have data about more than two nodes; that is, they may be n-ary tuples, such as those used with respect to shared media connections described herein.

A "singly-heard host" (shh) refers to a host, such as a workstation, PC, terminal, printer, other device, etc., that is connected directly to a connector, such as a switching device. A singly-heard host link (shhl) refers to the link, also referred to as a segment, between a connector and an shh. A "multi-heard host" (mhh) refers to hosts that are heard by a connector on the same port that other hosts are heard. A multi-heard host link (mhhl) refers to the link between the connector and an mhh. A link generally refers to the connection between nodes. A segment is a link that may include a shared media connection.

Figure 1 is a drawing of a typical topological bus segment 100 for representing the connectivity of nodes on a network 110. In Figure 1, first and second hosts 121, 122, as well as a first port 131 of a first connector 140 are interconnected via the network 110. The bus segment

100 comprises the first and second hosts 121, 122 connected to the first port 131 of the first connector 140.

Figure 2 is a drawing of a typical topological serial segment 200 for representing the connectivity of nodes on the network 110. In Figure 2, the first host 121 comprises a second port 132 on a second connector 145 which is connected via the network 110 to the first port 131 on the first connector 140. The serial segment 200 comprises the second port 132 on the second connector 145 connected to the first port 131 on the first connector 140. Figure 2 is an example of a connector-to-connector ("conn-to-conn") relationship.

Figure 3 is a drawing of a typical topological star segment 301 for representing the connectivity of nodes on the network 110. In Figure 3, the first host 121 is connected to the first port 131 of the first connector 140. The star segment 301 comprises the first host 121 connected to the first port 131 of the first connector 140. Figure 3 is an example of a connector-to-host ("conn-to-host") relationship.

Figure 4 is a drawing of another typical topological star segment 301 for representing the connectivity of nodes on the network 110. In addition to the connections described with respect to Figure 3, a third host 123 is connected to a third port 133 of the first connector 140 and a fourth host 124 is connected to a fourth port 134 of the first connector 140. In Figure 4, the star segment 301 comprises the first host 121 connected to the first port 131 of the first connector 140, the third host 123 connected to the third port 133 of the first connector 140, and the fourth host 124 connected to the fourth port 134 of the first connector 140. Thus, the star segment 301 comprises, on a given connector, at least one port, wherein one and only one host is connected to that port, and that host. In the more general case, the star segment 301 comprises, on a given connector, all ports having one and only one host connected to each port, and those connected hosts. Since the segments, or links, drawn using the topological methods of Figure 4 resemble a star, they are referred to as star segments.

For illustrative purposes, nodes in the figures described above and in subsequent figures are shown as individual electronic devices or ports on connectors. Also, in the figures the nodes are

represented as terminals. However, they could also be workstations, personal computers, printers, scanners, or any other electronic device that can be connected to networks 110.

Figure 5 is a drawing of the connectivity of an example network system. In Figure 5, first, third, and fourth hosts 121, 123, 124 are connected via the network 110 to first, third, and fourth ports 131, 133, 134 respectively, wherein the first, third, and fourth ports 131, 133, 134 are located on the first connector 140.

The first, third and fourth hosts 121, 123, 124 are singly-heard hosts connected to separate ports 131, 133, 134 of a common connector 140 – the first connector 140. The fifth and sixth hosts 125, 126 are singly-heard hosts connected to the third and fourth connectors 142, 143. The seventh and eighth hosts 127, 128 are multi-heard hosts connected to the same port 139 of the fifth connector 144. The multi-heard hosts 127, 128 illustrate a shared media segment 180, also referred to as a bus 180.

The second, third, fourth, and fifth connectors 141, 142, 143, 144 are interconnected and illustrate a switch mesh 181. Each of the connectors in the switch mesh 181 is connected to each other, either directly or indirectly, to create a fully meshed connection. In the mesh, traffic may be dynamically routed to create an efficient flow.

Figure 5 also shows an example of a port aggregation 182, also referred to as trunking 182. The first connector 140 is connected via the network 110 to the second connector 141 by two direct links, each of which is connected to different ports on the connectors. One link is connected to the sixth port 136 of the first connector 140 and to the seventh port of the second connector 137. The other link is connected to fifth port 135 of the first connector 140 and to the eighth port 138 of the second connector 141. In this example, two connectors illustrate the multiple connectivity between nodes. Depending upon the device specifications, devices such as connectors may be connected via any number of connectors. As explained herein, the system resolves multiple connectivity problems by tracking port information for each connection.

Figure 6 is a drawing of the connectivity of a portion of a network having three connectors 171, 172, 173. A first host 151 is connected directly to the first port 161 of the first connector 171 and the second host 152 is connected to a sixth port 166 of the third connector 173. The second

port 162 of the first connector 171 is connected directly to the third port 163 of the second, or intermediate, connector 172. The fourth port 164 of the intermediate connector 172 is connected directly to the fifth port 165 of the third connector 173.

Figure 7 shows a block diagram of the system. Figure 8 shows a flow chart of the method used by the system to retrieve and update the topology of the network. A tuple manager 300, also referred to as a data miner 300, gathers 902 data from network nodes and builds 904 tuples to update the current topology. The topology database "topodb" 350 stores the current topology for use by the system. The "neighbor data" database 310 stores new tuple data retrieved by the tuple manager 300. The connection calculator 320 processes the data in the neighbor data database 310 to determine the new network topology. The connection calculator 320 reduces 906 the tuple data and sends it to the reduced topology relationships database 330. The topology converter 340 then updates 908 the topology database 350 based on the new tuples sent to the reduced topology relationships database 330 by the connection calculator 320.

Figure 9 shows a flow chart of one operation of the tuple manager 300, as described generally by the data gathering 902 and tuple building 904 steps of the method shown in Figure 8. The tuple manager 300 receives 910 a signal to gather tuple data. The tuple manager 300 then retrieves 912 node information of the current topology stored in the topology database 350. This information tells the tuple manager 300 which devices or nodes are believed to exist in the system based on the nodes that were detected during a previous query. The tuple manager 300 then queries 914 the known nodes to gather the desired information. For example, the connectors may maintain forwarding tables that store connectivity data used to perform the connectors' ordinary functions, such as switching. Other devices may allow the system to perform queries to gather information about the flow of network traffic. This data identifies the devices heard by a connector and the port on which the device was heard. The tuple manager 300 gathers this data by accessing forwarding tables and other information sources for the nodes to determine such information as their physical address, interface information, and the port from which they "hear" other devices. Based on this information, the tuple manager 300 builds 916 tuples and stores 918 them in the "neighbor data" database 310. Some nodes may have incomplete information. In this case, the partial

information is assembled into a tuple and may be used as a "hint" to determine its connectivity later, based on other connections. The tuple manager 300 may also gather 920 additional information about the network or about particular nodes as needed. For example, the connection calculator 320 may require additional node information and may signal the tuple manager 300 to gather that information.

After the data is gathered and the tuples are stored in the neighbor database 310, the connection calculator 320 processes the tuples to reduce them to relationships in the topology. Figure 10 shows a flow chart of the process of the connection calculator 320, as shown generally in the reduction step 906 of the method shown in Figure 8. The connection calculator 320 performs a first weeding phase 922 to identify singly-heard hosts to distinguish them from multi-heard hosts. Singly-heard hosts refer to host devices connected directly to a connector. The connection calculator 320 then performs an infrastructure-building phase 924 to remove redundant connector-to-connector links and to complete the details for partial tuples that are missing information. Then, the connection calculator 320 performs a second weeding phase 926 to resolve conflicting reports of singly-heard hosts. The connection calculator 320 then performs a noise reduction phase 928 to remove redundant neighbor information for connector-to-host links. If clarification of device connectivity is required, the connection calculator 320 performs a "look for" phase 930 to ask the tuple manager 300 to gather additional data. The tuple data is then consolidated 932 into segment and network containment relationships. The connection calculator 320 may also tag redundant tuples to indicate their relevance to actual connectivity. These redundant tuples may still provide hints to connectivity of other tuples. As part of the consolidation phase 932, the connection calculator 320 creates new n-ary tuples (tuples having references to three or more tucos) for shared media segments.

Figure 11 is a flow chart of the connection calculator's first weeding process 922 for distinguishing singly-heard hosts. The purpose of the first weeding process 922 is to identify the direct connections between connectors and hosts; that is, those tuples having a first tuco that is a connector and a second tuco that is a host. The connection calculator 320 looks through the tuple list in the neighbor database 310, and for each tuple 402, the connection calculator 320 determines

404 whether the tuple is a connector-to-host (conn-to-host) link tuple. If it is not a conn-to-host link, the connection calculator 320 concludes 418 that it is a conn-to-conn link and processes 402 the next tuple. If the tuple is a conn-to-host link tuple, then the connection calculator 320 determines 406 whether the connector hears only this particular host on the port identified in the tuple. If the connector hears other hosts on this port, then the tuple is classified 416 as a multi-heard host link (mhhl) tuple.

If the connector hears only the one host on the port – that is, if the host is a singly-heard host – then the connection calculator 320 determines 408 whether the host is heard singly by any other connectors. If no other connectors hear the host as a singly-heard host, then the tuple is classified as a singly-heard host link (shhl) tuple 412 and other tuples for this host are classified 414 as extra host links (ehl). Another tuple for this host may be, for example, an intermediate connector connected indirectly to a host. For example, Figure 6 shows three connectors 171, 172, 173 the first connector is connected directly to the first host 151. This connection therefore forms an shhl tuple. The intermediate connector 172 is indirectly connected to the first host 151. The tuple data indicates that the intermediate connector 172 is indirectly connected to the host and hears the host from a particular port. An extra host links tuple is created so that this data may be used later in conjunction with other extra host links tuples from devices across the network, to verify connectivity by providing hints about connections.

The first weeding process also attempts to identify conflicts. If other connectors hear the host as a singly-heard host, then a conflict arises and the tuple is classified 410 as a singly-heard conflict link (shcl) tuple to be resolved later. This conflict may arise, for example, if a host has been moved within the network, in which case the forwarding table data may no longer be valid. Certain connectors previously connected directly to the host may still indicate that the moved host is connected. When all tuples have been processed 402 to identify singly-heard host links, the first weeding phase 922 is complete.

Figures 12a-d show a flow chart of the infrastructure building phase 924 of the connection calculator 320. The purpose of the infrastructure building phase 924 is to determine how the connectors are set up in the network. The first part of the infrastructure building phase 924

manufactures tuples based on the list of singly-heard host link tuples identified in the first weeding phase 922. The purpose is to identify the relationship between the connectors in the extra host links tuples and the connectors directly connected to the singly-heard hosts. For each singly-heard host link 420, the connection calculator 320 processes 422 each extra host link that refers to the host. In the illustration of Figure 6, a conn-to-conn link tuple would represent the connection between the first connector 171 and the intermediate connector 172. An extra host link tuple would represent the indirect connection between the intermediate connector 172 and the first host 151. The conn-to-conn link tuple between the first connector 171 and the intermediate connector 172 is an example of an ehlConn-to-shhlConn tuple. If a conn-to-conn link tuple exists 424 for the extra host link connector to the singly-heard host link connector (ehlConn-to-shhlConn), then the connection calculator 320 updates 428 the tuple if it is incomplete. It is possible that the tuple data may be incomplete and a conn-to-conn link may not exist. In that case, a conn-to-conn tuple does not exist for the ehlConn-to-shhlConn, then such a tuple is created 426.

After processing extra host links for singly-heard host links, the connection calculator 320 considers 430 each connector (referred to as conn1) in the tuples to determine the relationship between connectors. As illustrated in Figure 6, a single connector may be connected directly and indirectly to multiple other connectors. In Figure 6, the first connector 151 is connected to the intermediate connector 171 directly and also to the third connector 173 indirectly. The third connector 173 hears the first host 151 on the same part 165 that it hears the first connector 171 and the intermediate connector 172. The infrastructure building phase 924 tries to determine the relationship between other connectors heard on the same port of conn1. In a series of interconnected connectors, the connector on one end may not hear a connector on another end, but it may hear intermediate connectors, that in turn hear their own intermediate connectors. Tuples are created to represent the interconnection of conn-to-conn relationships. Based on this data, the connection calculator 320 can make inferences regarding the overall connection between connectors.

For every conn1, the connection calculator 320 considers 432 every other connector (conn2) to determine whether a conn1-to-conn2 tuple exists. If conn1-to-conn2 does not exist,

then the connection calculator 320 considers 436 every other conn-to-conn tuple containing conn2. The other connector on this tuple may be referred to as conn3. If conn2 hears conn3 on a unique port 438 and if conn1 also hears conn3 440, then the connection calculator 320 creates 442 a tuple for conn1-to-conn2 in the connector-to-connector links tuple list.

After processing all of the conn1 tuples, the connection calculator 320 processes 444 each conn1-to-conn2 links tuple to ensure that they have complete port data. For each incomplete tuple 446, the connection calculator 320 looks 448 for a different tuple involving conn1 in the extra host links tupleson a different port. If a different tuple is found 450, then the connection calculator 320 determines 452 whether conn2 also hears the host. If conn2 does hear the host, then the connection calculator 320 completes the missing port data for conn2. If conn2 does not also hear the host 452, then the connection calculator 320 continues looking 448 through different tuples involving conn1 in extra host links on different ports.

After attempting to complete the missing data in each of the conn-to-conn links tuples, the connection calculator 320 processes 456 each conn-to-conn links tuple. The purpose of this sub-phase is to attempt to disprove invalid conn-to-conn links. The connection calculator 320 considers 458 conn1 and conn2 of each conn-to-conn links tuple. Every other connector in conn-to-conn links may be referred to as testconn. For each testconn 460, the connection calculator 320 determines 462 whether the testconn hears conn1 and conn2 on different groups/ports. If testconn hears conn1 and conn2 on different ports, then the tuple is moved to extraconnlinks (ecl) 464. Otherwise, the connection calculator 320 continues processing 460 the remaining testconns.

Figure 13 shows a flow chart of the second weeding phase 926. The purpose of the second weeding phase 926 is to attempt to resolve conflicts involving singly-heard hosts identified in the first weeding phase 922. In the situation described herein in which more than one connector reports that a host is singly-heard, the second weeding phase 926 reviews the tuples created during the infrastructure-building phase 924 involving the connector and host in question and attempts to disprove the reported conflict. The connection calculator 320 processes 466 each singleConflictLinks (scl) tuple (sometimes referred to as the search tuple) and considers 468 conn1 and host1 of the tuple. For each extra host links tuple containing host1 470, the connection

calculator 320 considers 472 conn2 of the tuple. If there is a tuple in conn-to-conn links for conn2 and conn1 474, and if there is a conn2-to-conn1 tuple in the extra host links tuples 476, and if the port is the same for conn2 hearing conn1 and host1 478, then the search tuple is moved 480 into the singly heard host links and other tuples containing host1 are removed 482 from the singleConflictLinks.

Figure 14 shows a flow chart of the noise reduction phase 928. The purpose of the noise reduction phase 928 is to handle those connections in which a connector is not directly connected to a host or to another connector. For example, networking technology may employ shared media connections between connectors, rather than dedicated media connectors. With a shared media connection, the entries in the forwarding tables for connectors attached to the shared media connection will include every node accessing the shared media connection and may not present a useful or accurate representation of the nodal connection. For example, if the network configuration in Figure 6 used a shared media connection between the first connector 171 and the intermediate connector 172, then the first connector is not really connected directly to the intermediate connector because other devices (not shown in Figure 6) may also use the shared media connection. These other devices may include web servers, other connectors, other subnetworks, etc. Tuples will be created for the connectors 171, 172 on opposing ends of the shared media. In this situation, it is inefficient to maintain point-to-point binary tuples for every connection. The noise reduction phase 928 disproves invalid tuples created by the shared media connections.

For each multi-heard host links (mhhl) tuple, also referred to as multiHeardLinks (mhl) tuples (sometimes referred to as the search tuple) 484, conn1 and host1 are considered 486. For each extra host links tuple containing host1 488, conn2 is considered 490. If there is a tuple in conn-to-conn links for conn2 and conn1 492, and if there is a conn2-to-host1 tuple in extraHostLinks 494, and if the group/port for conn2 hearing conn1 and host1 is different 496, then the search tuple is moved 498 to extraHostLinks.

Figure 15 shows a flow chart for the "look for" phase 930. The purpose of this phase is to complete missing data for mhhl tuples. There may exist connections on the network that have incomplete tuple data. For example, the network may simply have no traffic between certain nodes,

in which case data might not be stored in forwarding tables. In another example, a forwarding table may not have sufficient room to store all of the required information and might delete data on a FIFO basis. In the look for phase 930, the connection calculator 320 instructs the tuple manager 300 to query specific nodes to retrieve the missing data. Data that was not stored in a forwarding table on the first interrogation may be present on a subsequent query. For each mhhl tuple 500, the connection calculator 320 considers 502 conn1 and host1. If the conn1 group/port is already in an "alreadyDidLookfors" list, then a list is created 508 for all connectors in conn-to-conn links that are heard by conn1 on the same group/port as host1. For each connector (conn2) in the list 510, the connection calculator 320 determines 512 whether there is a conn2-to-host1 tuple in the mhhl tuples. If there is not such a tuple, then the connection calculator 320 initiates a look-for for conn2-to-host1 via the tuple manager 300. When each connector in the list has been processed 510, the conn1 group/port tuco is added 516 to an alreadyDidLookfors list. As an additional portion of the look for phase 930 (not shown in figures) the system may ask a user to verify or clarify information about connectivity. For example, the system may show the user the perceived connectivity or the unresolved connectivity issues and request the user to add information as appropriate.

The connection calculator 330 process described above collects the tuple information from the tuple manager 300, builds tuples new tuples and removes redundant or unnecessary tuples to produce the new topology. This topology may have incomplete tuples possibly resulting from extraneous information that the connection calculator 330 could not disprove. To refine the new topology, the connection calculator 330 can request the tuple manager 300 to obtain additional information about particular nodes or it may also request a user to refine the topology by adding or removing tuples. Using the process of the connection calculator 330, tuples marked as non-essential may be removed from the new topology to save space and to simply the topology. The connection calculator 330 is not confused by multiple connectivity situations such as port aggregation 182 or switch meshing 181 as shown in Figure 5, because the tuples represent point-to-point, or neighbor-to-neighbor, connectivity showing each connection in the network. This point-to-point connectivity concept also helps enable the system to avoid difficulties that occur in

systems that track higher levels of abstraction, such as layer 3 connectivity. Also, the tuples may contain only selected information to minimize the storage space required for the topology.

Figures 16a-b show a flow chart of the consolidation phase 932. The purpose of this phase is to consolidate the tuples that involve shared media connections. After the noise reduction phase 928, a considerable number of tuples involving shared media may remain. Rather than maintain a binary tuple for each of the connections, an n-ary tuple is created for the link using a tuco for each connector and each host connected thereto. For each mhhl tuple 518, conn1 and host1 are considered 520. If there are more conn1 group/port tuples in multiHeardLinks, and if are not any n-ary multiHeardSegments (mhs) tuples 524, then an mhs tuple is created 526. If host1 is not already in this particular mhs tuple 528, then conn2 of the tuple is considered 534. If there is a conn1-to-conn2 conn-to-connLinks tuple on the same port as conn1-to-host1 536, then all multiHeardLinks tuples for conn2-to-host1 with the same conn2 group/port as the conn1-to-conn2 are added 538 to the current mhs tuple.

After processing each mhhl tuple 518, each singly-heard host links (shhl) tuple, also referred to as a singlyHeardLinks (shl) tuple, is considered 540. For each shhl tuple, the connector and host are considered 542. If there is no existing singlyHeardSegments (shs) tuple for the connector 544, then an shs tuple is created 546. The host tuco is then added to the shs 548.

Figure 17 shows a flow chart of the method used by the topology converter 340, as described generally by the topology update step 908 of the method shown in Figure 8. The topology converter 340 converts 934 the topology into tuple lists, also referred to as the "morph topo" phase 934. It then compares 936 the list from the topology currently stored in the topology database 350 with the new list generated by the connection calculator 320 and discards 936 identical tuples in what is also referred to as the "discard duplicates" phase 936. It then takes action 938 on the changes in the topology as determined by the changes in the tuple lists, in what is also referred to as the "identify different tuples" phase 938.

Figure 18a shows a flow chart for the "morph topo" phase 934. For each node in the topology 550, the topology converter 340 determines 552 whether the node is a connector. If the node is a connector, then for each connected interface (conniface) of the connector (conn1) 554,

the topology converter 340 determines 556 whether the conniface is connected to a star segment. If it is connected to a star segment, then for every other interface in the segment 558, the topology converter 340 determines 560 whether there is an existing shs tuple, referred to as the "topo tuple" for the segment. If there is no such tuple, then the topology converter 340 creates 562 a topo shs tuple. The tuco for the interface's host-to-topo shs is then added 564 to the topo shs tuple.

If the connector node is not connected to a star segment 556 and is connected to a bus segment 566, the topology converter 340 determines 568 whether there is an existing mhs tuple for conn1. If there is not an existing mhs tuple for conn1, then a topo mhs tuple is created 570. A tuco is added 572 for the host to the mhs tuple.

If the connector node is not connected to either a star segment 556 or to a bus segment 566, then the topology converter knows that it is connected to another connector (conn2). If such a connector does not already have an existing connLinks tuple for conn1 and conn2 576, then a connLinks tuple is created 578. After processing the bus segment, star segment, and conn-to-conn segment, for each conniface 554, the topology converter 340 proceeds to the next node 550.

Figure 18b shows a continuation of the flow chart of Figure 18a showing the steps of the method when the topology converter 340 determines that the node is not a connector 552. If the node is in the default segment, then an "unheardOfLinks" tuple is created 582 and the topology converter proceeds to the next node 550. If the node is not in the default segment 580, then the topology converter 340 determines whether the node is in a star segment 584. If the node is in a star segment, then if there is not already an shs tuple, the topology converter 340 creates 588 an shs tuple. The tuco for the node is then added 590 to the shs tuple, and the topology converter 340 proceeds to the next node 550.

If the node is not in a star segment, then the topology converter 340 knows that it is in the bus segment. If there is not already an mhs tuple for the node, 594, then the topology converter 340 creates 596 an mhs tuple. The tuco for the node is then added 598 to the mhs tuple, and the topology converter proceeds to the next node 550.

Figure 19 shows a flow chart for the discard duplicates phase 936 of the topology converter 340. For each tuple in the new tuples (nt) 600, the topology converter looks for 602 an

exact match in the current tuples stored in the topodb. If an exact match is found 604, then the new tuple is marked 606 as "no change" indicating that this is an identical tuple.

Figures 20a-d show a flow chart for the identify different tuples phase 938. The system looks through each tuple in the new SinglyHeardSegments (newSHS) tuple list 608 and tries to identify and fix 610 swapped ports on connectors. Swapped ports are identified by considering those segment tuples in both the new topology and the existing topology that differ only by the port specification in the tuco. Each tuple that is fixed as a swapped port is marked 612 as "handled." The system also looks through each tuple in the new multiHeardSegments tuple list (newMHS) 614 and tries to identify and fix 616 swapped ports on connectors. Each tuple that is fixed as a swapped port is marked 618 as "handled."

The system then processes 620 each unmarked tuple in the newSHL tuples. Four cases are possible for the host of the newSHL tuples. The host of the newSHL can be found in the current singlyHeardLinks (curSHL) 622, the current multiHeardLinks (curMHL) 630, the current connLinks (curCL) 638, or the current UnheardOfLinks (curUOL) 642. If the host of a newSHL tuple is found 622 in the current SinglyHeardLinks (curSHL) tuples, then the system determines 624 if there is a matching connector tuco between the newSHL tuples and the curSHL tuples. If there is a matching tuco, then the system changes 626 the host connection attribute. If there is not a matching tuco, then the host connection is moved 628 in the topology.

If the host is found in the curMHL tuples 630, then the system determines 632 whether there is a matching connector tuco between the newSHL tuples and the curSHL tuples. If there is a matching connector, then the segment type of connection is changed 634. If there is not a matching connector, then the host connection is moved 636 in the topology. If the host is found in the curCL tuples 638, then the host is moved 640 into a star segment of the connector. If it is found in the curUOL 642, then the host is moved 644 into the star segment of the connector.

Figure 20c shows another stage of the processing undertaken during the identify different tuples phase 938. For each unmarked tuple in the new multiHeardLinks tuples (newMHL) 946, four cases are possible for the host of the newMHL. The host of the newMHL may be found in the curSHL 648, the curMHL 656, the curCL 664, or the curUOL 668. If the host is found in the

curSHL 648, then the system determines 650 whether there is a matching connector tuco between the newMHL and the curMHL. If there is a matching tuco, then the segment type of connection is changed 652. If there is not a matching tuco, then the host connection is moved 654 in the topology.

If the host is found in the curMHL tuples 656, then the system determines 658 whether there is a matching connector tuco in both the curMHL tuples and the newMHL tuples. If there is a matching connector tuco, then the host connection attribute is changed 660. If there is not a matching tuco, then the host connection is moved 662 in the topology. If the host is found in the curCL tuples 664, then the host is moved into a bus segment of a connector. If the host is found in the curUOL tuples 668, then the host connection is moved 670 in the topology.

Figure 20d shows another portion of the identify different tuples phase 938. For each unmarked tuple in the newCL tuples 672, there are three possibilities for the connector. The connector of the unmarked tuple in newCL can be found in the curSHL or curMHL 674, in the curCL 678, or in the curUOL 682. If each connector is found in the curSHL or curMHL list 674, then the system creates 676 a new point-to-point segment for the connectors. If the connectors are found in the curCL 678, then the connection attributes of the connectors are changed 680. If each connector is found in the curUOL tuples 682, then the host connection is moved 684 in the topology.

Another part of the identify different tuples phase 938 is shown in blocks 686 and 688 of Figure 20d. For each unmarked tuple in the newUOL tuples 686, the system checks 688 the timer/configuration to determine whether the host/conn should move into the default segment from its current segment.

An advantage of the system is that it may be schedulable. The system may map network topology continuously, as done by existing systems, or it may be scheduled to run only at certain intervals, as desired by the user. A further advantage of the system is that it is capable of processing multiple connections between the same devices and of processing connection meshes, because it tracks each nodal connection independently, without limitations on the types of connections that are permitted to exist.

Although the present invention has been described with respect to particular embodiments thereof, variations are possible. The present invention may be embodied in specific forms without departing from the essential spirit or attributes thereof. It is desired that the embodiments described herein be considered in all respects illustrative and not restrictive and that reference be made to the appended claims for determining the scope of the invention.

Claims

1. In a network having interconnected nodes with data tuples that represent nodal connections, a method for mapping a network topology by identifying changes between an existing topology and a new topology, the method comprising:

converting an existing topology into a list of existing tuples that represent existing nodal connections;

receiving new tuples that represent new nodal connections; and

comparing the list of existing tuples with the new tuples to identify changes to the topology.

2. The method of claim 1, further comprising updating a topology database with a new topology.

3. The method of claim 1, further comprising taking action on the changes to the topology.

4. The method of claim 1, wherein the tuples include information about a host identifier, a connector interface, and a port specification.

5. The method of claim 1, wherein the step of comparing comprises identifying duplicate tuples that appear both in the list of existing tuples and in the new tuples, and maintaining a current status of the topology for these tuples.

6. The method of claim 1, wherein the step of comparing comprises identifying a swapped port condition on a connector.

7. The method of claim 1, wherein the step of comparing comprises searching for a host of a new singly-heard host link tuple or a new multi-heard host link tuple in the list of existing tuples.

8. A system for mapping a network topology by identifying changes between an existing topology and a new topology, based on changes to data tuples that represent nodal connections comprising:

a topology database that stores an existing topology of a network; and

a topology converter connected to the topology database that receives new tuples that represent new nodal connections; and compares the new tuples with the existing topology to identify changes in the network.

9. The system of claim 8, wherein the topology converter converts the existing topology into a list of existing tuples that represent existing nodal connections.

10. The system of claim 8, wherein the topology converter updates the topology database with a new topology based on the new tuples.

11. The system of claim 8, wherein the topology converter attempts to identify swapped ports on connectors.

12. The system of claim 8, wherein the topology converter identifies duplicate tuples that appear both in the list of existing tuples and in the new tuples, and maintains a current status of the topology for these tuples.

13. The system of claim 8, wherein the topology converter searches for a host of a new singly-heard host link tuple or a new multi-heard host link tuple in the list of existing tuples.

14. The system of claim 8, wherein the topology converter searches for a connector of a new conflict links tuple in the list of existing tuples.

15. A computer-readable medium having computer-executable instructions for performing a method for mapping a network topology by identifying changes between an existing topology and a new topology in a network having a interconnected nodes, the method comprising:

converting an existing topology into a list of existing tuples that represent existing nodal connections;

receiving new tuples that represent new nodal connections;

comparing the list of existing tuples with the new tuples to identify changes to the topology; and

updating a topology database with a new topology.

16. The method of claim 15, wherein a topology converter receives the new tuples from a connection calculator that calculates connections between nodes.

17. The method of claim 15, wherein the step of comparing comprises identifying duplicate tuples that appear both in the list of existing tuples and in the new tuples, and maintaining a current status of the topology for these tuples.

18. The method of claim 15, wherein the step of comparing comprises identifying a swapped port condition on a connector.

19. The method of claim 15, wherein the step of comparing comprises searching for a host of a new singly-heard host link tuple or a new multi-heard host link tuple in the list of existing tuples.

20. The method of claim 15, wherein the step of comparing comprises searching for a connector of a new conflict links tuple in the list of existing tuples.

Abstract

A method and system are disclosed for mapping the topology of a network having interconnected nodes by identifying changes in the network and updating a stored network topology based on the changes. The nodal connections are represented by data tuples that store information such as a host identifier, a connector interface, and a port specification for each connection. A topology database stores an existing topology of a network. A topology converter accesses the topology database and converts the existing topology into a list of current tuples. A connection calculator calculates tuples to represent connections in the new topology. The topology converter receives the new tuples, identifies changes to the topology, and updates the topology database using the new tuples. The topology converter identifies duplicate tuples that appear in both the new tuples and the existing tuples and marks the duplicate tuples to reflect that no change has occurred to these connections. The topology converter attempts to resolve swapped port conditions and searches for new singly-heard and multi-heard host link tuples in the list of existing tuples. The topology converter also searches for new conflict link tuples in the existing tuples. The topology converter updates the topology database with the new topology.

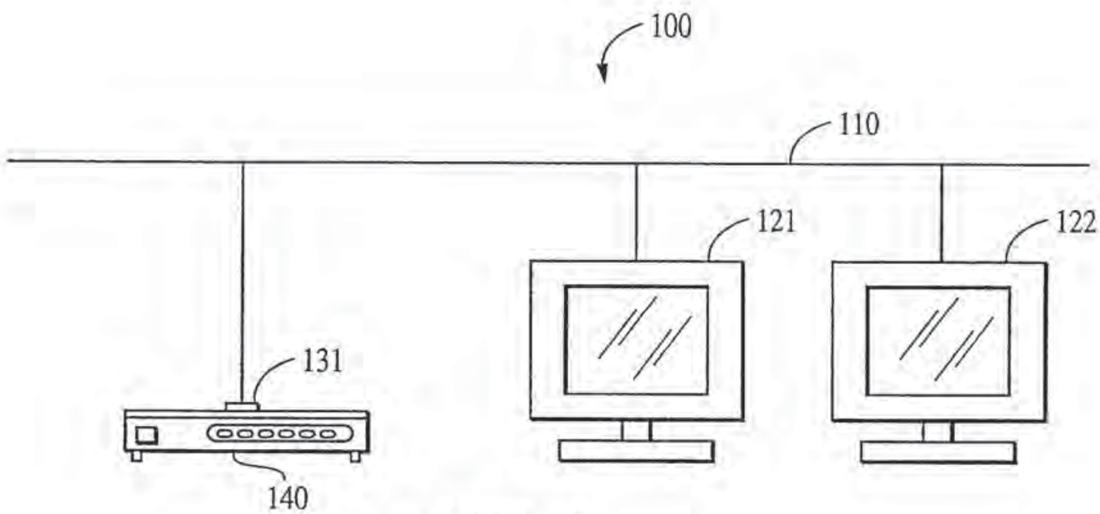


FIG. 1

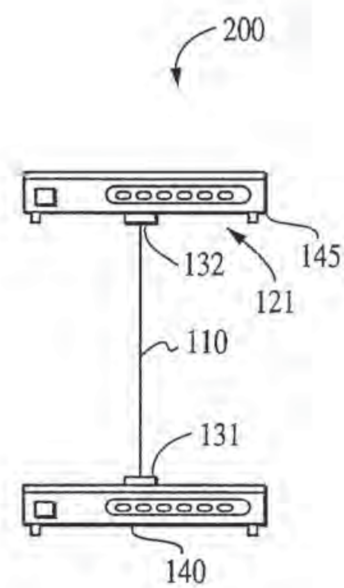


FIG. 2

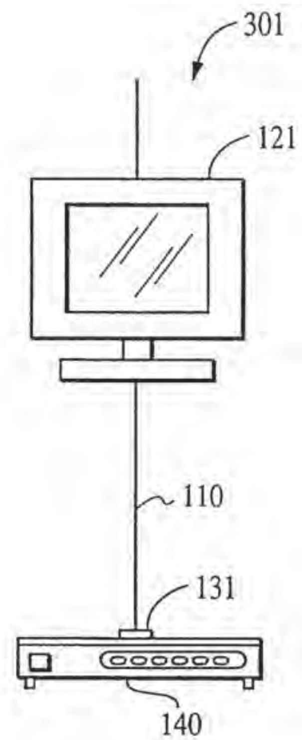


FIG. 3

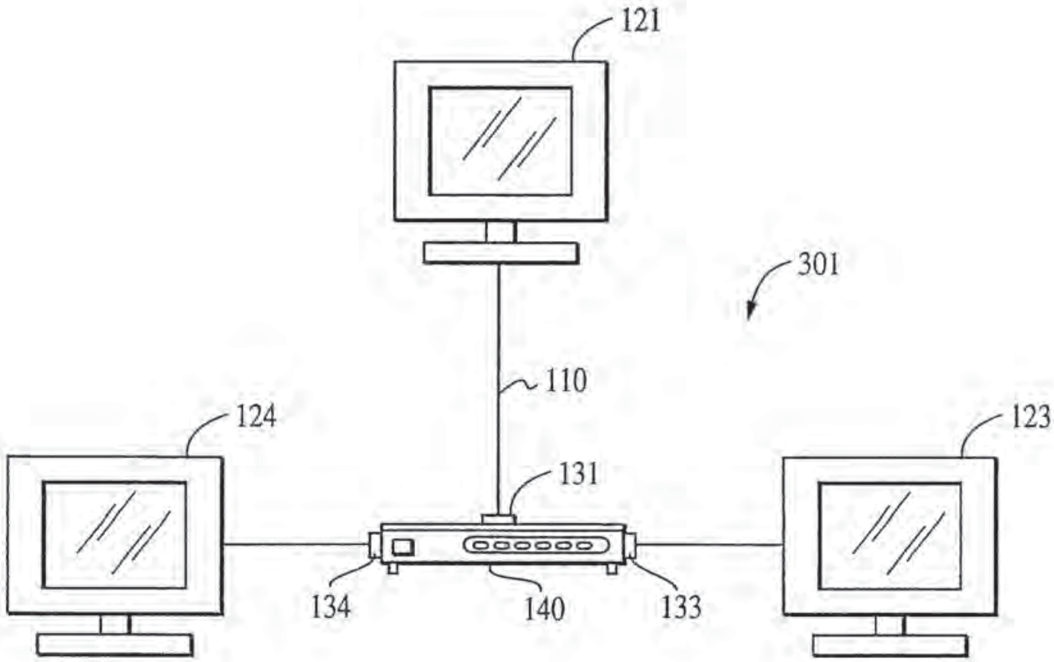


FIG. 4

FIG. 5

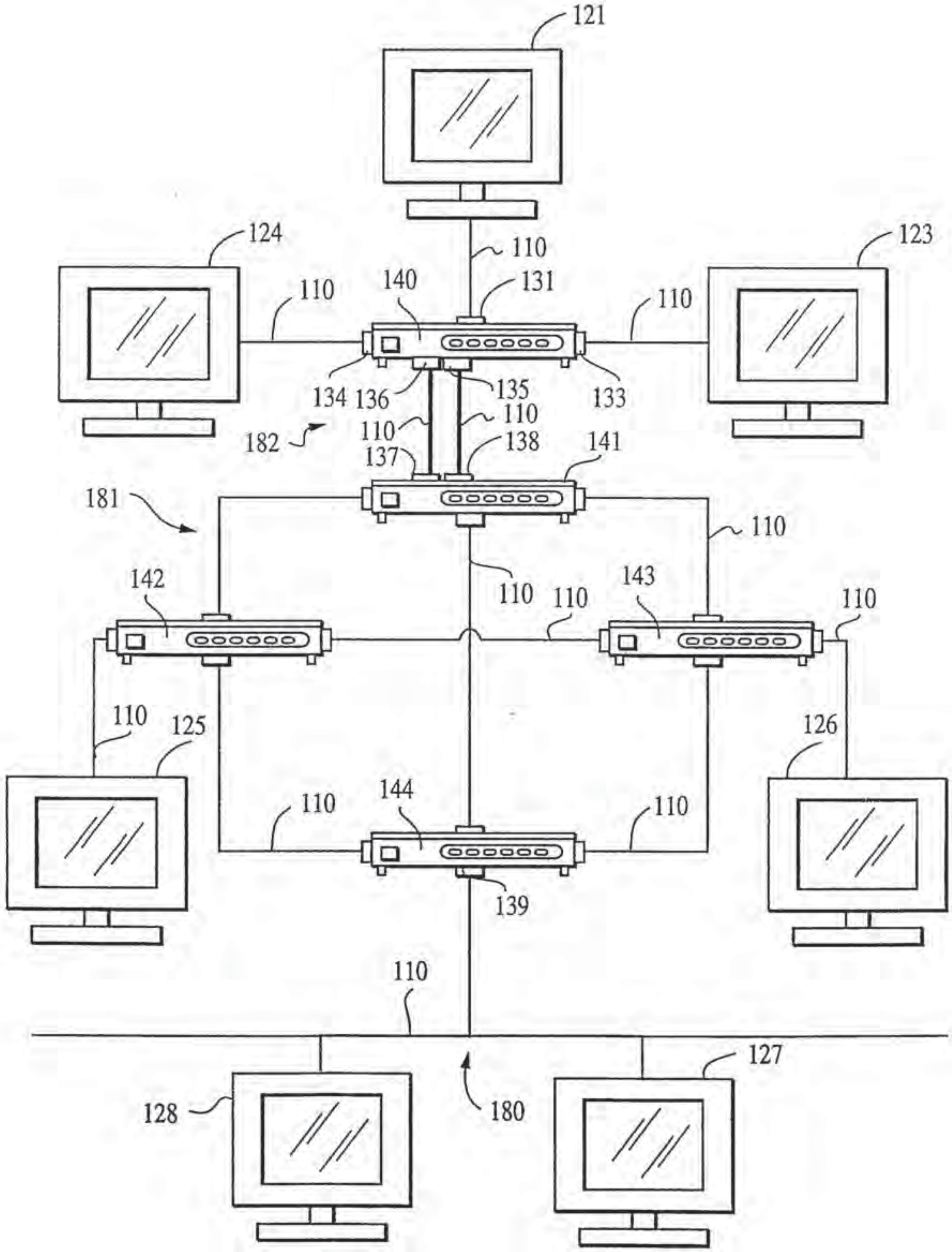
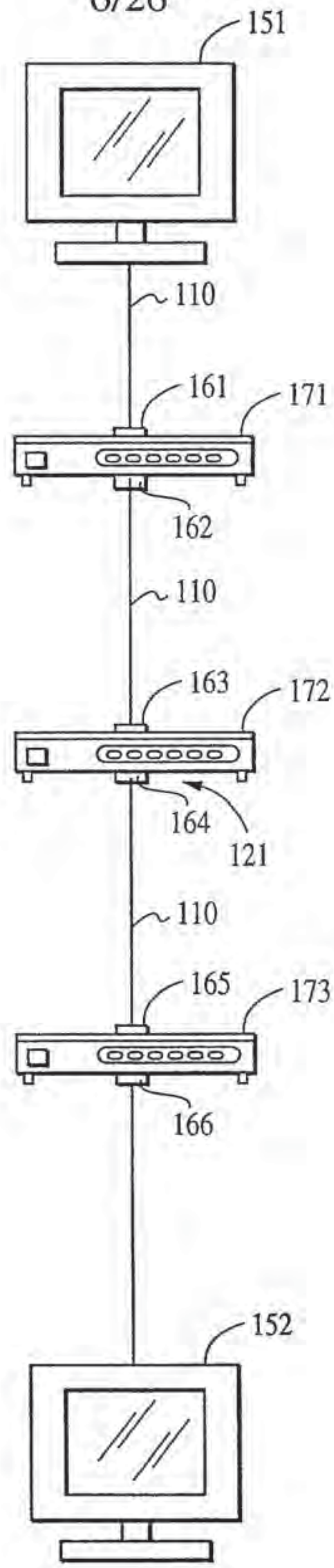

121
124
123
110
140
131
134
136
135
133
182
137
138
141
181
142
143
125
126
144
139
180
128
127

FIG. 6



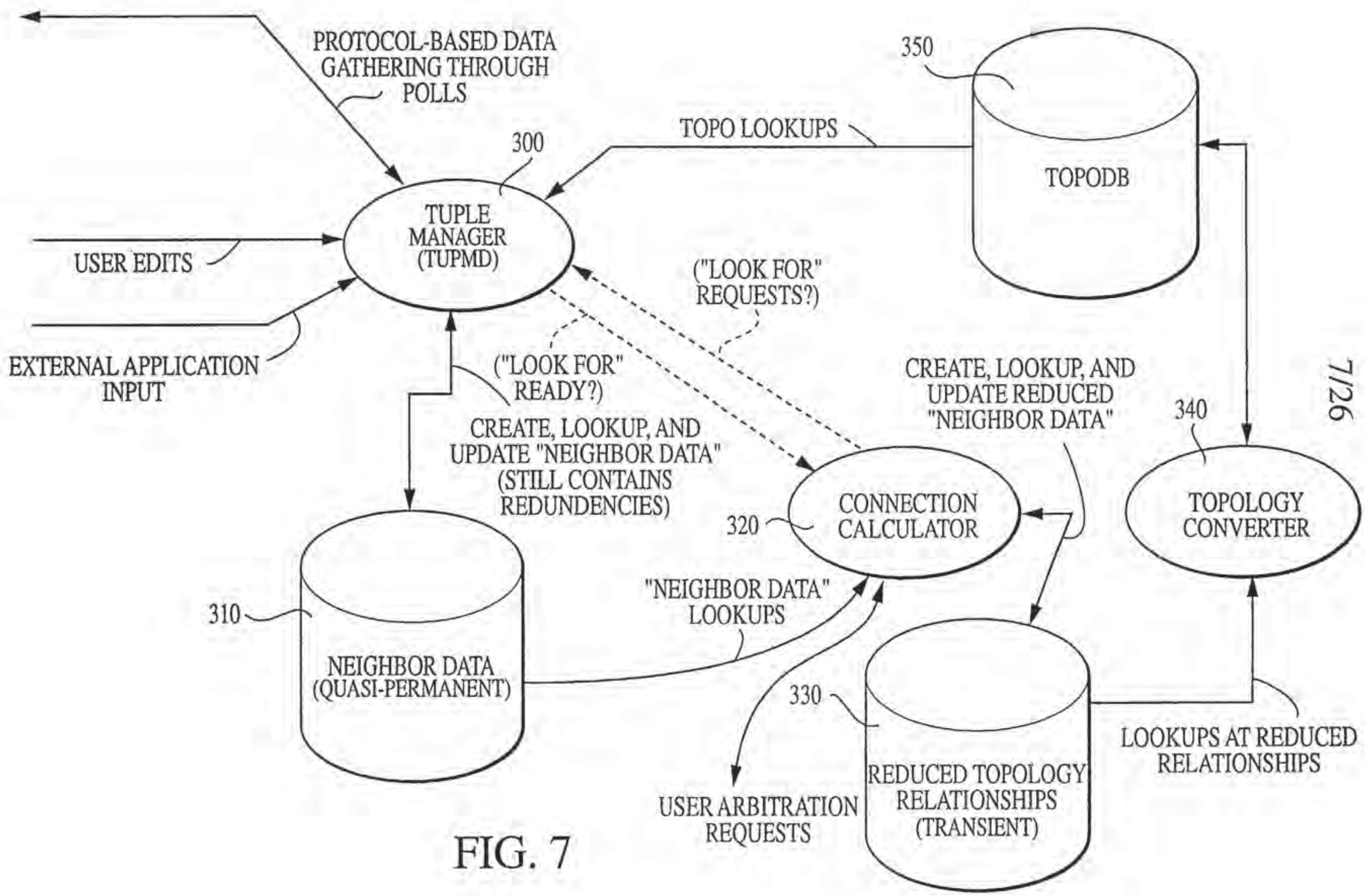

PROTOCOL-BASED DATA GATHERING THROUGH POLLS
USER EDITS
EXTERNAL APPLICATION INPUT
300
TUPLE MANAGER (TUPMD)
TOPO LOOKUPS
350
TOPODB
("LOOK FOR" REQUESTS?)
("LOOK FOR" READY?)
CREATE, LOOKUP, AND UPDATE "NEIGHBOR DATA" (STILL CONTAINS REDUNDENCIES)
CREATE, LOOKUP, AND UPDATE REDUCED "NEIGHBOR DATA"
340
TOPOLOGY CONVERTER
320
CONNECTION CALCULATOR
310
NEIGHBOR DATA (QUASI-PERMANENT)
"NEIGHBOR DATA" LOOKUPS
330
REDUCED TOPOLOGY RELATIONSHIPS (TRANSIENT)
LOOKUPS AT REDUCED RELATIONSHIPS
USER ARBITRATION REQUESTS

FIG. 7

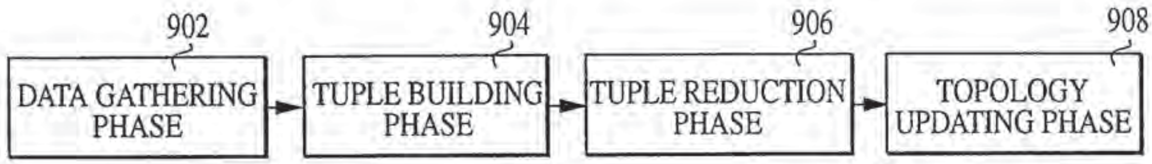


FIG. 8

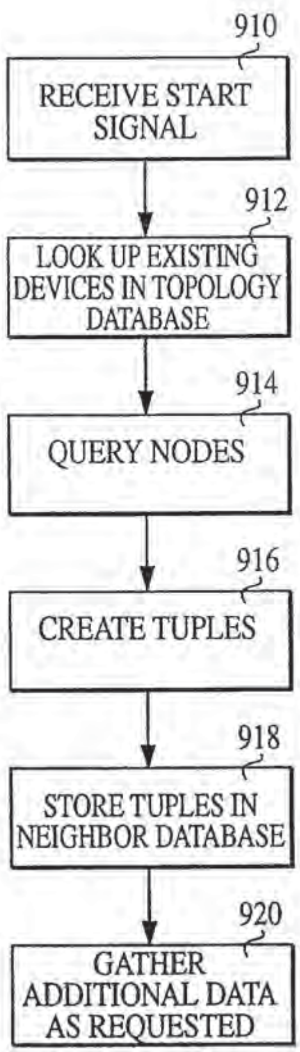


FIG. 9

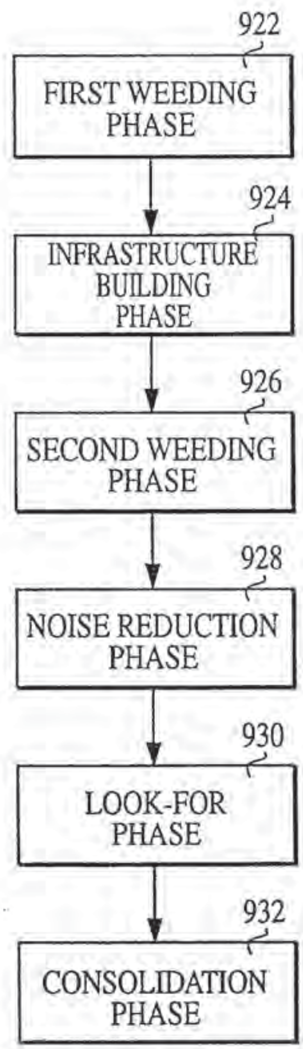
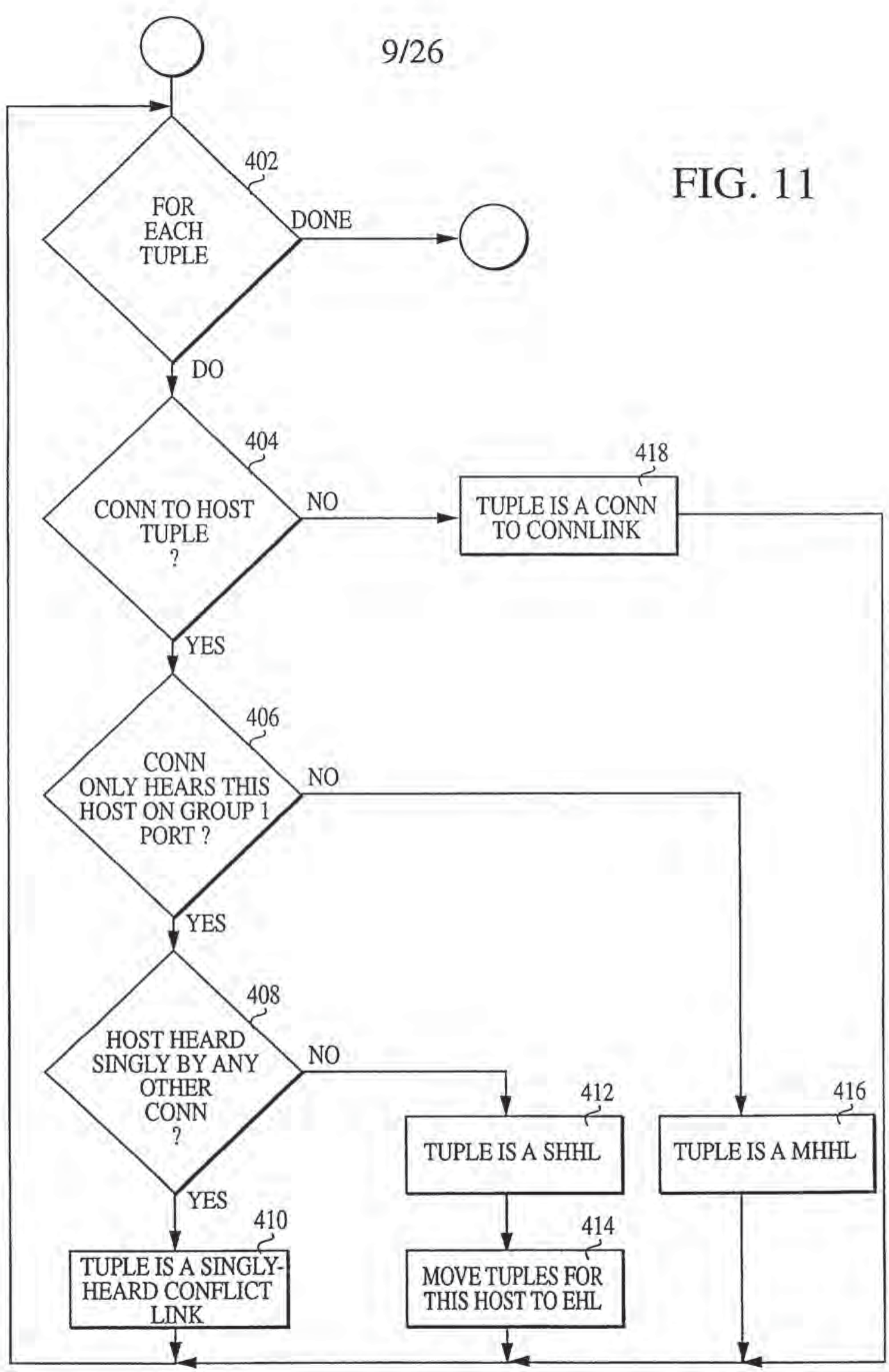


FIG. 10

FIG. 11



402
FOR EACH TUPLE
DONE
DO
404
CONN TO HOST TUPLE ?
NO
418
TUPLE IS A CONN TO CONNLINK
YES
406
CONN ONLY HEARS THIS HOST ON GROUP 1 PORT ?
NO
YES
408
HOST HEARD SINGLY BY ANY OTHER CONN ?
NO
412
TUPLE IS A SHHL
416
TUPLE IS A MHHL
YES
410
TUPLE IS A SINGLY-HEARD CONFLICT LINK
414
MOVE TUPLES FOR THIS HOST TO EHL

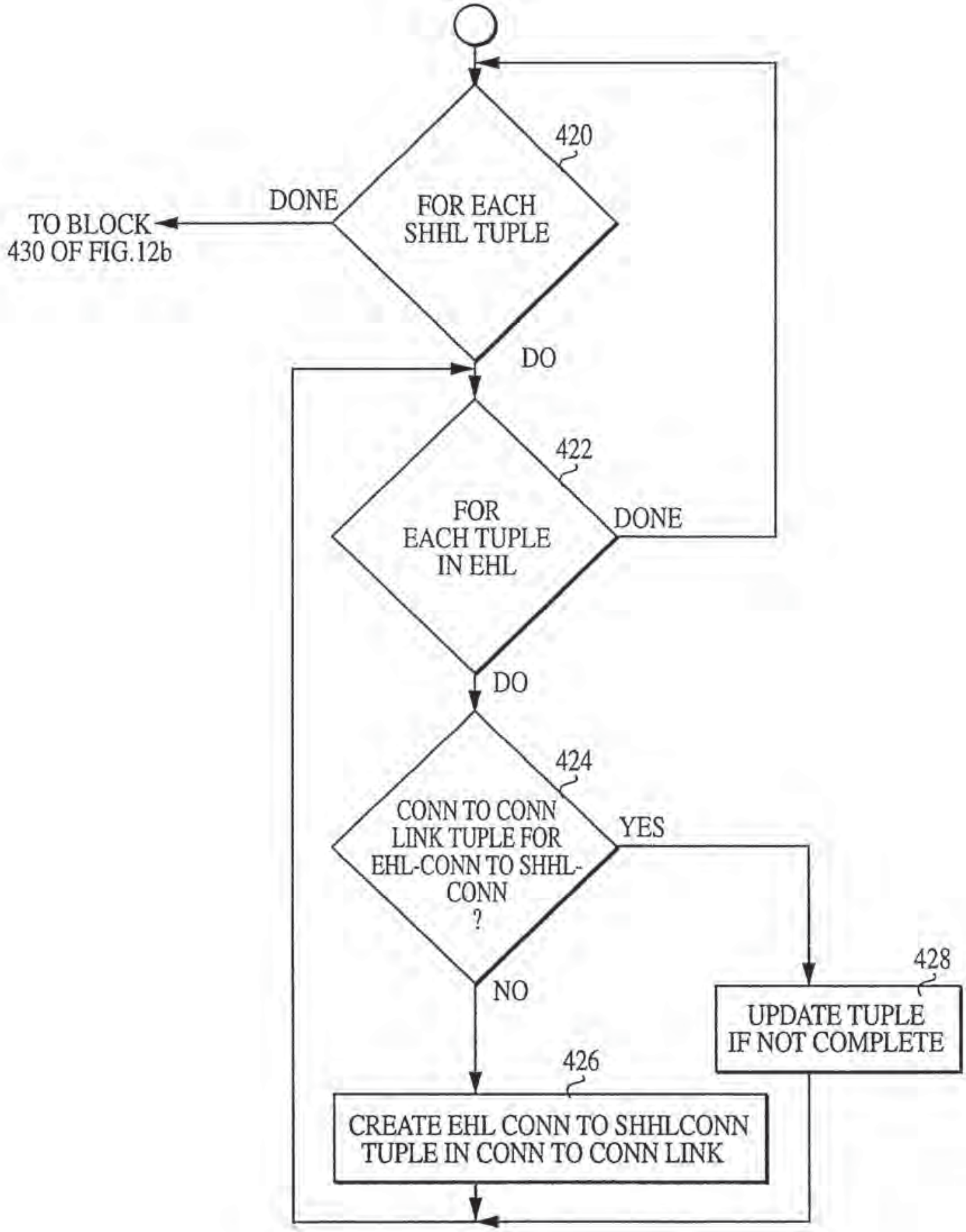
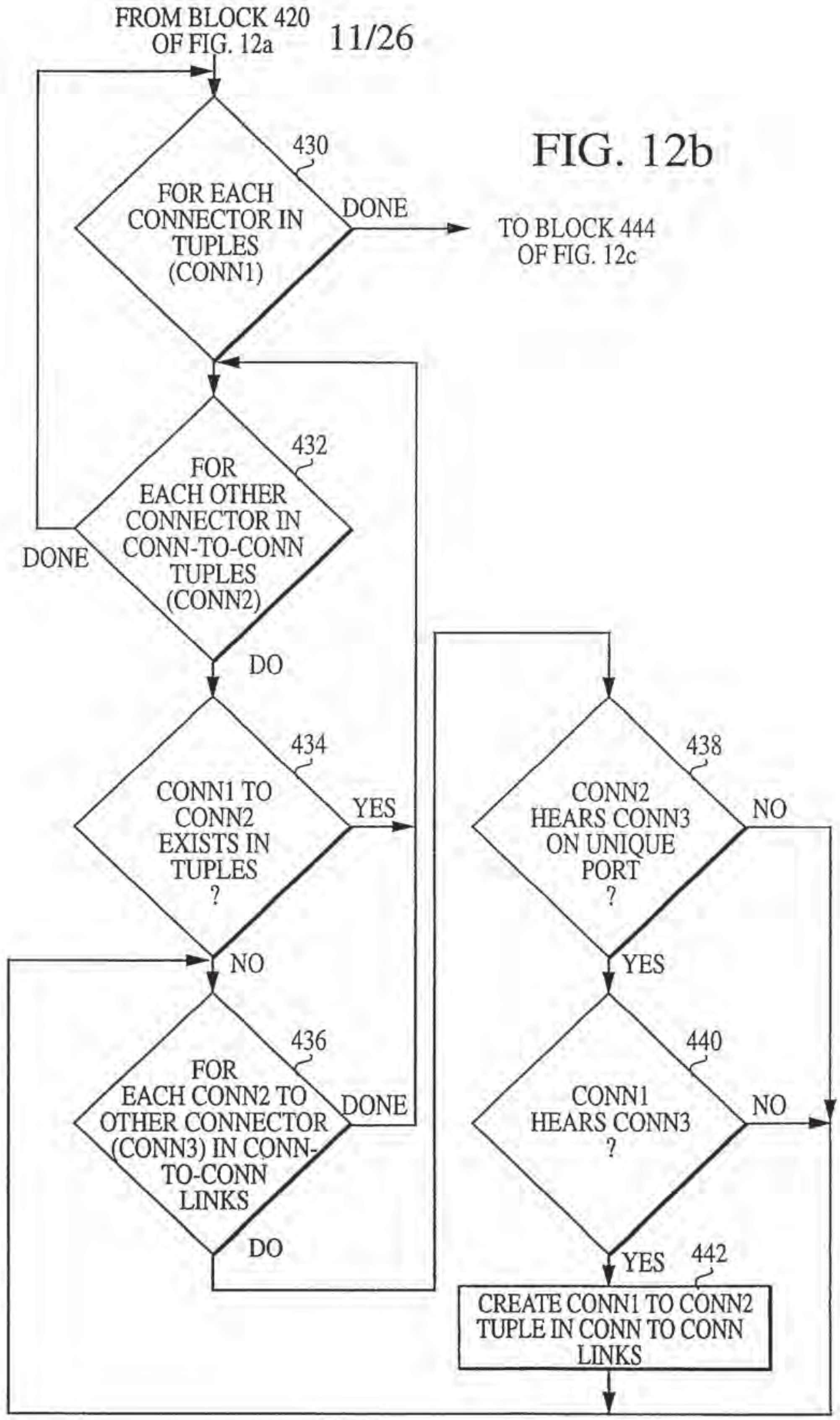

420
DONE
TO BLOCK 430 OF FIG.12b
FOR EACH SHHL TUPLE
DO
422
FOR EACH TUPLE IN EHL
DONE
DO
424
CONN TO CONN LINK TUPLE FOR EHL-CONN TO SHHL-CONN ?
YES
NO
428
UPDATE TUPLE IF NOT COMPLETE
426
CREATE EHL CONN TO SHHLCONN TUPLE IN CONN TO CONN LINK

FIG. 12a



FROM BLOCK 420 OF FIG. 12a
430
FOR EACH CONNECTOR IN TUPLES (CONN1)
DONE
TO BLOCK 444 OF FIG. 12c
432
FOR EACH OTHER CONNECTOR IN CONN-TO-CONN TUPLES (CONN2)
DONE
DO
434
CONN1 TO CONN2 EXISTS IN TUPLES ?
YES
NO
436
FOR EACH CONN2 TO OTHER CONNECTOR (CONN3) IN CONN-TO-CONN LINKS
DONE
DO
438
CONN2 HEARS CONN3 ON UNIQUE PORT ?
NO
YES
440
CONN1 HEARS CONN3 ?
NO
YES
442
CREATE CONN1 TO CONN2 TUPLE IN CONN TO CONN LINKS

FIG. 12b

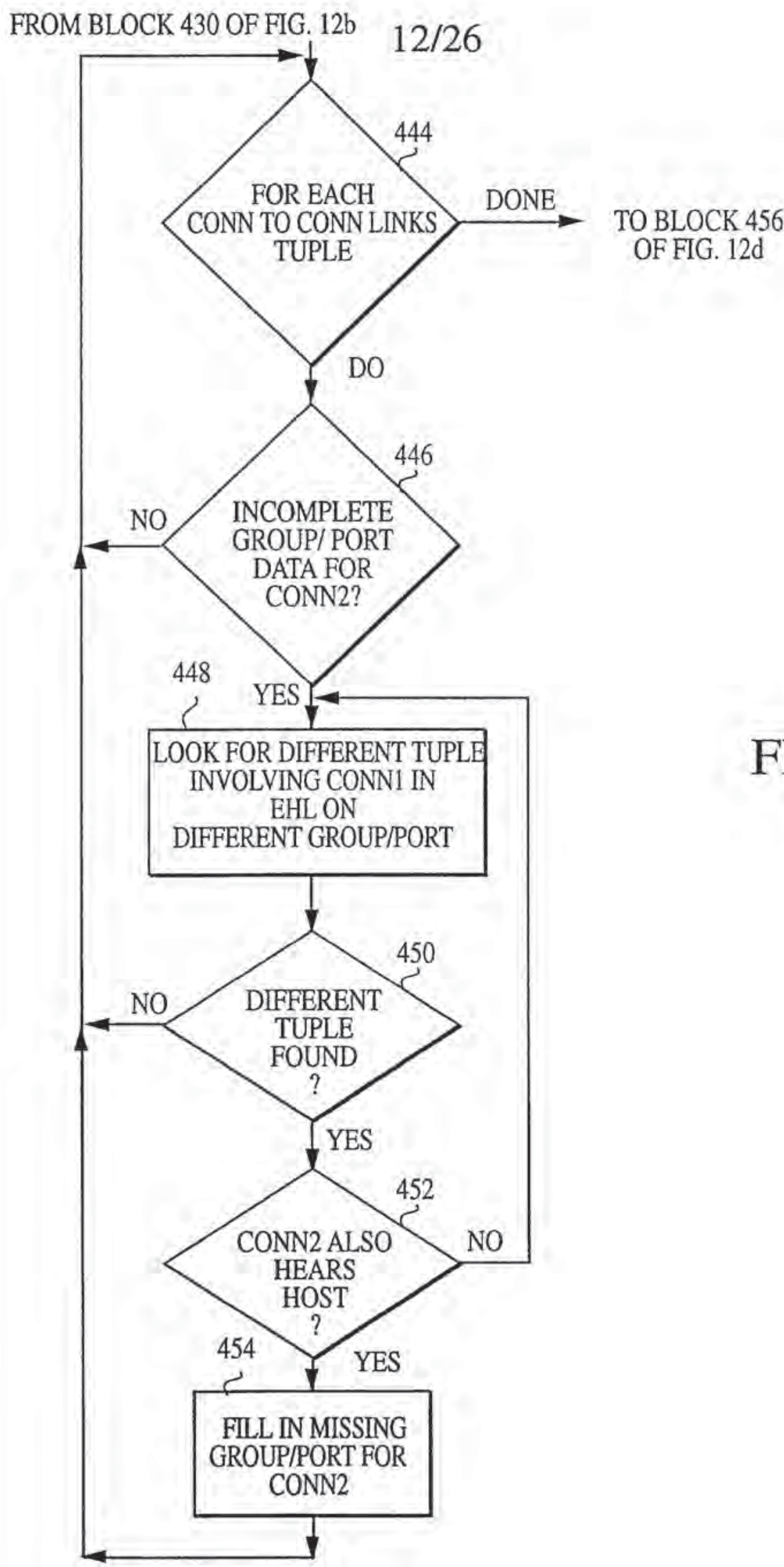

FROM BLOCK 430 OF FIG. 12b
444
FOR EACH CONN TO CONN LINKS TUPLE
DONE
TO BLOCK 456 OF FIG. 12d
DO
446
INCOMPLETE GROUP/ PORT DATA FOR CONN2?
NO
YES
448
LOOK FOR DIFFERENT TUPLE INVOLVING CONN1 IN EHL ON DIFFERENT GROUP/PORT
450
DIFFERENT TUPLE FOUND ?
NO
YES
452
CONN2 ALSO HEARS HOST ?
NO
YES
454
FILL IN MISSING GROUP/PORT FOR CONN2

FIG. 12c

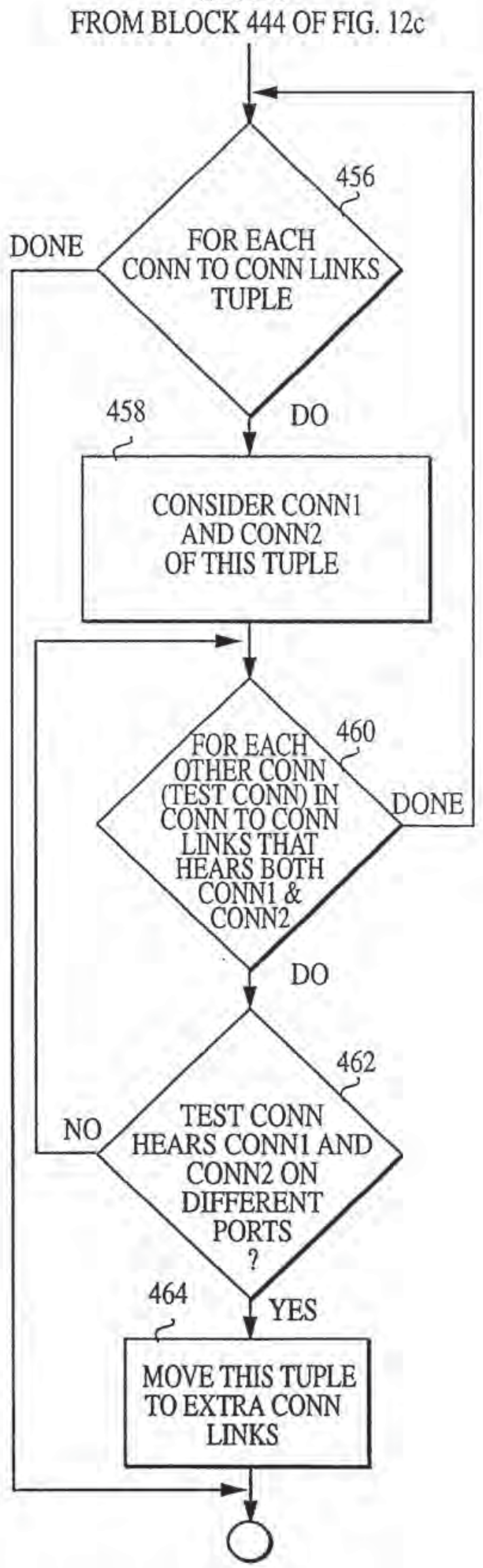

FROM BLOCK 444 OF FIG. 12c
456
FOR EACH CONN TO CONN LINKS TUPLE
DONE
DO
458
CONSIDER CONN1 AND CONN2 OF THIS TUPLE
460
FOR EACH OTHER CONN (TEST CONN) IN CONN TO CONN LINKS THAT HEARS BOTH CONN1 & CONN2
DONE
DO
462
TEST CONN HEARS CONN1 AND CONN2 ON DIFFERENT PORTS ?
NO
YES
464
MOVE THIS TUPLE TO EXTRA CONN LINKS

FIG. 12d

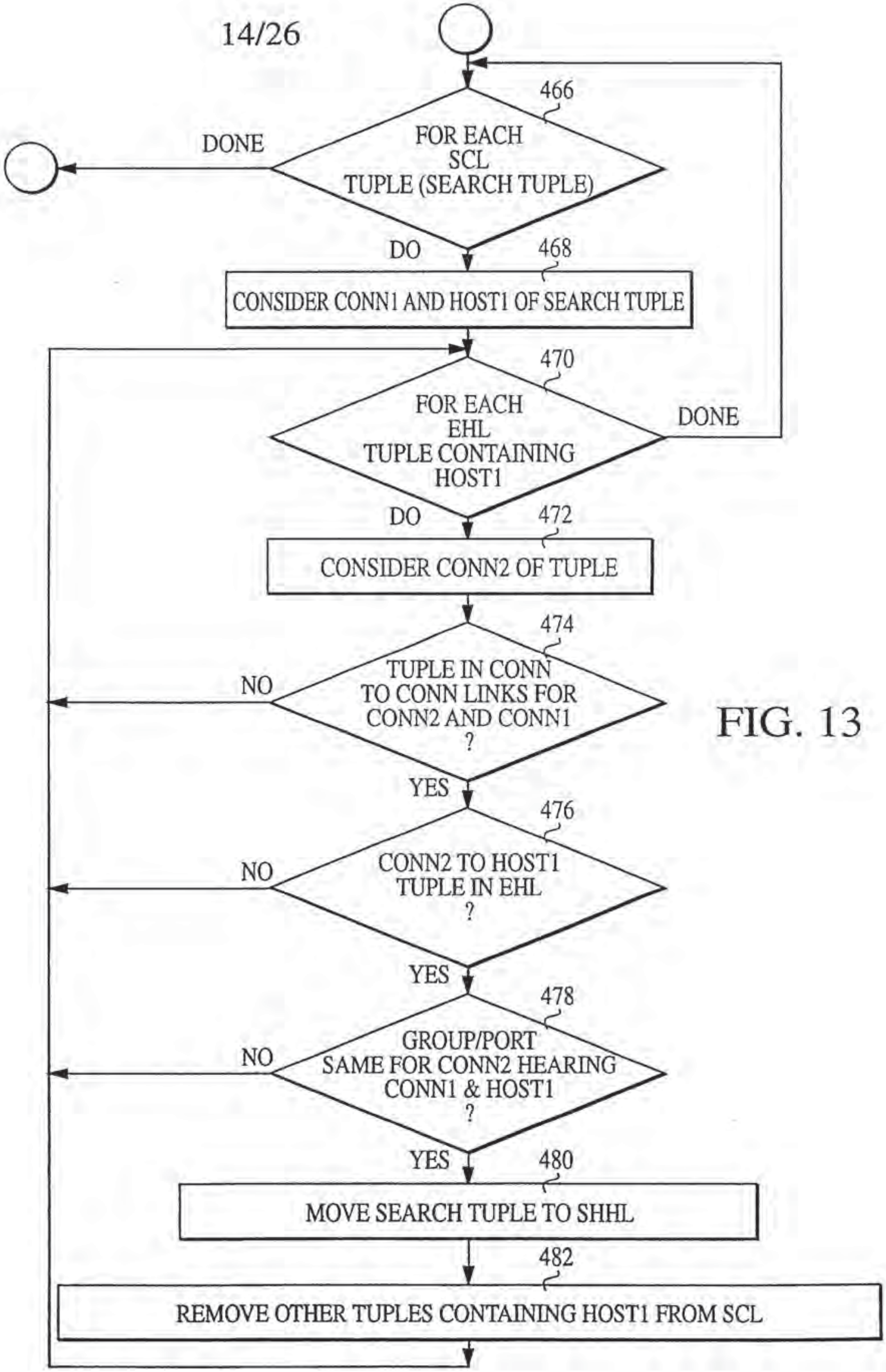

466
FOR EACH SCL TUPLE (SEARCH TUPLE)
DONE
DO
468
CONSIDER CONN1 AND HOST1 OF SEARCH TUPLE
470
FOR EACH EHL TUPLE CONTAINING HOST1
DONE
DO
472
CONSIDER CONN2 OF TUPLE
474
TUPLE IN CONN TO CONN LINKS FOR CONN2 AND CONN1 ?
NO
YES
476
CONN2 TO HOST1 TUPLE IN EHL ?
NO
YES
478
GROUP/PORT SAME FOR CONN2 HEARING CONN1 & HOST1 ?
NO
YES
480
MOVE SEARCH TUPLE TO SHHL
482
REMOVE OTHER TUPLES CONTAINING HOST1 FROM SCL

FIG. 13

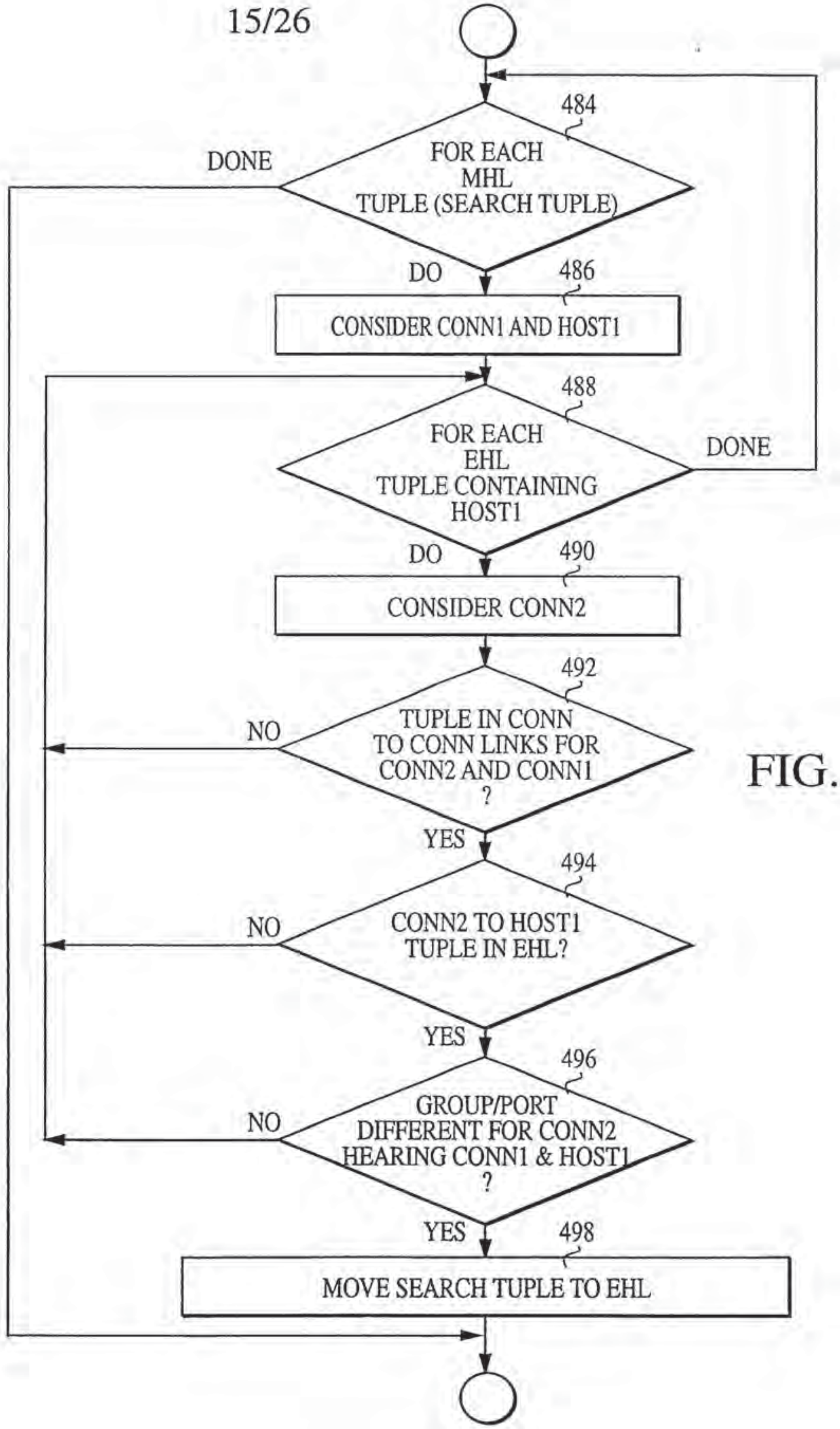
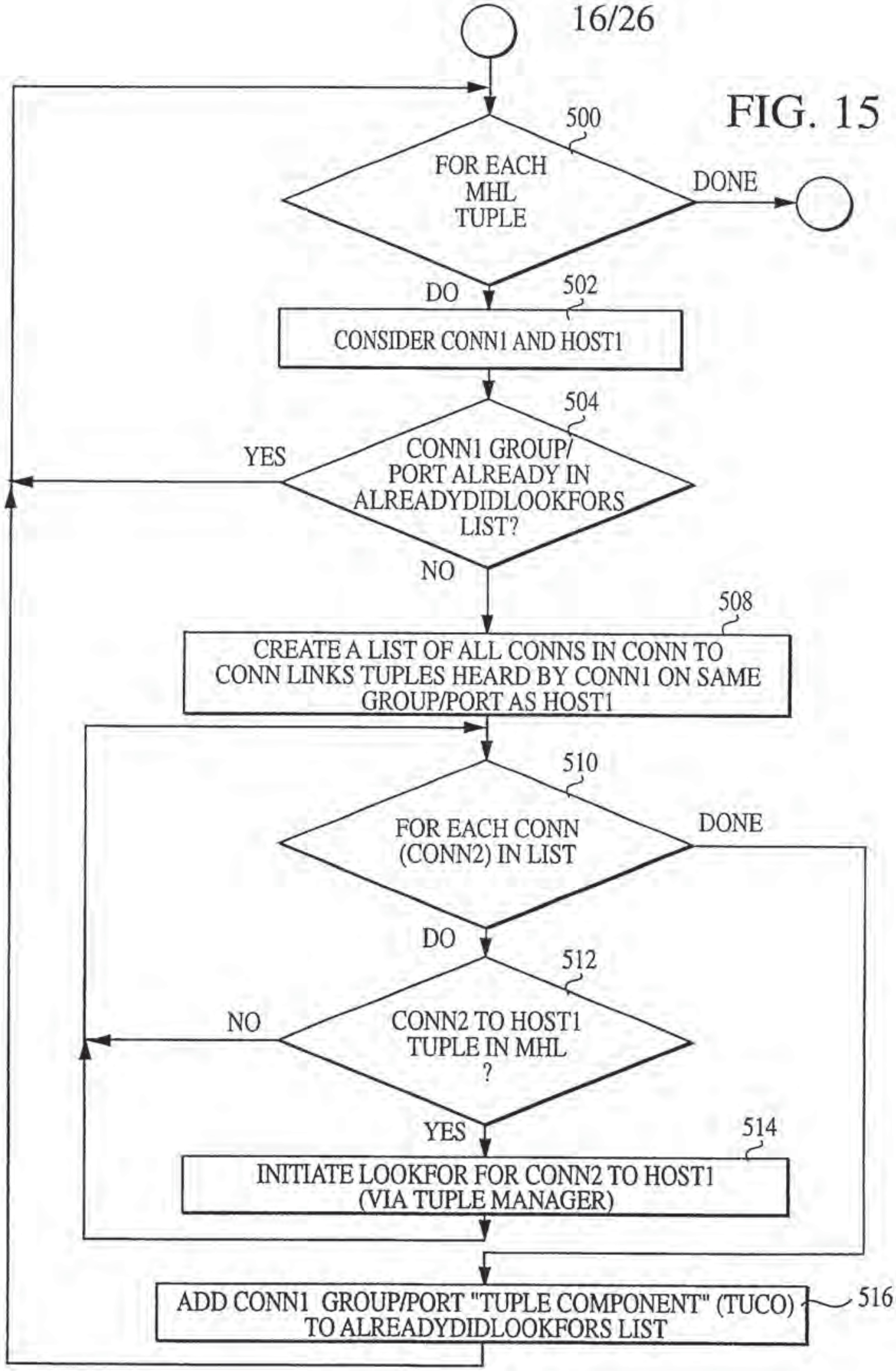

484
FOR EACH
MHL
TUPLE (SEARCH TUPLE)
DONE
DO
486
CONSIDER CONN1 AND HOST1
488
FOR EACH
EHL
TUPLE CONTAINING
HOST1
DONE
DO
490
CONSIDER CONN2
492
TUPLE IN CONN
TO CONN LINKS FOR
CONN2 AND CONN1
?
NO
YES
494
CONN2 TO HOST1
TUPLE IN EHL?
NO
YES
496
GROUP/PORT
DIFFERENT FOR CONN2
HEARING CONN1 & HOST1
?
NO
YES
498
MOVE SEARCH TUPLE TO EHL

FIG. 14

16/26

FIG. 15


500
FOR EACH MHL TUPLE
DONE
DO
502
CONSIDER CONN1 AND HOST1
504
YES
CONN1 GROUP/ PORT ALREADY IN ALREADYDIDLOOKFORS LIST?
NO
508
CREATE A LIST OF ALL CONNS IN CONN TO CONN LINKS TUPLES HEARD BY CONN1 ON SAME GROUP/PORT AS HOST1
510
FOR EACH CONN (CONN2) IN LIST
DONE
DO
512
NO
CONN2 TO HOST1 TUPLE IN MHL ?
YES
514
INITIATE LOOKFOR FOR CONN2 TO HOST1 (VIA TUPLE MANAGER)
ADD CONN1 GROUP/PORT "TUPLE COMPONENT" (TUCO) TO ALREADYDIDLOOKFORS LIST
516

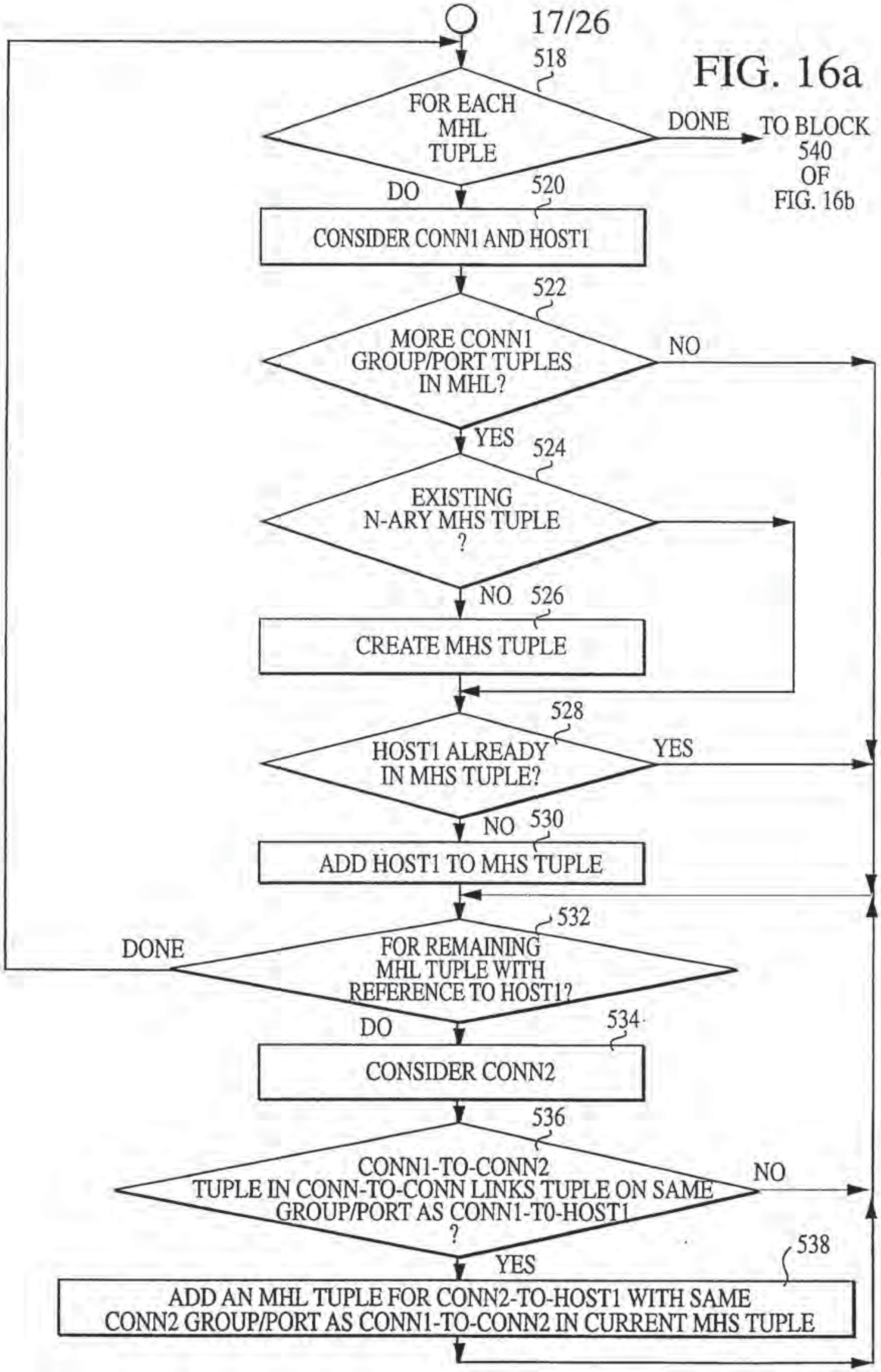

518
FOR EACH MHL TUPLE
DONE
TO BLOCK 540 OF FIG. 16b
DO
520
CONSIDER CONN1 AND HOST1
522
MORE CONN1 GROUP/PORT TUPLES IN MHL?
NO
YES
524
EXISTING N-ARY MHS TUPLE ?
NO
526
CREATE MHS TUPLE
528
HOST1 ALREADY IN MHS TUPLE?
YES
NO
530
ADD HOST1 TO MHS TUPLE
532
FOR REMAINING MHL TUPLE WITH REFERENCE TO HOST1?
DONE
DO
534
CONSIDER CONN2
536
CONN1-TO-CONN2 TUPLE IN CONN-TO-CONN LINKS TUPLE ON SAME GROUP/PORT AS CONN1-T0-HOST1 ?
NO
YES
538
ADD AN MHL TUPLE FOR CONN2-TO-HOST1 WITH SAME CONN2 GROUP/PORT AS CONN1-TO-CONN2 IN CURRENT MHS TUPLE

FIG. 16a

FIG. 16b

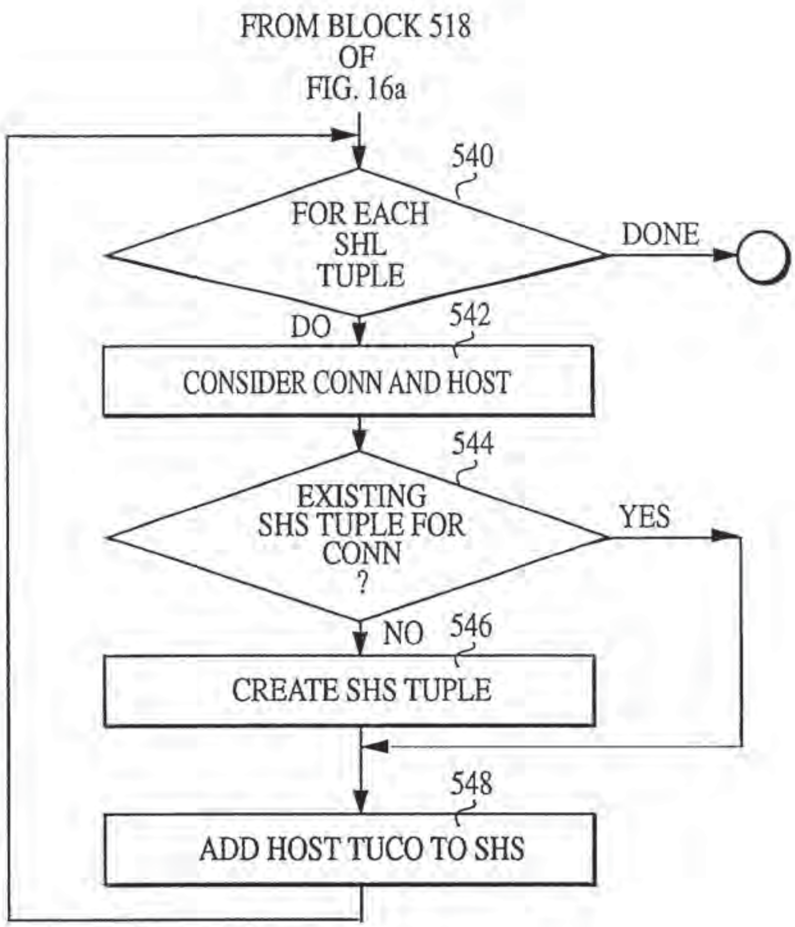

FROM BLOCK 518
OF
FIG. 16a
540
FOR EACH
SHL
TUPLE
DONE
DO
542
CONSIDER CONN AND HOST
544
EXISTING
SHS TUPLE FOR
CONN
?
YES
NO
546
CREATE SHS TUPLE
548
ADD HOST TUCO TO SHS

FIG. 17

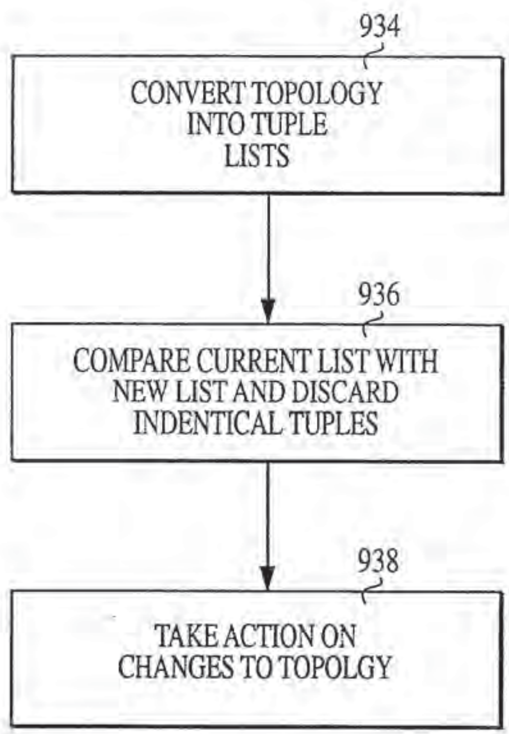
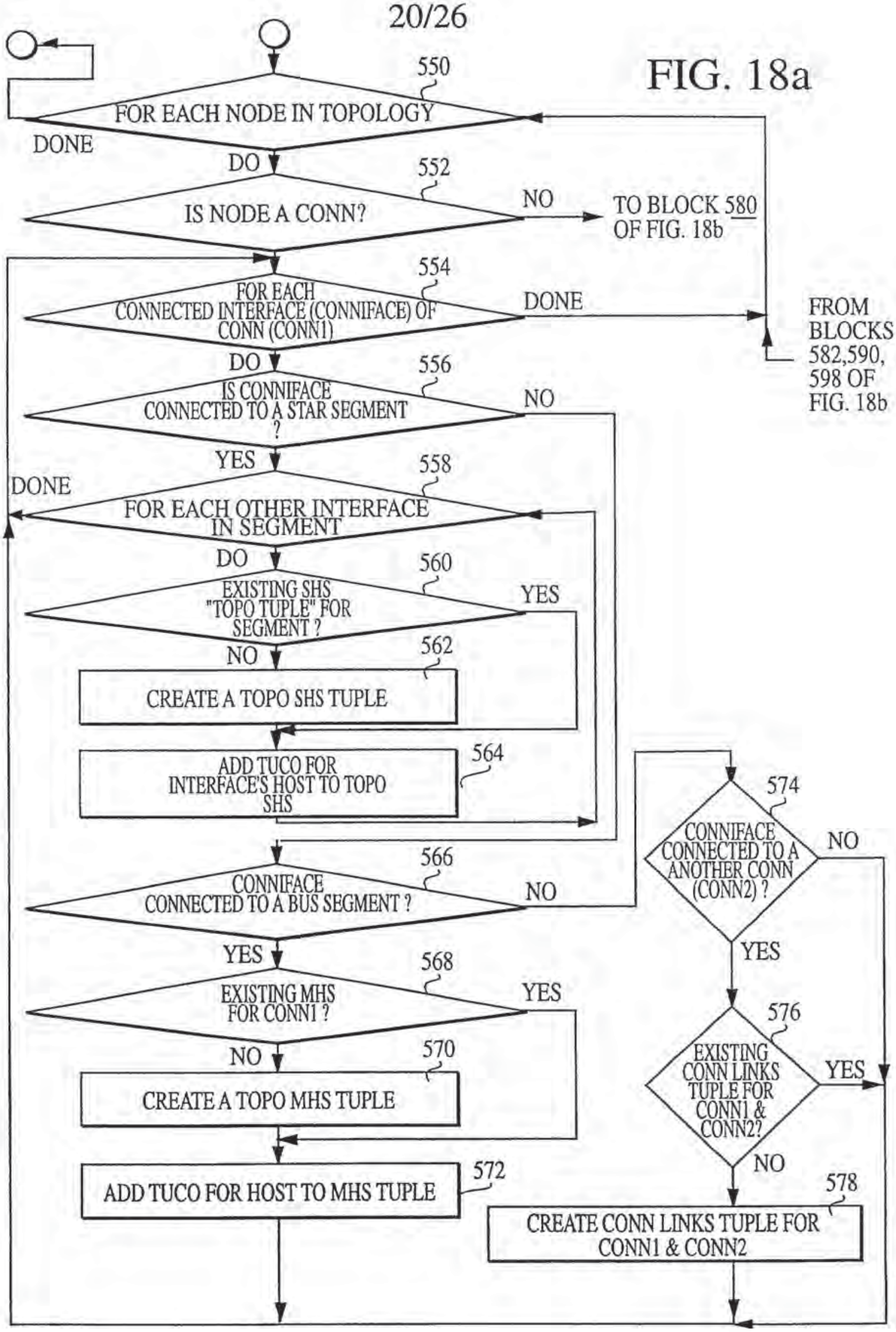

934
CONVERT TOPOLOGY INTO TUPLE LISTS
936
COMPARE CURRENT LIST WITH NEW LIST AND DISCARD INDENTICAL TUPLES
938
TAKE ACTION ON CHANGES TO TOPOLGY

FIG. 18a


550
FOR EACH NODE IN TOPOLOGY
DONE
DO
552
IS NODE A CONN?
NO
TO BLOCK 580 OF FIG. 18b
554
FOR EACH CONNECTED INTERFACE (CONNIFACE) OF CONN (CONN1)
DONE
FROM BLOCKS 582,590, 598 OF FIG. 18b
DO
556
IS CONNIFACE CONNECTED TO A STAR SEGMENT ?
NO
YES
558
DONE
FOR EACH OTHER INTERFACE IN SEGMENT
DO
560
EXISTING SHS "TOPO TUPLE" FOR SEGMENT ?
YES
NO
562
CREATE A TOPO SHS TUPLE
564
ADD TUCO FOR INTERFACE'S HOST TO TOPO SHS
574
CONNIFACE CONNECTED TO A ANOTHER CONN (CONN2) ?
NO
566
CONNIFACE CONNECTED TO A BUS SEGMENT ?
NO
YES
568
EXISTING MHS FOR CONN1 ?
YES
YES
576
EXISTING CONN LINKS TUPLE FOR CONN1 & CONN2?
YES
NO
570
CREATE A TOPO MHS TUPLE
572
ADD TUCO FOR HOST TO MHS TUPLE
NO
578
CREATE CONN LINKS TUPLE FOR CONN1 & CONN2

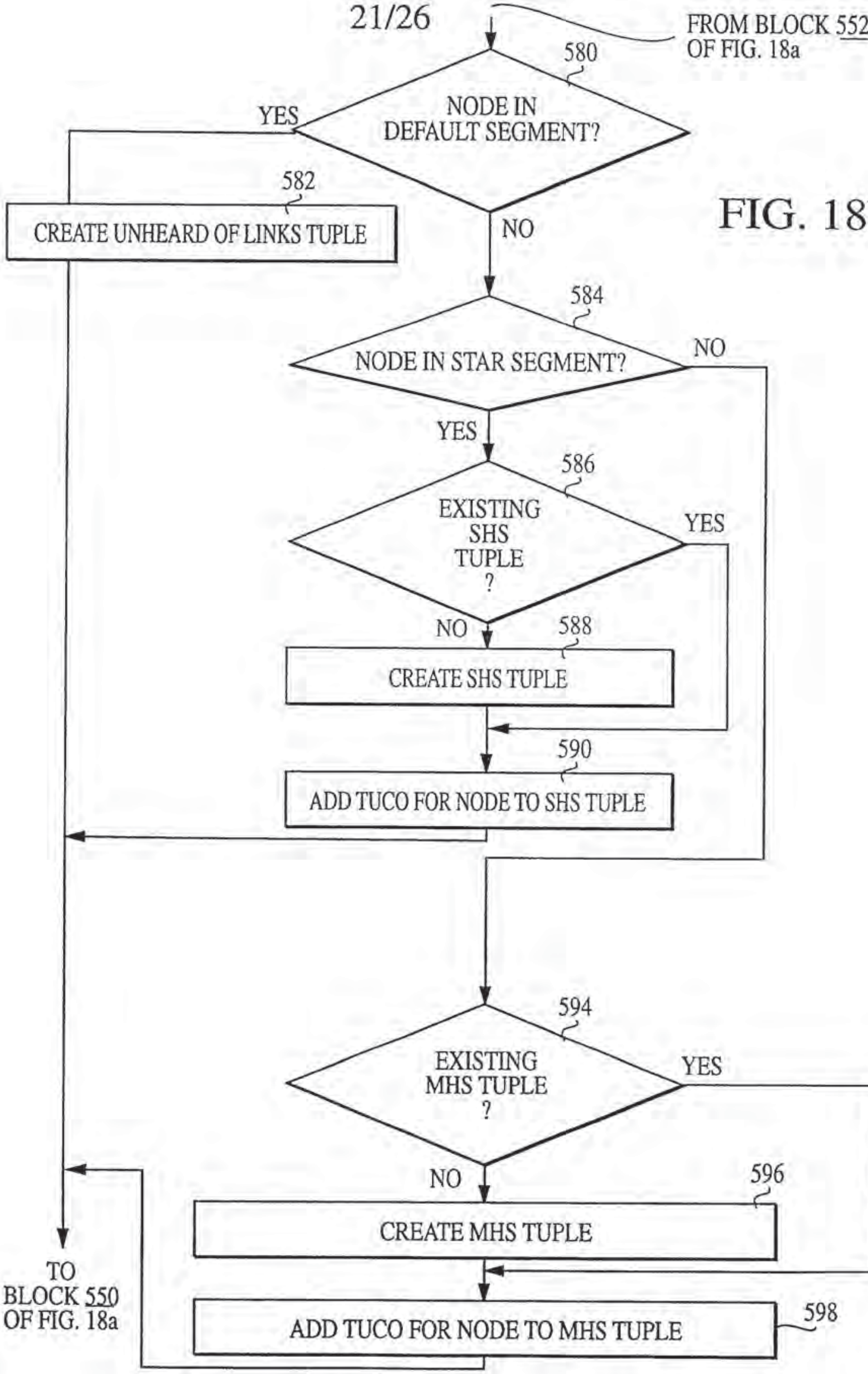

FROM BLOCK 552 OF FIG. 18a
580
NODE IN DEFAULT SEGMENT?
YES
NO
582
CREATE UNHEARD OF LINKS TUPLE
584
NODE IN STAR SEGMENT?
NO
YES
586
EXISTING SHS TUPLE ?
YES
NO
588
CREATE SHS TUPLE
590
ADD TUCO FOR NODE TO SHS TUPLE
594
EXISTING MHS TUPLE ?
YES
NO
596
CREATE MHS TUPLE
598
ADD TUCO FOR NODE TO MHS TUPLE
TO BLOCK 550 OF FIG. 18a

FIG. 18b

FIG. 19

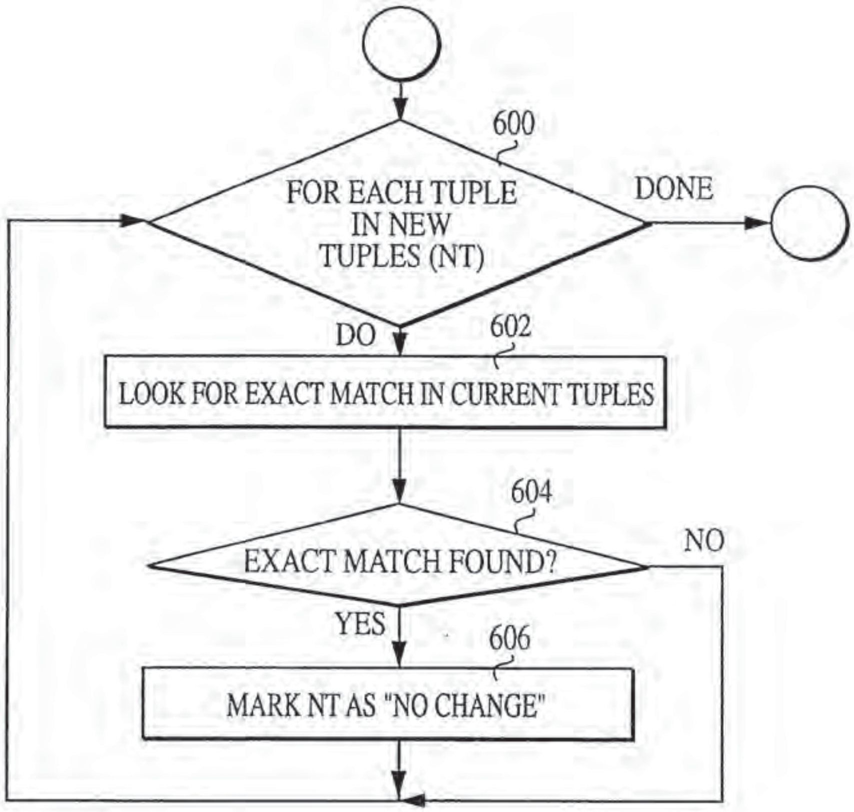
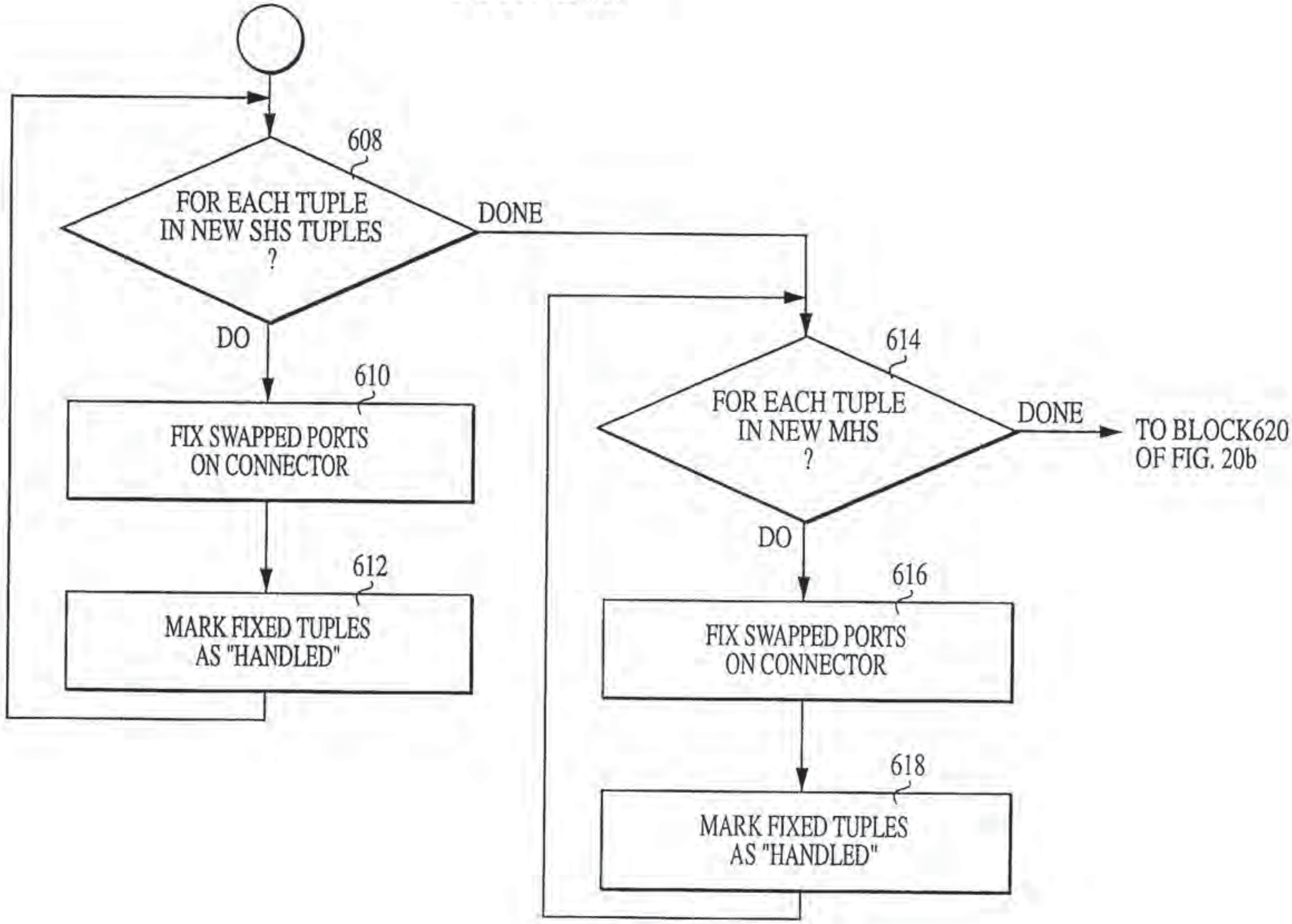

600
FOR EACH TUPLE IN NEW TUPLES (NT)
DONE
DO
602
LOOK FOR EXACT MATCH IN CURRENT TUPLES
604
EXACT MATCH FOUND?
NO
YES
606
MARK NT AS "NO CHANGE"

FIG. 20a


608
FOR EACH TUPLE IN NEW SHS TUPLES ?
DONE
DO
610
FIX SWAPPED PORTS ON CONNECTOR
612
MARK FIXED TUPLES AS "HANDLED"
614
FOR EACH TUPLE IN NEW MHS ?
DONE
TO BLOCK620 OF FIG. 20b
DO
616
FIX SWAPPED PORTS ON CONNECTOR
618
MARK FIXED TUPLES AS "HANDLED"

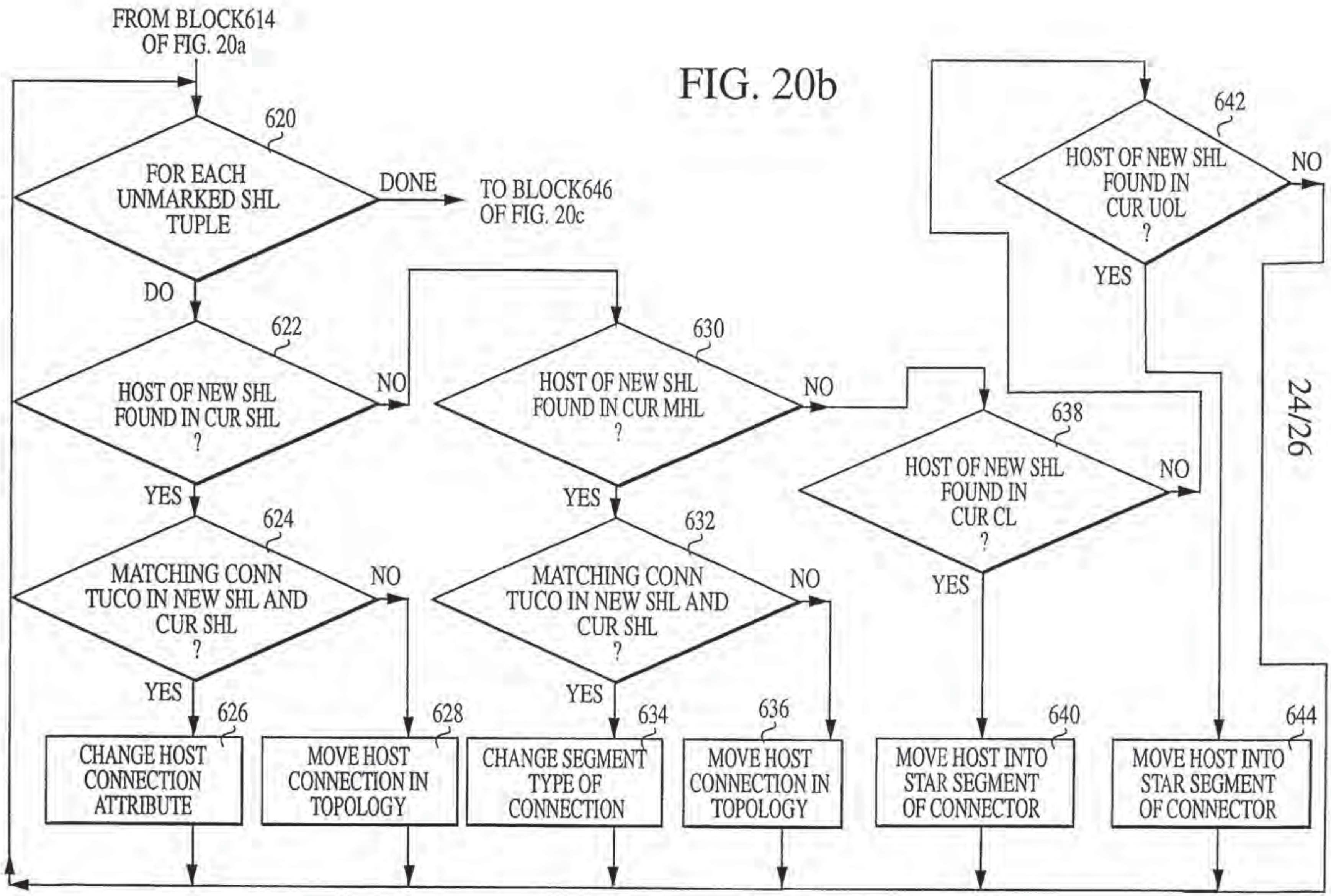

FROM BLOCK614
OF FIG. 20a
620
FOR EACH
UNMARKED SHL
TUPLE
DONE
TO BLOCK646
OF FIG. 20c
DO
622
HOST OF NEW SHL
FOUND IN CUR SHL
?
NO
YES
624
MATCHING CONN
TUCO IN NEW SHL AND
CUR SHL
?
NO
YES
626
CHANGE HOST
CONNECTION
ATTRIBUTE
628
MOVE HOST
CONNECTION IN
TOPOLOGY
630
HOST OF NEW SHL
FOUND IN CUR MHL
?
NO
YES
632
MATCHING CONN
TUCO IN NEW SHL AND
CUR SHL
?
NO
YES
634
CHANGE SEGMENT
TYPE OF
CONNECTION
636
MOVE HOST
CONNECTION IN
TOPOLOGY
638
HOST OF NEW SHL
FOUND IN
CUR CL
?
NO
YES
640
MOVE HOST INTO
STAR SEGMENT
OF CONNECTOR
642
HOST OF NEW SHL
FOUND IN
CUR UOL
?
NO
YES
644
MOVE HOST INTO
STAR SEGMENT
OF CONNECTOR

FIG. 20b

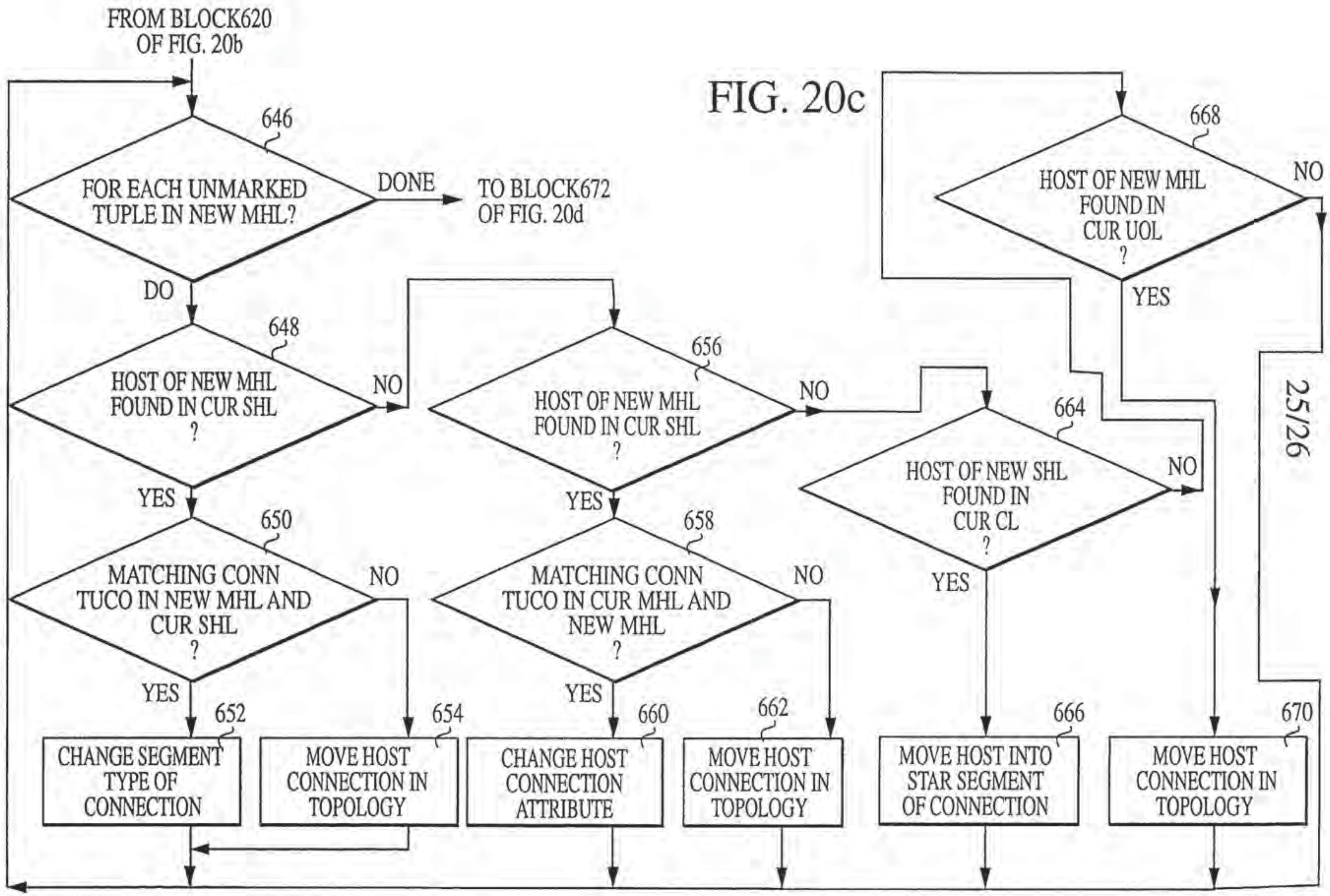
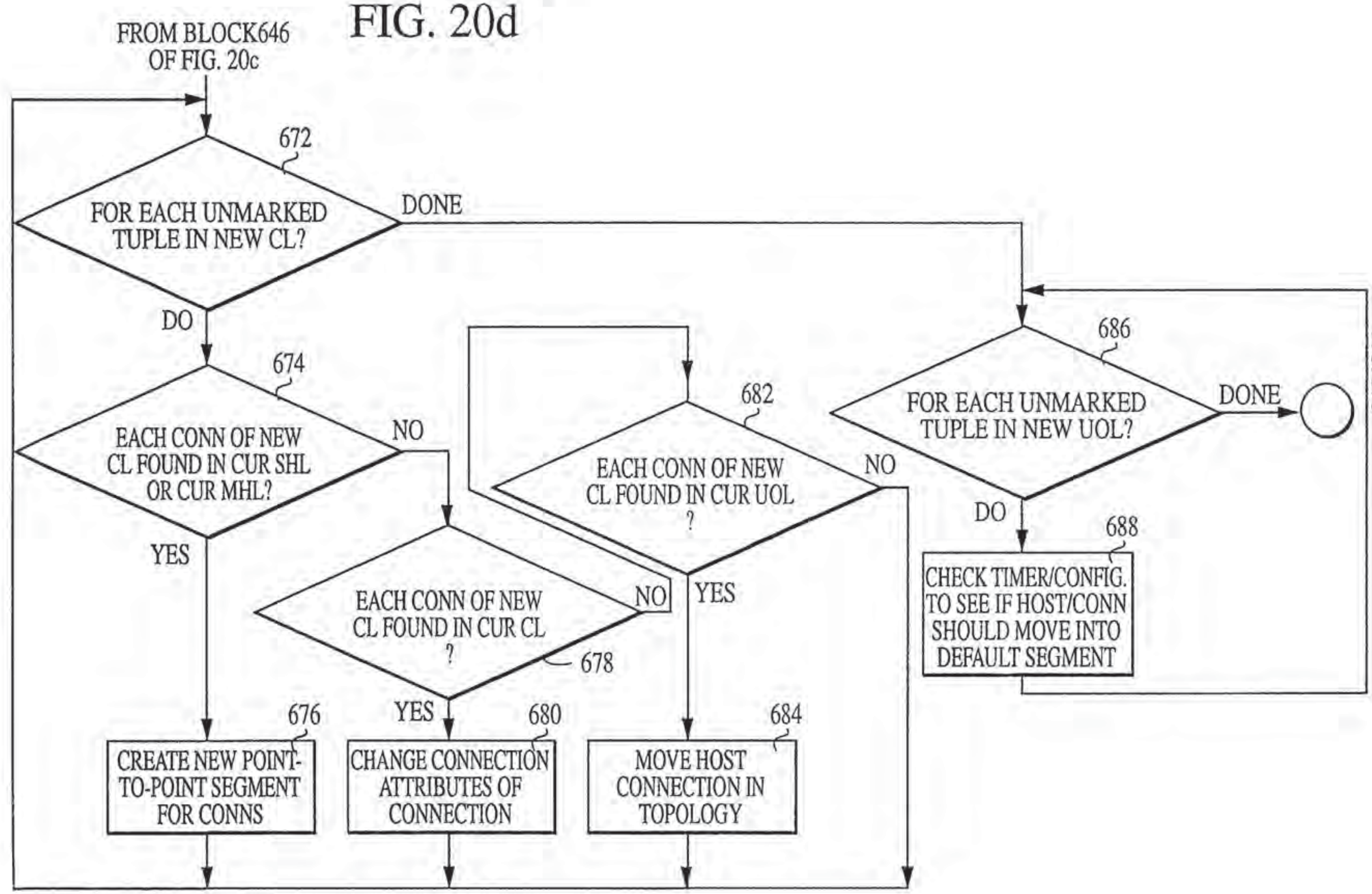

FROM BLOCK620
OF FIG. 20b
646
FOR EACH UNMARKED
TUPLE IN NEW MHL?
DONE
TO BLOCK672
OF FIG. 20d
DO
648
HOST OF NEW MHL
FOUND IN CUR SHL
?
NO
YES
650
MATCHING CONN
TUCO IN NEW MHL AND
CUR SHL
?
NO
YES
652
CHANGE SEGMENT
TYPE OF
CONNECTION
654
MOVE HOST
CONNECTION IN
TOPOLOGY
656
HOST OF NEW MHL
FOUND IN CUR SHL
?
NO
YES
658
MATCHING CONN
TUCO IN CUR MHL AND
NEW MHL
?
NO
YES
660
CHANGE HOST
CONNECTION
ATTRIBUTE
662
MOVE HOST
CONNECTION IN
TOPOLOGY
664
HOST OF NEW SHL
FOUND IN
CUR CL
?
NO
YES
666
MOVE HOST INTO
STAR SEGMENT
OF CONNECTION
668
HOST OF NEW MHL
FOUND IN
CUR UOL
?
NO
YES
670
MOVE HOST
CONNECTION IN
TOPOLOGY
25/26

FIG. 20c

FIG. 20d


FROM BLOCK646
OF FIG. 20c
672
FOR EACH UNMARKED
TUPLE IN NEW CL?
DONE
DO
674
EACH CONN OF NEW
CL FOUND IN CUR SHL
OR CUR MHL?
NO
YES
678
EACH CONN OF NEW
CL FOUND IN CUR CL
?
NO
YES
682
EACH CONN OF NEW
CL FOUND IN CUR UOL
?
NO
YES
676
CREATE NEW POINT-
TO-POINT SEGMENT
FOR CONNS
680
CHANGE CONNECTION
ATTRIBUTES OF
CONNECTION
684
MOVE HOST
CONNECTION IN
TOPOLOGY
686
FOR EACH UNMARKED
TUPLE IN NEW UOL?
DONE
DO
688
CHECK TIMER/CONFIG.
TO SEE IF HOST/CONN
SHOULD MOVE INTO
DEFAULT SEGMENT

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

ATTORNEY DOCKET NO. 10008102-1

As a below named inventor, I hereby declare that:

My residence/post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

Method And System For Identifying And Processing Changes To A Network Topology

the specification of which is attached hereto unless the following box is checked:

( ) was filed on ________________ as US Application Serial No. or PCT International Application Number ______________ and was amended on ________________ (if applicable).

I hereby state that I have reviewed and understood the contents of the above-identified specification, including the claims, as amended by any amendment(s) referred to above. I acknowledge the duty to disclose all information which is material to patentability as defined in 37 CFR 1.56.

Foreign Application(s) and/or Claim of Foreign Priority

I hereby claim foreign priority benefits under Title 35, United States Code Section 119 of any foreign application(s) for patent or inventor(s) certificate listed below and have also identified below any foreign application for patent or inventor(s) certificate having a filing date before that of the application on which priority is claimed:

COUNTRY	APPLICATION NUMBER	DATE FILED	PRIORITY CLAIMED UNDER 35 U.S.C. 119
N/A			YES: ___ NO: ___
			YES: ___ NO: ___

Provisional Application

I hereby claim the benefit under Title 35, United States Code Section 119(e) of any United States provisional application(s) listed below:

APPLICATION SERIAL NUMBER	FILING DATE
N/A	

U. S. Priority Claim

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code Section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, Section 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

APPLICATION SERIAL NUMBER	FILING DATE	STATUS (patented/pending/abandoned)
N/A		

POWER OF ATTORNEY:

As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith:

Customer Number 022879

Place Customer Number Bar Code Label here

Send Correspondence to:
HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

Direct Telephone Calls To:
T. Grant Ritz
(970) 898-0697

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Inventor: Eric A Pulsipher **Citizenship:** US

Residence: 2937 Redburn Drive Ft Collins CO 80525

Post Office Address: Same as residence

[signature] 10/31/2000

Inventor's Signature **Date**

Rev 10/00 (DecPwr) (Use Page Two For Additional Inventor(s) Signature(s)) Page 1 of 2

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION (continued)

ATTORNEY DOCKET NO. 10008102-1

Full Name of # 2 joint inventor: Joseph R Hunt Citizenship: US

Residence: ~~5841 Meadow Creek Ln~~ 5239 Hahns Peak Dr #205 Loveland, CO 80538

Post Office Address: Same as Residence

Inventor's Signature: Joseph R Hunt Date: 10/31/00

Full Name of # 3 joint inventor: ______ Citizenship: ______

Residence: ______

Post Office Address: ______

Inventor's Signature ______ Date ______

Full Name of # 4 joint inventor: ______ Citizenship: ______

Residence: ______

Post Office Address: ______

Inventor's Signature ______ Date ______

Full Name of # 5 joint inventor: ______ Citizenship: ______

Residence: ______

Post Office Address: ______

Inventor's Signature ______ Date ______

Full Name of # 6 joint inventor: ______ Citizenship: ______

Residence: ______

Post Office Address: ______

Inventor's Signature ______ Date ______

Full Name of # 7 joint inventor: ______ Citizenship: ______

Residence: ______

Post Office Address: ______

Inventor's Signature ______ Date ______

Full Name of # 8 joint inventor: ______ Citizenship: ______

Residence: ______

Post Office Address: ______

Inventor's Signature ______ Date ______

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P. O. Box 272400
Fort Collins, Colorado 80527-2400

11-02-00

PATENT APPLICATION

A

ATTORNEY DOCKET NO. 10008102-1

10/31/00 JC956 U.S. PTO

IN THE U.S. PATENT AND TRADEMARK OFFICE
Patent Application Transmittal Letter

COMMISSIONER FOR PATENTS
Washington, D.C. 20231

JC825 U.S. PTO 09/703942 10/31/00

Sir:

Transmitted herewith for filing under 37 CFR 1.53(b) is a(n): (X) Utility ( ) Design
(X) original patent application,
( ) continuation-in-part application

INVENTOR(S): **Eric A Pulsipher et al**

TITLE: **Method And System For Identifying And Processing Changes To A Network Topology**

Enclosed are:

(X) The Declaration and Power of Attorney. (X) signed ( ) unsigned or partially signed

(X) 26 sheets of drawings (one set) ( ) Associate Power of Attorney

( ) Form PTO-1449 ( ) Information Disclosure Statement and Form PTO-1449

( ) Priority document(s) ( ) (Other) ______ (fee $______)

00 FEDF " 2495 0 0 2 5 0 0

CLAIMS AS FILED BY OTHER THAN A SMALL ENTITY				
(1) FOR	(2) NUMBER FILED	(3) NUMBER EXTRA	(4) RATE	(5) TOTALS
TOTAL CLAIMS	20 — 20	0	X $18	$ 0
INDEPENDENT CLAIMS	3 — 3	0	X $80	$ 0
ANY MULTIPLE DEPENDENT CLAIMS	0		$270	$ 0
BASIC FEE: Design ($320.00); Utility ($710.00)				$ 710
TOTAL FILING FEE				$ 710
OTHER FEES				$
TOTAL CHARGES TO DEPOSIT ACCOUNT				$ 710

Charge $ 710 to Deposit Account 08-2025. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16, 1.17,1.19, 1.20 and 1.21. A duplicate copy of this sheet is enclosed.

"Express Mail" label no. EL523338183US

Date of Deposit Oct. 31, 2000

I hereby certify that this is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to: Commissioner for Patents, Washington, D.C. 20231.

By Laura M Clark

Typed Name: Laura M. Clark

Respectfully submitted,

Eric A Pulsipher et al

By T.G.R.

T. Grant Ritz

Attorney/Agent for Applicant(s)
Reg. No. **39,819**
Date: **Oct. 31, 2000**
Telephone No.: **(970) 898-0697**

Title

Method and System for Identifying and Processing Changes to a Network Topology

Field of Invention

The present invention relates generally to computer networks. More particularly, it relates to a method and system for identifying changes to a network topology and for acting upon the network based on the changes.

Background

As communications networks, such as the Internet, carry more and more traffic, efficient use of the bandwidth available in the network becomes more and more important. Switching technology was developed in order to reduce congestion and associated competition for the available bandwidth. Switching technology works by restricting traffic. Instead of broadcasting a given data packet to all parts of the network, switches are used to control data flow such that the data packet is sent only along those network segments necessary to deliver it to the target node. The smaller volume of traffic on any given segment results in few packet collisions on that segment and, thus, the smoother and faster delivery of data. A choice between alternative paths is usually possible and is typically made based upon current traffic patterns.

The intelligent routing of data packets with resultant reduction in network congestion can only be effected if the network topology is known. The topology of a network is a description of the network which includes the location of and interconnections between nodes on the network. The word "topology" refers to either the physical or logical layout of the network, including devices, and their connections in relationship to one another. Information necessary to create the topology layout can be derived from tables stored in network devices such as hubs, bridges, and switches. The information in these tables is in a constant state of flux as new entries are being added and old entries time out. Many times there simply is not enough information to determine where to place a particular device.

Switches examine each data packet that they receive, read the source addresses, and log those addresses into tables along with the switch ports on which the packets were received. If a packet is received with a target address without an entry in the switches table, the switch receiving it

broadcasts that packet to each of its ports. When the switch receives a reply, it will have identified where the new node lies.

In a large network with multiple possible paths from the switch to the target node, this table can become quite large and may require a significant amount of the switch's resources to develop and maintain. As an additional complication, the physical layout of devices and their connections are typically in a state of constant change. Devices are continually being removed from, added to, and moved to new physical locations on the network. To be effectively managed, the topology of a network must be accurately and efficiently ascertained, as well as maintained.

Existing mapping methods have limitations that prevent them from accurately mapping topological relationships. Multiple connectivity problems are one sort of difficulty encountered by existing methods. For example, connectors such as routers, switches, and bridges may be interconnected devices in a network. Some existing methods assume that these devices have only a single connection between them. In newer devices, however, it is common for manufacturers to provide multiple connections between devices to improve network efficiency and to increase capacity of links between the devices. The multiple connectivity allows the devices to maintain connection in case one connection fails. Methods that do not consider multiple connectivity do not present a complete and accurate topological map of the network.

Another limitation of existing topology methods is the use of a single reference to identify a device. Existing methods use a reference interface or a reference address in a set of devices to orient all other devices in the same area. These methods assumed that every working device would be able to identify, or "hear," this reference and identify it with a particular port of the device. With newer devices, however, it is possible that the same address or reference may be heard out of multiple ports of the same device. It is also possible that the address or reference may not be heard from any ports, for example, if switching technology is used.

Still another limitation of existing mapping systems is that they require a complete copy of the topological database to be stored in memory. In larger networks, the database is so large that this really is not feasible, because it requires the computer to be very large and expensive.

Still another difficulty with existing systems is that they focus on the minutia without considering the larger mapping considerations. Whenever an individual change in the system is detected, existing methods immediately act on that change, rather than taking a broader view of the change in the context of other system changes. For example, a device may be removed from the network temporarily and replaced with its ports reversed. In existing systems, this swapped port scenario could require hundreds or thousands of changes because the reference addresses will have changed for all interconnected devices.

Still another disadvantage of existing methods is that they use a continuous polling paradigm. These methods continuously poll network addresses throughout the day and make decisions based on those continuous polling results. This creates traffic on the network that slows other processes.

Still another limitation of existing methods is the assumption that network parts of a particular layer would be physically separated from other parts. Network layer 1 may represent the physical cabling of the network, layer 2 may represent the device connectivity, and layer 3 may represent a higher level of abstraction, such as the groupings of devices into regions. Existing methods assume that all layer 3 region groupings are self-contained, running on the same unique physical networking. However, in an internet protocol (IP) network, multiple IP domains may co-exist on the same lower layer networking infrastructure. It has become common for a network to employ a virtual local area network (LAN) to improve security or to simplify network maintenance, for example. Using virtual LANs, a system may have any number of different IP domains sharing the same physical connectivity. As a result, existing methods create confusion with respect to topological mapping because networks with multiple IP addresses in different subnets for the infrastructure devices cannot be properly represented because they assume the physical separation of connectivity for separate IP domains. Still another limitation of existing methods is that they do not allow topological loops, such as port aggregation or trunking, and switch meshing.

Summary of Invention

A method and system are disclosed for mapping the topology of a network having interconnected nodes by identifying changes in the network and updating a stored network topology

based on the changes. The nodal connections are represented by data tuples that store information such as a host identifier, a connector interface, and a port specification for each connection. A topology database stores an existing topology of a network. A topology converter accesses the topology database and converts the existing topology into a list of current tuples. A connection calculator calculates tuples to represent connections in the new topology. The topology converter receives the new tuples, identifies changes to the topology, and updates the topology database using the new tuples. The topology converter identifies duplicate tuples that appear in both the new tuples and the existing tuples and marks the duplicate tuples to reflect that no change has occurred to these connections. The topology converter attempts to resolve swapped port conditions and searches for new singly-heard and multi-heard host link tuples in the list of existing tuples. The topology converter also searches for new conflict link tuples in the existing tuples. The topology converter updates the topology database with the new topology.

Summary of Drawings

Figure 1 is a drawing of a typical topological bus segment for representing the connectivity of nodes on a network.

Figure 2 is a drawing of a typical topological serial segment for representing the connectivity of nodes on a network.

Figure 3 is a drawing of a typical topological star segment for representing the connectivity of nodes on a network.

Figure 4 is a drawing of another typical topological star segment for representing the connectivity of nodes on a network.

Figure 5 is a drawing of the connectivity of an example network system.

Figure 6 is a drawing of the connectivity of another example network system.

Figure 7 is a block diagram of the system.

Figure 8 is a flow chart of the method of the system.

Figure 9 is a flow chart of the method used by the tuple manager.

Figure 10 is a flow chart of the method used by the connection calculator.

Figure 11 is a flow chart of the first weeding phase of the method used by the connection calculator.

Figures 12a-d are flow charts of an infrastructure-building phase of the method used by the connection calculator.

Figure 13 is a flow chart of a second weeding phase of the method used by the connection calculator.

Figure 14 is a flow chart of the noise reduction phase of the method used by the connection calculator.

Figure 15 is a flow chart of the look-for phase of the method used by the connection calculator.

Figures 16a-b are flow charts of the consolidation phase of the method used by the connection calculator.

Figure 17 is a flow chart of the method used by the topology converter.

Figures 18a-b are flow charts of the morph topo phase of the method used by the topology converter.

Figure 19 is a flow chart of the duplication discard phase of the method used by the topology converter.

Figures 20a-d are flow charts of the identify different tuples phase of the method used by the topology converter.

Detailed Description

The system provides an improved method for creating topological maps of communication networks based. Connectivity information is retrieved from the network nodes and stored as "tuples" to track specifically the desired information necessary to map the topology. These light weight data structures may store the host identifier, interface index, and a port. From this tuple information, the topology may be determined. A tuple may be a binary element insofar as it has two parts representing the two nodes on either end of a network link or segment. A "tuco" refers to a tuple component, such as half of a binary tuple.

As used herein, a node is any electronic component, such as a connector or a host, or combination of electronic components with their interconnections. A connector is any network device other than a host, including a switching device. A switching device is one type of connector and refers to any device that controls the flow of messages on a network. Switching devices include, but are not limited to, any of the following devices: repeaters, hubs, routers, bridges, and switches.

As used herein, the term "tuple" refers to any collection of assorted data. Tuples may be used to track information about network topology by storing data from network nodes. In one use, tuples may include a host identifier, interface information, and a port specification for each node. The port specification (also described as the group/port) may include a group number and a port number, or just a port number, depending upon the manufacturer's specifications. A binary tuple may include this information about two nodes as a means of showing the connectivity between them, whether the nodes are connected directly or indirectly through other nodes. A "conn-to-conn" tuple refers to a tuple that has connectivity data about connector nodes. A "conn-to-host" tuple refers to a tuple that has connectivity data about a connector node and a host node. In one use, tuples may have data about more than two nodes; that is, they may be n-ary tuples, such as those used with respect to shared media connections described herein.

A "singly-heard host" (shh) refers to a host, such as a workstation, PC, terminal, printer, other device, etc., that is connected directly to a connector, such as a switching device. A singly-heard host link (shhl) refers to the link, also referred to as a segment, between a connector and an shh. A "multi-heard host" (mhh) refers to hosts that are heard by a connector on the same port that other hosts are heard. A multi-heard host link (mhhl) refers to the link between the connector and an mhh. A link generally refers to the connection between nodes. A segment is a link that may include a shared media connection.

Figure 1 is a drawing of a typical topological bus segment 100 for representing the connectivity of nodes on a network 110. In Figure 1, first and second hosts 121, 122, as well as a first port 131 of a first connector 140 are interconnected via the network 110. The bus segment

100 comprises the first and second hosts 121, 122 connected to the first port 131 of the first connector 140.

Figure 2 is a drawing of a typical topological serial segment 200 for representing the connectivity of nodes on the network 110. In Figure 2, the first host 121 comprises a second port 132 on a second connector 145 which is connected via the network 110 to the first port 131 on the first connector 140. The serial segment 200 comprises the second port 132 on the second connector 145 connected to the first port 131 on the first connector 140. Figure 2 is an example of a connector-to-connector ("conn-to-conn") relationship.

Figure 3 is a drawing of a typical topological star segment 301 for representing the connectivity of nodes on the network 110. In Figure 3, the first host 121 is connected to the first port 131 of the first connector 140. The star segment 301 comprises the first host 121 connected to the first port 131 of the first connector 140. Figure 3 is an example of a connector-to-host ("conn-to-host") relationship.

Figure 4 is a drawing of another typical topological star segment 301 for representing the connectivity of nodes on the network 110. In addition to the connections described with respect to Figure 3, a third host 123 is connected to a third port 133 of the first connector 140 and a fourth host 124 is connected to a fourth port 134 of the first connector 140. In Figure 4, the star segment 301 comprises the first host 121 connected to the first port 131 of the first connector 140, the third host 123 connected to the third port 133 of the first connector 140, and the fourth host 124 connected to the fourth port 134 of the first connector 140. Thus, the star segment 301 comprises, on a given connector, at least one port, wherein one and only one host is connected to that port, and that host. In the more general case, the star segment 301 comprises, on a given connector, all ports having one and only one host connected to each port, and those connected hosts. Since the segments, or links, drawn using the topological methods of Figure 4 resemble a star, they are referred to as star segments.

For illustrative purposes, nodes in the figures described above and in subsequent figures are shown as individual electronic devices or ports on connectors. Also, in the figures the nodes are

represented as terminals. However, they could also be workstations, personal computers, printers, scanners, or any other electronic device that can be connected to networks 110.

Figure 5 is a drawing of the connectivity of an example network system. In Figure 5, first, third, and fourth hosts 121, 123, 124 are connected via the network 110 to first, third, and fourth ports 131, 133, 134 respectively, wherein the first, third, and fourth ports 131, 133, 134 are located on the first connector 140.

The first, third and fourth hosts 121, 123, 124 are singly-heard hosts connected to separate ports 131, 133, 134 of a common connector 140 – the first connector 140. The fifth and sixth hosts 125, 126 are singly-heard hosts connected to the third and fourth connectors 142, 143. The seventh and eighth hosts 127, 128 are multi-heard hosts connected to the same port 139 of the fifth connector 144. The multi-heard hosts 127, 128 illustrate a shared media segment 180, also referred to as a bus 180.

The second, third, fourth, and fifth connectors 141, 142, 143, 144 are interconnected and illustrate a switch mesh 181. Each of the connectors in the switch mesh 181 is connected to each other, either directly or indirectly, to create a fully meshed connection. In the mesh, traffic may be dynamically routed to create an efficient flow.

Figure 5 also shows an example of a port aggregation 182, also referred to as trunking 182. The first connector 140 is connected via the network 110 to the second connector 141 by two direct links, each of which is connected to different ports on the connectors. One link is connected to the sixth port 136 of the first connector 140 and to the seventh port of the second connector 137. The other link is connected to fifth port 135 of the first connector 140 and to the eighth port 138 of the second connector 141. In this example, two connectors illustrate the multiple connectivity between nodes. Depending upon the device specifications, devices such as connectors may be connected via any number of connectors. As explained herein, the system resolves multiple connectivity problems by tracking port information for each connection.

Figure 6 is a drawing of the connectivity of a portion of a network having three connectors 171, 172, 173. A first host 151 is connected directly to the first port 161 of the first connector 171 and the second host 152 is connected to a sixth port 166 of the third connector 173. The second

port 162 of the first connector 171 is connected directly to the third port 163 of the second, or intermediate, connector 172. The fourth port 164 of the intermediate connector 172 is connected directly to the fifth port 165 of the third connector 173.

Figure 7 shows a block diagram of the system. Figure 8 shows a flow chart of the method used by the system to retrieve and update the topology of the network. A tuple manager 300, also referred to as a data miner 300, gathers 902 data from network nodes and builds 904 tuples to update the current topology. The topology database "topodb" 350 stores the current topology for use by the system. The "neighbor data" database 310 stores new tuple data retrieved by the tuple manager 300. The connection calculator 320 processes the data in the neighbor data database 310 to determine the new network topology. The connection calculator 320 reduces 906 the tuple data and sends it to the reduced topology relationships database 330. The topology converter 340 then updates 908 the topology database 350 based on the new tuples sent to the reduced topology relationships database 330 by the connection calculator 320.

Figure 9 shows a flow chart of one operation of the tuple manager 300, as described generally by the data gathering 902 and tuple building 904 steps of the method shown in Figure 8. The tuple manager 300 receives 910 a signal to gather tuple data. The tuple manager 300 then retrieves 912 node information of the current topology stored in the topology database 350. This information tells the tuple manager 300 which devices or nodes are believed to exist in the system based on the nodes that were detected during a previous query. The tuple manager 300 then queries 914 the known nodes to gather the desired information. For example, the connectors may maintain forwarding tables that store connectivity data used to perform the connectors' ordinary functions, such as switching. Other devices may allow the system to perform queries to gather information about the flow of network traffic. This data identifies the devices heard by a connector and the port on which the device was heard. The tuple manager 300 gathers this data by accessing forwarding tables and other information sources for the nodes to determine such information as their physical address, interface information, and the port from which they "hear" other devices. Based on this information, the tuple manager 300 builds 916 tuples and stores 918 them in the "neighbor data" database 310. Some nodes may have incomplete information. In this case, the partial

information is assembled into a tuple and may be used as a "hint" to determine its connectivity later, based on other connections. The tuple manager 300 may also gather 920 additional information about the network or about particular nodes as needed. For example, the connection calculator 320 may require additional node information and may signal the tuple manager 300 to gather that information.

After the data is gathered and the tuples are stored in the neighbor database 310, the connection calculator 320 processes the tuples to reduce them to relationships in the topology. Figure 10 shows a flow chart of the process of the connection calculator 320, as shown generally in the reduction step 906 of the method shown in Figure 8. The connection calculator 320 performs a first weeding phase 922 to identify singly-heard hosts to distinguish them from multi-heard hosts. Singly-heard hosts refer to host devices connected directly to a connector. The connection calculator 320 then performs an infrastructure-building phase 924 to remove redundant connector-to-connector links and to complete the details for partial tuples that are missing information. Then, the connection calculator 320 performs a second weeding phase 926 to resolve conflicting reports of singly-heard hosts. The connection calculator 320 then performs a noise reduction phase 928 to remove redundant neighbor information for connector-to-host links. If clarification of device connectivity is required, the connection calculator 320 performs a "look for" phase 930 to ask the tuple manager 300 to gather additional data. The tuple data is then consolidated 932 into segment and network containment relationships. The connection calculator 320 may also tag redundant tuples to indicate their relevance to actual connectivity. These redundant tuples may still provide hints to connectivity of other tuples. As part of the consolidation phase 932, the connection calculator 320 creates new n-ary tuples (tuples having references to three or more tucos) for shared media segments.

Figure 11 is a flow chart of the connection calculator's first weeding process 922 for distinguishing singly-heard hosts. The purpose of the first weeding process 922 is to identify the direct connections between connectors and hosts; that is, those tuples having a first tuco that is a connector and a second tuco that is a host. The connection calculator 320 looks through the tuple list in the neighbor database 310, and for each tuple 402, the connection calculator 320 determines

404 whether the tuple is a connector-to-host (conn-to-host) link tuple. If it is not a conn-to-host link, the connection calculator 320 concludes 418 that it is a conn-to-conn link and processes 402 the next tuple. If the tuple is a conn-to-host link tuple, then the connection calculator 320 determines 406 whether the connector hears only this particular host on the port identified in the tuple. If the connector hears other hosts on this port, then the tuple is classified 416 as a multi-heard host link (mhhl) tuple.

If the connector hears only the one host on the port – that is, if the host is a singly-heard host – then the connection calculator 320 determines 408 whether the host is heard singly by any other connectors. If no other connectors hear the host as a singly-heard host, then the tuple is classified as a singly-heard host link (shhl) tuple 412 and other tuples for this host are classified 414 as extra host links (ehl). Another tuple for this host may be, for example, an intermediate connector connected indirectly to a host. For example, Figure 6 shows three connectors 171, 172, 173 the first connector is connected directly to the first host 151. This connection therefore forms an shhl tuple. The intermediate connector 172 is indirectly connected to the first host 151. The tuple data indicates that the intermediate connector 172 is indirectly connected to the host and hears the host from a particular port. An extra host links tuple is created so that this data may be used later in conjunction with other extra host links tuples from devices across the network, to verify connectivity by providing hints about connections.

The first weeding process also attempts to identify conflicts. If other connectors hear the host as a singly-heard host, then a conflict arises and the tuple is classified 410 as a singly-heard conflict link (shcl) tuple to be resolved later. This conflict may arise, for example, if a host has been moved within the network, in which case the forwarding table data may no longer be valid. Certain connectors previously connected directly to the host may still indicate that the moved host is connected. When all tuples have been processed 402 to identify singly-heard host links, the first weeding phase 922 is complete.

Figures 12a-d show a flow chart of the infrastructure building phase 924 of the connection calculator 320. The purpose of the infrastructure building phase 924 is to determine how the connectors are set up in the network. The first part of the infrastructure building phase 924

manufactures tuples based on the list of singly-heard host link tuples identified in the first weeding phase 922. The purpose is to identify the relationship between the connectors in the extra host links tuples and the connectors directly connected to the singly-heard hosts. For each singly-heard host link 420, the connection calculator 320 processes 422 each extra host link that refers to the host. In the illustration of Figure 6, a conn-to-conn link tuple would represent the connection between the first connector 171 and the intermediate connector 172. An extra host link tuple would represent the indirect connection between the intermediate connector 172 and the first host 151. The conn-to-conn link tuple between the first connector 171 and the intermediate connector 172 is an example of an ehlConn-to-shhlConn tuple. If a conn-to-conn link tuple exists 424 for the extra host link connector to the singly-heard host link connector (ehlConn-to-shhlConn), then the connection calculator 320 updates 428 the tuple if it is incomplete. It is possible that the tuple data may be incomplete and a conn-to-conn link may not exist. In that case, a conn-to-conn tuple does not exist for the ehlConn-to-shhlConn, then such a tuple is created 426.

After processing extra host links for singly-heard host links, the connection calculator 320 considers 430 each connector (referred to as conn1) in the tuples to determine the relationship between connectors. As illustrated in Figure 6, a single connector may be connected directly and indirectly to multiple other connectors. In Figure 6, the first connector 151 is connected to the intermediate connector 171 directly and also to the third connector 173 indirectly. The third connector 173 hears the first host 151 on the same part 165 that it hears the first connector 171 and the intermediate connector 172. The infrastructure building phase 924 tries to determine the relationship between other connectors heard on the same port of conn1. In a series of interconnected connectors, the connector on one end may not hear a connector on another end, but it may hear intermediate connectors, that in turn hear their own intermediate connectors. Tuples are created to represent the interconnection of conn-to-conn relationships. Based on this data, the connection calculator 320 can make inferences regarding the overall connection between connectors.

For every conn1, the connection calculator 320 considers 432 every other connector (conn2) to determine whether a conn1-to-conn2 tuple exists. If conn1-to-conn2 does not exist,

then the connection calculator 320 considers 436 every other conn-to-conn tuple containing conn2. The other connector on this tuple may be referred to as conn3. If conn2 hears conn3 on a unique port 438 and if conn1 also hears conn3 440, then the connection calculator 320 creates 442 a tuple for conn1-to-conn2 in the connector-to-connector links tuple list.

After processing all of the conn1 tuples, the connection calculator 320 processes 444 each conn1-to-conn2 links tuple to ensure that they have complete port data. For each incomplete tuple 446, the connection calculator 320 looks 448 for a different tuple involving conn1 in the extra host links tupleson a different port. If a different tuple is found 450, then the connection calculator 320 determines 452 whether conn2 also hears the host. If conn2 does hear the host, then the connection calculator 320 completes the missing port data for conn2. If conn2 does not also hear the host 452, then the connection calculator 320 continues looking 448 through different tuples involving conn1 in extra host links on different ports.

After attempting to complete the missing data in each of the conn-to-conn links tuples, the connection calculator 320 processes 456 each conn-to-conn links tuple. The purpose of this sub-phase is to attempt to disprove invalid conn-to-conn links. The connection calculator 320 considers 458 conn1 and conn2 of each conn-to-conn links tuple. Every other connector in conn-to-conn links may be referred to as testconn. For each testconn 460, the connection calculator 320 determines 462 whether the testconn hears conn1 and conn2 on different groups/ports. If testconn hears conn1 and conn2 on different ports, then the tuple is moved to extraconnlinks (ecl) 464. Otherwise, the connection calculator 320 continues processing 460 the remaining testconns.

Figure 13 shows a flow chart of the second weeding phase 926. The purpose of the second weeding phase 926 is to attempt to resolve conflicts involving singly-heard hosts identified in the first weeding phase 922. In the situation described herein in which more than one connector reports that a host is singly-heard, the second weeding phase 926 reviews the tuples created during the infrastructure-building phase 924 involving the connector and host in question and attempts to disprove the reported conflict. The connection calculator 320 processes 466 each singleConflictLinks (scl) tuple (sometimes referred to as the search tuple) and considers 468 conn1 and host1 of the tuple. For each extra host links tuple containing host1 470, the connection

calculator 320 considers 472 conn2 of the tuple. If there is a tuple in conn-to-conn links for conn2 and conn1 474, and if there is a conn2-to-conn1 tuple in the extra host links tuples 476, and if the port is the same for conn2 hearing conn1 and host1 478, then the search tuple is moved 480 into the singly heard host links and other tuples containing host1 are removed 482 from the singleConflictLinks.

Figure 14 shows a flow chart of the noise reduction phase 928. The purpose of the noise reduction phase 928 is to handle those connections in which a connector is not directly connected to a host or to another connector. For example, networking technology may employ shared media connections between connectors, rather than dedicated media connectors. With a shared media connection, the entries in the forwarding tables for connectors attached to the shared media connection will include every node accessing the shared media connection and may not present a useful or accurate representation of the nodal connection. For example, if the network configuration in Figure 6 used a shared media connection between the first connector 171 and the intermediate connector 172, then the first connector is not really connected directly to the intermediate connector because other devices (not shown in Figure 6) may also use the shared media connection. These other devices may include web servers, other connectors, other subnetworks, etc. Tuples will be created for the connectors 171, 172 on opposing ends of the shared media. In this situation, it is inefficient to maintain point-to-point binary tuples for every connection. The noise reduction phase 928 disproves invalid tuples created by the shared media connections.

For each multi-heard host links (mhhl) tuple, also referred to as multiHeardLinks (mhl) tuples (sometimes referred to as the search tuple) 484, conn1 and host1 are considered 486. For each extra host links tuple containing host1 488, conn2 is considered 490. If there is a tuple in conn-to-conn links for conn2 and conn1 492, and if there is a conn2-to-host1 tuple in extraHostLinks 494, and if the group/port for conn2 hearing conn1 and host1 is different 496, then the search tuple is moved 498 to extraHostLinks.

Figure 15 shows a flow chart for the "look for" phase 930. The purpose of this phase is to complete missing data for mhhl tuples. There may exist connections on the network that have incomplete tuple data. For example, the network may simply have no traffic between certain nodes,

in which case data might not be stored in forwarding tables. In another example, a forwarding table may not have sufficient room to store all of the required information and might delete data on a FIFO basis. In the look for phase 930, the connection calculator 320 instructs the tuple manager 300 to query specific nodes to retrieve the missing data. Data that was not stored in a forwarding table on the first interrogation may be present on a subsequent query. For each mhhl tuple 500, the connection calculator 320 considers 502 conn1 and host1. If the conn1 group/port is already in an "alreadyDidLookfors" list, then a list is created 508 for all connectors in conn-to-conn links that are heard by conn1 on the same group/port as host1. For each connector (conn2) in the list 510, the connection calculator 320 determines 512 whether there is a conn2-to-host1 tuple in the mhhl tuples. If there is not such a tuple, then the connection calculator 320 initiates a look-for for conn2-to-host1 via the tuple manager 300. When each connector in the list has been processed 510, the conn1 group/port tuco is added 516 to an alreadyDidLookfors list. As an additional portion of the look for phase 930 (not shown in figures) the system may ask a user to verify or clarify information about connectivity. For example, the system may show the user the perceived connectivity or the unresolved connectivity issues and request the user to add information as appropriate.

The connection calculator 330 process described above collects the tuple information from the tuple manager 300, builds tuples new tuples and removes redundant or unnecessary tuples to produce the new topology. This topology may have incomplete tuples possibly resulting from extraneous information that the connection calculator 330 could not disprove. To refine the new topology, the connection calculator 330 can request the tuple manager 300 to obtain additional information about particular nodes or it may also request a user to refine the topology by adding or removing tuples. Using the process of the connection calculator 330, tuples marked as non-essential may be removed from the new topology to save space and to simply the topology. The connection calculator 330 is not confused by multiple connectivity situations such as port aggregation 182 or switch meshing 181 as shown in Figure 5, because the tuples represent point-to-point, or neighbor-to-neighbor, connectivity showing each connection in the network. This point-to-point connectivity concept also helps enable the system to avoid difficulties that occur in

systems that track higher levels of abstraction, such as layer 3 connectivity. Also, the tuples may contain only selected information to minimize the storage space required for the topology.

Figures 16a-b show a flow chart of the consolidation phase 932. The purpose of this phase is to consolidate the tuples that involve shared media connections. After the noise reduction phase 928, a considerable number of tuples involving shared media may remain. Rather than maintain a binary tuple for each of the connections, an n-ary tuple is created for the link using a tuco for each connector and each host connected thereto. For each mhhl tuple 518, conn1 and host1 are considered 520. If there are more conn1 group/port tuples in multiHeardLinks, and if are not any n-ary multiHeardSegments (mhs) tuples 524, then an mhs tuple is created 526. If host1 is not already in this particular mhs tuple 528, then conn2 of the tuple is considered 534. If there is a conn1-to-conn2 conn-to-connLinks tuple on the same port as conn1-to-host1 536, then all multiHeardLinks tuples for conn2-to-host1 with the same conn2 group/port as the conn1-to-conn2 are added 538 to the current mhs tuple.

After processing each mhhl tuple 518, each singly-heard host links (shhl) tuple, also referred to as a singlyHeardLinks (shl) tuple, is considered 540. For each shhl tuple, the connector and host are considered 542. If there is no existing singlyHeardSegments (shs) tuple for the connector 544, then an shs tuple is created 546. The host tuco is then added to the shs 548.

Figure 17 shows a flow chart of the method used by the topology converter 340, as described generally by the topology update step 908 of the method shown in Figure 8. The topology converter 340 converts 934 the topology into tuple lists, also referred to as the "morph topo" phase 934. It then compares 936 the list from the topology currently stored in the topology database 350 with the new list generated by the connection calculator 320 and discards 936 identical tuples in what is also referred to as the "discard duplicates" phase 936. It then takes action 938 on the changes in the topology as determined by the changes in the tuple lists, in what is also referred to as the "identify different tuples" phase 938.

Figure 18a shows a flow chart for the "morph topo" phase 934. For each node in the topology 550, the topology converter 340 determines 552 whether the node is a connector. If the node is a connector, then for each connected interface (conniface) of the connector (conn1) 554,

the topology converter 340 determines 556 whether the conniface is connected to a star segment. If it is connected to a star segment, then for every other interface in the segment 558, the topology converter 340 determines 560 whether there is an existing shs tuple, referred to as the "topo tuple" for the segment. If there is no such tuple, then the topology converter 340 creates 562 a topo shs tuple. The tuco for the interface's host-to-topo shs is then added 564 to the topo shs tuple.

If the connector node is not connected to a star segment 556 and is connected to a bus segment 566, the topology converter 340 determines 568 whether there is an existing mhs tuple for conn1. If there is not an existing mhs tuple for conn1, then a topo mhs tuple is created 570. A tuco is added 572 for the host to the mhs tuple.

If the connector node is not connected to either a star segment 556 or to a bus segment 566, then the topology converter knows that it is connected to another connector (conn2). If such a connector does not already have an existing connLinks tuple for conn1 and conn2 576, then a connLinks tuple is created 578. After processing the bus segment, star segment, and conn-to-conn segment, for each conniface 554, the topology converter 340 proceeds to the next node 550.

Figure 18b shows a continuation of the flow chart of Figure 18a showing the steps of the method when the topology converter 340 determines that the node is not a connector 552. If the node is in the default segment, then an "unheardOfLinks" tuple is created 582 and the topology converter proceeds to the next node 550. If the node is not in the default segment 580, then the topology converter 340 determines whether the node is in a star segment 584. If the node is in a star segment, then if there is not already an shs tuple, the topology converter 340 creates 588 an shs tuple. The tuco for the node is then added 590 to the shs tuple, and the topology converter 340 proceeds to the next node 550.

If the node is not in a star segment, then the topology converter 340 knows that it is in the bus segment. If there is not already an mhs tuple for the node, 594, then the topology converter 340 creates 596 an mhs tuple. The tuco for the node is then added 598 to the mhs tuple, and the topology converter proceeds to the next node 550.

Figure 19 shows a flow chart for the discard duplicates phase 936 of the topology converter 340. For each tuple in the new tuples (nt) 600, the topology converter looks for 602 an

exact match in the current tuples stored in the topodb. If an exact match is found 604, then the new tuple is marked 606 as "no change" indicating that this is an identical tuple.

Figures 20a-d show a flow chart for the identify different tuples phase 938. The system looks through each tuple in the new SinglyHeardSegments (newSHS) tuple list 608 and tries to identify and fix 610 swapped ports on connectors. Swapped ports are identified by considering those segment tuples in both the new topology and the existing topology that differ only by the port specification in the tuco. Each tuple that is fixed as a swapped port is marked 612 as "handled." The system also looks through each tuple in the new multiHeardSegments tuple list (newMHS) 614 and tries to identify and fix 616 swapped ports on connectors. Each tuple that is fixed as a swapped port is marked 618 as "handled."

The system then processes 620 each unmarked tuple in the newSHL tuples. Four cases are possible for the host of the newSHL tuples. The host of the newSHL can be found in the current singlyHeardLinks (curSHL) 622, the current multiHeardLinks (curMHL) 630, the current connLinks (curCL) 638, or the current UnheardOfLinks (curUOL) 642. If the host of a newSHL tuple is found 622 in the current SinglyHeardLinks (curSHL) tuples, then the system determines 624 if there is a matching connector tuco between the newSHL tuples and the curSHL tuples. If there is a matching tuco, then the system changes 626 the host connection attribute. If there is not a matching tuco, then the host connection is moved 628 in the topology.

If the host is found in the curMHL tuples 630, then the system determines 632 whether there is a matching connector tuco between the newSHL tuples and the curSHL tuples. If there is a matching connector, then the segment type of connection is changed 634. If there is not a matching connector, then the host connection is moved 636 in the topology. If the host is found in the curCL tuples 638, then the host is moved 640 into a star segment of the connector. If it is found in the curUOL 642, then the host is moved 644 into the star segment of the connector.

Figure 20c shows another stage of the processing undertaken during the identify different tuples phase 938. For each unmarked tuple in the new multiHeardLinks tuples (newMHL) 946, four cases are possible for the host of the newMHL. The host of the newMHL may be found in the curSHL 648, the curMHL 656, the curCL 664, or the curUOL 668. If the host is found in the

curSHL 648, then the system determines 650 whether there is a matching connector tuco between the newMHL and the curMHL. If there is a matching tuco, then the segment type of connection is changed 652. If there is not a matching tuco, then the host connection is moved 654 in the topology.

If the host is found in the curMHL tuples 656, then the system determines 658 whether there is a matching connector tuco in both the curMHL tuples and the newMHL tuples. If there is a matching connector tuco, then the host connection attribute is changed 660. If there is not a matching tuco, then the host connection is moved 662 in the topology. If the host is found in the curCL tuples 664, then the host is moved into a bus segment of a connector. If the host is found in the curUOL tuples 668, then the host connection is moved 670 in the topology.

Figure 20d shows another portion of the identify different tuples phase 938. For each unmarked tuple in the newCL tuples 672, there are three possibilities for the connector. The connector of the unmarked tuple in newCL can be found in the curSHL or curMHL 674, in the curCL 678, or in the curUOL 682. If each connector is found in the curSHL or curMHL list 674, then the system creates 676 a new point-to-point segment for the connectors. If the connectors are found in the curCL 678, then the connection attributes of the connectors are changed 680. If each connector is found in the curUOL tuples 682, then the host connection is moved 684 in the topology.

Another part of the identify different tuples phase 938 is shown in blocks 686 and 688 of Figure 20d. For each unmarked tuple in the newUOL tuples 686, the system checks 688 the timer/configuration to determine whether the host/conn should move into the default segment from its current segment.

An advantage of the system is that it may be schedulable. The system may map network topology continuously, as done by existing systems, or it may be scheduled to run only at certain intervals, as desired by the user. A further advantage of the system is that it is capable of processing multiple connections between the same devices and of processing connection meshes, because it tracks each nodal connection independently, without limitations on the types of connections that are permitted to exist.

Although the present invention has been described with respect to particular embodiments thereof, variations are possible. The present invention may be embodied in specific forms without departing from the essential spirit or attributes thereof. It is desired that the embodiments described herein be considered in all respects illustrative and not restrictive and that reference be made to the appended claims for determining the scope of the invention.

Claims

1. In a network having interconnected nodes with data tuples that represent nodal connections, a method for mapping a network topology by identifying changes between an existing topology and a new topology, the method comprising:

converting an existing topology into a list of existing tuples that represent existing nodal connections;

receiving new tuples that represent new nodal connections; and

comparing the list of existing tuples with the new tuples to identify changes to the topology.

2. The method of claim 1, further comprising updating a topology database with a new topology.

3. The method of claim 1, further comprising taking action on the changes to the topology.

4. The method of claim 1, wherein the tuples include information about a host identifier, a connector interface, and a port specification.

5. The method of claim 1, wherein the step of comparing comprises identifying duplicate tuples that appear both in the list of existing tuples and in the new tuples, and maintaining a current status of the topology for these tuples.

6. The method of claim 1, wherein the step of comparing comprises identifying a swapped port condition on a connector.

7. The method of claim 1, wherein the step of comparing comprises searching for a host of a new singly-heard host link tuple or a new multi-heard host link tuple in the list of existing tuples.

8. A system for mapping a network topology by identifying changes between an existing topology and a new topology, based on changes to data tuples that represent nodal connections comprising:

a topology database that stores an existing topology of a network; and

a topology converter connected to the topology database that receives new tuples that represent new nodal connections; and compares the new tuples with the existing topology to identify changes in the network.

9. The system of claim 8, wherein the topology converter converts the existing topology into a list of existing tuples that represent existing nodal connections.

10. The system of claim 8, wherein the topology converter updates the topology database with a new topology based on the new tuples.

11. The system of claim 8, wherein the topology converter attempts to identify swapped ports on connectors.

12. The system of claim 8, wherein the topology converter identifies duplicate tuples that appear both in the list of existing tuples and in the new tuples, and maintains a current status of the topology for these tuples.

13. The system of claim 8, wherein the topology converter searches for a host of a new singly-heard host link tuple or a new multi-heard host link tuple in the list of existing tuples.

14. The system of claim 8, wherein the topology converter searches for a connector of a new conflict links tuple in the list of existing tuples.

15. A computer-readable medium having computer-executable instructions for performing a method for mapping a network topology by identifying changes between an existing topology and a new topology in a network having a interconnected nodes, the method comprising:

converting an existing topology into a list of existing tuples that represent existing nodal connections;

receiving new tuples that represent new nodal connections;

comparing the list of existing tuples with the new tuples to identify changes to the topology; and

updating a topology database with a new topology.

16. The method of claim 15, wherein a topology converter receives the new tuples from a connection calculator that calculates connections between nodes.

17. The method of claim 15, wherein the step of comparing comprises identifying duplicate tuples that appear both in the list of existing tuples and in the new tuples, and maintaining a current status of the topology for these tuples.

18. The method of claim 15, wherein the step of comparing comprises identifying a swapped port condition on a connector.

19. The method of claim 15, wherein the step of comparing comprises searching for a host of a new singly-heard host link tuple or a new multi-heard host link tuple in the list of existing tuples.

20. The method of claim 15, wherein the step of comparing comprises searching for a connector of a new conflict links tuple in the list of existing tuples.

Abstract

A method and system are disclosed for mapping the topology of a network having interconnected nodes by identifying changes in the network and updating a stored network topology based on the changes. The nodal connections are represented by data tuples that store information such as a host identifier, a connector interface, and a port specification for each connection. A topology database stores an existing topology of a network. A topology converter accesses the topology database and converts the existing topology into a list of current tuples. A connection calculator calculates tuples to represent connections in the new topology. The topology converter receives the new tuples, identifies changes to the topology, and updates the topology database using the new tuples. The topology converter identifies duplicate tuples that appear in both the new tuples and the existing tuples and marks the duplicate tuples to reflect that no change has occurred to these connections. The topology converter attempts to resolve swapped port conditions and searches for new singly-heard and multi-heard host link tuples in the list of existing tuples. The topology converter also searches for new conflict link tuples in the existing tuples. The topology converter updates the topology database with the new topology.

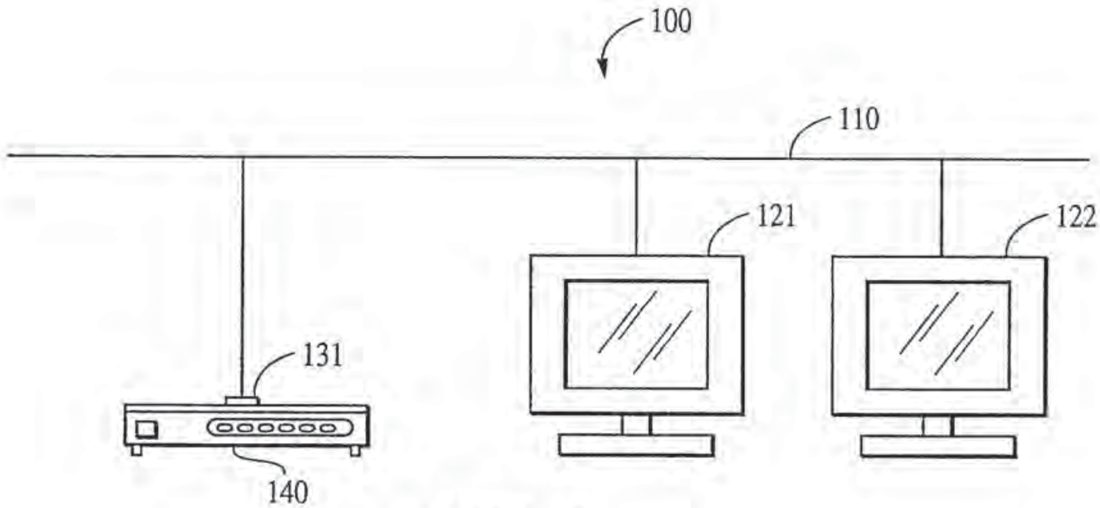


FIG. 1

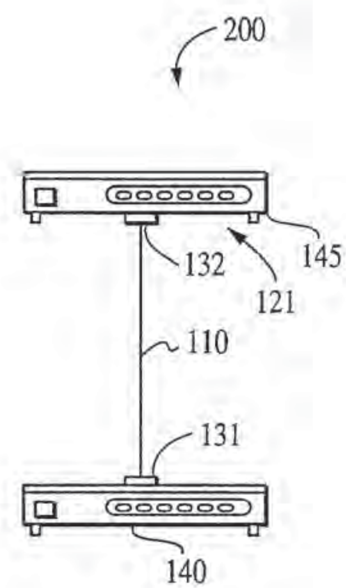


FIG. 2

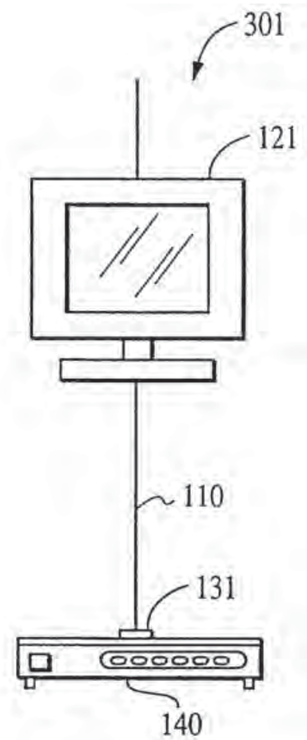


FIG. 3

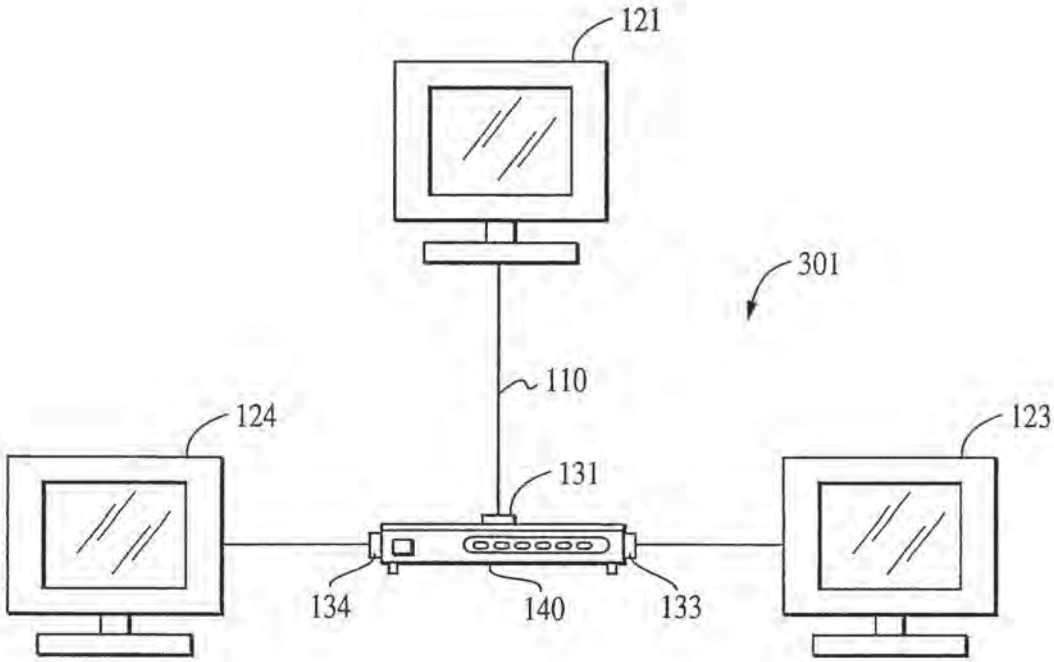


FIG. 4

FIG. 5

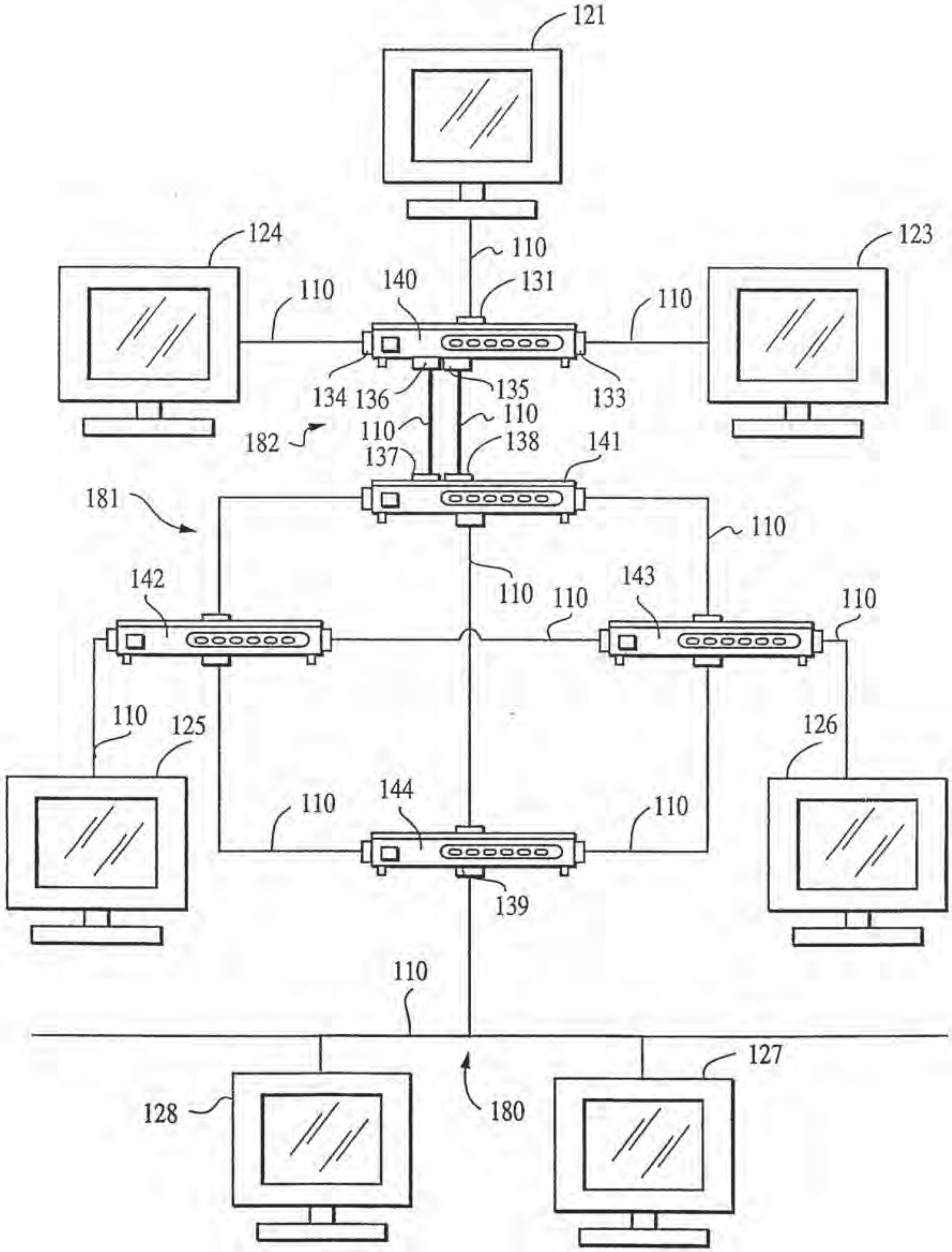
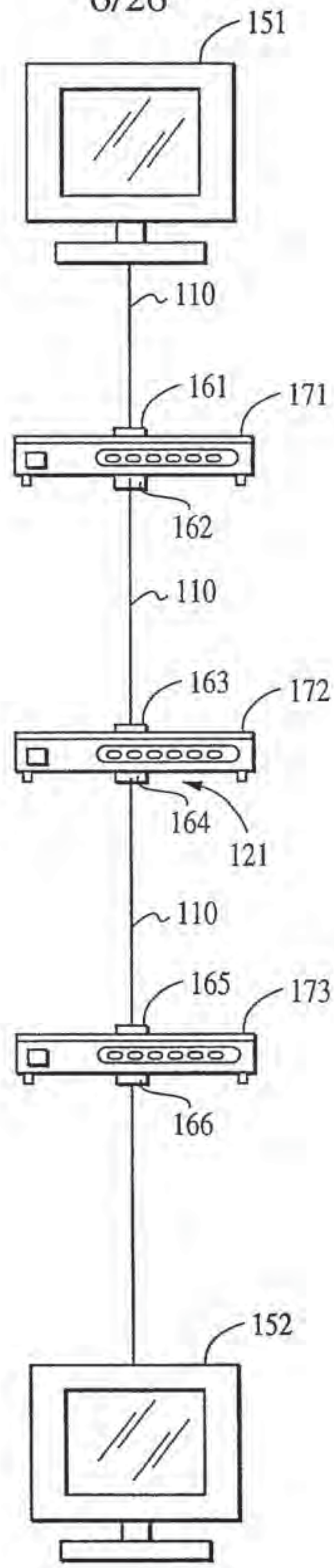

121
124
123
110
140
131
134 136
135
133
182
137
138
141
181
142
143
125
126
144
139
128
180
127

FIG. 6



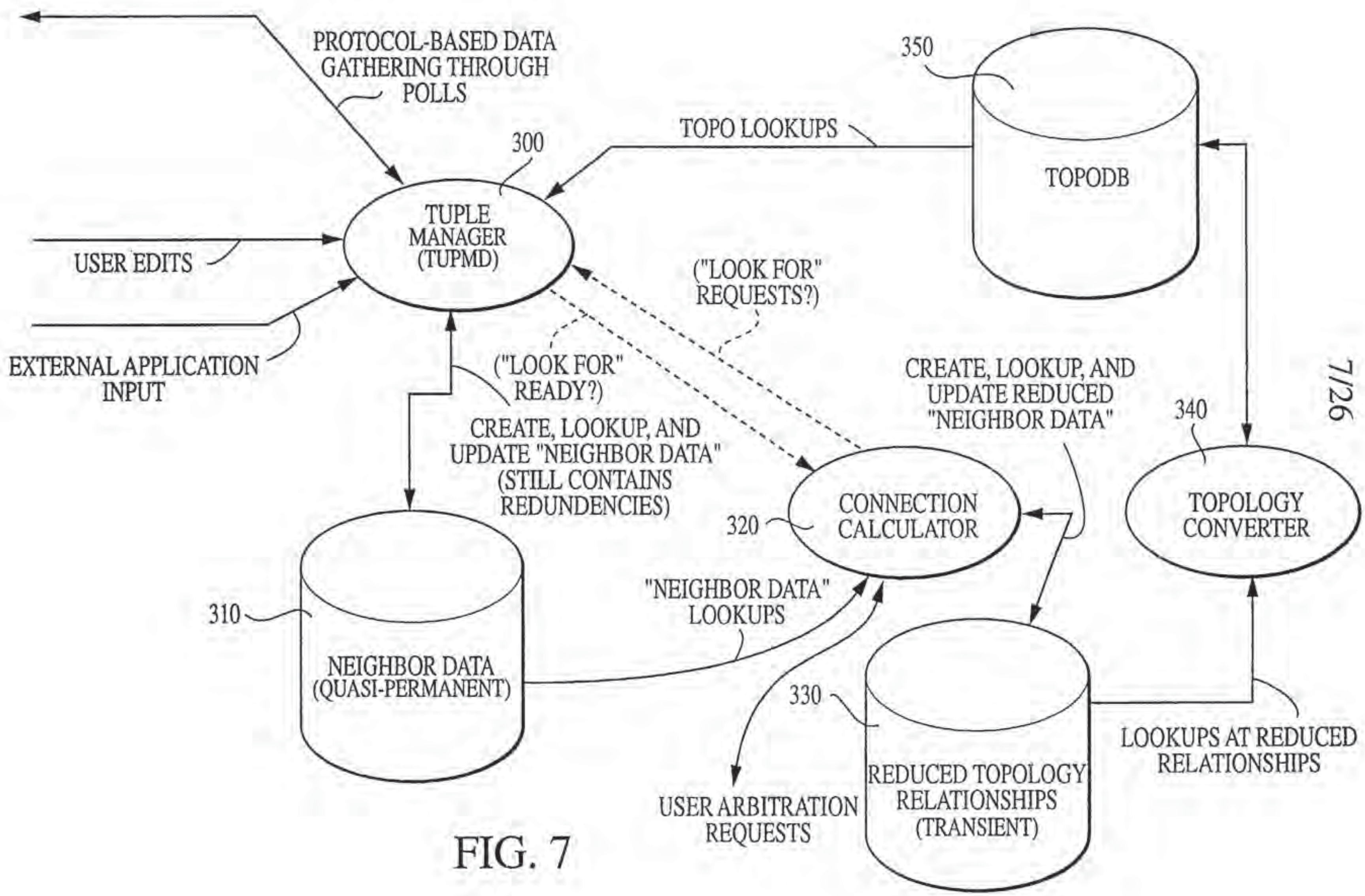

PROTOCOL-BASED DATA GATHERING THROUGH POLLS
USER EDITS
EXTERNAL APPLICATION INPUT
300
TUPLE MANAGER (TUPMD)
TOPO LOOKUPS
350
TOPODB
("LOOK FOR" REQUESTS?)
("LOOK FOR" READY?)
CREATE, LOOKUP, AND UPDATE "NEIGHBOR DATA" (STILL CONTAINS REDUNDENCIES)
CREATE, LOOKUP, AND UPDATE REDUCED "NEIGHBOR DATA"
340
TOPOLOGY CONVERTER
320
CONNECTION CALCULATOR
310
NEIGHBOR DATA (QUASI-PERMANENT)
"NEIGHBOR DATA" LOOKUPS
330
REDUCED TOPOLOGY RELATIONSHIPS (TRANSIENT)
LOOKUPS AT REDUCED RELATIONSHIPS
USER ARBITRATION REQUESTS

FIG. 7

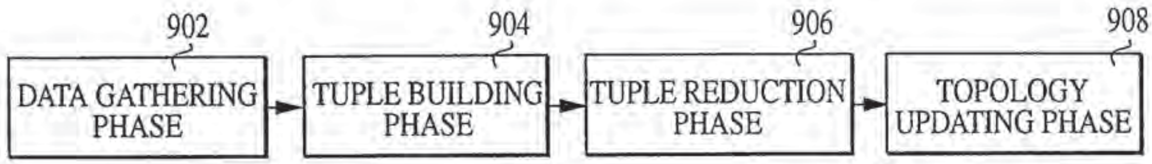


FIG. 8

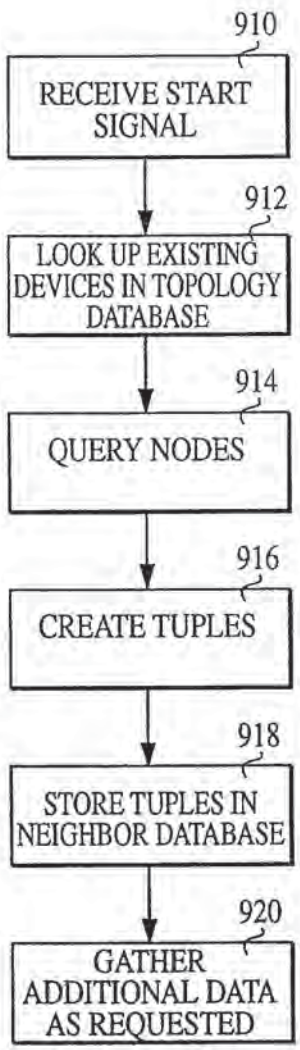


FIG. 9

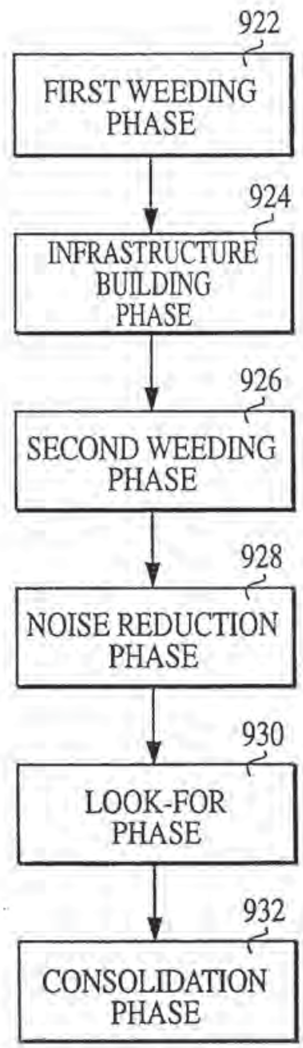


FIG. 10

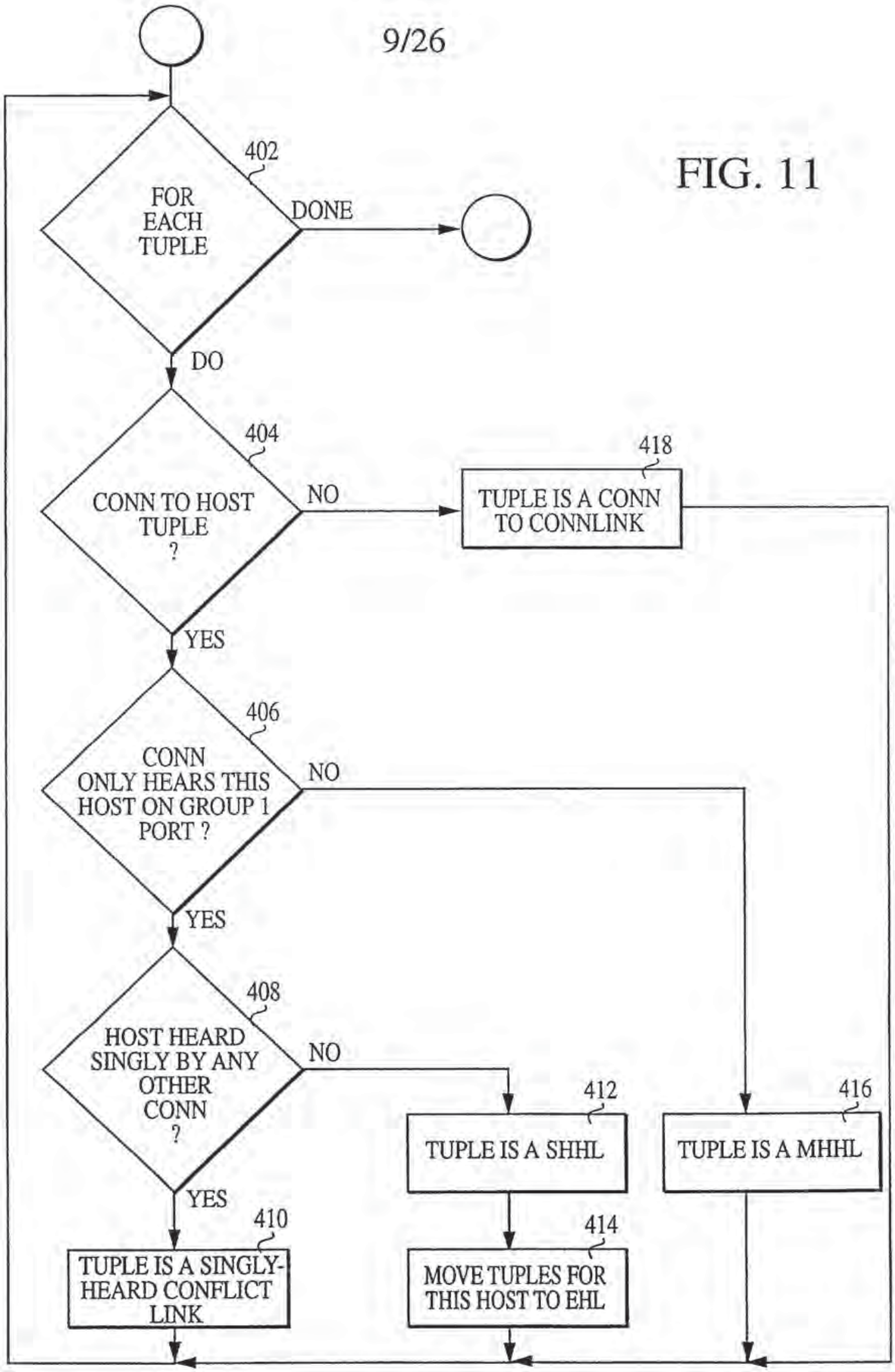

402
FOR EACH TUPLE
DONE
DO
404
CONN TO HOST TUPLE ?
NO
418
TUPLE IS A CONN TO CONNLINK
YES
406
CONN ONLY HEARS THIS HOST ON GROUP 1 PORT ?
NO
YES
408
HOST HEARD SINGLY BY ANY OTHER CONN ?
NO
412
TUPLE IS A SHHL
416
TUPLE IS A MHHL
YES
410
TUPLE IS A SINGLY-HEARD CONFLICT LINK
414
MOVE TUPLES FOR THIS HOST TO EHL

FIG. 11

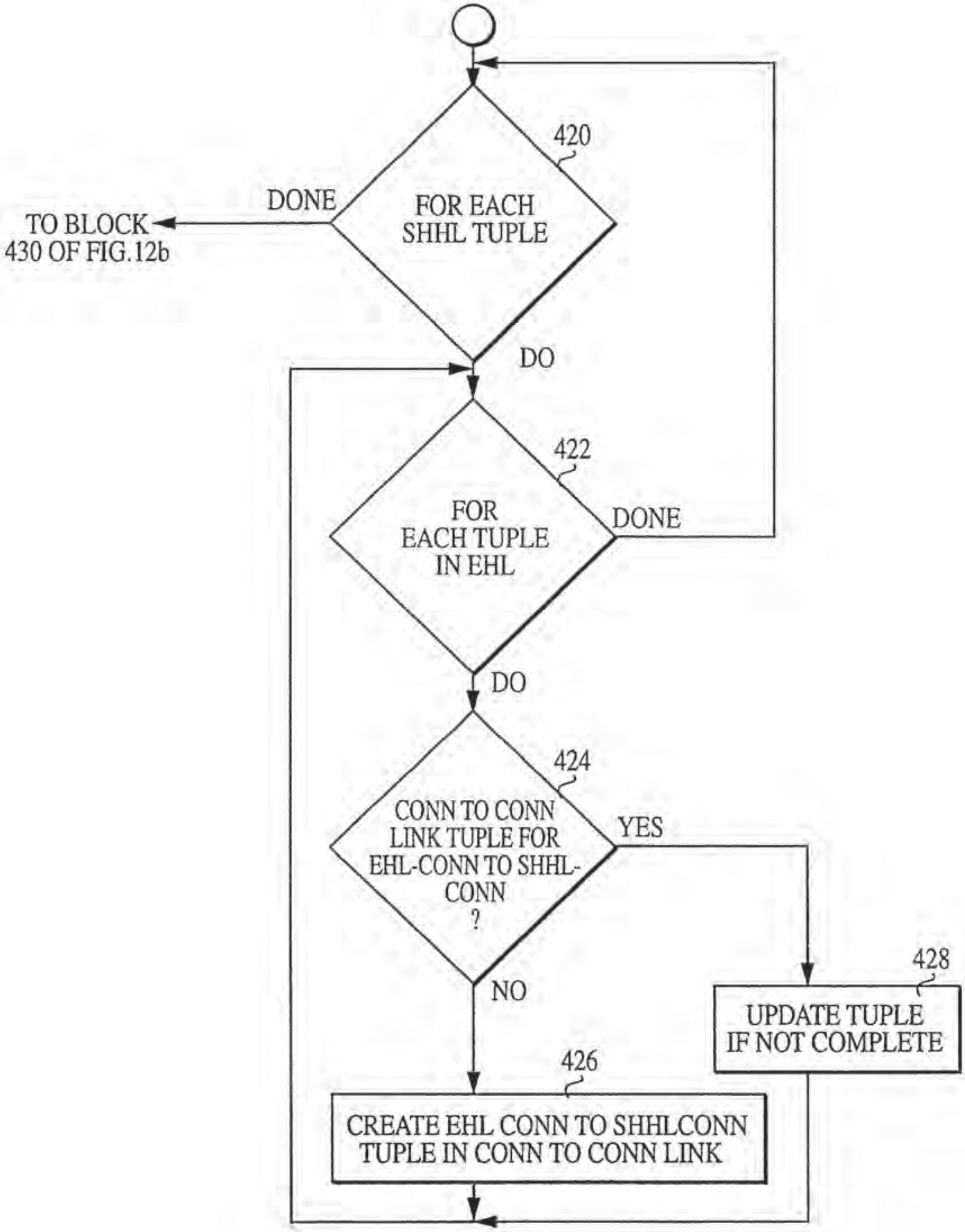

TO BLOCK
430 OF FIG.12b
DONE
FOR EACH
SHHL TUPLE
420
DO
FOR
EACH TUPLE
IN EHL
422
DONE
DO
CONN TO CONN
LINK TUPLE FOR
EHL-CONN TO SHHL-
CONN
?
424
YES
NO
UPDATE TUPLE
IF NOT COMPLETE
428
CREATE EHL CONN TO SHHLCONN
TUPLE IN CONN TO CONN LINK
426

FIG. 12a

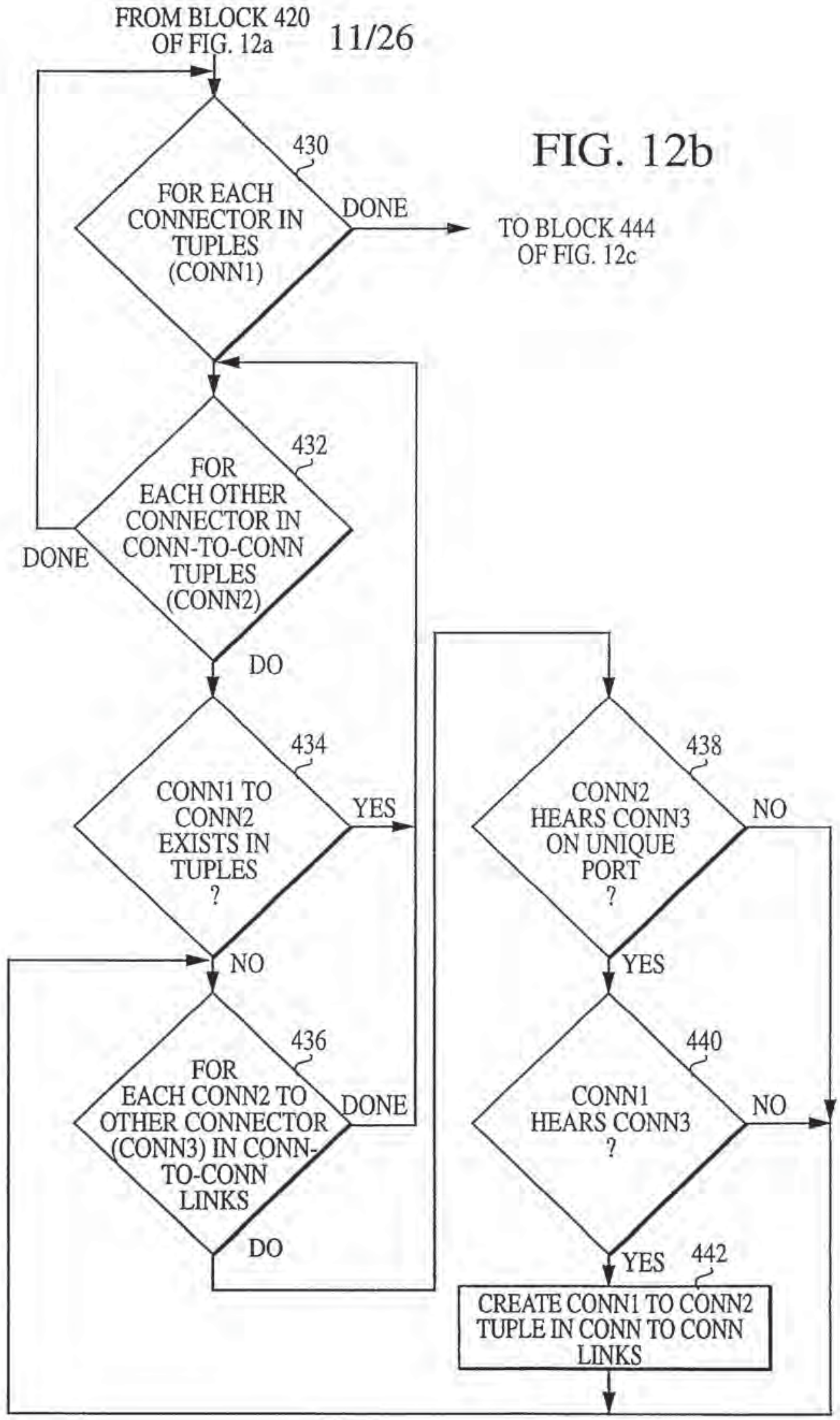

FROM BLOCK 420
OF FIG. 12a
430
FOR EACH CONNECTOR IN TUPLES (CONN1)
DONE
TO BLOCK 444
OF FIG. 12c
432
FOR EACH OTHER CONNECTOR IN CONN-TO-CONN TUPLES (CONN2)
DONE
DO
434
CONN1 TO CONN2 EXISTS IN TUPLES ?
YES
NO
436
FOR EACH CONN2 TO OTHER CONNECTOR (CONN3) IN CONN-TO-CONN LINKS
DONE
DO
438
CONN2 HEARS CONN3 ON UNIQUE PORT ?
NO
YES
440
CONN1 HEARS CONN3 ?
NO
YES
442
CREATE CONN1 TO CONN2 TUPLE IN CONN TO CONN LINKS

FIG. 12b

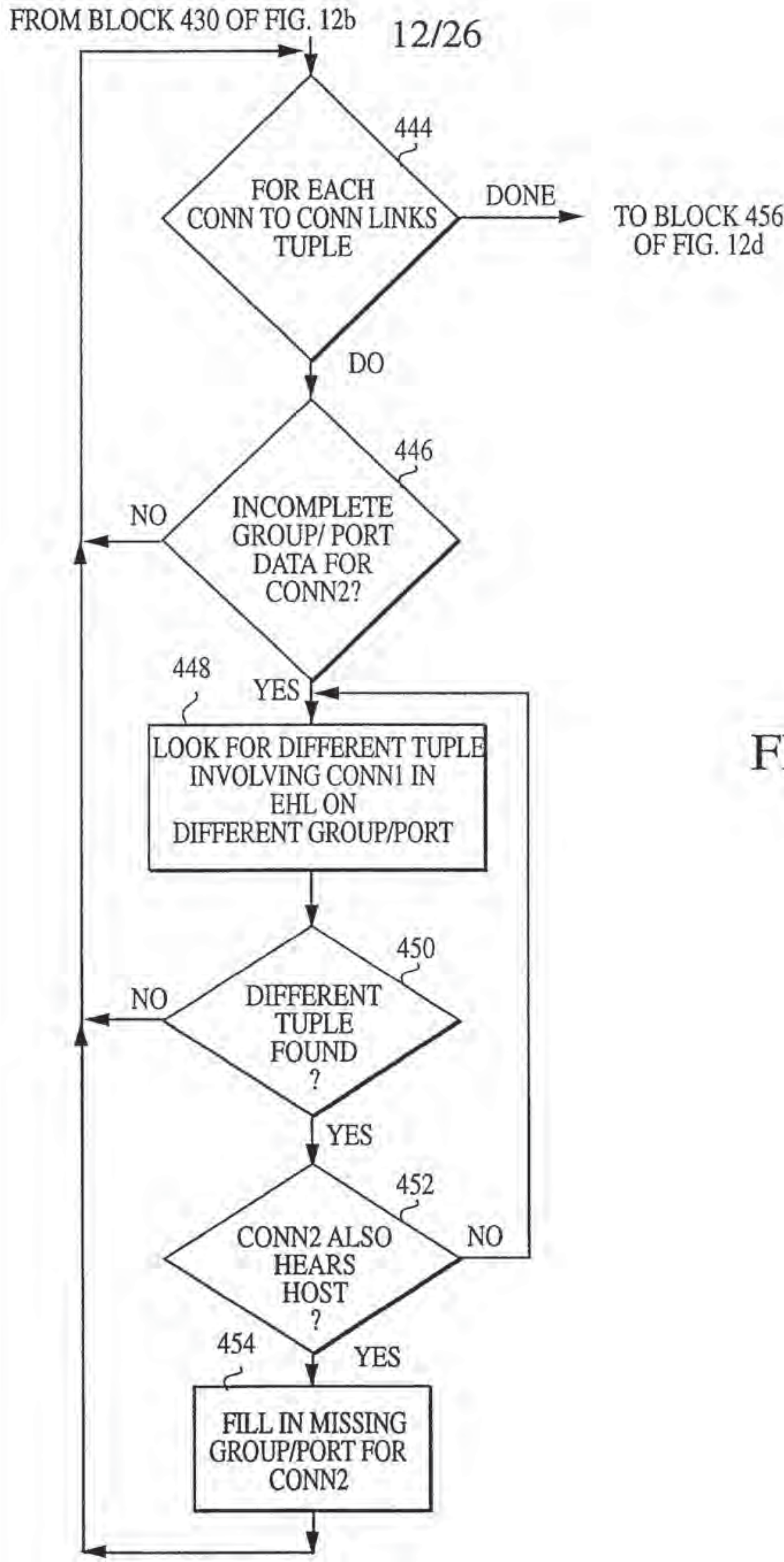

FROM BLOCK 430 OF FIG. 12b
444
FOR EACH CONN TO CONN LINKS TUPLE
DONE
TO BLOCK 456 OF FIG. 12d
DO
446
INCOMPLETE GROUP/ PORT DATA FOR CONN2?
NO
448
YES
LOOK FOR DIFFERENT TUPLE INVOLVING CONN1 IN EHL ON DIFFERENT GROUP/PORT
450
DIFFERENT TUPLE FOUND ?
NO
YES
452
CONN2 ALSO HEARS HOST ?
NO
454
YES
FILL IN MISSING GROUP/PORT FOR CONN2

FIG. 12c

FROM BLOCK 444 OF FIG. 12c

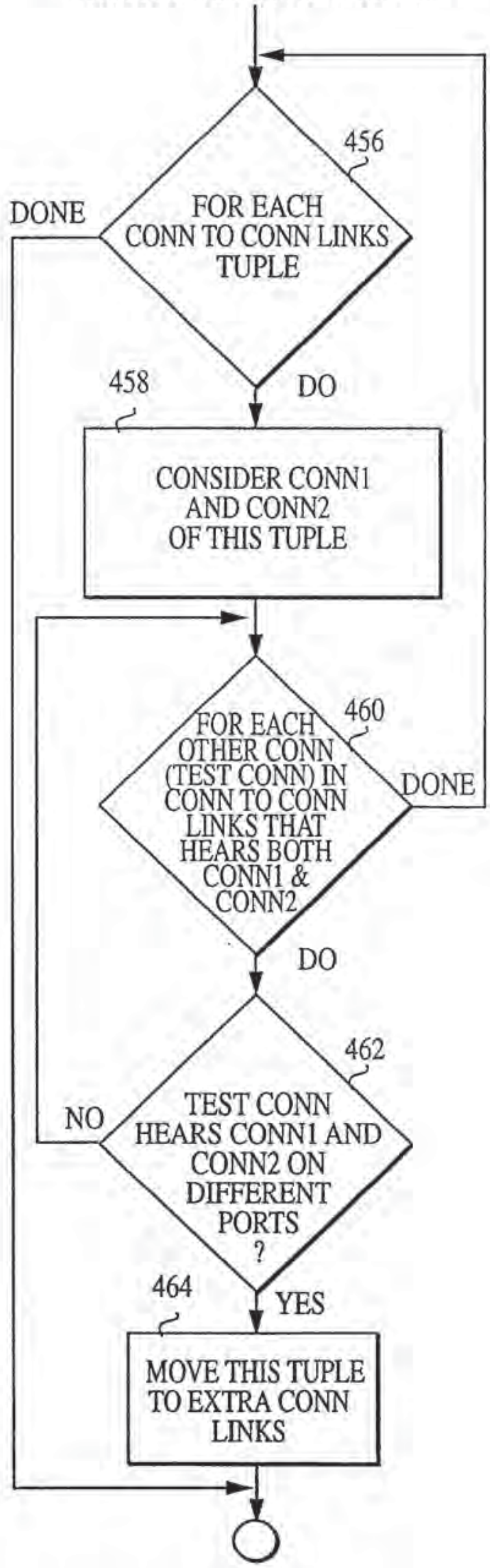


FIG. 12d

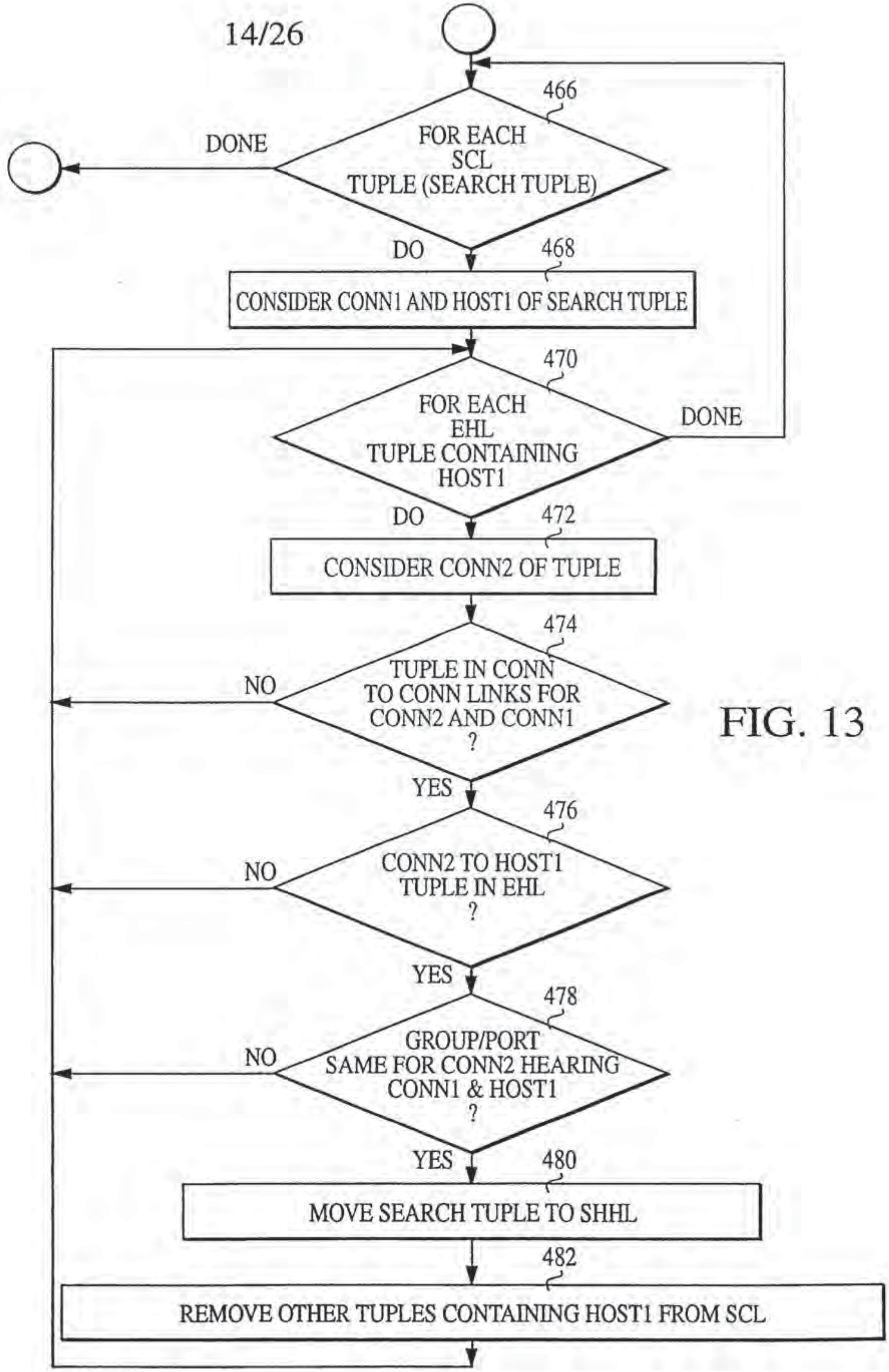
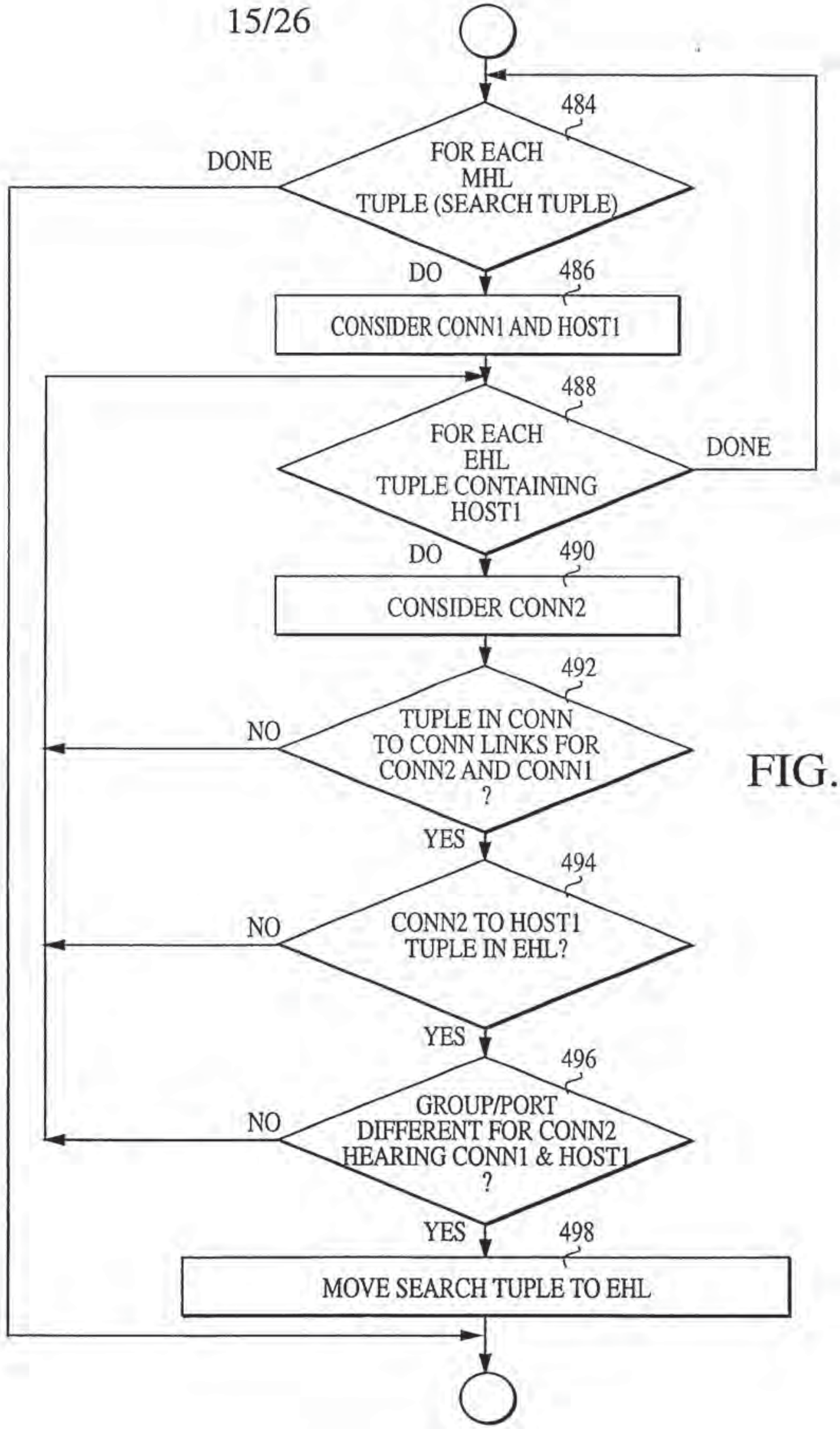

466
FOR EACH SCL TUPLE (SEARCH TUPLE)
DONE
DO
468
CONSIDER CONN1 AND HOST1 OF SEARCH TUPLE
470
FOR EACH EHL TUPLE CONTAINING HOST1
DONE
DO
472
CONSIDER CONN2 OF TUPLE
474
TUPLE IN CONN TO CONN LINKS FOR CONN2 AND CONN1 ?
NO
YES
476
CONN2 TO HOST1 TUPLE IN EHL ?
NO
YES
478
GROUP/PORT SAME FOR CONN2 HEARING CONN1 & HOST1 ?
NO
YES
480
MOVE SEARCH TUPLE TO SHHL
482
REMOVE OTHER TUPLES CONTAINING HOST1 FROM SCL

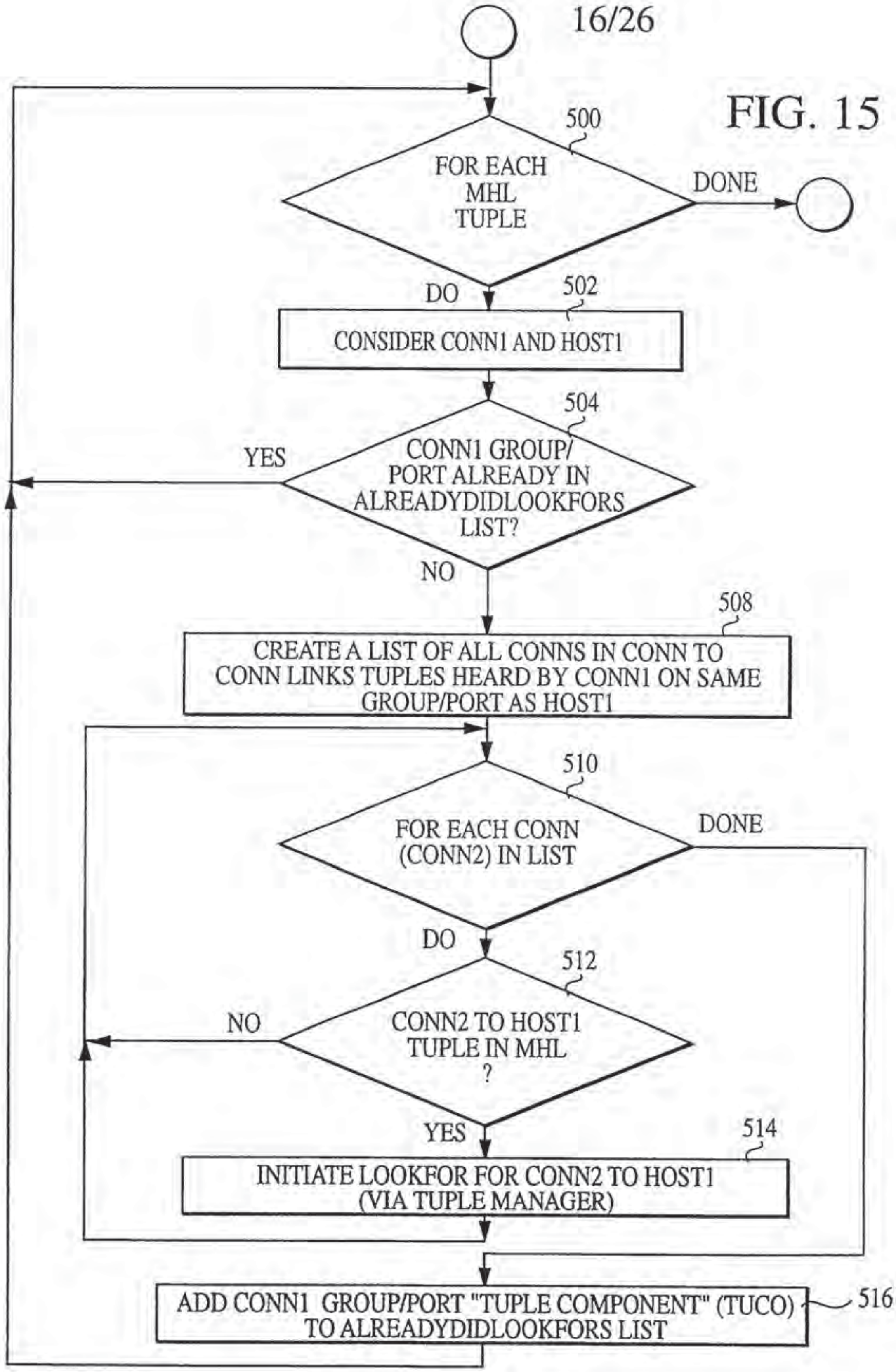
FIG. 13



484
FOR EACH MHL TUPLE (SEARCH TUPLE)
DONE
DO
486
CONSIDER CONN1 AND HOST1
488
FOR EACH EHL TUPLE CONTAINING HOST1
DONE
DO
490
CONSIDER CONN2
492
TUPLE IN CONN TO CONN LINKS FOR CONN2 AND CONN1 ?
NO
YES
494
CONN2 TO HOST1 TUPLE IN EHL?
NO
YES
496
GROUP/PORT DIFFERENT FOR CONN2 HEARING CONN1 & HOST1 ?
NO
YES
498
MOVE SEARCH TUPLE TO EHL

FIG. 14

FIG. 15


500
FOR EACH MHL TUPLE
DONE
DO
502
CONSIDER CONN1 AND HOST1
504
CONN1 GROUP/ PORT ALREADY IN ALREADYDIDLOOKFORS LIST?
YES
NO
508
CREATE A LIST OF ALL CONNS IN CONN TO CONN LINKS TUPLES HEARD BY CONN1 ON SAME GROUP/PORT AS HOST1
510
FOR EACH CONN (CONN2) IN LIST
DONE
DO
512
CONN2 TO HOST1 TUPLE IN MHL ?
NO
YES
514
INITIATE LOOKFOR FOR CONN2 TO HOST1 (VIA TUPLE MANAGER)
ADD CONN1 GROUP/PORT "TUPLE COMPONENT" (TUCO) TO ALREADYDIDLOOKFORS LIST
516

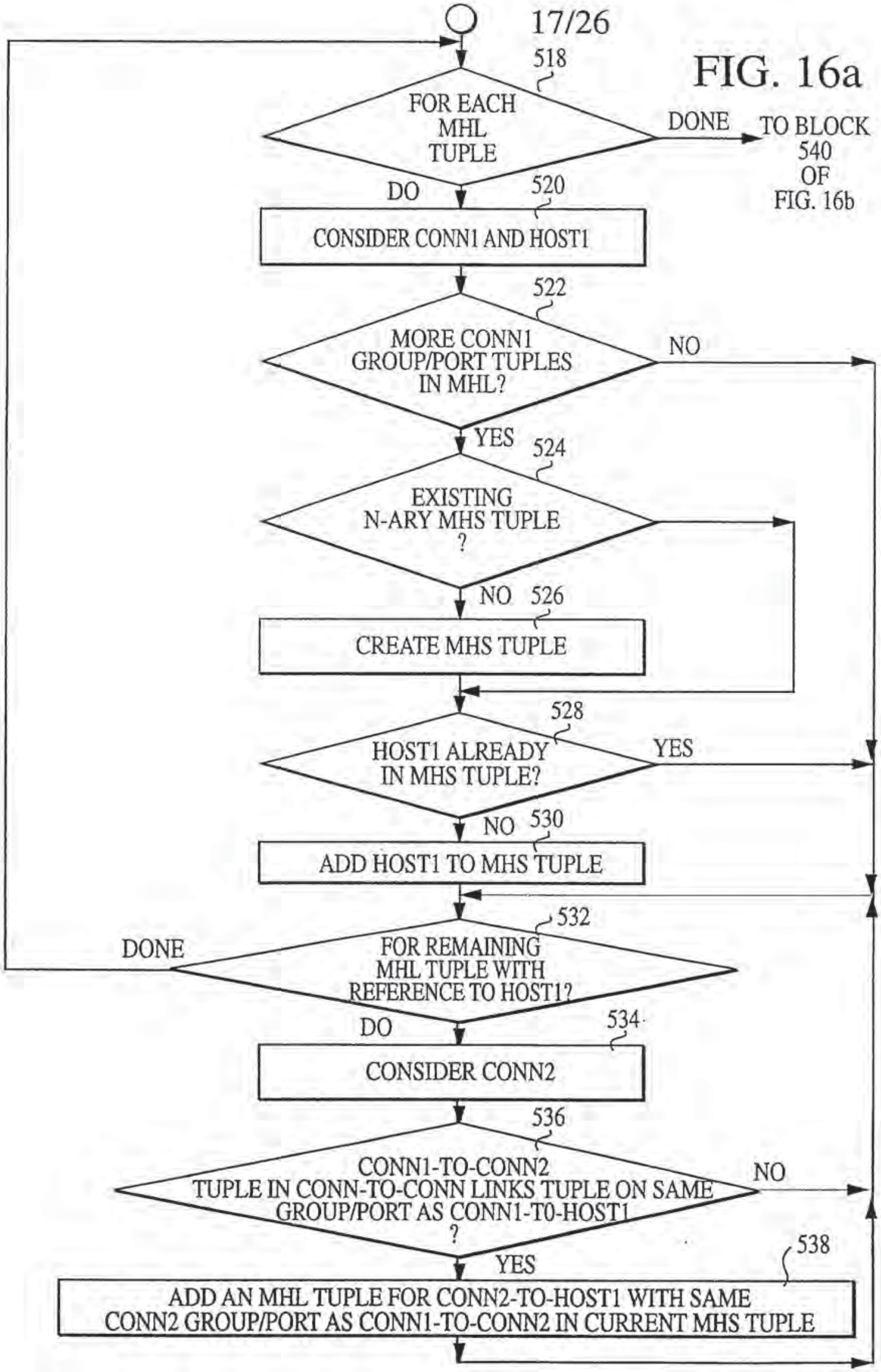

518
FOR EACH MHL TUPLE
DONE
TO BLOCK 540 OF FIG. 16b
DO
520
CONSIDER CONN1 AND HOST1
522
MORE CONN1 GROUP/PORT TUPLES IN MHL?
NO
YES
524
EXISTING N-ARY MHS TUPLE ?
NO
526
CREATE MHS TUPLE
528
HOST1 ALREADY IN MHS TUPLE?
YES
NO
530
ADD HOST1 TO MHS TUPLE
532
FOR REMAINING MHL TUPLE WITH REFERENCE TO HOST1?
DONE
DO
534
CONSIDER CONN2
536
CONN1-TO-CONN2 TUPLE IN CONN-TO-CONN LINKS TUPLE ON SAME GROUP/PORT AS CONN1-T0-HOST1 ?
NO
YES
538
ADD AN MHL TUPLE FOR CONN2-TO-HOST1 WITH SAME CONN2 GROUP/PORT AS CONN1-TO-CONN2 IN CURRENT MHS TUPLE

FIG. 16a

FIG. 16b

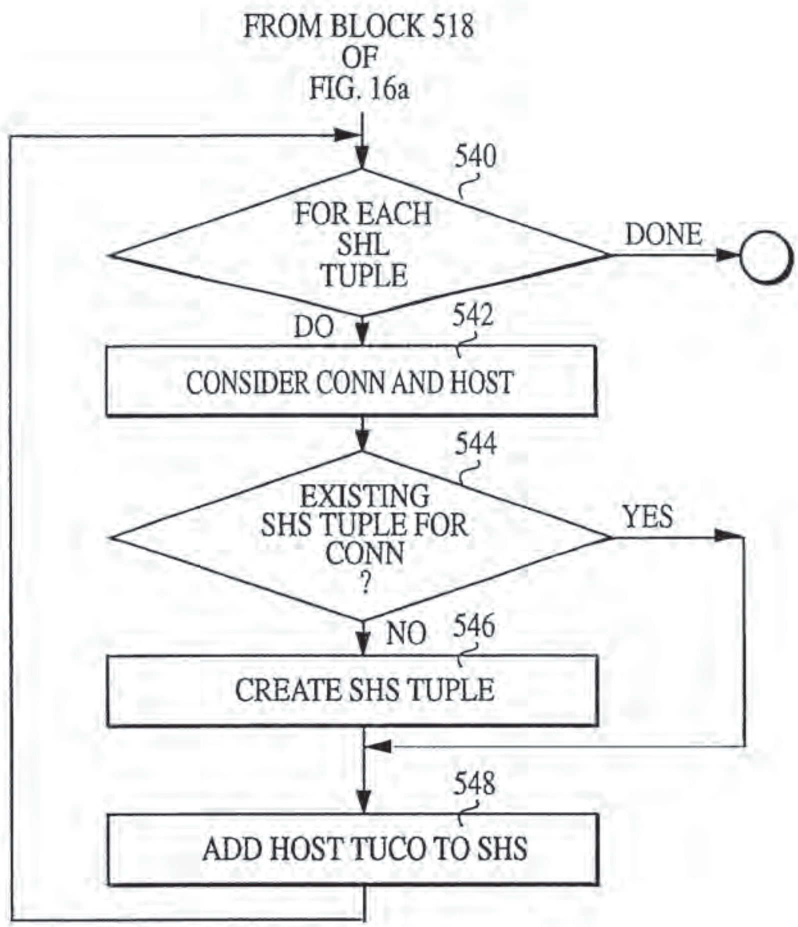

FROM BLOCK 518 OF FIG. 16a
540
FOR EACH SHL TUPLE
DONE
DO
542
CONSIDER CONN AND HOST
544
EXISTING SHS TUPLE FOR CONN ?
YES
NO
546
CREATE SHS TUPLE
548
ADD HOST TUCO TO SHS

FIG. 17

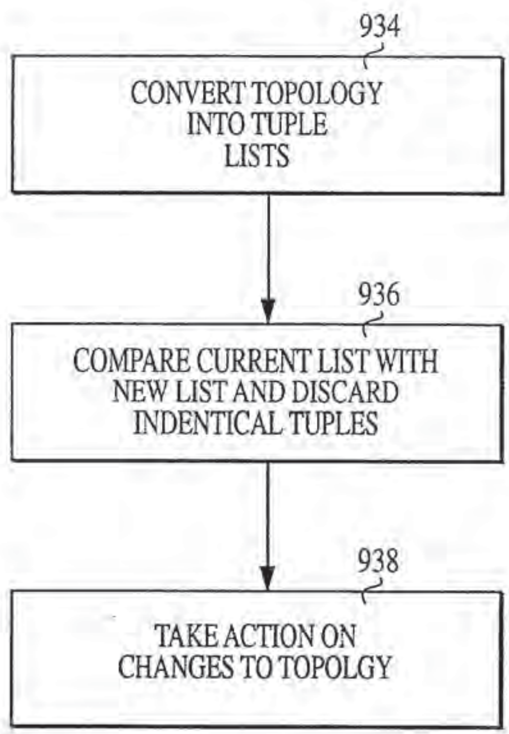
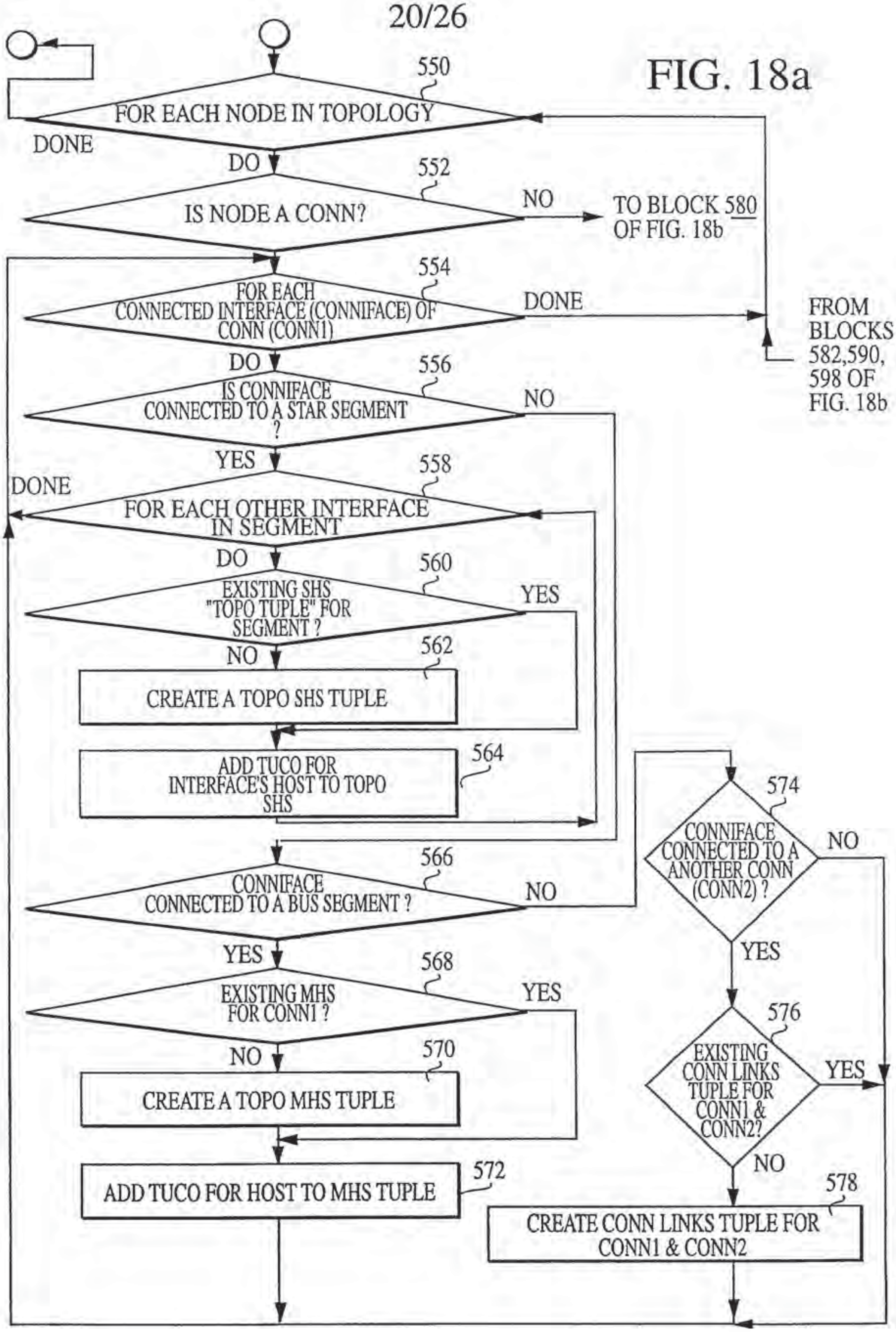

934
CONVERT TOPOLOGY INTO TUPLE LISTS
936
COMPARE CURRENT LIST WITH NEW LIST AND DISCARD INDENTICAL TUPLES
938
TAKE ACTION ON CHANGES TO TOPOLGY

FIG. 18a


550
FOR EACH NODE IN TOPOLOGY
DONE
DO
552
IS NODE A CONN?
NO
TO BLOCK 580 OF FIG. 18b
554
FOR EACH CONNECTED INTERFACE (CONNIFACE) OF CONN (CONN1)
DONE
FROM BLOCKS 582,590, 598 OF FIG. 18b
DO
556
IS CONNIFACE CONNECTED TO A STAR SEGMENT ?
NO
YES
558
DONE
FOR EACH OTHER INTERFACE IN SEGMENT
DO
560
EXISTING SHS "TOPO TUPLE" FOR SEGMENT ?
YES
NO
562
CREATE A TOPO SHS TUPLE
564
ADD TUCO FOR INTERFACE'S HOST TO TOPO SHS
566
CONNIFACE CONNECTED TO A BUS SEGMENT ?
NO
574
CONNIFACE CONNECTED TO A ANOTHER CONN (CONN2) ?
NO
YES
YES
568
EXISTING MHS FOR CONN1 ?
YES
NO
570
CREATE A TOPO MHS TUPLE
576
EXISTING CONN LINKS TUPLE FOR CONN1 & CONN2?
YES
NO
572
ADD TUCO FOR HOST TO MHS TUPLE
578
CREATE CONN LINKS TUPLE FOR CONN1 & CONN2

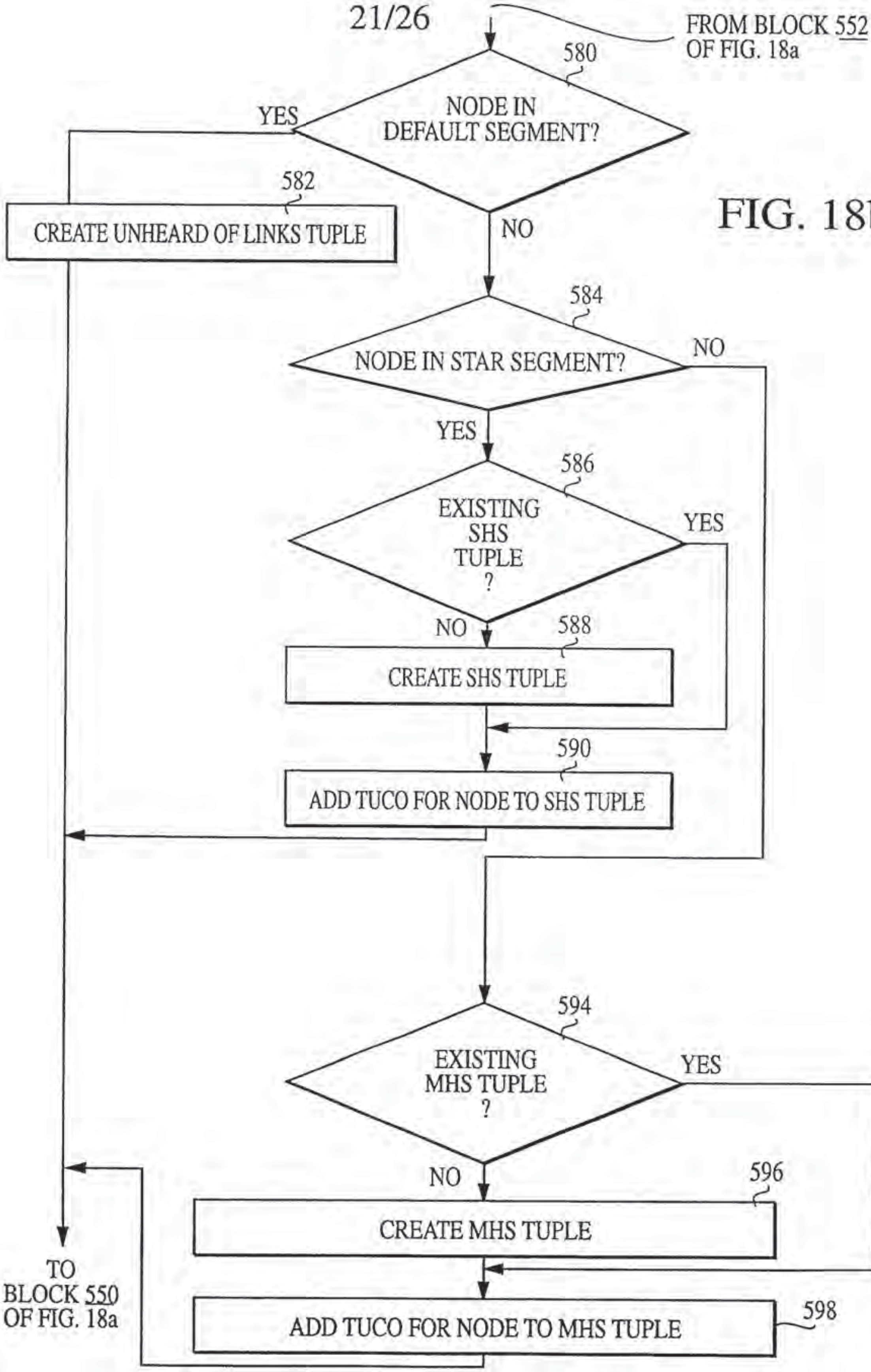

FROM BLOCK 552 OF FIG. 18a
580
NODE IN DEFAULT SEGMENT?
YES
NO
582
CREATE UNHEARD OF LINKS TUPLE
584
NODE IN STAR SEGMENT?
NO
YES
586
EXISTING SHS TUPLE ?
YES
NO
588
CREATE SHS TUPLE
590
ADD TUCO FOR NODE TO SHS TUPLE
594
EXISTING MHS TUPLE ?
YES
NO
596
CREATE MHS TUPLE
598
ADD TUCO FOR NODE TO MHS TUPLE
TO BLOCK 550 OF FIG. 18a

FIG. 18b

FIG. 19

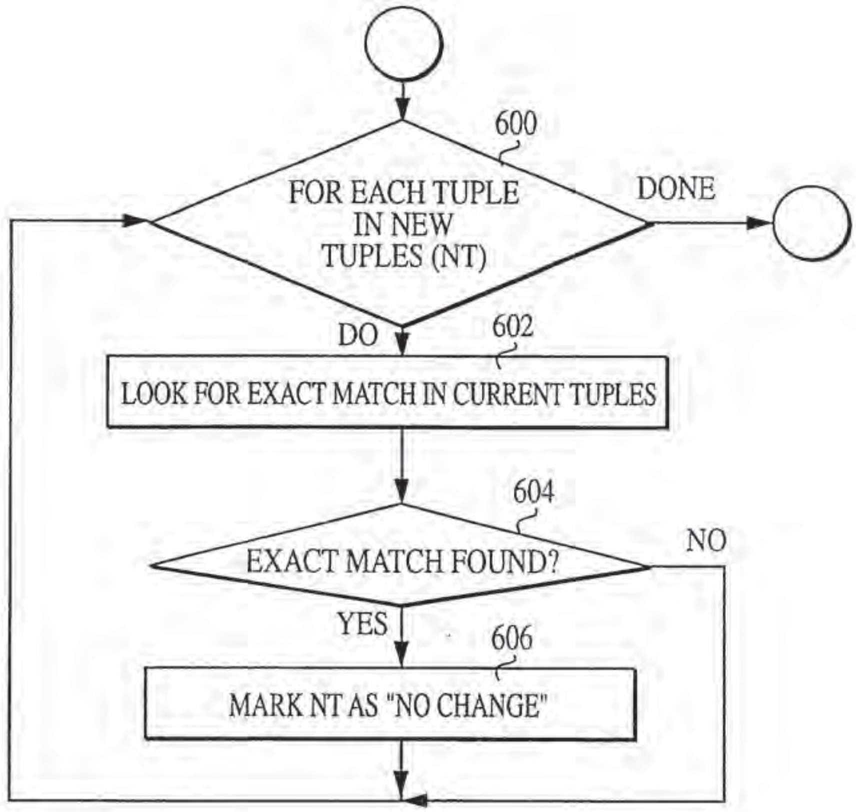
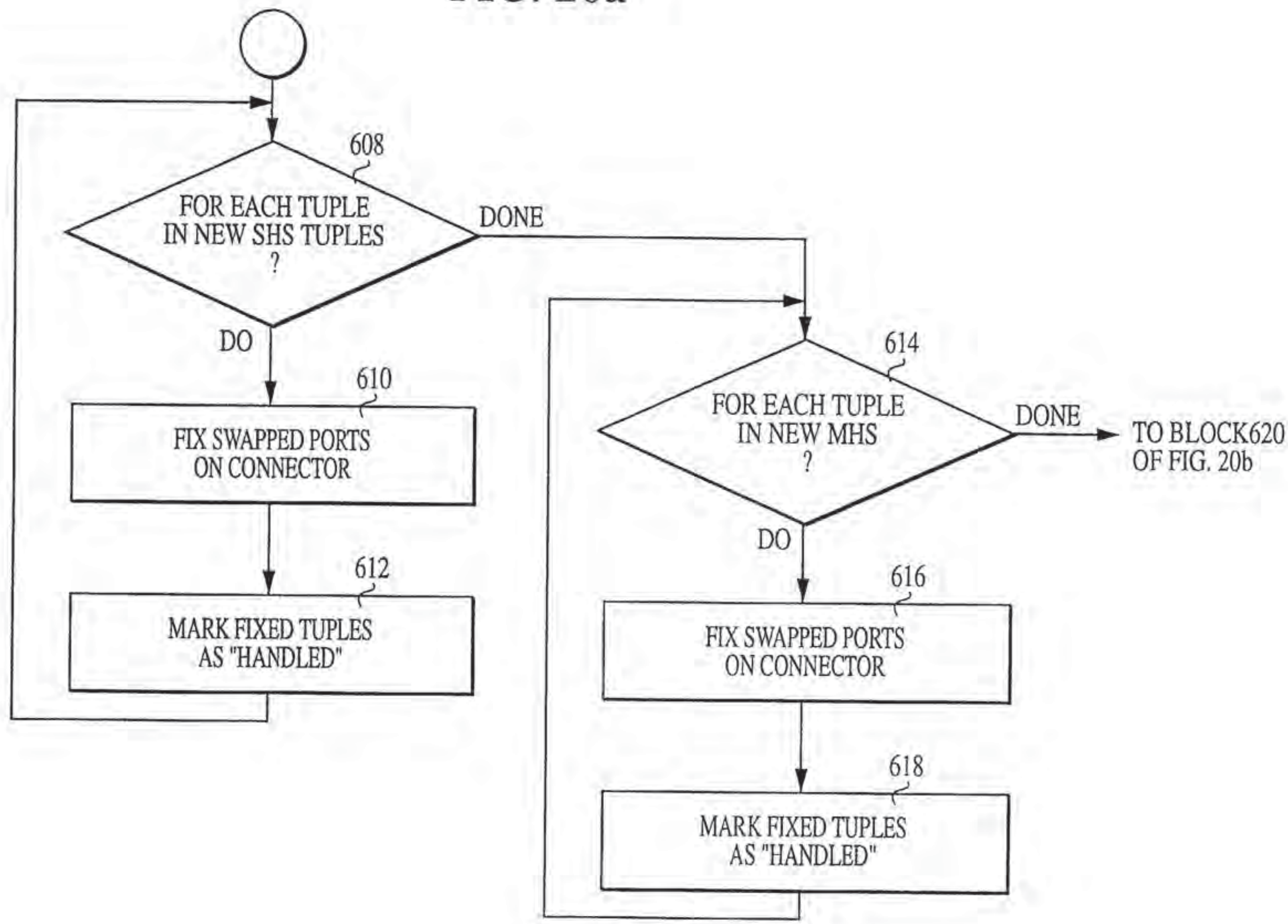
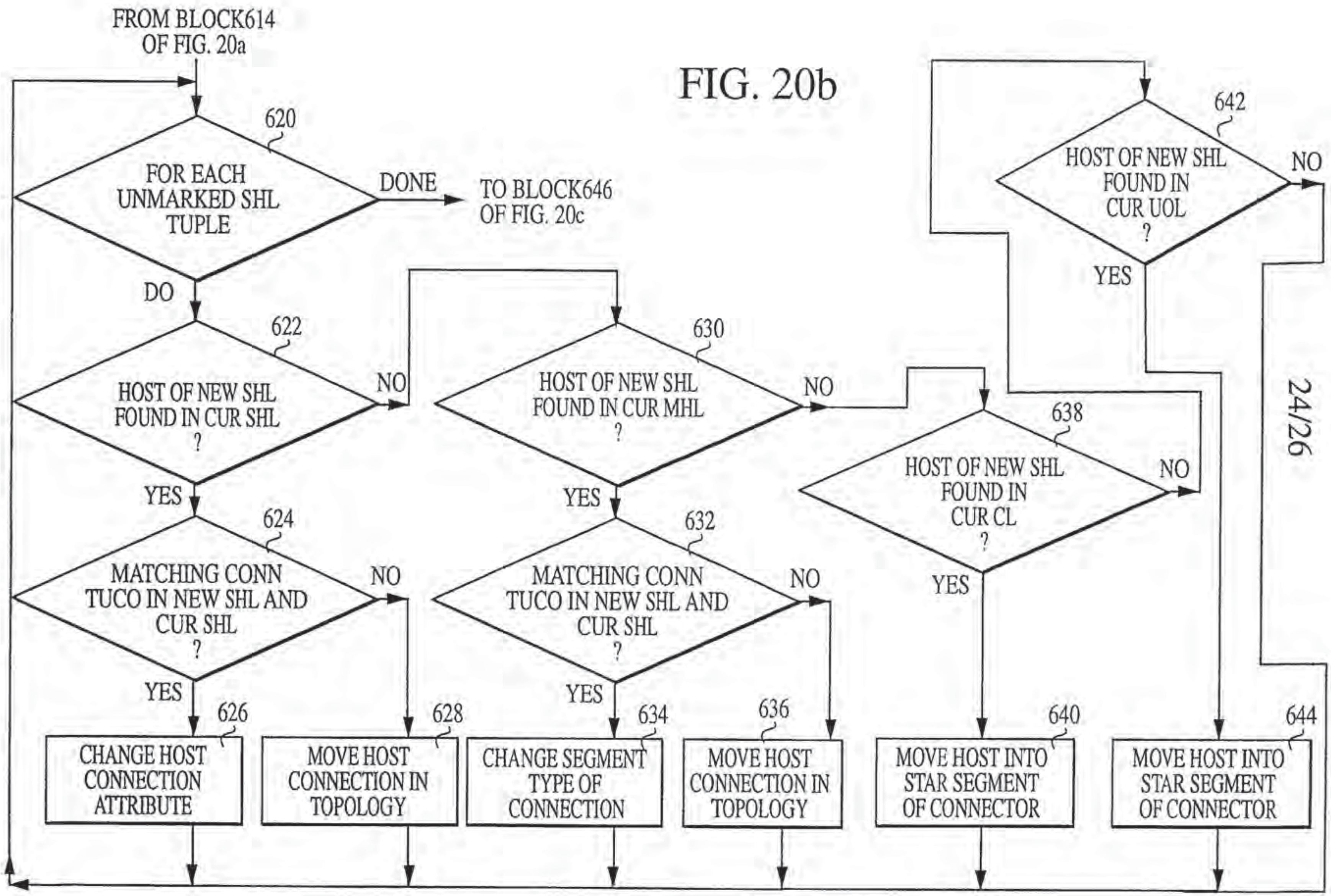

600
FOR EACH TUPLE
IN NEW
TUPLES (NT)
DONE
DO
602
LOOK FOR EXACT MATCH IN CURRENT TUPLES
604
EXACT MATCH FOUND?
NO
YES
606
MARK NT AS "NO CHANGE"

FIG. 20a



608
FOR EACH TUPLE IN NEW SHS TUPLES ?
DONE
DO
610
FIX SWAPPED PORTS ON CONNECTOR
612
MARK FIXED TUPLES AS "HANDLED"
614
FOR EACH TUPLE IN NEW MHS ?
DONE
TO BLOCK620 OF FIG. 20b
DO
616
FIX SWAPPED PORTS ON CONNECTOR
618
MARK FIXED TUPLES AS "HANDLED"

FIG. 20b



FROM BLOCK614 OF FIG. 20a
620
FOR EACH UNMARKED SHL TUPLE
DONE
TO BLOCK646 OF FIG. 20c
DO
622
HOST OF NEW SHL FOUND IN CUR SHL ?
YES
NO
624
MATCHING CONN TUCO IN NEW SHL AND CUR SHL ?
YES
NO
626
CHANGE HOST CONNECTION ATTRIBUTE
628
MOVE HOST CONNECTION IN TOPOLOGY
630
HOST OF NEW SHL FOUND IN CUR MHL ?
YES
NO
632
MATCHING CONN TUCO IN NEW SHL AND CUR SHL ?
YES
NO
634
CHANGE SEGMENT TYPE OF CONNECTION
636
MOVE HOST CONNECTION IN TOPOLOGY
638
HOST OF NEW SHL FOUND IN CUR CL ?
YES
NO
640
MOVE HOST INTO STAR SEGMENT OF CONNECTOR
642
HOST OF NEW SHL FOUND IN CUR UOL ?
YES
NO
644
MOVE HOST INTO STAR SEGMENT OF CONNECTOR

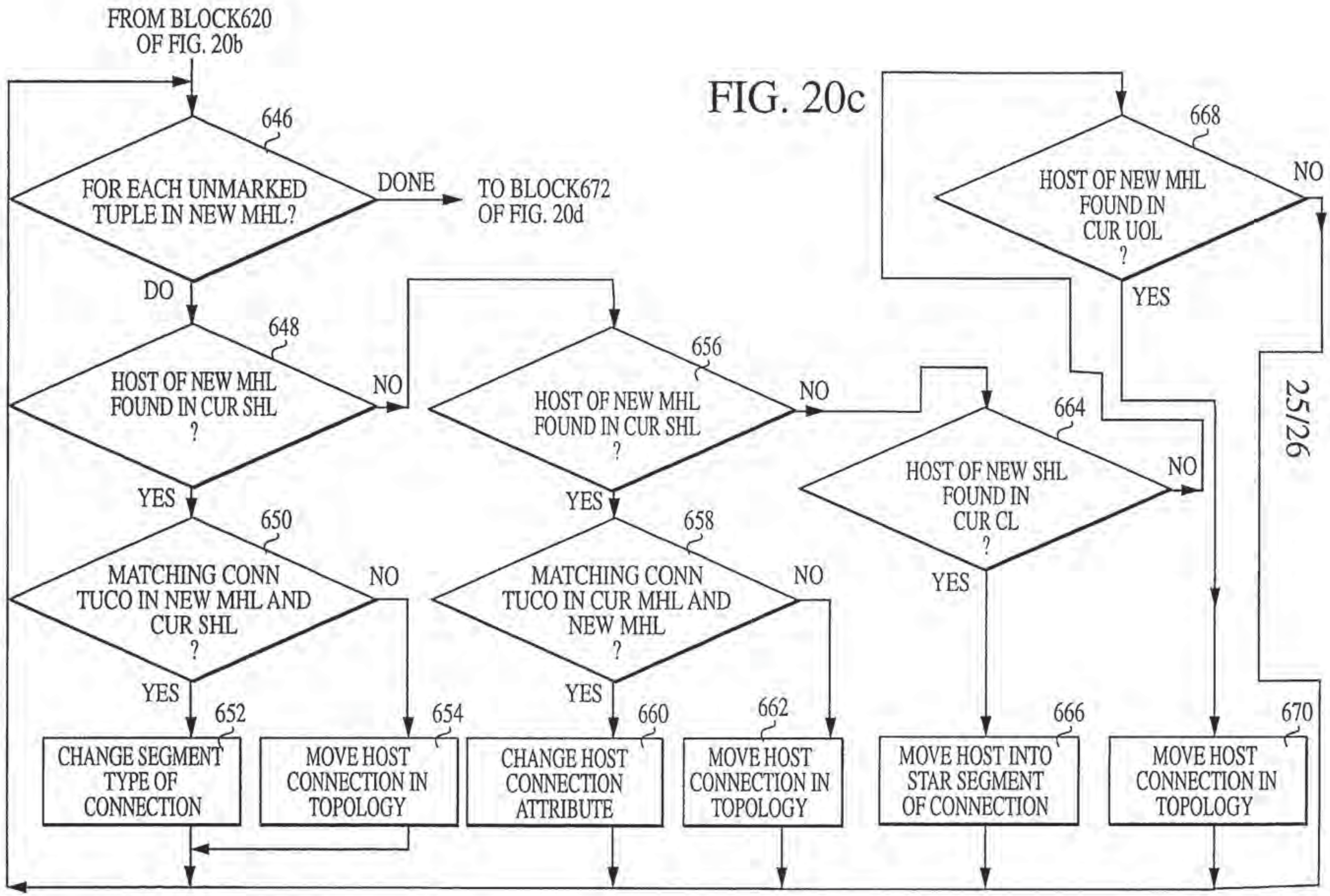
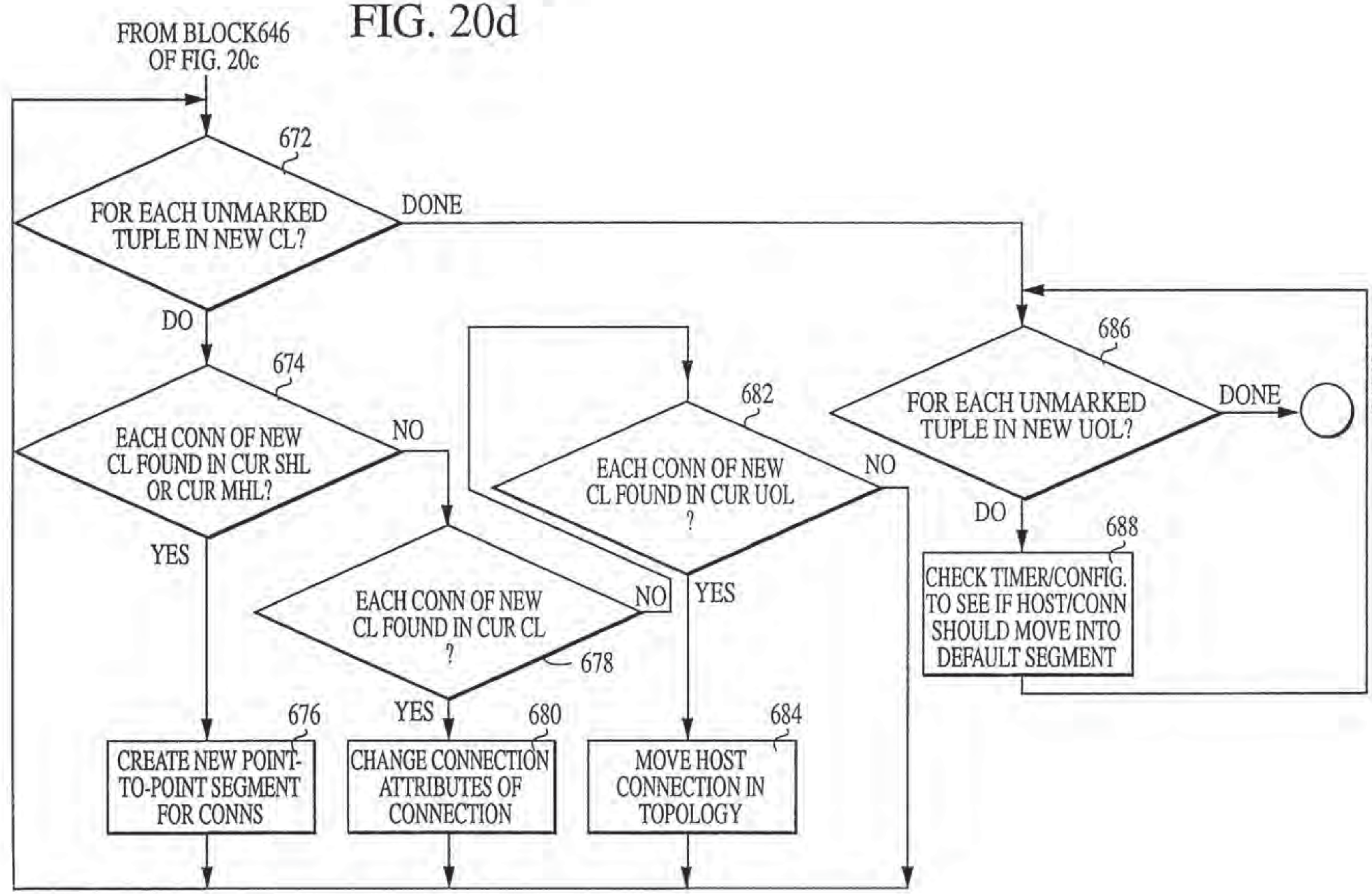

FROM BLOCK620
OF FIG. 20b
646
FOR EACH UNMARKED
TUPLE IN NEW MHL?
DONE
TO BLOCK672
OF FIG. 20d
DO
648
HOST OF NEW MHL
FOUND IN CUR SHL
?
NO
YES
650
MATCHING CONN
TUCO IN NEW MHL AND
CUR SHL
?
NO
YES
652
CHANGE SEGMENT
TYPE OF
CONNECTION
654
MOVE HOST
CONNECTION IN
TOPOLOGY
656
HOST OF NEW MHL
FOUND IN CUR SHL
?
NO
YES
658
MATCHING CONN
TUCO IN CUR MHL AND
NEW MHL
?
NO
YES
660
CHANGE HOST
CONNECTION
ATTRIBUTE
662
MOVE HOST
CONNECTION IN
TOPOLOGY
664
HOST OF NEW SHL
FOUND IN
CUR CL
?
NO
YES
666
MOVE HOST INTO
STAR SEGMENT
OF CONNECTION
668
HOST OF NEW MHL
FOUND IN
CUR UOL
?
NO
YES
670
MOVE HOST
CONNECTION IN
TOPOLOGY

FIG. 20c

FIG. 20d


FROM BLOCK646
OF FIG. 20c
672
FOR EACH UNMARKED TUPLE IN NEW CL?
DONE
DO
674
EACH CONN OF NEW CL FOUND IN CUR SHL OR CUR MHL?
NO
YES
EACH CONN OF NEW CL FOUND IN CUR CL ?
678
NO
YES
682
EACH CONN OF NEW CL FOUND IN CUR UOL ?
NO
YES
676
CREATE NEW POINT-TO-POINT SEGMENT FOR CONNS
680
CHANGE CONNECTION ATTRIBUTES OF CONNECTION
684
MOVE HOST CONNECTION IN TOPOLOGY
686
FOR EACH UNMARKED TUPLE IN NEW UOL?
DONE
DO
688
CHECK TIMER/CONFIG. TO SEE IF HOST/CONN SHOULD MOVE INTO DEFAULT SEGMENT

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

ATTORNEY DOCKET NO. 10008102-1

As a below named inventor, I hereby declare that:

My residence/post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

Method And System For Identifying And Processing Changes To A Network Topology

the specification of which is attached hereto unless the following box is checked:

( ) was filed on ______________ as US Application Serial No. or PCT International Application Number ____________ and was amended on ______________ (if applicable).

I hereby state that I have reviewed and understood the contents of the above-identified specification, including the claims, as amended by any amendment(s) referred to above. I acknowledge the duty to disclose all information which is material to patentability as defined in 37 CFR 1.56.

Foreign Application(s) and/or Claim of Foreign Priority

I hereby claim foreign priority benefits under Title 35, United States Code Section 119 of any foreign application(s) for patent or inventor(s) certificate listed below and have also identified below any foreign application for patent or inventor(s) certificate having a filing date before that of the application on which priority is claimed:

COUNTRY	APPLICATION NUMBER	DATE FILED	PRIORITY CLAIMED UNDER 35 U.S.C. 119
N/A			YES: ___ NO: ___
			YES: ___ NO: ___

Provisional Application

I hereby claim the benefit under Title 35, United States Code Section 119(e) of any United States provisional application(s) listed below:

APPLICATION SERIAL NUMBER	FILING DATE
N/A	

U. S. Priority Claim

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code Section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, Section 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

APPLICATION SERIAL NUMBER	FILING DATE	STATUS (patented/pending/abandoned)
N/A		

POWER OF ATTORNEY:

As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith:

Customer Number 022879

Place Customer Number Bar Code Label here

Send Correspondence to:
HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

Direct Telephone Calls To:
T. Grant Ritz
(970) 898-0697

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Inventor: Eric A Pulsipher **Citizenship:** US

Residence: 2937 Redburn Drive Ft Collins CO 80525

Post Office Address: Same as residence

[signature] 10/31/2000

Inventor's Signature **Date**

Rev 10/00 (DecPwr) (Use Page Two For Additional Inventor(s) Signature(s))

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION (continued)

ATTORNEY DOCKET NO. 10008102-1

Full Name of # 2 joint inventor: Joseph R Hunt

Citizenship: US

Residence: ~~5841 Meadow Creek Ln~~ 5239 Hahns Peak Dr #205 Loveland, CO 80538

Post Office Address: Same as Residence

Inventor's Signature: Joseph R Hunt

Date: 10/31/00

Full Name of # 3 joint inventor:

Citizenship:

Residence:

Post Office Address:

Inventor's Signature

Date

Full Name of # 4 joint inventor:

Citizenship:

Residence:

Post Office Address:

Inventor's Signature

Date

Full Name of # 5 joint inventor:

Citizenship:

Residence:

Post Office Address:

Inventor's Signature

Date

Full Name of # 6 joint inventor:

Citizenship:

Residence:

Post Office Address:

Inventor's Signature

Date

Full Name of # 7 joint inventor:

Citizenship:

Residence:

Post Office Address:

Inventor's Signature

Date

Full Name of # 8 joint inventor:

Citizenship:

Residence:

Post Office Address:

Inventor's Signature

Date

JC825 U.S. PTO
09/703942
10/31/00

ISSUE CLASSIFICATION

Class	Subclass

PATENT NUMBER

U.S. UTILITY Patent Application

O.I.P.E.

SCANNED Am² Q.A. GTH

PATENT DATE

APPLICATION NO.	CONT/PRIOR	CLASS	SUBCLASS	ART UNIT	EXAMINER
09/703942		370	254	~~2661~~ 2664	~~TRAN, M~~ Schultz

APPLICANTS
Eric Pulsipher
Joseph Hunt

TITLE
Method and system for identifying and processing changes to a network topology

PTO-2040 12/99

Best Available Copy

ISSUING CLASSIFICATION

ORIGINAL		CROSS REFERENCE(S)						
CLASS	SUBCLASS	CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)					
INTERNATIONAL CLASSIFICATION								

☐ Continued on Issue Slip Inside File Jacket

☐ TERMINAL DISCLAIMER	DRAWINGS			CLAIMS ALLOWED	
	Sheets Drwg.	Figs. Drwg.	Print Fig.	Total Claims	Print Claim for O.G.

☐ The term of this patent subsequent to ________ (date) has been disclaimed.

☐ The term of this patent shall not extend beyond the expiration date of U.S Patent. No. ________

☐ The terminal ___ months of this patent have been disclaimed.

________ (Assistant Examiner) ________ (Date)

________ (Primary Examiner) ________ (Date)

________ (Legal Instruments Examiner) ________ (Date)

NOTICE OF ALLOWANCE MAILED

ISSUE FEE

Amount Due	Date Paid

ISSUE BATCH NUMBER

WARNING:
The information disclosed herein may be restricted. Unauthorized disclosure may be prohibited by the United States Code Title 35, Sections 122, 181 and 368. Possession outside the U.S. Patent & Trademark Office is restricted to authorized employees and contractors only.

Form PTO-436A (Rev. 6/99)

FILED WITH: ☐ DISK (CRF) ☐ FICHE ☐ CD-ROM
(Attached in pocket on right inside flap)

(FACE)

[redacted]

Best Available Copy

SEARCHED			
Class	Sub.	Date	Exmr.
370	229	6/21/04	wcs
↓	216	↓	↓
	217		
	221		
	225		
	254		
	255		
	256		
	257		
	258		

SEARCH NOTES (INCLUDING SEARCH STRATEGY)		
	Date	Exmr.
uspat, Epo, Jpo	6/21/04	wcs

INTERFERENCE SEARCHED			
Class	Sub.	Date	Exmr.

(RIGHT OUTSIDE)

ISSUE SLIP STAPLE AREA (for additional cross references)

POSITION	INITIALS	ID NO.	DATE
FEE DETERMINATION			
O.I.P.E. CLASSIFIER	[illegible]		11-25-00
FORMALITY REVIEW	[illegible]	75353	2-5-01
RESPONSE FORMALITY REVIEW			

INDEX OF CLAIMS

✔	Rejected	N	Non-elected
=	Allowed	I	Interference
— (Through numeral)	Canceled	A	Appeal
÷	Restricted	O	Objected

Claim Final	Claim Original	Date [illegible]	Claim Final	Claim Original	Date	Claim Final	Claim Original	Date
	1	✔		51			101	
	2	✔		52			102	
	3	✔		53			103	
	4	✔		54			104	
	5	✔		55			105	
	6	✔		56			106	
	7	✔		57			107	
	8	✔		58			108	
	9	✔		59			109	
	10	✔		60			110	
	11	✔		61			111	
	12	✔		62			112	
	13	✔		63			113	
	14	✔		64			114	
	15	✔		65			115	
	16	✔		66			116	
	17	✔		67			117	
	18	✔		68			118	
	19	✔		69			119	
	20	✔		70			120	
	21			71			121	
	22			72			122	
	23			73			123	
	24			74			124	
	25			75			125	
	26			76			126	
	27			77			127	
	28			78			128	
	29			79			129	
	30			80			130	
	31			81			131	
	32			82			132	
	33			83			133	
	34			84			134	
	35			85			135	
	36			86			136	
	37			87			137	
	38			88			138	
	39			89			139	
	40			90			140	
	41			91			141	
	42			92			142	
	43			93			143	
	44			94			144	
	45			95			145	
	46			96			146	
	47			97			147	
	48			98			148	
	49			99			149	
	50			100			150	

If more than 150 claims or 10 actions
staple additional sheet here

(LEFT INSIDE)

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P. O. Box 272400
Fort Collins, Colorado 80527-2400

11-02-00

PATENT APPLICATION

ATTORNEY DOCKET NO. 10008102-1

A

IN THE U.S. PATENT AND TRADEMARK OFFICE
Patent Application Transmittal Letter

COMMISSIONER FOR PATENTS
Washington, D.C. 20231

Sir:

Transmitted herewith for filing under 37 CFR 1.53(b) is a(n): (X) Utility ( ) Design
(X) original patent application,
( ) continuation-in-part application

INVENTOR(S): **Eric A Pulsipher et al**

TITLE: **Method And System For Identifying And Processing Changes To A Network Topology**

Enclosed are:

(X) The Declaration and Power of Attorney. (X) signed ( ) unsigned or partially signed

(X) 26 sheets of drawings (one set) ( ) Associate Power of Attorney

( ) Form PTO-1449 ( ) Information Disclosure Statement and Form PTO-1449

( ) Priority document(s) ( ) (Other) ______ (fee $______)

CLAIMS AS FILED BY OTHER THAN A SMALL ENTITY				
(1) FOR	(2) NUMBER FILED	(3) NUMBER EXTRA	(4) RATE	(5) TOTALS
TOTAL CLAIMS	20 — 20	0	X $18	$ 0
INDEPENDENT CLAIMS	3 — 3	0	X $80	$ 0
ANY MULTIPLE DEPENDENT CLAIMS	0		$270	$ 0
BASIC FEE: Design ($320.00); Utility ($710.00)				$ 710
TOTAL FILING FEE				$ 710
OTHER FEES				$
TOTAL CHARGES TO DEPOSIT ACCOUNT				$ 710

Charge $ 710 to Deposit Account 08-2025. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16, 1.17,1.19, 1.20 and 1.21. A duplicate copy of this sheet is enclosed.

"Express Mail" label no. EL523338183US

Date of Deposit Oct. 31, 2000

I hereby certify that this is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to: Commissioner for Patents, Washington, D.C. 20231.

By Laura M. Clark

Typed Name: Laura M. Clark

Respectfully submitted,

Eric A Pulsipher et al

By T. G. Ritz

T. Grant Ritz
Attorney/Agent for Applicant(s)
Reg. No. 39,819
Date: Oct. 31, 2000
Telephone No.: (970) 898-0697

Rev 10/00 (TransNew)

- Attach as First Page to Transmitted Papers -

11-02-00

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P. O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 10008102-1

A

IN THE U.S. PATENT AND TRADEMARK OFFICE
Patent Application Transmittal Letter

COMMISSIONER FOR PATENTS
Washington, D.C. 20231

Sir:

Transmitted herewith for filing under 37 CFR 1.53(b) is a(n): (X) Utility ( ) Design
(X) original patent application,
( ) continuation-in-part application

INVENTOR(S): **Eric A Pulsipher et al**

TITLE: **Method And System For Identifying And Processing Changes To A Network Topology**

Enclosed are:

(X) The Declaration and Power of Attorney. (X) signed ( ) unsigned or partially signed

(X) 26 sheets of drawings (one set) ( ) Associate Power of Attorney

( ) Form PTO-1449 ( ) Information Disclosure Statement and Form PTO-1449

( ) Priority document(s) ( ) (Other) ____________ (fee $________)

CLAIMS AS FILED BY OTHER THAN A SMALL ENTITY				
(1) FOR	(2) NUMBER FILED	(3) NUMBER EXTRA	(4) RATE	(5) TOTALS
TOTAL CLAIMS	20 — 20	0	X $18	$ 0
INDEPENDENT CLAIMS	3 — 3	0	X $80	$ 0
ANY MULTIPLE DEPENDENT CLAIMS	0		$270	$ 0
BASIC FEE: Design ($320.00); Utility ($710.00)				$ 710
TOTAL FILING FEE				$ 710
OTHER FEES				$
TOTAL CHARGES TO DEPOSIT ACCOUNT				$ 710

Charge $ 710 to Deposit Account 08-2025. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16, 1.17,1.19, 1.20 and 1.21. A duplicate copy of this sheet is enclosed.

"Express Mail" label no. EL523338183US

Date of Deposit Oct. 31, 2000

I hereby certify that this is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to: Commissioner for Patents, Washington, D.C. 20231.

By [signature]

Typed Name: Laura M. Clark

Respectfully submitted,

Eric A Pulsipher et al

By [signature]

T. Grant Ritz

Attorney/Agent for Applicant(s)
Reg. No. 39,819
Date: Oct. 31, 2000
Telephone No.: (970) 898-0697

Rev 10/00 (TransNew)

- Attach as First Page to Transmitted Papers -

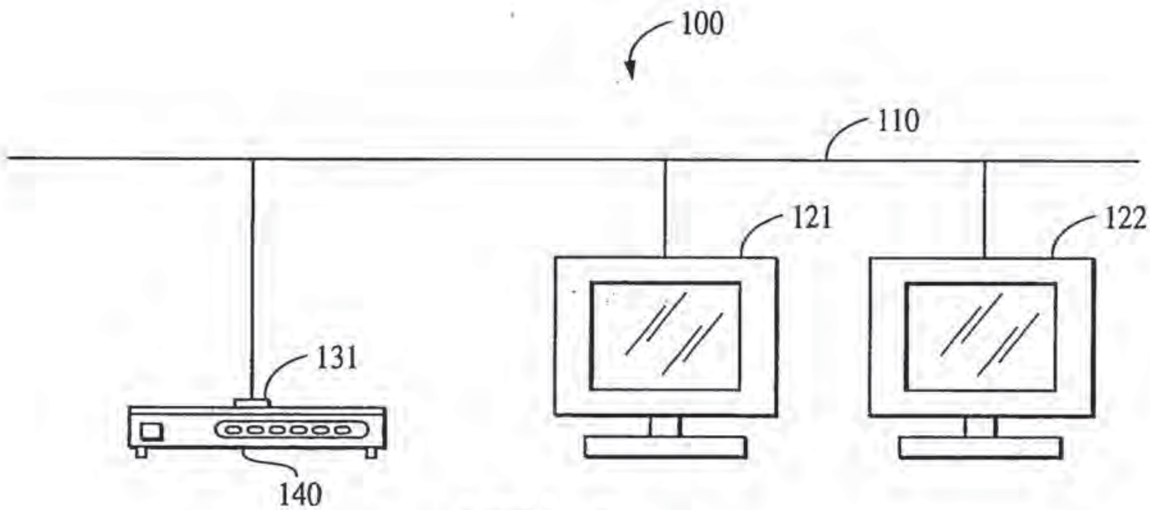


FIG. 1

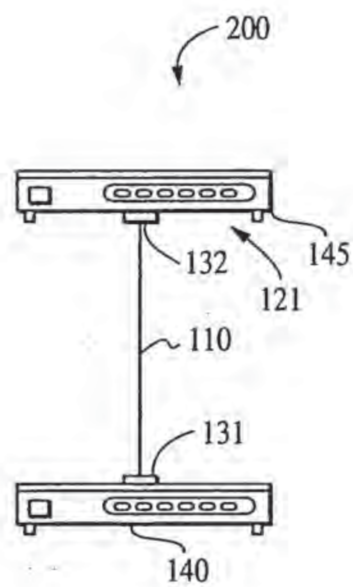


FIG. 2

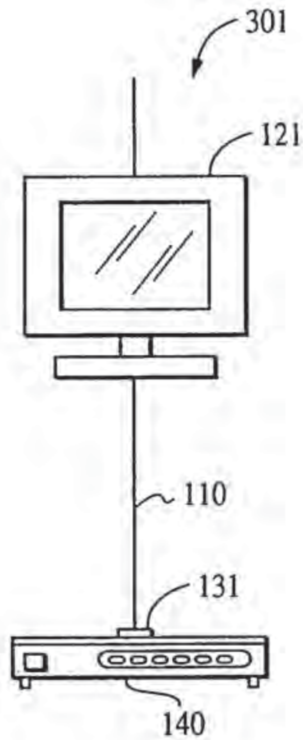


FIG. 3

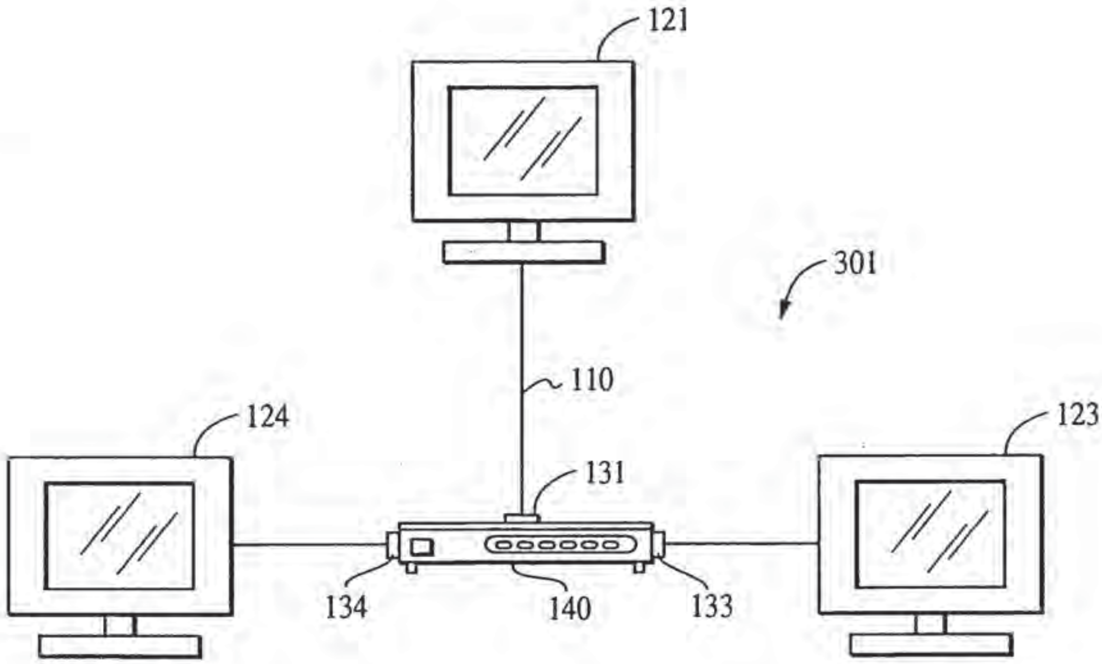


FIG. 4

FIG. 5

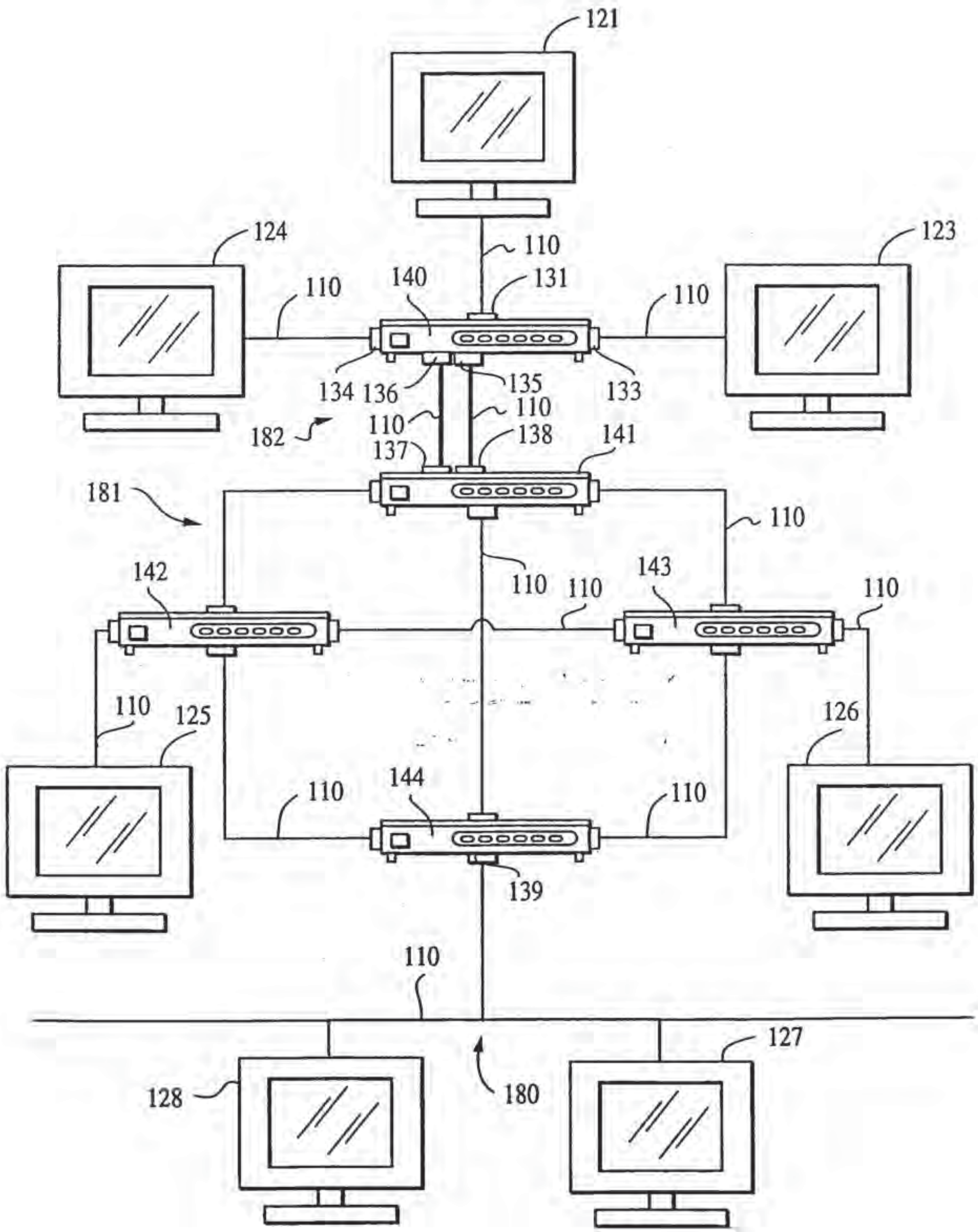
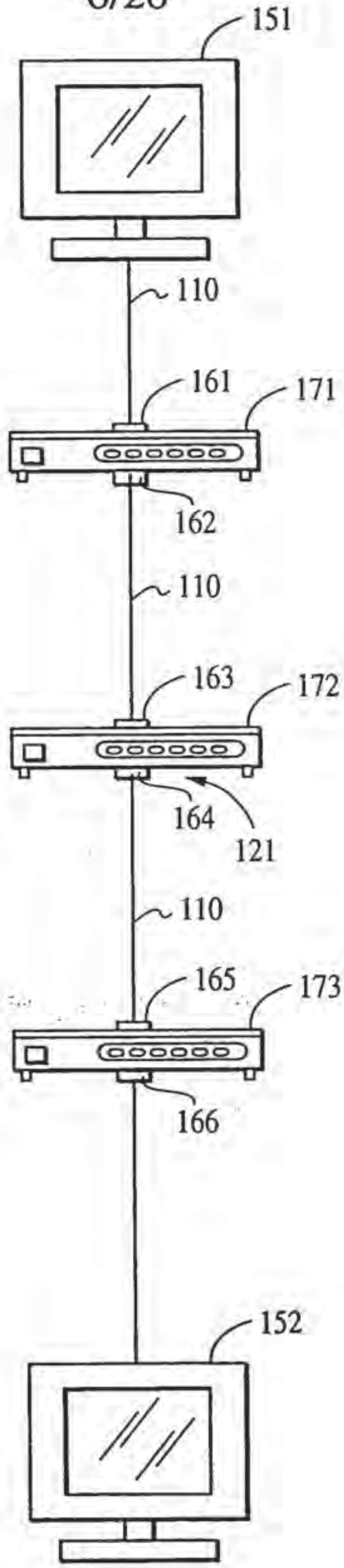

121
124
123
110
140
131
110
110
134 136
135
133
110
182
110
138
141
137
181
110
142
110
110
143
110
110
125
126
144
110
110
139
110
180
127
128

FIG. 6



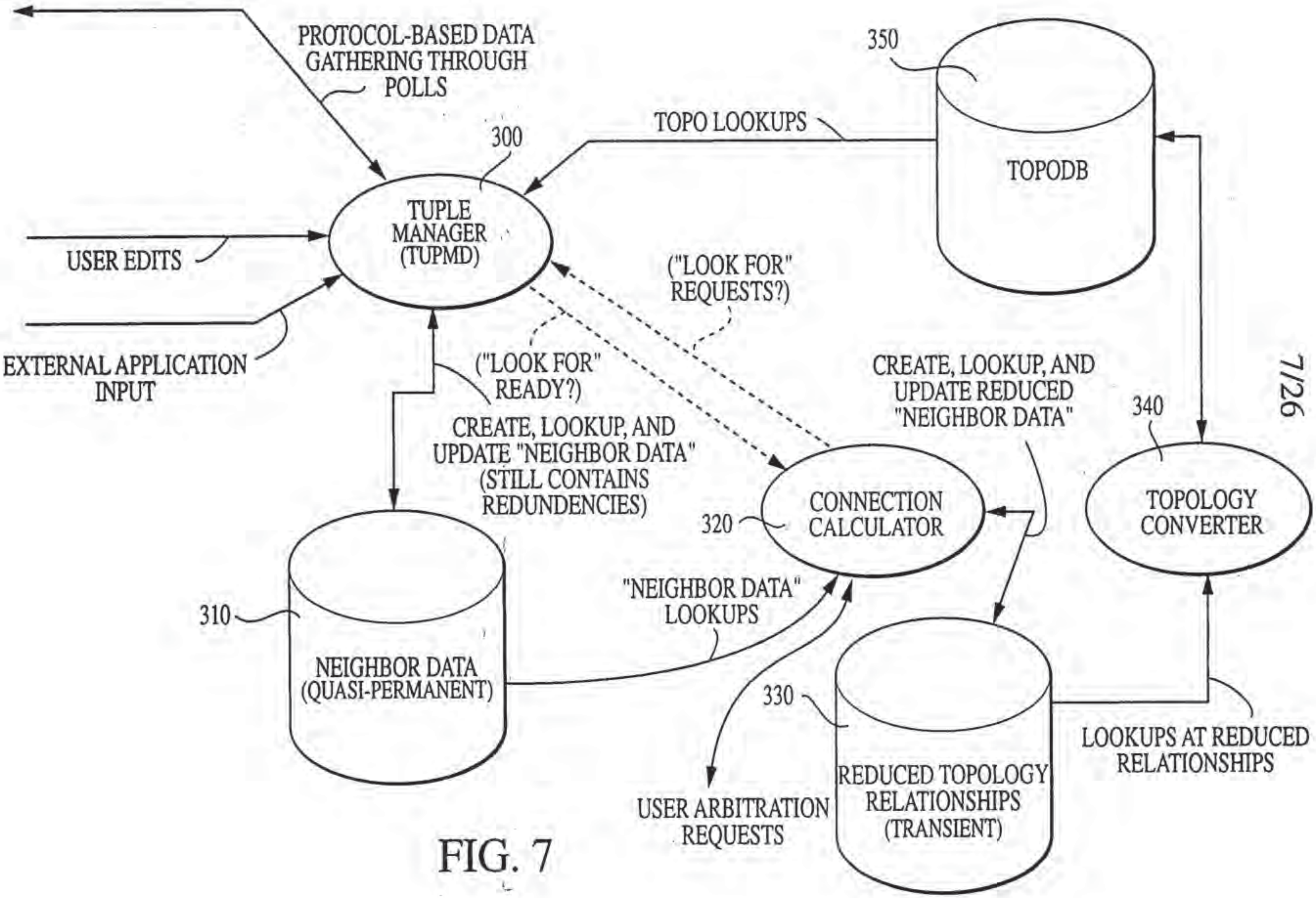

PROTOCOL-BASED DATA GATHERING THROUGH POLLS
300
TUPLE MANAGER (TUPMD)
USER EDITS
EXTERNAL APPLICATION INPUT
TOPO LOOKUPS
350
TOPODB
("LOOK FOR" REQUESTS?)
("LOOK FOR" READY?)
CREATE, LOOKUP, AND UPDATE "NEIGHBOR DATA" (STILL CONTAINS REDUNDENCIES)
310
NEIGHBOR DATA (QUASI-PERMANENT)
"NEIGHBOR DATA" LOOKUPS
320
CONNECTION CALCULATOR
CREATE, LOOKUP, AND UPDATE REDUCED "NEIGHBOR DATA"
340
TOPOLOGY CONVERTER
330
REDUCED TOPOLOGY RELATIONSHIPS (TRANSIENT)
USER ARBITRATION REQUESTS
LOOKUPS AT REDUCED RELATIONSHIPS

FIG. 7

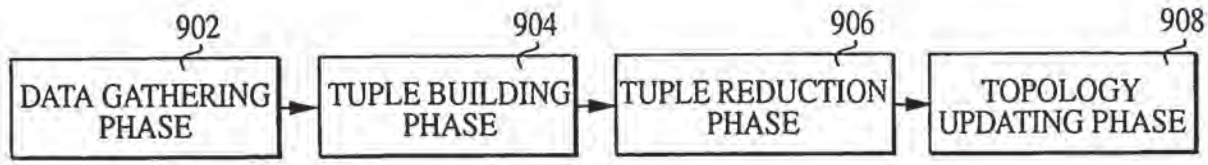


FIG. 8

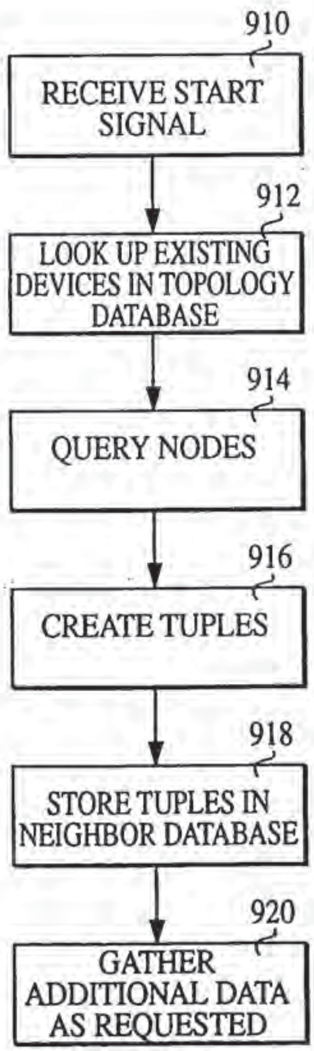


FIG. 9

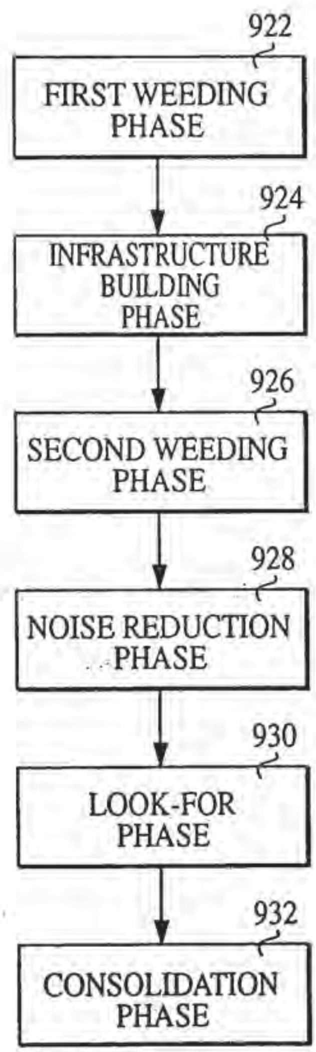


FIG. 10

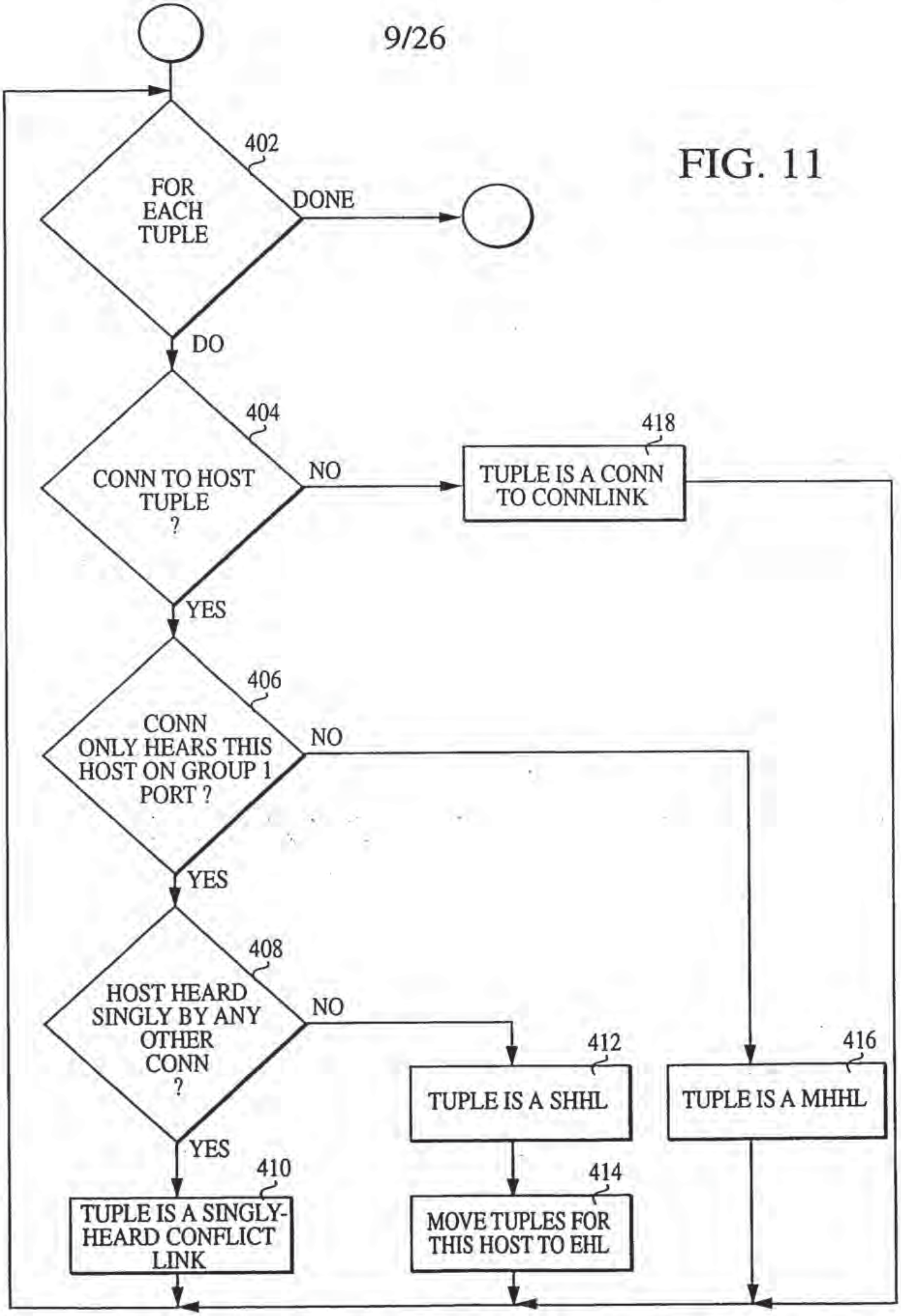
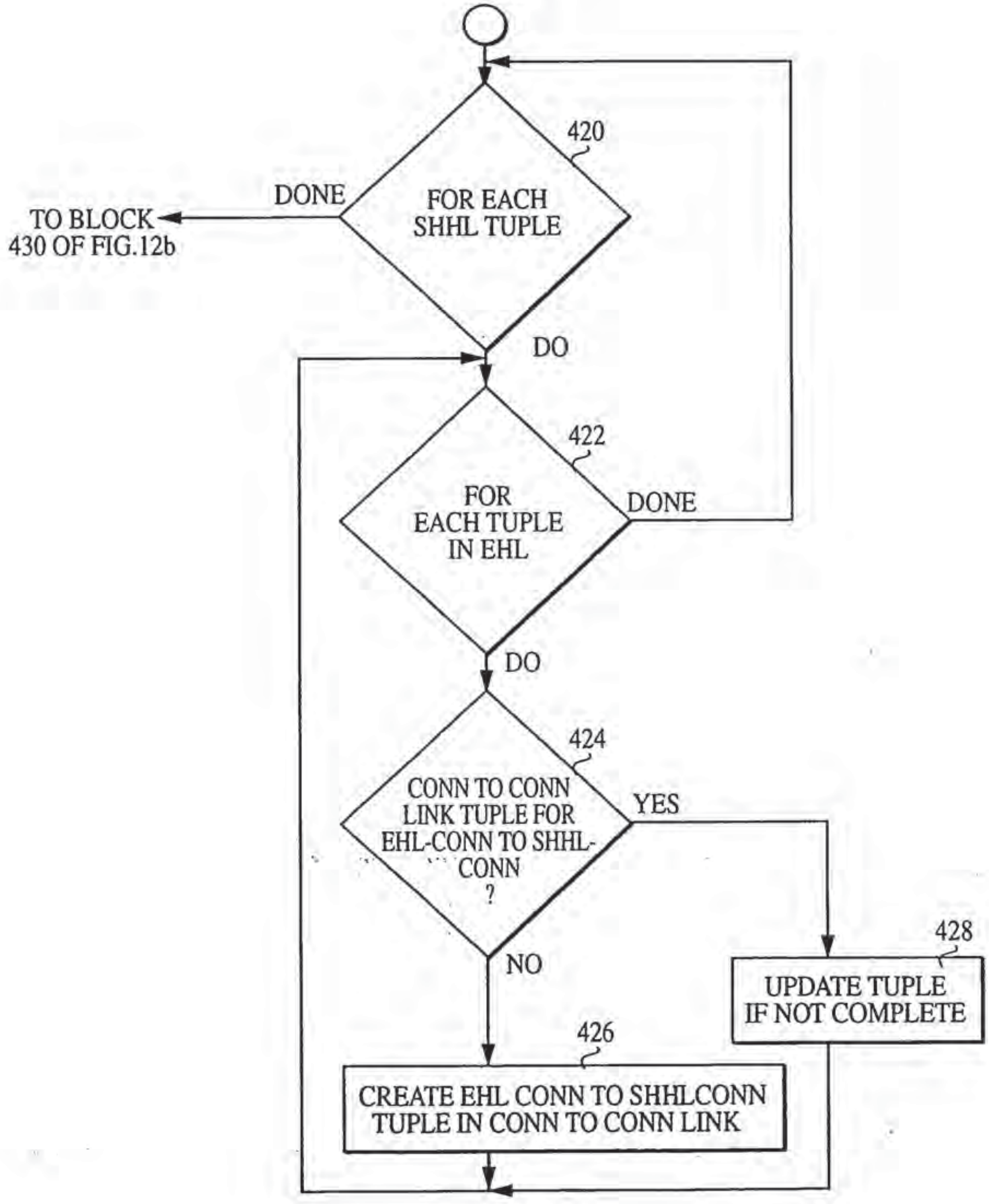

402
FOR EACH TUPLE
DONE
DO
404
CONN TO HOST TUPLE ?
NO
418
TUPLE IS A CONN TO CONNLINK
YES
406
CONN ONLY HEARS THIS HOST ON GROUP 1 PORT ?
NO
YES
408
HOST HEARD SINGLY BY ANY OTHER CONN ?
NO
412
TUPLE IS A SHHL
416
TUPLE IS A MHHL
YES
410
TUPLE IS A SINGLY-HEARD CONFLICT LINK
414
MOVE TUPLES FOR THIS HOST TO EHL

FIG. 11



420
FOR EACH SHHL TUPLE
DONE
TO BLOCK 430 OF FIG.12b
DO
422
FOR EACH TUPLE IN EHL
DONE
DO
424
CONN TO CONN LINK TUPLE FOR EHL-CONN TO SHHL-CONN ?
YES
NO
428
UPDATE TUPLE IF NOT COMPLETE
426
CREATE EHL CONN TO SHHLCONN TUPLE IN CONN TO CONN LINK

FIG. 12a

11/26

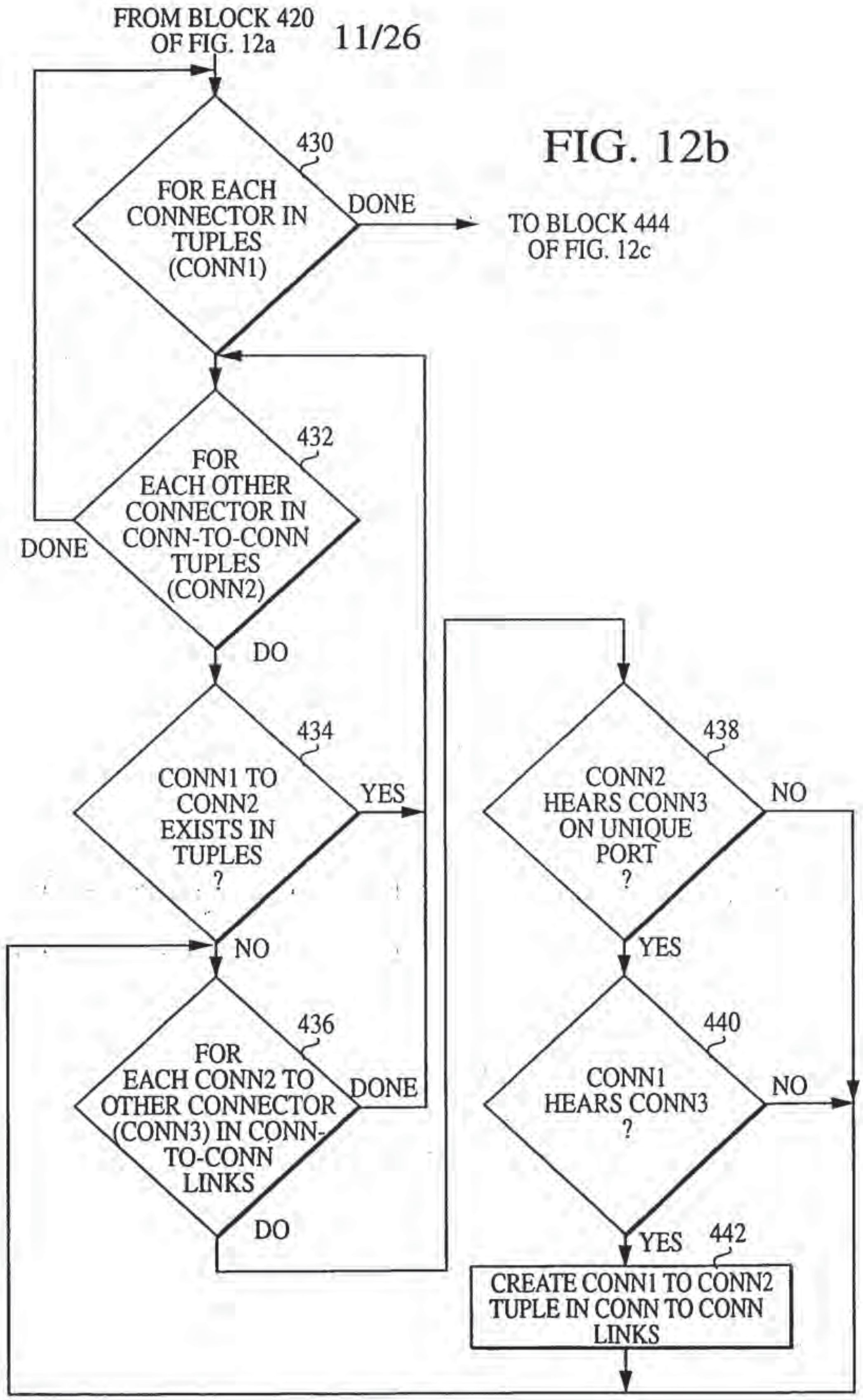

FROM BLOCK 420
OF FIG. 12a
430
FOR EACH CONNECTOR IN TUPLES (CONN1)
DONE
TO BLOCK 444
OF FIG. 12c
432
FOR EACH OTHER CONNECTOR IN CONN-TO-CONN TUPLES (CONN2)
DONE
DO
434
CONN1 TO CONN2 EXISTS IN TUPLES ?
YES
NO
436
FOR EACH CONN2 TO OTHER CONNECTOR (CONN3) IN CONN-TO-CONN LINKS
DONE
DO
438
CONN2 HEARS CONN3 ON UNIQUE PORT ?
NO
YES
440
CONN1 HEARS CONN3 ?
NO
YES
442
CREATE CONN1 TO CONN2 TUPLE IN CONN TO CONN LINKS

FIG. 12b

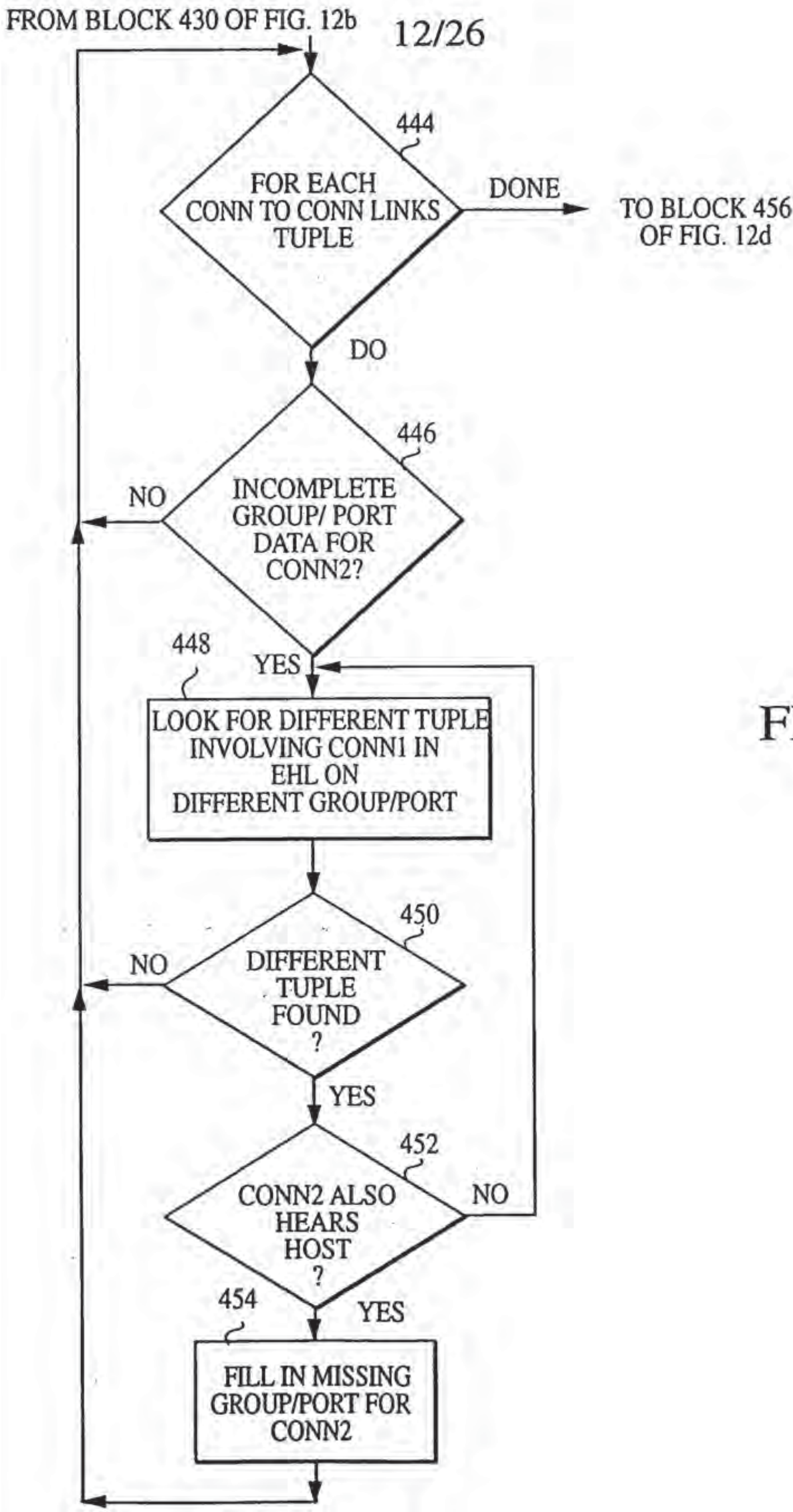

FROM BLOCK 430 OF FIG. 12b
444
FOR EACH CONN TO CONN LINKS TUPLE
DONE
TO BLOCK 456 OF FIG. 12d
DO
446
INCOMPLETE GROUP/ PORT DATA FOR CONN2?
NO
448
YES
LOOK FOR DIFFERENT TUPLE INVOLVING CONN1 IN EHL ON DIFFERENT GROUP/PORT
450
DIFFERENT TUPLE FOUND ?
NO
YES
452
CONN2 ALSO HEARS HOST ?
NO
454
YES
FILL IN MISSING GROUP/PORT FOR CONN2

FIG. 12c

FROM BLOCK 444 OF FIG. 12c

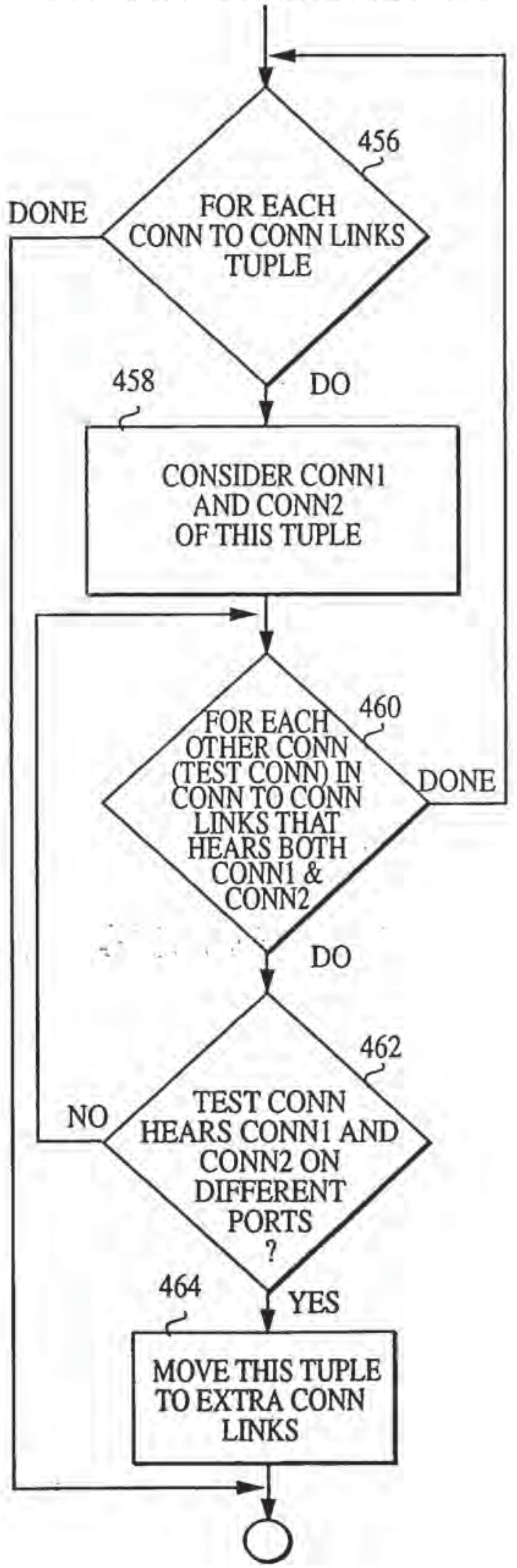


FIG. 12d

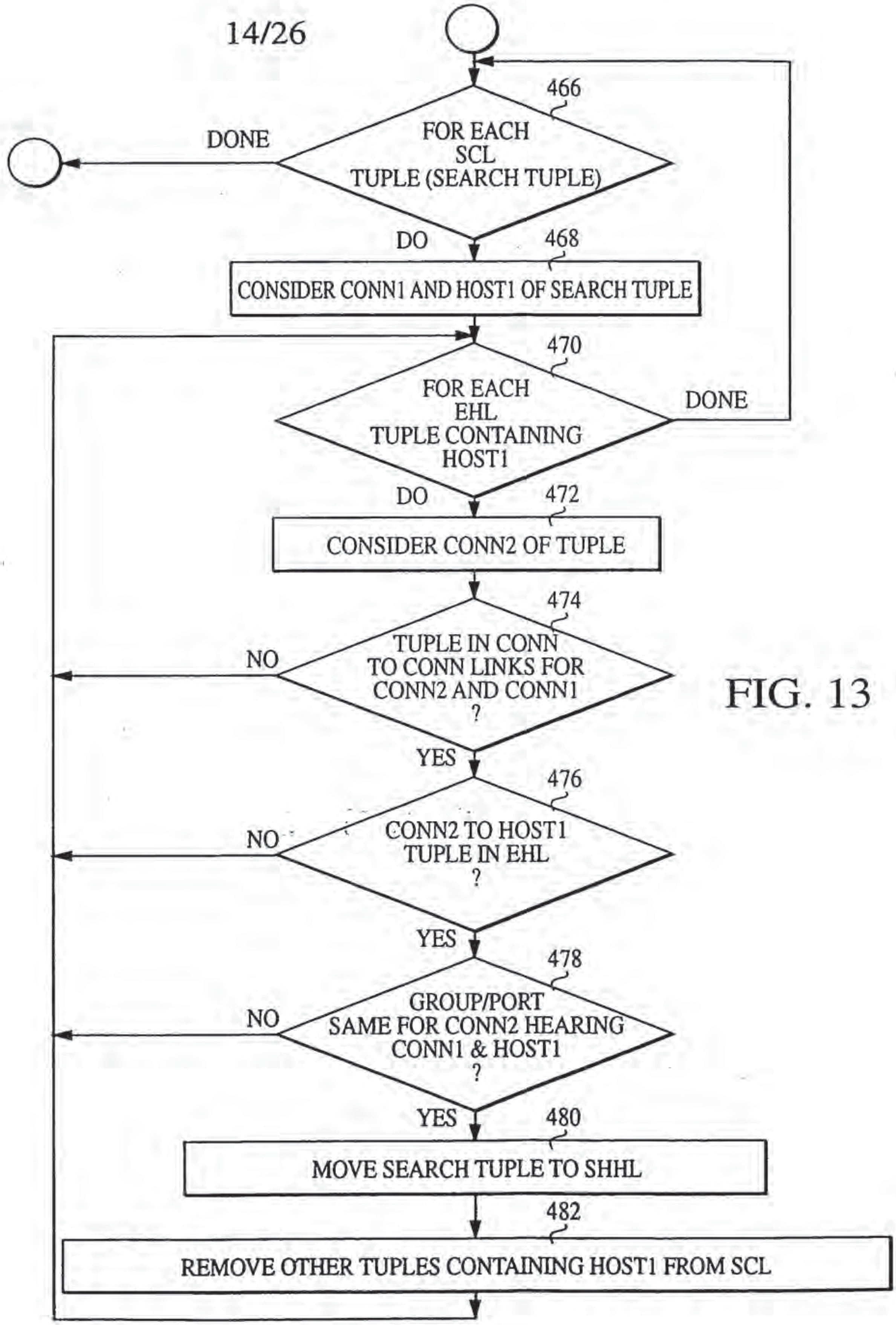

FOR EACH SCL TUPLE (SEARCH TUPLE)
466
DONE
DO
468
CONSIDER CONN1 AND HOST1 OF SEARCH TUPLE
470
FOR EACH EHL TUPLE CONTAINING HOST1
DONE
DO
472
CONSIDER CONN2 OF TUPLE
474
TUPLE IN CONN TO CONN LINKS FOR CONN2 AND CONN1 ?
NO
YES
476
CONN2 TO HOST1 TUPLE IN EHL ?
NO
YES
478
GROUP/PORT SAME FOR CONN2 HEARING CONN1 & HOST1 ?
NO
YES
480
MOVE SEARCH TUPLE TO SHHL
482
REMOVE OTHER TUPLES CONTAINING HOST1 FROM SCL

FIG. 13

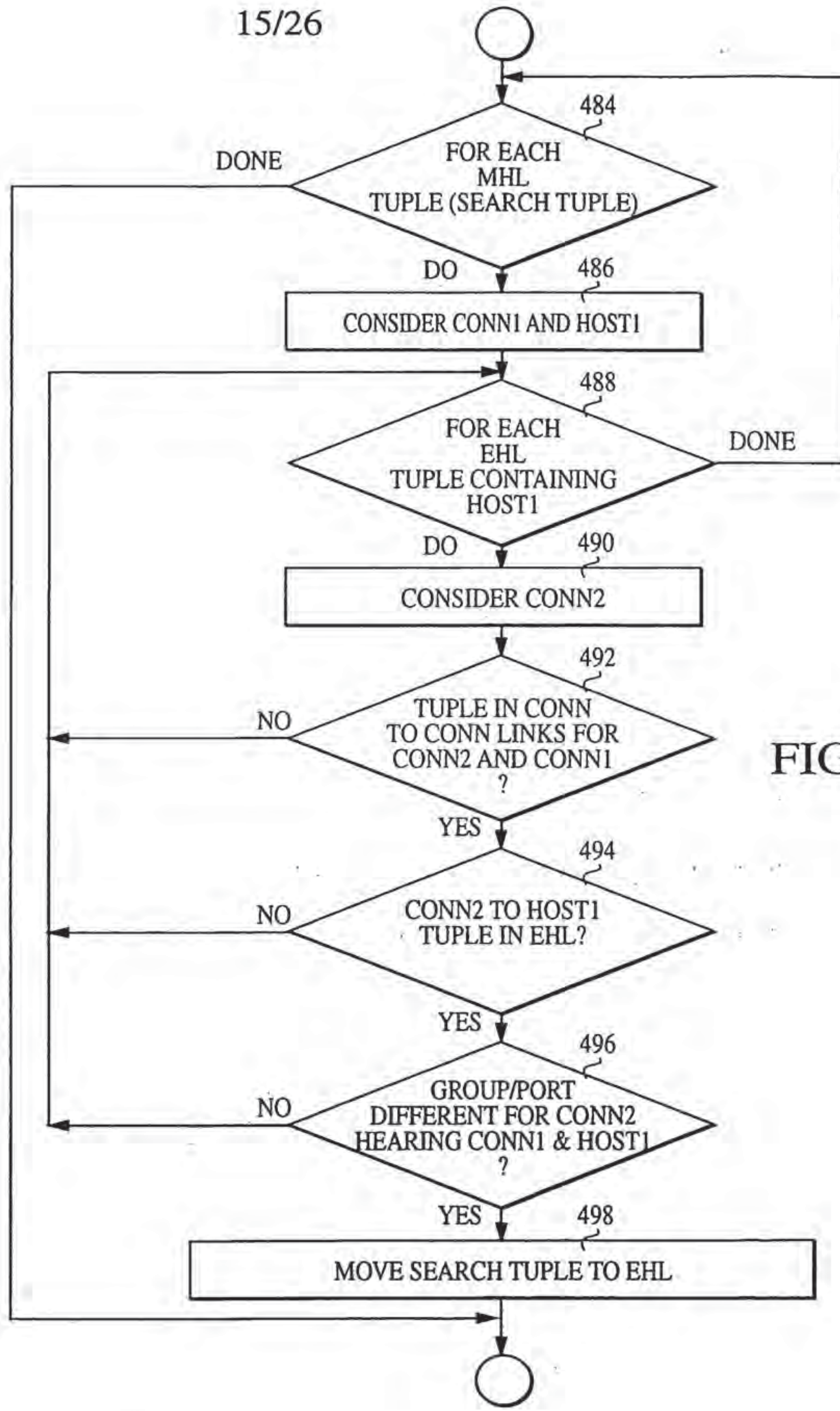
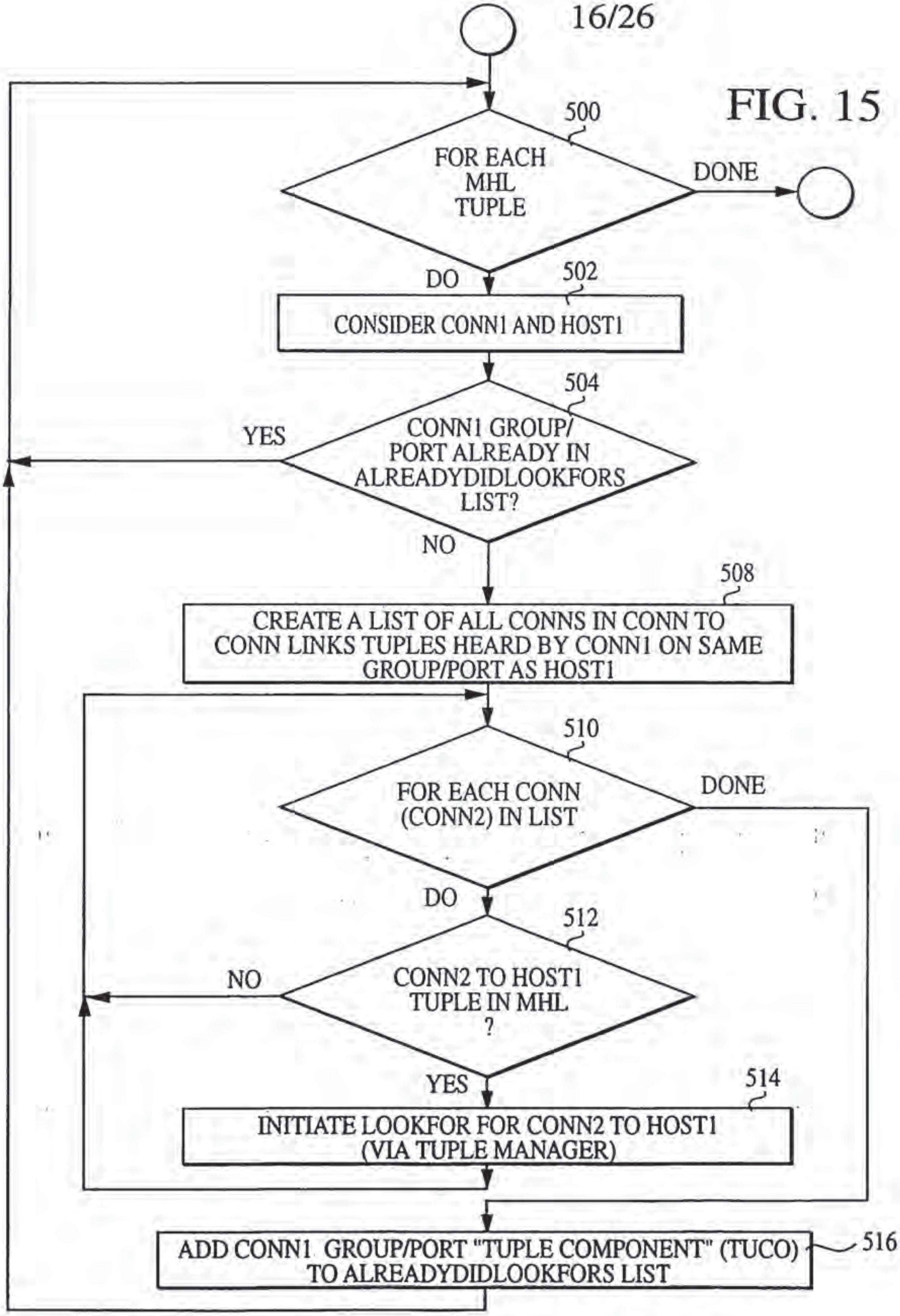

484
FOR EACH
MHL
TUPLE (SEARCH TUPLE)
DONE
DO
486
CONSIDER CONN1 AND HOST1
488
FOR EACH
EHL
TUPLE CONTAINING
HOST1
DONE
DO
490
CONSIDER CONN2
492
TUPLE IN CONN
TO CONN LINKS FOR
CONN2 AND CONN1
?
NO
YES
494
CONN2 TO HOST1
TUPLE IN EHL?
NO
YES
496
GROUP/PORT
DIFFERENT FOR CONN2
HEARING CONN1 & HOST1
?
NO
YES
498
MOVE SEARCH TUPLE TO EHL

FIG. 14

FIG. 15


500
FOR EACH MHL TUPLE
DONE
DO
502
CONSIDER CONN1 AND HOST1
504
YES
CONN1 GROUP/ PORT ALREADY IN ALREADYDIDLOOKFORS LIST?
NO
508
CREATE A LIST OF ALL CONNS IN CONN TO CONN LINKS TUPLES HEARD BY CONN1 ON SAME GROUP/PORT AS HOST1
510
FOR EACH CONN (CONN2) IN LIST
DONE
DO
512
NO
CONN2 TO HOST1 TUPLE IN MHL ?
YES
514
INITIATE LOOKFOR FOR CONN2 TO HOST1 (VIA TUPLE MANAGER)
ADD CONN1 GROUP/PORT "TUPLE COMPONENT" (TUCO) TO ALREADYDIDLOOKFORS LIST
516

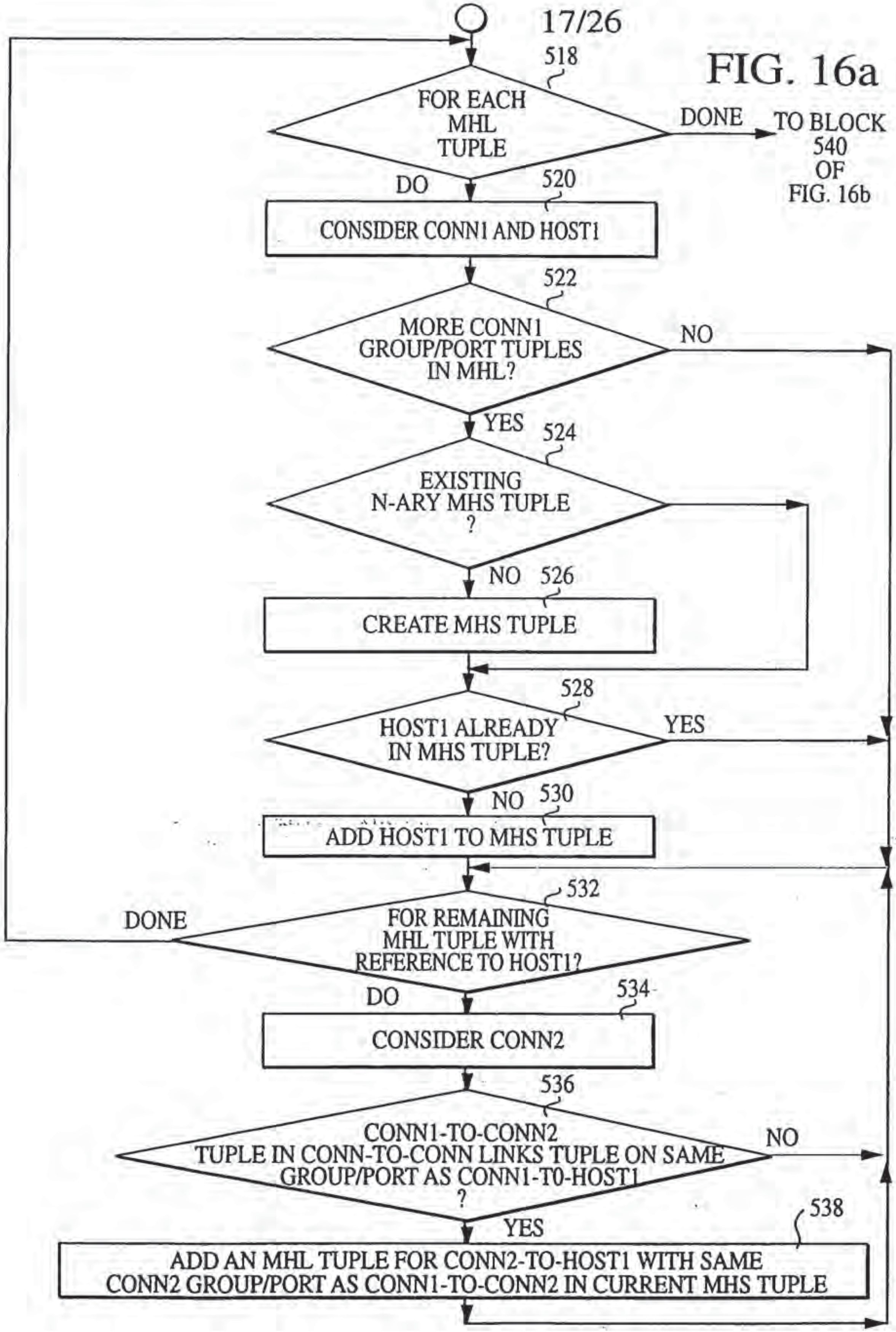

518
FOR EACH MHL TUPLE
DONE
TO BLOCK 540 OF FIG. 16b
DO
520
CONSIDER CONN1 AND HOST1
522
MORE CONN1 GROUP/PORT TUPLES IN MHL?
NO
YES
524
EXISTING N-ARY MHS TUPLE ?
NO
526
CREATE MHS TUPLE
528
HOST1 ALREADY IN MHS TUPLE?
YES
NO
530
ADD HOST1 TO MHS TUPLE
532
DONE
FOR REMAINING MHL TUPLE WITH REFERENCE TO HOST1?
DO
534
CONSIDER CONN2
536
CONN1-TO-CONN2 TUPLE IN CONN-TO-CONN LINKS TUPLE ON SAME GROUP/PORT AS CONN1-T0-HOST1 ?
NO
YES
538
ADD AN MHL TUPLE FOR CONN2-TO-HOST1 WITH SAME CONN2 GROUP/PORT AS CONN1-TO-CONN2 IN CURRENT MHS TUPLE

FIG. 16a

FIG. 16b

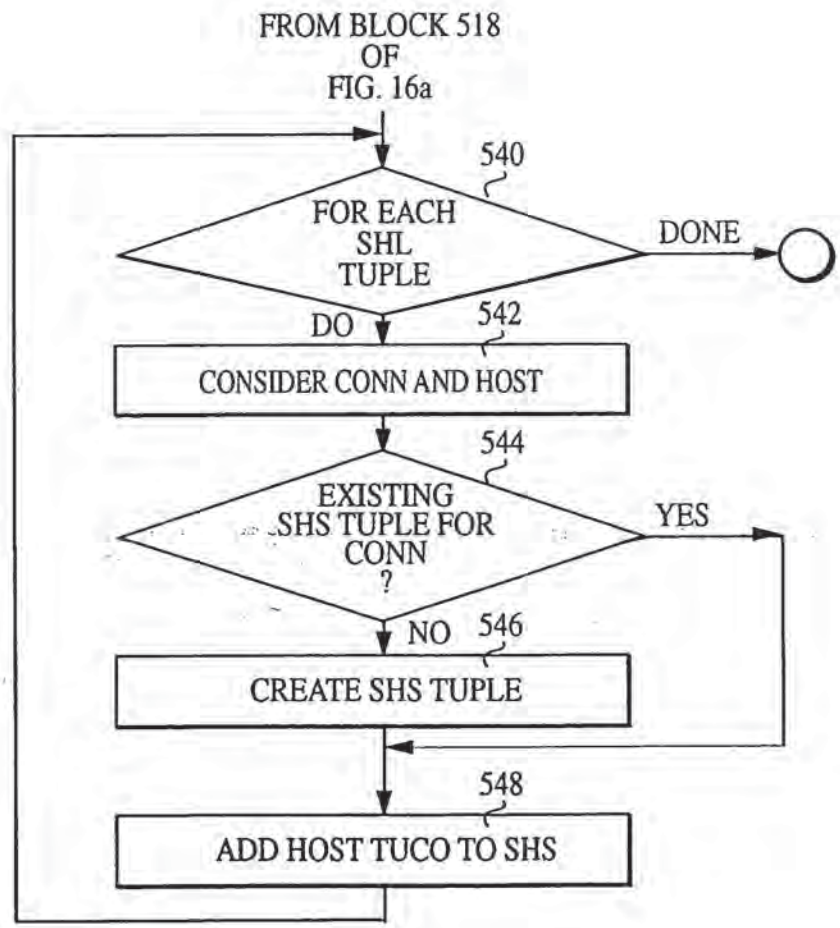

FROM BLOCK 518
OF
FIG. 16a
540
FOR EACH
SHL
TUPLE
DONE
DO
542
CONSIDER CONN AND HOST
544
EXISTING
SHS TUPLE FOR
CONN
?
YES
NO
546
CREATE SHS TUPLE
548
ADD HOST TUCO TO SHS

FIG. 17

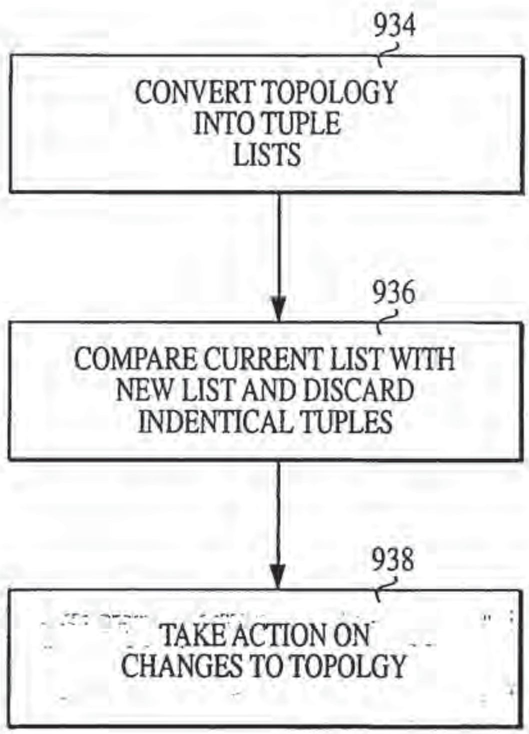
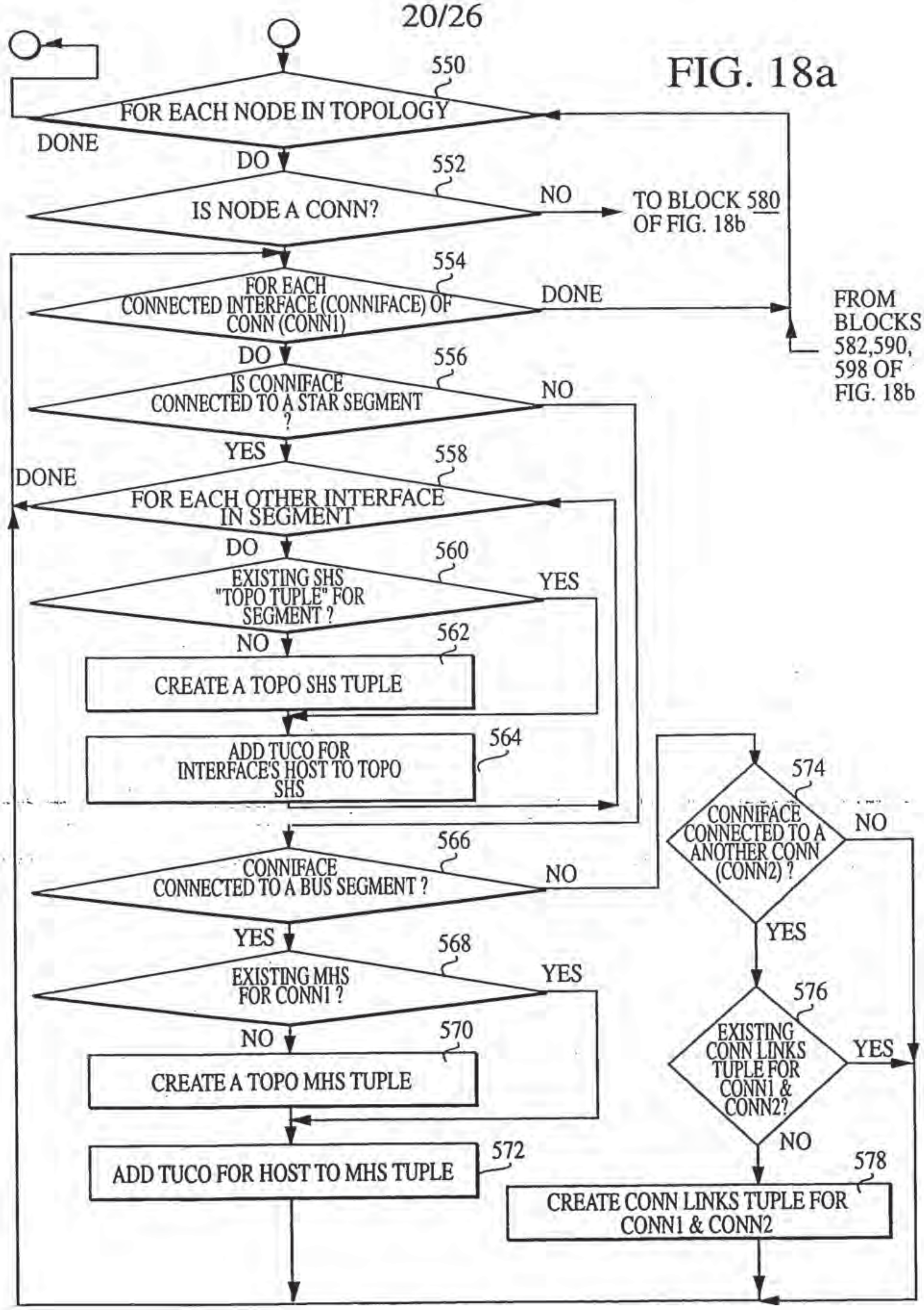

934
CONVERT TOPOLOGY INTO TUPLE LISTS
936
COMPARE CURRENT LIST WITH NEW LIST AND DISCARD INDENTICAL TUPLES
938
TAKE ACTION ON CHANGES TO TOPOLGY

FIG. 18a



550
FOR EACH NODE IN TOPOLOGY
DONE
DO
552
IS NODE A CONN?
NO
TO BLOCK 580 OF FIG. 18b
554
FOR EACH CONNECTED INTERFACE (CONNIFACE) OF CONN (CONN1)
DONE
FROM BLOCKS 582,590, 598 OF FIG. 18b
DO
556
IS CONNIFACE CONNECTED TO A STAR SEGMENT ?
NO
YES
558
DONE
FOR EACH OTHER INTERFACE IN SEGMENT
DO
560
EXISTING SHS "TOPO TUPLE" FOR SEGMENT ?
YES
NO
562
CREATE A TOPO SHS TUPLE
564
ADD TUCO FOR INTERFACE'S HOST TO TOPO SHS
566
CONNIFACE CONNECTED TO A BUS SEGMENT ?
NO
YES
568
EXISTING MHS FOR CONN1 ?
YES
NO
570
CREATE A TOPO MHS TUPLE
572
ADD TUCO FOR HOST TO MHS TUPLE
574
CONNIFACE CONNECTED TO A ANOTHER CONN (CONN2) ?
NO
YES
576
EXISTING CONN LINKS TUPLE FOR CONN1 & CONN2?
YES
NO
578
CREATE CONN LINKS TUPLE FOR CONN1 & CONN2

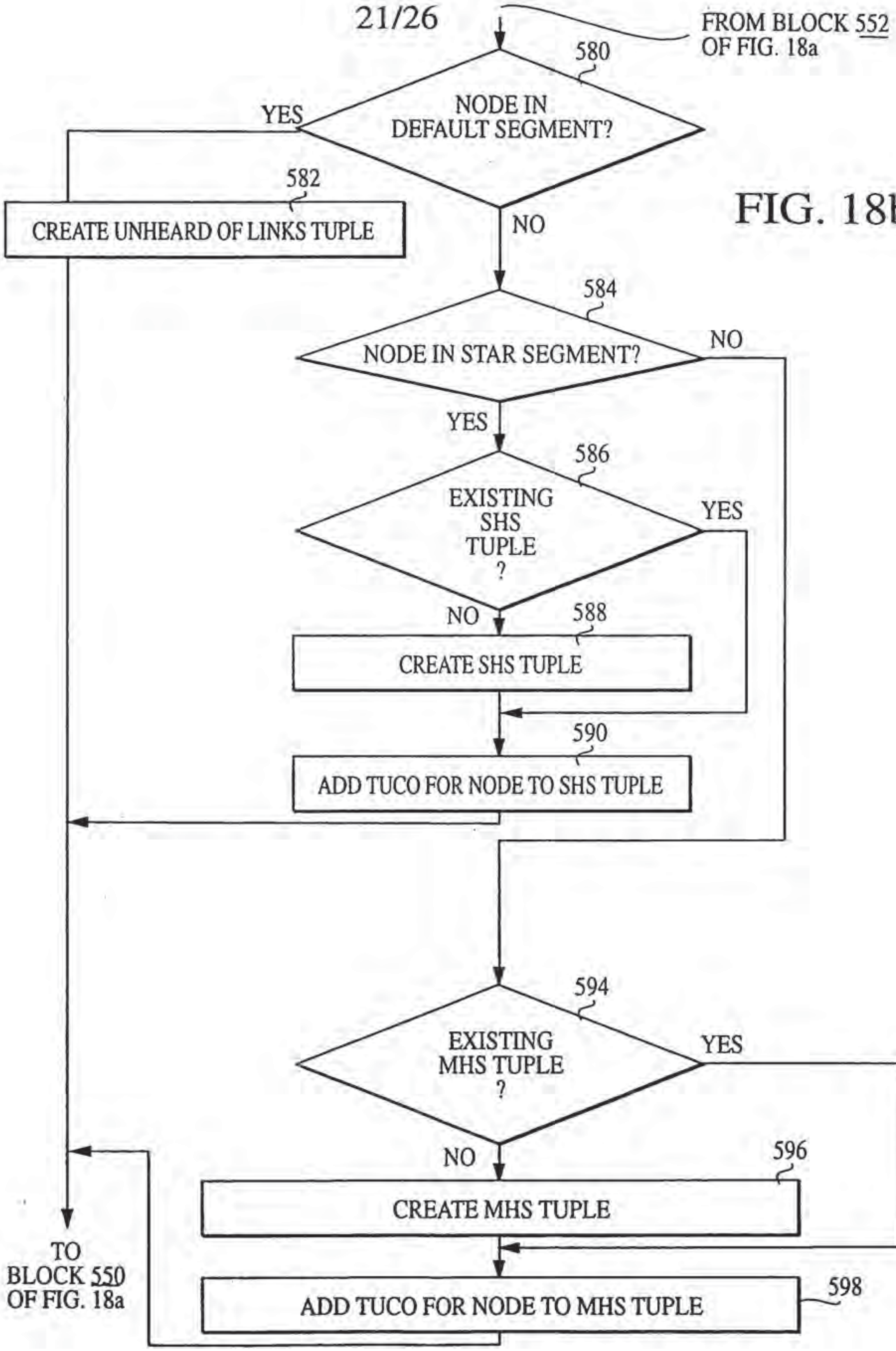

FROM BLOCK 552
OF FIG. 18a
580
NODE IN
DEFAULT SEGMENT?
YES
582
CREATE UNHEARD OF LINKS TUPLE
NO
584
NODE IN STAR SEGMENT?
NO
YES
586
EXISTING
SHS
TUPLE
?
YES
NO
588
CREATE SHS TUPLE
590
ADD TUCO FOR NODE TO SHS TUPLE
594
EXISTING
MHS TUPLE
?
YES
NO
596
CREATE MHS TUPLE
598
ADD TUCO FOR NODE TO MHS TUPLE
TO
BLOCK 550
OF FIG. 18a

FIG. 18b

FIG. 19

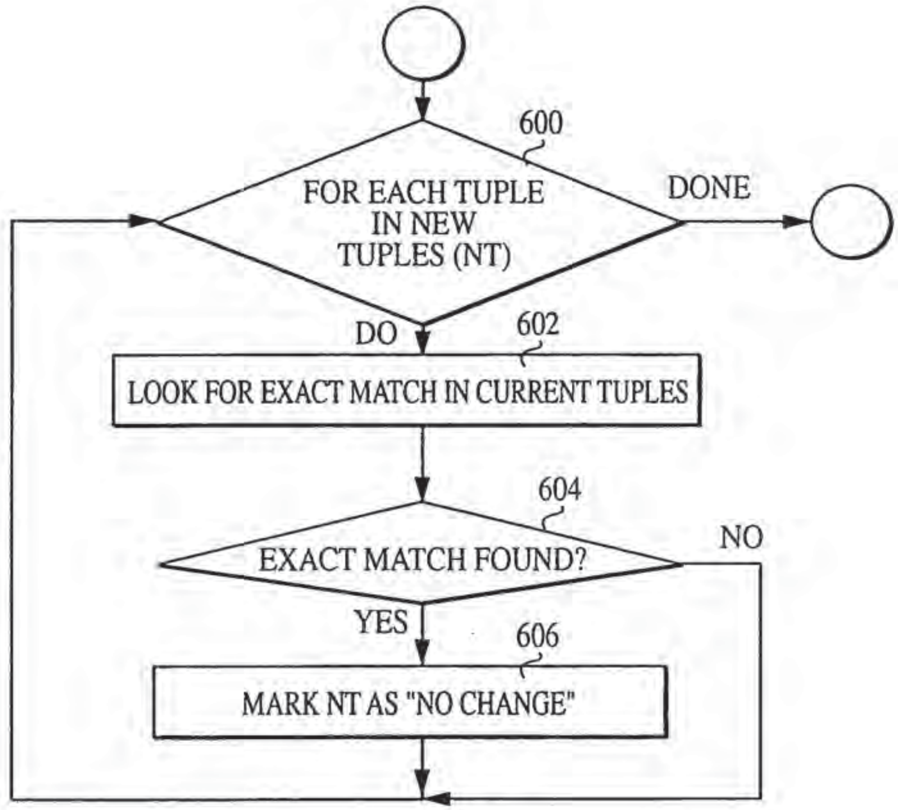
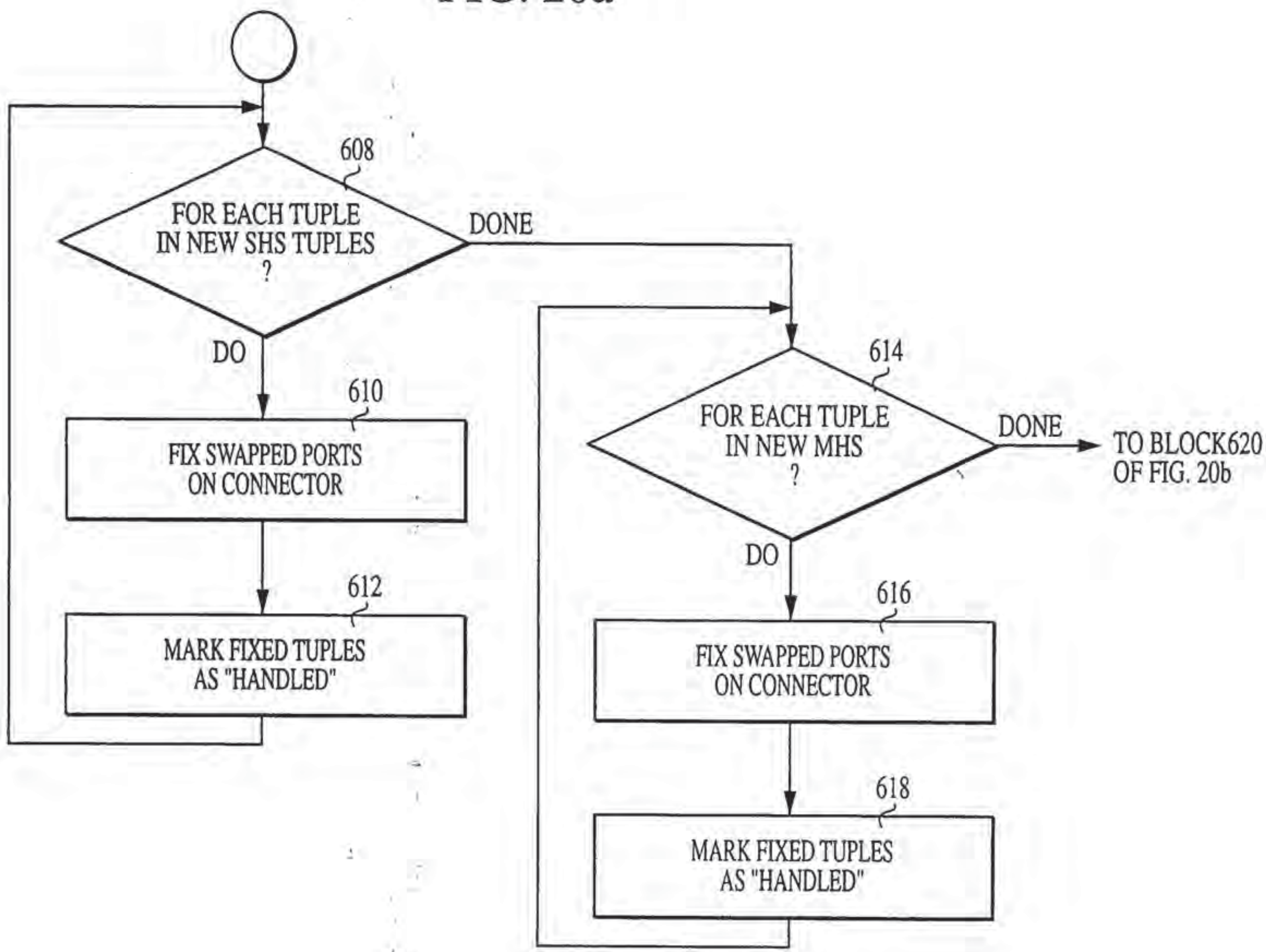

600
FOR EACH TUPLE IN NEW TUPLES (NT)
DONE
DO
602
LOOK FOR EXACT MATCH IN CURRENT TUPLES
604
EXACT MATCH FOUND?
NO
YES
606
MARK NT AS "NO CHANGE"

FIG. 20a


608
FOR EACH TUPLE IN NEW SHS TUPLES ?
DONE
DO
610
FIX SWAPPED PORTS ON CONNECTOR
612
MARK FIXED TUPLES AS "HANDLED"
614
FOR EACH TUPLE IN NEW MHS ?
DONE
TO BLOCK620 OF FIG. 20b
DO
616
FIX SWAPPED PORTS ON CONNECTOR
618
MARK FIXED TUPLES AS "HANDLED"

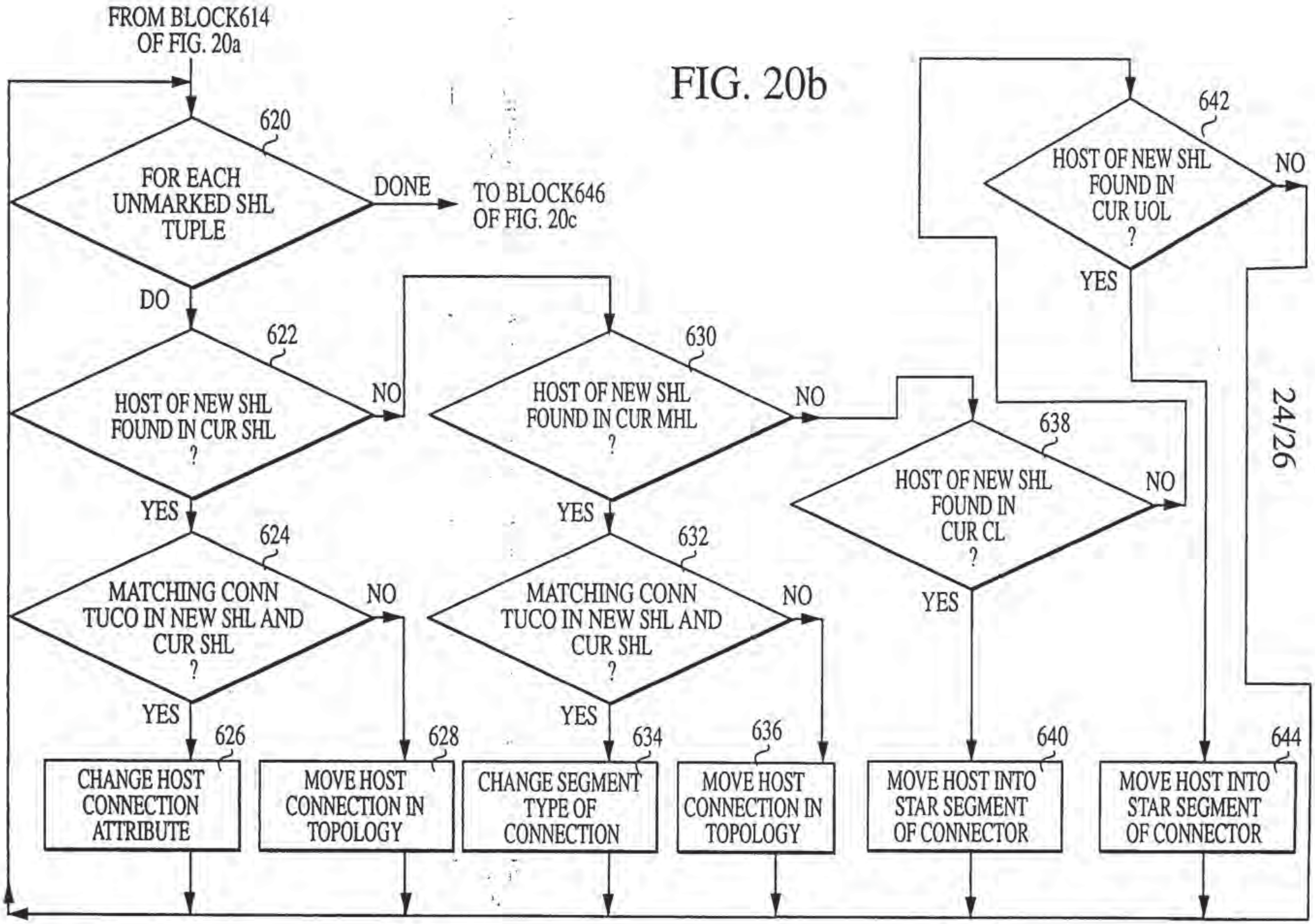
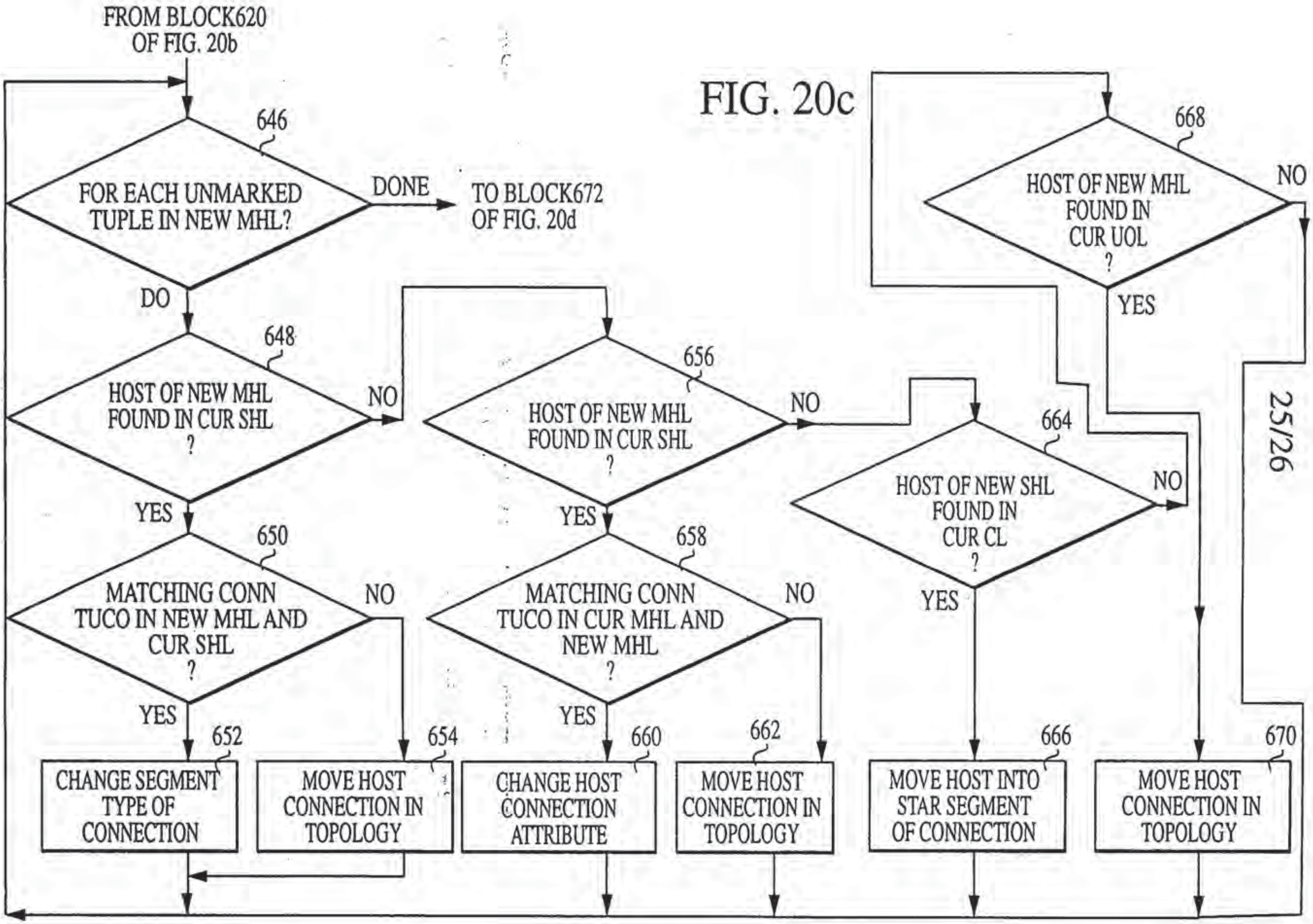

FROM BLOCK614
OF FIG. 20a
620
FOR EACH
UNMARKED SHL
TUPLE
DONE
TO BLOCK646
OF FIG. 20c
DO
622
HOST OF NEW SHL
FOUND IN CUR SHL
?
NO
YES
624
MATCHING CONN
TUCO IN NEW SHL AND
CUR SHL
?
NO
YES
626
CHANGE HOST
CONNECTION
ATTRIBUTE
628
MOVE HOST
CONNECTION IN
TOPOLOGY
630
HOST OF NEW SHL
FOUND IN CUR MHL
?
NO
YES
632
MATCHING CONN
TUCO IN NEW SHL AND
CUR SHL
?
NO
YES
634
CHANGE SEGMENT
TYPE OF
CONNECTION
636
MOVE HOST
CONNECTION IN
TOPOLOGY
638
HOST OF NEW SHL
FOUND IN
CUR CL
?
NO
YES
640
MOVE HOST INTO
STAR SEGMENT
OF CONNECTOR
642
HOST OF NEW SHL
FOUND IN
CUR UOL
?
NO
YES
644
MOVE HOST INTO
STAR SEGMENT
OF CONNECTOR

FIG. 20b

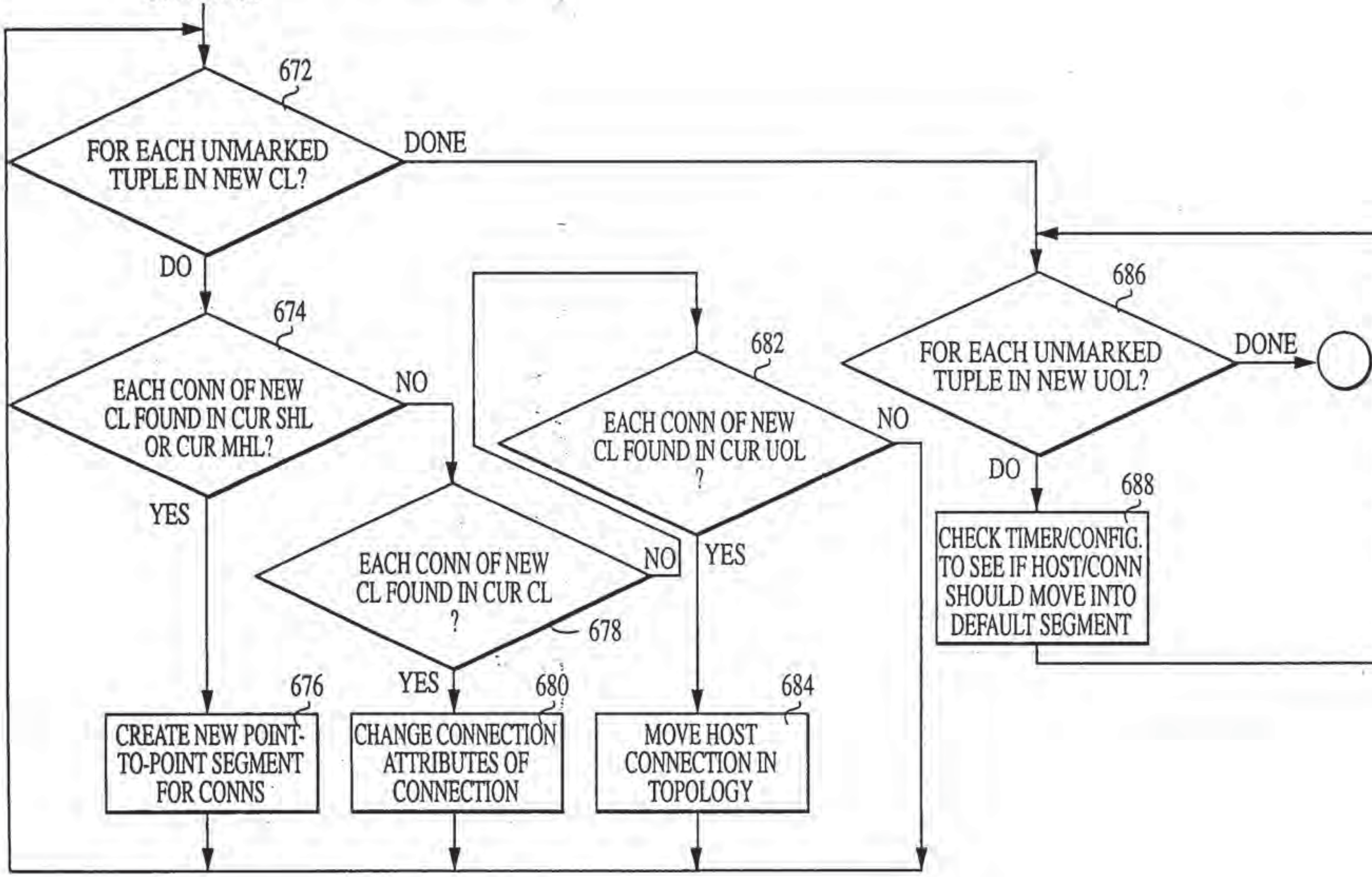


FROM BLOCK620 OF FIG. 20b
646
FOR EACH UNMARKED TUPLE IN NEW MHL?
DONE
TO BLOCK672 OF FIG. 20d
DO
648
HOST OF NEW MHL FOUND IN CUR SHL ?
NO
YES
650
MATCHING CONN TUCO IN NEW MHL AND CUR SHL ?
NO
YES
652
CHANGE SEGMENT TYPE OF CONNECTION
654
MOVE HOST CONNECTION IN TOPOLOGY
656
HOST OF NEW MHL FOUND IN CUR SHL ?
NO
YES
658
MATCHING CONN TUCO IN CUR MHL AND NEW MHL ?
NO
YES
660
CHANGE HOST CONNECTION ATTRIBUTE
662
MOVE HOST CONNECTION IN TOPOLOGY
664
HOST OF NEW SHL FOUND IN CUR CL ?
NO
YES
666
MOVE HOST INTO STAR SEGMENT OF CONNECTION
668
HOST OF NEW MHL FOUND IN CUR UOL ?
NO
YES
670
MOVE HOST CONNECTION IN TOPOLOGY

FIG. 20c

25/26

FIG. 20d


FROM BLOCK646
OF FIG. 20c
672
FOR EACH UNMARKED
TUPLE IN NEW CL?
DONE
DO
674
EACH CONN OF NEW
CL FOUND IN CUR SHL
OR CUR MHL?
NO
YES
678
EACH CONN OF NEW
CL FOUND IN CUR CL
?
NO
YES
682
EACH CONN OF NEW
CL FOUND IN CUR UOL
?
NO
YES
676
CREATE NEW POINT-
TO-POINT SEGMENT
FOR CONNS
680
CHANGE CONNECTION
ATTRIBUTES OF
CONNECTION
684
MOVE HOST
CONNECTION IN
TOPOLOGY
686
FOR EACH UNMARKED
TUPLE IN NEW UOL?
DONE
DO
688
CHECK TIMER/CONFIG.
TO SEE IF HOST/CONN
SHOULD MOVE INTO
DEFAULT SEGMENT

Title

Method and System for Identifying and Processing Changes to a Network Topology

Field of Invention

The present invention relates generally to computer networks. More particularly, it relates to a method and system for identifying changes to a network topology and for acting upon the network based on the changes.

Background

As communications networks, such as the Internet, carry more and more traffic, efficient use of the bandwidth available in the network becomes more and more important. Switching technology was developed in order to reduce congestion and associated competition for the available bandwidth. Switching technology works by restricting traffic. Instead of broadcasting a given data packet to all parts of the network, switches are used to control data flow such that the data packet is sent only along those network segments necessary to deliver it to the target node. The smaller volume of traffic on any given segment results in few packet collisions on that segment and, thus, the smoother and faster delivery of data. A choice between alternative paths is usually possible and is typically made based upon current traffic patterns.

The intelligent routing of data packets with resultant reduction in network congestion can only be effected if the network topology is known. The topology of a network is a description of the network which includes the location of and interconnections between nodes on the network. The word "topology" refers to either the physical or logical layout of the network, including devices, and their connections in relationship to one another. Information necessary to create the topology layout can be derived from tables stored in network devices such as hubs, bridges, and switches. The information in these tables is in a constant state of flux as new entries are being added and old entries time out. Many times there simply is not enough information to determine where to place a particular device.

Switches examine each data packet that they receive, read the source addresses, and log those addresses into tables along with the switch ports on which the packets were received. If a packet is received with a target address without an entry in the switches table, the switch receiving it

broadcasts that packet to each of its ports. When the switch receives a reply, it will have identified where the new node lies.

In a large network with multiple possible paths from the switch to the target node, this table can become quite large and may require a significant amount of the switch's resources to develop and maintain. As an additional complication, the physical layout of devices and their connections are typically in a state of constant change. Devices are continually being removed from, added to, and moved to new physical locations on the network. To be effectively managed, the topology of a network must be accurately and efficiently ascertained, as well as maintained.

Existing mapping methods have limitations that prevent them from accurately mapping topological relationships. Multiple connectivity problems are one sort of difficulty encountered by existing methods. For example, connectors such as routers, switches, and bridges may be interconnected devices in a network. Some existing methods assume that these devices have only a single connection between them. In newer devices, however, it is common for manufacturers to provide multiple connections between devices to improve network efficiency and to increase capacity of links between the devices. The multiple connectivity allows the devices to maintain connection in case one connection fails. Methods that do not consider multiple connectivity do not present a complete and accurate topological map of the network.

Another limitation of existing topology methods is the use of a single reference to identify a device. Existing methods use a reference interface or a reference address in a set of devices to orient all other devices in the same area. These methods assumed that every working device would be able to identify, or "hear," this reference and identify it with a particular port of the device. With newer devices, however, it is possible that the same address or reference may be heard out of multiple ports of the same device. It is also possible that the address or reference may not be heard from any ports, for example, if switching technology is used.

Still another limitation of existing mapping systems is that they require a complete copy of the topological database to be stored in memory. In larger networks, the database is so large that this really is not feasible, because it requires the computer to be very large and expensive.

Still another difficulty with existing systems is that they focus on the minutia without considering the larger mapping considerations. Whenever an individual change in the system is detected, existing methods immediately act on that change, rather than taking a broader view of the change in the context of other system changes. For example, a device may be removed from the network temporarily and replaced with its ports reversed. In existing systems, this swapped port scenario could require hundreds or thousands of changes because the reference addresses will have changed for all interconnected devices.

Still another disadvantage of existing methods is that they use a continuous polling paradigm. These methods continuously poll network addresses throughout the day and make decisions based on those continuous polling results. This creates traffic on the network that slows other processes.

Still another limitation of existing methods is the assumption that network parts of a particular layer would be physically separated from other parts. Network layer 1 may represent the physical cabling of the network, layer 2 may represent the device connectivity, and layer 3 may represent a higher level of abstraction, such as the groupings of devices into regions. Existing methods assume that all layer 3 region groupings are self-contained, running on the same unique physical networking. However, in an internet protocol (IP) network, multiple IP domains may co-exist on the same lower layer networking infrastructure. It has become common for a network to employ a virtual local area network (LAN) to improve security or to simplify network maintenance, for example. Using virtual LANs, a system may have any number of different IP domains sharing the same physical connectivity. As a result, existing methods create confusion with respect to topological mapping because networks with multiple IP addresses in different subnets for the infrastructure devices cannot be properly represented because they assume the physical separation of connectivity for separate IP domains. Still another limitation of existing methods is that they do not allow topological loops, such as port aggregation or trunking, and switch meshing.

Summary of Invention

A method and system are disclosed for mapping the topology of a network having interconnected nodes by identifying changes in the network and updating a stored network topology

based on the changes. The nodal connections are represented by data tuples that store information such as a host identifier, a connector interface, and a port specification for each connection. A topology database stores an existing topology of a network. A topology converter accesses the topology database and converts the existing topology into a list of current tuples. A connection calculator calculates tuples to represent connections in the new topology. The topology converter receives the new tuples, identifies changes to the topology, and updates the topology database using the new tuples. The topology converter identifies duplicate tuples that appear in both the new tuples and the existing tuples and marks the duplicate tuples to reflect that no change has occurred to these connections. The topology converter attempts to resolve swapped port conditions and searches for new singly-heard and multi-heard host link tuples in the list of existing tuples. The topology converter also searches for new conflict link tuples in the existing tuples. The topology converter updates the topology database with the new topology.

Summary of Drawings

Figure 1 is a drawing of a typical topological bus segment for representing the connectivity of nodes on a network.

Figure 2 is a drawing of a typical topological serial segment for representing the connectivity of nodes on a network.

Figure 3 is a drawing of a typical topological star segment for representing the connectivity of nodes on a network.

Figure 4 is a drawing of another typical topological star segment for representing the connectivity of nodes on a network.

Figure 5 is a drawing of the connectivity of an example network system.

Figure 6 is a drawing of the connectivity of another example network system.

Figure 7 is a block diagram of the system.

Figure 8 is a flow chart of the method of the system.

Figure 9 is a flow chart of the method used by the tuple manager.

Figure 10 is a flow chart of the method used by the connection calculator.

Figure 11 is a flow chart of the first weeding phase of the method used by the connection calculator.

Figures 12a-d are flow charts of an infrastructure-building phase of the method used by the connection calculator.

Figure 13 is a flow chart of a second weeding phase of the method used by the connection calculator.

Figure 14 is a flow chart of the noise reduction phase of the method used by the connection calculator.

Figure 15 is a flow chart of the look-for phase of the method used by the connection calculator.

Figures 16a-b are flow charts of the consolidation phase of the method used by the connection calculator.

Figure 17 is a flow chart of the method used by the topology converter.

Figures 18a-b are flow charts of the morph topo phase of the method used by the topology converter.

Figure 19 is a flow chart of the duplication discard phase of the method used by the topology converter.

Figures 20a-d are flow charts of the identify different tuples phase of the method used by the topology converter.

Detailed Description

The system provides an improved method for creating topological maps of communication networks based. Connectivity information is retrieved from the network nodes and stored as "tuples" to track specifically the desired information necessary to map the topology. These light weight data structures may store the host identifier, interface index, and a port. From this tuple information, the topology may be determined. A tuple may be a binary element insofar as it has two parts representing the two nodes on either end of a network link or segment. A "tuco" refers to a tuple component, such as half of a binary tuple.

As used herein, a node is any electronic component, such as a connector or a host, or combination of electronic components with their interconnections. A connector is any network device other than a host, including a switching device. A switching device is one type of connector and refers to any device that controls the flow of messages on a network. Switching devices include, but are not limited to, any of the following devices: repeaters, hubs, routers, bridges, and switches.

As used herein, the term "tuple" refers to any collection of assorted data. Tuples may be used to track information about network topology by storing data from network nodes. In one use, tuples may include a host identifier, interface information, and a port specification for each node. The port specification (also described as the group/port) may include a group number and a port number, or just a port number, depending upon the manufacturer's specifications. A binary tuple may include this information about two nodes as a means of showing the connectivity between them, whether the nodes are connected directly or indirectly through other nodes. A "conn-to-conn" tuple refers to a tuple that has connectivity data about connector nodes. A "conn-to-host" tuple refers to a tuple that has connectivity data about a connector node and a host node. In one use, tuples may have data about more than two nodes; that is, they may be n-ary tuples, such as those used with respect to shared media connections described herein.

A "singly-heard host" (shh) refers to a host, such as a workstation, PC, terminal, printer, other device, etc., that is connected directly to a connector, such as a switching device. A singly-heard host link (shhl) refers to the link, also referred to as a segment, between a connector and an shh. A "multi-heard host" (mhh) refers to hosts that are heard by a connector on the same port that other hosts are heard. A multi-heard host link (mhhl) refers to the link between the connector and an mhh. A link generally refers to the connection between nodes. A segment is a link that may include a shared media connection.

Figure 1 is a drawing of a typical topological bus segment 100 for representing the connectivity of nodes on a network 110. In Figure 1, first and second hosts 121, 122, as well as a first port 131 of a first connector 140 are interconnected via the network 110. The bus segment

100 comprises the first and second hosts 121, 122 connected to the first port 131 of the first connector 140.

Figure 2 is a drawing of a typical topological serial segment 200 for representing the connectivity of nodes on the network 110. In Figure 2, the first host 121 comprises a second port 132 on a second connector 145 which is connected via the network 110 to the first port 131 on the first connector 140. The serial segment 200 comprises the second port 132 on the second connector 145 connected to the first port 131 on the first connector 140. Figure 2 is an example of a connector-to-connector ("conn-to-conn") relationship.

Figure 3 is a drawing of a typical topological star segment 301 for representing the connectivity of nodes on the network 110. In Figure 3, the first host 121 is connected to the first port 131 of the first connector 140. The star segment 301 comprises the first host 121 connected to the first port 131 of the first connector 140. Figure 3 is an example of a connector-to-host ("conn-to-host") relationship.

Figure 4 is a drawing of another typical topological star segment 301 for representing the connectivity of nodes on the network 110. In addition to the connections described with respect to Figure 3, a third host 123 is connected to a third port 133 of the first connector 140 and a fourth host 124 is connected to a fourth port 134 of the first connector 140. In Figure 4, the star segment 301 comprises the first host 121 connected to the first port 131 of the first connector 140, the third host 123 connected to the third port 133 of the first connector 140, and the fourth host 124 connected to the fourth port 134 of the first connector 140. Thus, the star segment 301 comprises, on a given connector, at least one port, wherein one and only one host is connected to that port, and that host. In the more general case, the star segment 301 comprises, on a given connector, all ports having one and only one host connected to each port, and those connected hosts. Since the segments, or links, drawn using the topological methods of Figure 4 resemble a star, they are referred to as star segments.

For illustrative purposes, nodes in the figures described above and in subsequent figures are shown as individual electronic devices or ports on connectors. Also, in the figures the nodes are

represented as terminals. However, they could also be workstations, personal computers, printers, scanners, or any other electronic device that can be connected to networks 110.

Figure 5 is a drawing of the connectivity of an example network system. In Figure 5, first, third, and fourth hosts 121, 123, 124 are connected via the network 110 to first, third, and fourth ports 131, 133, 134 respectively, wherein the first, third, and fourth ports 131, 133, 134 are located on the first connector 140.

The first, third and fourth hosts 121, 123, 124 are singly-heard hosts connected to separate ports 131, 133, 134 of a common connector 140 – the first connector 140. The fifth and sixth hosts 125, 126 are singly-heard hosts connected to the third and fourth connectors 142, 143. The seventh and eighth hosts 127, 128 are multi-heard hosts connected to the same port 139 of the fifth connector 144. The multi-heard hosts 127, 128 illustrate a shared media segment 180, also referred to as a bus 180.

The second, third, fourth, and fifth connectors 141, 142, 143, 144 are interconnected and illustrate a switch mesh 181. Each of the connectors in the switch mesh 181 is connected to each other, either directly or indirectly, to create a fully meshed connection. In the mesh, traffic may be dynamically routed to create an efficient flow.

Figure 5 also shows an example of a port aggregation 182, also referred to as trunking 182. The first connector 140 is connected via the network 110 to the second connector 141 by two direct links, each of which is connected to different ports on the connectors. One link is connected to the sixth port 136 of the first connector 140 and to the seventh port of the second connector 137. The other link is connected to fifth port 135 of the first connector 140 and to the eighth port 138 of the second connector 141. In this example, two connectors illustrate the multiple connectivity between nodes. Depending upon the device specifications, devices such as connectors may be connected via any number of connectors. As explained herein, the system resolves multiple connectivity problems by tracking port information for each connection.

Figure 6 is a drawing of the connectivity of a portion of a network having three connectors 171, 172, 173. A first host 151 is connected directly to the first port 161 of the first connector 171 and the second host 152 is connected to a sixth port 166 of the third connector 173. The second

port 162 of the first connector 171 is connected directly to the third port 163 of the second, or intermediate, connector 172. The fourth port 164 of the intermediate connector 172 is connected directly to the fifth port 165 of the third connector 173.

Figure 7 shows a block diagram of the system. Figure 8 shows a flow chart of the method used by the system to retrieve and update the topology of the network. A tuple manager 300, also referred to as a data miner 300, gathers 902 data from network nodes and builds 904 tuples to update the current topology. The topology database "topodb" 350 stores the current topology for use by the system. The "neighbor data" database 310 stores new tuple data retrieved by the tuple manager 300. The connection calculator 320 processes the data in the neighbor data database 310 to determine the new network topology. The connection calculator 320 reduces 906 the tuple data and sends it to the reduced topology relationships database 330. The topology converter 340 then updates 908 the topology database 350 based on the new tuples sent to the reduced topology relationships database 330 by the connection calculator 320.

Figure 9 shows a flow chart of one operation of the tuple manager 300, as described generally by the data gathering 902 and tuple building 904 steps of the method shown in Figure 8. The tuple manager 300 receives 910 a signal to gather tuple data. The tuple manager 300 then retrieves 912 node information of the current topology stored in the topology database 350. This information tells the tuple manager 300 which devices or nodes are believed to exist in the system based on the nodes that were detected during a previous query. The tuple manager 300 then queries 914 the known nodes to gather the desired information. For example, the connectors may maintain forwarding tables that store connectivity data used to perform the connectors' ordinary functions, such as switching. Other devices may allow the system to perform queries to gather information about the flow of network traffic. This data identifies the devices heard by a connector and the port on which the device was heard. The tuple manager 300 gathers this data by accessing forwarding tables and other information sources for the nodes to determine such information as their physical address, interface information, and the port from which they "hear" other devices. Based on this information, the tuple manager 300 builds 916 tuples and stores 918 them in the "neighbor data" database 310. Some nodes may have incomplete information. In this case, the partial

information is assembled into a tuple and may be used as a "hint" to determine its connectivity later, based on other connections. The tuple manager 300 may also gather 920 additional information about the network or about particular nodes as needed. For example, the connection calculator 320 may require additional node information and may signal the tuple manager 300 to gather that information.

After the data is gathered and the tuples are stored in the neighbor database 310, the connection calculator 320 processes the tuples to reduce them to relationships in the topology. Figure 10 shows a flow chart of the process of the connection calculator 320, as shown generally in the reduction step 906 of the method shown in Figure 8. The connection calculator 320 performs a first weeding phase 922 to identify singly-heard hosts to distinguish them from multi-heard hosts. Singly-heard hosts refer to host devices connected directly to a connector. The connection calculator 320 then performs an infrastructure-building phase 924 to remove redundant connector-to-connector links and to complete the details for partial tuples that are missing information. Then, the connection calculator 320 performs a second weeding phase 926 to resolve conflicting reports of singly-heard hosts. The connection calculator 320 then performs a noise reduction phase 928 to remove redundant neighbor information for connector-to-host links. If clarification of device connectivity is required, the connection calculator 320 performs a "look for" phase 930 to ask the tuple manager 300 to gather additional data. The tuple data is then consolidated 932 into segment and network containment relationships. The connection calculator 320 may also tag redundant tuples to indicate their relevance to actual connectivity. These redundant tuples may still provide hints to connectivity of other tuples. As part of the consolidation phase 932, the connection calculator 320 creates new n-ary tuples (tuples having references to three or more tucos) for shared media segments.

Figure 11 is a flow chart of the connection calculator's first weeding process 922 for distinguishing singly-heard hosts. The purpose of the first weeding process 922 is to identify the direct connections between connectors and hosts; that is, those tuples having a first tuco that is a connector and a second tuco that is a host. The connection calculator 320 looks through the tuple list in the neighbor database 310, and for each tuple 402, the connection calculator 320 determines

404 whether the tuple is a connector-to-host (conn-to-host) link tuple. If it is not a conn-to-host link, the connection calculator 320 concludes 418 that it is a conn-to-conn link and processes 402 the next tuple. If the tuple is a conn-to-host link tuple, then the connection calculator 320 determines 406 whether the connector hears only this particular host on the port identified in the tuple. If the connector hears other hosts on this port, then the tuple is classified 416 as a multi-heard host link (mhhl) tuple.

If the connector hears only the one host on the port – that is, if the host is a singly-heard host – then the connection calculator 320 determines 408 whether the host is heard singly by any other connectors. If no other connectors hear the host as a singly-heard host, then the tuple is classified as a singly-heard host link (shhl) tuple 412 and other tuples for this host are classified 414 as extra host links (ehl). Another tuple for this host may be, for example, an intermediate connector connected indirectly to a host. For example, Figure 6 shows three connectors 171, 172, 173 the first connector is connected directly to the first host 151. This connection therefore forms an shhl tuple. The intermediate connector 172 is indirectly connected to the first host 151. The tuple data indicates that the intermediate connector 172 is indirectly connected to the host and hears the host from a particular port. An extra host links tuple is created so that this data may be used later in conjunction with other extra host links tuples from devices across the network, to verify connectivity by providing hints about connections.

The first weeding process also attempts to identify conflicts. If other connectors hear the host as a singly-heard host, then a conflict arises and the tuple is classified 410 as a singly-heard conflict link (shcl) tuple to be resolved later. This conflict may arise, for example, if a host has been moved within the network, in which case the forwarding table data may no longer be valid. Certain connectors previously connected directly to the host may still indicate that the moved host is connected. When all tuples have been processed 402 to identify singly-heard host links, the first weeding phase 922 is complete.

Figures 12a-d show a flow chart of the infrastructure building phase 924 of the connection calculator 320. The purpose of the infrastructure building phase 924 is to determine how the connectors are set up in the network. The first part of the infrastructure building phase 924

manufactures tuples based on the list of singly-heard host link tuples identified in the first weeding phase 922. The purpose is to identify the relationship between the connectors in the extra host links tuples and the connectors directly connected to the singly-heard hosts. For each singly-heard host link 420, the connection calculator 320 processes 422 each extra host link that refers to the host. In the illustration of Figure 6, a conn-to-conn link tuple would represent the connection between the first connector 171 and the intermediate connector 172. An extra host link tuple would represent the indirect connection between the intermediate connector 172 and the first host 151. The conn-to-conn link tuple between the first connector 171 and the intermediate connector 172 is an example of an ehlConn-to-shhlConn tuple. If a conn-to-conn link tuple exists 424 for the extra host link connector to the singly-heard host link connector (ehlConn-to-shhlConn), then the connection calculator 320 updates 428 the tuple if it is incomplete. It is possible that the tuple data may be incomplete and a conn-to-conn link may not exist. In that case, a conn-to-conn tuple does not exist for the ehlConn-to-shhlConn, then such a tuple is created 426.

After processing extra host links for singly-heard host links, the connection calculator 320 considers 430 each connector (referred to as conn1) in the tuples to determine the relationship between connectors. As illustrated in Figure 6, a single connector may be connected directly and indirectly to multiple other connectors. In Figure 6, the first connector 151 is connected to the intermediate connector 171 directly and also to the third connector 173 indirectly. The third connector 173 hears the first host 151 on the same part 165 that it hears the first connector 171 and the intermediate connector 172. The infrastructure building phase 924 tries to determine the relationship between other connectors heard on the same port of conn1. In a series of interconnected connectors, the connector on one end may not hear a connector on another end, but it may hear intermediate connectors, that in turn hear their own intermediate connectors. Tuples are created to represent the interconnection of conn-to-conn relationships. Based on this data, the connection calculator 320 can make inferences regarding the overall connection between connectors.

For every conn1, the connection calculator 320 considers 432 every other connector (conn2) to determine whether a conn1-to-conn2 tuple exists. If conn1-to-conn2 does not exist,

then the connection calculator 320 considers 436 every other conn-to-conn tuple containing conn2. The other connector on this tuple may be referred to as conn3. If conn2 hears conn3 on a unique port 438 and if conn1 also hears conn3 440, then the connection calculator 320 creates 442 a tuple for conn1-to-conn2 in the connector-to-connector links tuple list.

After processing all of the conn1 tuples, the connection calculator 320 processes 444 each conn1-to-conn2 links tuple to ensure that they have complete port data. For each incomplete tuple 446, the connection calculator 320 looks 448 for a different tuple involving conn1 in the extra host links tupleson a different port. If a different tuple is found 450, then the connection calculator 320 determines 452 whether conn2 also hears the host. If conn2 does hear the host, then the connection calculator 320 completes the missing port data for conn2. If conn2 does not also hear the host 452, then the connection calculator 320 continues looking 448 through different tuples involving conn1 in extra host links on different ports.

After attempting to complete the missing data in each of the conn-to-conn links tuples, the connection calculator 320 processes 456 each conn-to-conn links tuple. The purpose of this sub-phase is to attempt to disprove invalid conn-to-conn links. The connection calculator 320 considers 458 conn1 and conn2 of each conn-to-conn links tuple. Every other connector in conn-to-conn links may be referred to as testconn. For each testconn 460, the connection calculator 320 determines 462 whether the testconn hears conn1 and conn2 on different groups/ports. If testconn hears conn1 and conn2 on different ports, then the tuple is moved to extraconnlinks (ecl) 464. Otherwise, the connection calculator 320 continues processing 460 the remaining testconns.

Figure 13 shows a flow chart of the second weeding phase 926. The purpose of the second weeding phase 926 is to attempt to resolve conflicts involving singly-heard hosts identified in the first weeding phase 922. In the situation described herein in which more than one connector reports that a host is singly-heard, the second weeding phase 926 reviews the tuples created during the infrastructure-building phase 924 involving the connector and host in question and attempts to disprove the reported conflict. The connection calculator 320 processes 466 each singleConflictLinks (scl) tuple (sometimes referred to as the search tuple) and considers 468 conn1 and host1 of the tuple. For each extra host links tuple containing host1 470, the connection

calculator 320 considers 472 conn2 of the tuple. If there is a tuple in conn-to-conn links for conn2 and conn1 474, and if there is a conn2-to-conn1 tuple in the extra host links tuples 476, and if the port is the same for conn2 hearing conn1 and host1 478, then the search tuple is moved 480 into the singly heard host links and other tuples containing host1 are removed 482 from the singleConflictLinks.

Figure 14 shows a flow chart of the noise reduction phase 928. The purpose of the noise reduction phase 928 is to handle those connections in which a connector is not directly connected to a host or to another connector. For example, networking technology may employ shared media connections between connectors, rather than dedicated media connectors. With a shared media connection, the entries in the forwarding tables for connectors attached to the shared media connection will include every node accessing the shared media connection and may not present a useful or accurate representation of the nodal connection. For example, if the network configuration in Figure 6 used a shared media connection between the first connector 171 and the intermediate connector 172, then the first connector is not really connected directly to the intermediate connector because other devices (not shown in Figure 6) may also use the shared media connection. These other devices may include web servers, other connectors, other subnetworks, etc. Tuples will be created for the connectors 171, 172 on opposing ends of the shared media. In this situation, it is inefficient to maintain point-to-point binary tuples for every connection. The noise reduction phase 928 disproves invalid tuples created by the shared media connections.

For each multi-heard host links (mhhl) tuple, also referred to as multiHeardLinks (mhl) tuples (sometimes referred to as the search tuple) 484, conn1 and host1 are considered 486. For each extra host links tuple containing host1 488, conn2 is considered 490. If there is a tuple in conn-to-conn links for conn2 and conn1 492, and if there is a conn2-to-host1 tuple in extraHostLinks 494, and if the group/port for conn2 hearing conn1 and host1 is different 496, then the search tuple is moved 498 to extraHostLinks.

Figure 15 shows a flow chart for the "look for" phase 930. The purpose of this phase is to complete missing data for mhhl tuples. There may exist connections on the network that have incomplete tuple data. For example, the network may simply have no traffic between certain nodes,

in which case data might not be stored in forwarding tables. In another example, a forwarding table may not have sufficient room to store all of the required information and might delete data on a FIFO basis. In the look for phase 930, the connection calculator 320 instructs the tuple manager 300 to query specific nodes to retrieve the missing data. Data that was not stored in a forwarding table on the first interrogation may be present on a subsequent query. For each mhhl tuple 500, the connection calculator 320 considers 502 conn1 and host1. If the conn1 group/port is already in an "alreadyDidLookfors" list, then a list is created 508 for all connectors in conn-to-conn links that are heard by conn1 on the same group/port as host1. For each connector (conn2) in the list 510, the connection calculator 320 determines 512 whether there is a conn2-to-host1 tuple in the mhhl tuples. If there is not such a tuple, then the connection calculator 320 initiates a look-for for conn2-to-host1 via the tuple manager 300. When each connector in the list has been processed 510, the conn1 group/port tuco is added 516 to an alreadyDidLookfors list. As an additional portion of the look for phase 930 (not shown in figures) the system may ask a user to verify or clarify information about connectivity. For example, the system may show the user the perceived connectivity or the unresolved connectivity issues and request the user to add information as appropriate.

The connection calculator 330 process described above collects the tuple information from the tuple manager 300, builds tuples new tuples and removes redundant or unnecessary tuples to produce the new topology. This topology may have incomplete tuples possibly resulting from extraneous information that the connection calculator 330 could not disprove. To refine the new topology, the connection calculator 330 can request the tuple manager 300 to obtain additional information about particular nodes or it may also request a user to refine the topology by adding or removing tuples. Using the process of the connection calculator 330, tuples marked as non-essential may be removed from the new topology to save space and to simply the topology. The connection calculator 330 is not confused by multiple connectivity situations such as port aggregation 182 or switch meshing 181 as shown in Figure 5, because the tuples represent point-to-point, or neighbor-to-neighbor, connectivity showing each connection in the network. This point-to-point connectivity concept also helps enable the system to avoid difficulties that occur in

systems that track higher levels of abstraction, such as layer 3 connectivity. Also, the tuples may contain only selected information to minimize the storage space required for the topology.

Figures 16a-b show a flow chart of the consolidation phase 932. The purpose of this phase is to consolidate the tuples that involve shared media connections. After the noise reduction phase 928, a considerable number of tuples involving shared media may remain. Rather than maintain a binary tuple for each of the connections, an n-ary tuple is created for the link using a tuco for each connector and each host connected thereto. For each mhhl tuple 518, conn1 and host1 are considered 520. If there are more conn1 group/port tuples in multiHeardLinks, and if are not any n-ary multiHeardSegments (mhs) tuples 524, then an mhs tuple is created 526. If host1 is not already in this particular mhs tuple 528, then conn2 of the tuple is considered 534. If there is a conn1-to-conn2 conn-to-connLinks tuple on the same port as conn1-to-host1 536, then all multiHeardLinks tuples for conn2-to-host1 with the same conn2 group/port as the conn1-to-conn2 are added 538 to the current mhs tuple.

After processing each mhhl tuple 518, each singly-heard host links (shhl) tuple, also referred to as a singlyHeardLinks (shl) tuple, is considered 540. For each shhl tuple, the connector and host are considered 542. If there is no existing singlyHeardSegments (shs) tuple for the connector 544, then an shs tuple is created 546. The host tuco is then added to the shs 548.

Figure 17 shows a flow chart of the method used by the topology converter 340, as described generally by the topology update step 908 of the method shown in Figure 8. The topology converter 340 converts 934 the topology into tuple lists, also referred to as the "morph topo" phase 934. It then compares 936 the list from the topology currently stored in the topology database 350 with the new list generated by the connection calculator 320 and discards 936 identical tuples in what is also referred to as the "discard duplicates" phase 936. It then takes action 938 on the changes in the topology as determined by the changes in the tuple lists, in what is also referred to as the "identify different tuples" phase 938.

Figure 18a shows a flow chart for the "morph topo" phase 934. For each node in the topology 550, the topology converter 340 determines 552 whether the node is a connector. If the node is a connector, then for each connected interface (conniface) of the connector (conn1) 554,

the topology converter 340 determines 556 whether the conniface is connected to a star segment. If it is connected to a star segment, then for every other interface in the segment 558, the topology converter 340 determines 560 whether there is an existing shs tuple, referred to as the "topo tuple" for the segment. If there is no such tuple, then the topology converter 340 creates 562 a topo shs tuple. The tuco for the interface's host-to-topo shs is then added 564 to the topo shs tuple.

If the connector node is not connected to a star segment 556 and is connected to a bus segment 566, the topology converter 340 determines 568 whether there is an existing mhs tuple for conn1. If there is not an existing mhs tuple for conn1, then a topo mhs tuple is created 570. A tuco is added 572 for the host to the mhs tuple.

If the connector node is not connected to either a star segment 556 or to a bus segment 566, then the topology converter knows that it is connected to another connector (conn2). If such a connector does not already have an existing connLinks tuple for conn1 and conn2 576, then a connLinks tuple is created 578. After processing the bus segment, star segment, and conn-to-conn segment, for each conniface 554, the topology converter 340 proceeds to the next node 550.

Figure 18b shows a continuation of the flow chart of Figure 18a showing the steps of the method when the topology converter 340 determines that the node is not a connector 552. If the node is in the default segment, then an "unheardOfLinks" tuple is created 582 and the topology converter proceeds to the next node 550. If the node is not in the default segment 580, then the topology converter 340 determines whether the node is in a star segment 584. If the node is in a star segment, then if there is not already an shs tuple, the topology converter 340 creates 588 an shs tuple. The tuco for the node is then added 590 to the shs tuple, and the topology converter 340 proceeds to the next node 550.

If the node is not in a star segment, then the topology converter 340 knows that it is in the bus segment. If there is not already an mhs tuple for the node, 594, then the topology converter 340 creates 596 an mhs tuple. The tuco for the node is then added 598 to the mhs tuple, and the topology converter proceeds to the next node 550.

Figure 19 shows a flow chart for the discard duplicates phase 936 of the topology converter 340. For each tuple in the new tuples (nt) 600, the topology converter looks for 602 an

exact match in the current tuples stored in the topodb. If an exact match is found 604, then the new tuple is marked 606 as "no change" indicating that this is an identical tuple.

Figures 20a-d show a flow chart for the identify different tuples phase 938. The system looks through each tuple in the new SinglyHeardSegments (newSHS) tuple list 608 and tries to identify and fix 610 swapped ports on connectors. Swapped ports are identified by considering those segment tuples in both the new topology and the existing topology that differ only by the port specification in the tuco. Each tuple that is fixed as a swapped port is marked 612 as "handled." The system also looks through each tuple in the new multiHeardSegments tuple list (newMHS) 614 and tries to identify and fix 616 swapped ports on connectors. Each tuple that is fixed as a swapped port is marked 618 as "handled."

The system then processes 620 each unmarked tuple in the newSHL tuples. Four cases are possible for the host of the newSHL tuples. The host of the newSHL can be found in the current singlyHeardLinks (curSHL) 622, the current multiHeardLinks (curMHL) 630, the current connLinks (curCL) 638, or the current UnheardOfLinks (curUOL) 642. If the host of a newSHL tuple is found 622 in the current SinglyHeardLinks (curSHL) tuples, then the system determines 624 if there is a matching connector tuco between the newSHL tuples and the curSHL tuples. If there is a matching tuco, then the system changes 626 the host connection attribute. If there is not a matching tuco, then the host connection is moved 628 in the topology.

If the host is found in the curMHL tuples 630, then the system determines 632 whether there is a matching connector tuco between the newSHL tuples and the curSHL tuples. If there is a matching connector, then the segment type of connection is changed 634. If there is not a matching connector, then the host connection is moved 636 in the topology. If the host is found in the curCL tuples 638, then the host is moved 640 into a star segment of the connector. If it is found in the curUOL 642, then the host is moved 644 into the star segment of the connector.

Figure 20c shows another stage of the processing undertaken during the identify different tuples phase 938. For each unmarked tuple in the new multiHeardLinks tuples (newMHL) 946, four cases are possible for the host of the newMHL. The host of the newMHL may be found in the curSHL 648, the curMHL 656, the curCL 664, or the curUOL 668. If the host is found in the

curSHL 648, then the system determines 650 whether there is a matching connector tuco between the newMHL and the curMHL. If there is a matching tuco, then the segment type of connection is changed 652. If there is not a matching tuco, then the host connection is moved 654 in the topology.

If the host is found in the curMHL tuples 656, then the system determines 658 whether there is a matching connector tuco in both the curMHL tuples and the newMHL tuples. If there is a matching connector tuco, then the host connection attribute is changed 660. If there is not a matching tuco, then the host connection is moved 662 in the topology. If the host is found in the curCL tuples 664, then the host is moved into a bus segment of a connector. If the host is found in the curUOL tuples 668, then the host connection is moved 670 in the topology.

Figure 20d shows another portion of the identify different tuples phase 938. For each unmarked tuple in the newCL tuples 672, there are three possibilities for the connector. The connector of the unmarked tuple in newCL can be found in the curSHL or curMHL 674, in the curCL 678, or in the curUOL 682. If each connector is found in the curSHL or curMHL list 674, then the system creates 676 a new point-to-point segment for the connectors. If the connectors are found in the curCL 678, then the connection attributes of the connectors are changed 680. If each connector is found in the curUOL tuples 682, then the host connection is moved 684 in the topology.

Another part of the identify different tuples phase 938 is shown in blocks 686 and 688 of Figure 20d. For each unmarked tuple in the newUOL tuples 686, the system checks 688 the timer/configuration to determine whether the host/conn should move into the default segment from its current segment.

An advantage of the system is that it may be schedulable. The system may map network topology continuously, as done by existing systems, or it may be scheduled to run only at certain intervals, as desired by the user. A further advantage of the system is that it is capable of processing multiple connections between the same devices and of processing connection meshes, because it tracks each nodal connection independently, without limitations on the types of connections that are permitted to exist.

Although the present invention has been described with respect to particular embodiments thereof, variations are possible. The present invention may be embodied in specific forms without departing from the essential spirit or attributes thereof. It is desired that the embodiments described herein be considered in all respects illustrative and not restrictive and that reference be made to the appended claims for determining the scope of the invention.

Claims

1. In a network having interconnected nodes with data tuples that represent nodal connections, a method for mapping a network topology by identifying changes between an existing topology and a new topology, the method comprising:

converting an existing topology into a list of existing tuples that represent existing nodal connections;

receiving new tuples that represent new nodal connections; and

comparing the list of existing tuples with the new tuples to identify changes to the topology.

2. The method of claim 1, further comprising updating a topology database with a new topology.

3. The method of claim 1, further comprising taking action on the changes to the topology.

4. The method of claim 1, wherein the tuples include information about a host identifier, a connector interface, and a port specification.

5. The method of claim 1, wherein the step of comparing comprises identifying duplicate tuples that appear both in the list of existing tuples and in the new tuples, and maintaining a current status of the topology for these tuples.

6. The method of claim 1, wherein the step of comparing comprises identifying a swapped port condition on a connector.

7. The method of claim 1, wherein the step of comparing comprises searching for a host of a new singly-heard host link tuple or a new multi-heard host link tuple in the list of existing tuples.

8. A system for mapping a network topology by identifying changes between an existing topology and a new topology, based on changes to data tuples that represent nodal connections comprising:

a topology database that stores an existing topology of a network; and

a topology converter connected to the topology database that receives new tuples that represent new nodal connections; and compares the new tuples with the existing topology to identify changes in the network.

9. The system of claim 8, wherein the topology converter converts the existing topology into a list of existing tuples that represent existing nodal connections.

10. The system of claim 8, wherein the topology converter updates the topology database with a new topology based on the new tuples.

11. The system of claim 8, wherein the topology converter attempts to identify swapped ports on connectors.

12. The system of claim 8, wherein the topology converter identifies duplicate tuples that appear both in the list of existing tuples and in the new tuples, and maintains a current status of the topology for these tuples.

13. The system of claim 8, wherein the topology converter searches for a host of a new singly-heard host link tuple or a new multi-heard host link tuple in the list of existing tuples.

14. The system of claim 8, wherein the topology converter searches for a connector of a new conflict links tuple in the list of existing tuples.

15. A computer-readable medium having computer-executable instructions for performing a method for mapping a network topology by identifying changes between an existing topology and a new topology in a network having a interconnected nodes, the method comprising:

converting an existing topology into a list of existing tuples that represent existing nodal connections;

receiving new tuples that represent new nodal connections;

comparing the list of existing tuples with the new tuples to identify changes to the topology; and

updating a topology database with a new topology.

16. The method of claim 15, wherein a topology converter receives the new tuples from a connection calculator that calculates connections between nodes.

17. The method of claim 15, wherein the step of comparing comprises identifying duplicate tuples that appear both in the list of existing tuples and in the new tuples, and maintaining a current status of the topology for these tuples.

18. The method of claim 15, wherein the step of comparing comprises identifying a swapped port condition on a connector.

19. The method of claim 15, wherein the step of comparing comprises searching for a host of a new singly-heard host link tuple or a new multi-heard host link tuple in the list of existing tuples.

20. The method of claim 15, wherein the step of comparing comprises searching for a connector of a new conflict links tuple in the list of existing tuples.

Abstract

A method and system are disclosed for mapping the topology of a network having interconnected nodes by identifying changes in the network and updating a stored network topology based on the changes. The nodal connections are represented by data tuples that store information such as a host identifier, a connector interface, and a port specification for each connection. A topology database stores an existing topology of a network. A topology converter accesses the topology database and converts the existing topology into a list of current tuples. A connection calculator calculates tuples to represent connections in the new topology. The topology converter receives the new tuples, identifies changes to the topology, and updates the topology database using the new tuples. The topology converter identifies duplicate tuples that appear in both the new tuples and the existing tuples and marks the duplicate tuples to reflect that no change has occurred to these connections. The topology converter attempts to resolve swapped port conditions and searches for new singly-heard and multi-heard host link tuples in the list of existing tuples. The topology converter also searches for new conflict link tuples in the existing tuples. The topology converter updates the topology database with the new topology.

PATENT APPLICATION

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

ATTORNEY DOCKET NO. 10008102-1

As a below named inventor, I hereby declare that:

My residence/post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

Method And System For Identifying And Processing Changes To A Network Topology

the specification of which is attached hereto unless the following box is checked:

( ) was filed on __________ as US Application Serial No. or PCT International Application Number __________ and was amended on __________ (if applicable).

I hereby state that I have reviewed and understood the contents of the above-identified specification, including the claims, as amended by any amendment(s) referred to above. I acknowledge the duty to disclose all information which is material to patentability as defined in 37 CFR 1.56.

Foreign Application(s) and/or Claim of Foreign Priority

I hereby claim foreign priority benefits under Title 35, United States Code Section 119 of any foreign application(s) for patent or inventor(s) certificate listed below and have also identified below any foreign application for patent or inventor(s) certificate having a filing date before that of the application on which priority is claimed:

COUNTRY	APPLICATION NUMBER	DATE FILED	PRIORITY CLAIMED UNDER 35 U.S.C. 119
N/A			YES: ___ NO: ___
			YES: ___ NO: ___

Provisional Application

I hereby claim the benefit under Title 35, United States Code Section 119(e) of any United States provisional application(s) listed below:

APPLICATION SERIAL NUMBER	FILING DATE
N/A	

U. S. Priority Claim

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code Section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, Section 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

APPLICATION SERIAL NUMBER	FILING DATE	STATUS (patented/pending/abandoned)
N/A		

09703246 . 103100

POWER OF ATTORNEY:

As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith:

Customer Number 022879

Place Customer Number Bar Code Label here

Send Correspondence to:
HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

Direct Telephone Calls To:

T. Grant Ritz

(970) 898-0697

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Inventor: Eric A Pulsipher **Citizenship:** US

Residence: 2937 Redburn Drive Ft Collins CO 80525

Post Office Address: Same as residence

Inventor's Signature **Date** 10/31/2000

Rev 10/00 (DecPwr) (Use Page Two For Additional Inventor(s) Signature(s)) Page 1 of 2

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION (continued)

ATTORNEY DOCKET NO. 10008102-1

Full Name of # 2 joint inventor: Joseph R Hunt JH Citizenship: US

Residence: ~~5841 Meadow Creek Ln~~ 5739 Hahns Peak Dr #205 Loveland, CO 80538

Post Office Address: Same as Residence

Inventor's Signature: Joseph R Hunt Date: 10/31/00

Full Name of # 3 joint inventor: ______ Citizenship: ______

Residence: ______

Post Office Address: ______

Inventor's Signature ______ Date ______

Full Name of # 4 joint inventor: ______ Citizenship: ______

Residence: ______

Post Office Address: ______

Inventor's Signature ______ Date ______

Full Name of # 5 joint inventor: ______ Citizenship: ______

Residence: ______

Post Office Address: ______

Inventor's Signature ______ Date ______

Full Name of # 6 joint inventor: ______ Citizenship: ______

Residence: ______

Post Office Address: ______

Inventor's Signature ______ Date ______

Full Name of # 7 joint inventor: ______ Citizenship: ______

Residence: ______

Post Office Address: ______

Inventor's Signature ______ Date ______

Full Name of # 8 joint inventor: ______ Citizenship: ______

Residence: ______

Post Office Address: ______

Inventor's Signature ______ Date ______

Rev 10/00 (DecPwr) (Use Page Two For Additional Inventor(s) Signature(s))

UNITED STATES PATENT AND TRADEMARK OFFICE

COMMISSIONER FOR PATENTS
UNITED STATES PATENT AND TRADEMARK OFFICE
WASHINGTON, D.C. 20231
www.uspto.gov

Bib Data Sheet

SERIAL NUMBER	FILING DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
09/703,942	10/31/2000 RULE –	370	2661	10008102-1

APPLICANTS

Eric A. Pulsipher, Ft Collins, CO ;
Joseph R. Hunt, Loveland, CO ;

** CONTINUING DATA *************************

** FOREIGN APPLICATIONS ********************

None / WCS

IF REQUIRED, FOREIGN FILING LICENSE GRANTED ** 02/05/2001 –

	STATE OR COUNTRY	SHEETS DRAWING	TOTAL CLAIMS	INDEPENDENT CLAIMS
Foreign Priority claimed ☐ yes ☒ no 35 USC 119 (a-d) conditions met ☐ yes ☒ no ☐ Met after Allowance Verified and Acknowledged WCS Examiner's Signature Initials	CO	26	20	3

ADDRESS

022879 –

TITLE

Method and system for identifying and processing changes to a network topology

FILING FEE RECEIVED 710	FEES: Authority has been given in Paper No. ________ to charge/credit DEPOSIT ACCOUNT No. ________ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other ________ ☐ Credit

PATENT APPLICATION SERIAL NO. ______________

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

11/06/2000 KZEWDIE 00000062 082025 09703942
01 FC:101 710.00 CH

PTO-1556
(5/87)

*U.S. GPO: 1999-459-082/19144

Application or Docket Number 09/703,942

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2000

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	20	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	20 minus 20=	* 0
INDEPENDENT CLAIMS	3 minus 3 =	* 0
MULTIPLE DEPENDENT CLAIM PRESENT		☐

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE ☐ RATE	FEE	OR	OTHER THAN SMALL ENTITY RATE	FEE
BASIC FEE	355.00	OR	BASIC FEE	710.00
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL		OR	TOTAL	

CLAIMS AS AMENDED - PART II

AMENDMENT A

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

SMALL ENTITY RATE	ADDITIONAL FEE	OR	OTHER THAN SMALL ENTITY RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE	OR	RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE	OR	RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
***If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

FORM PTO-875
(Rev. 8/00)

Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE
*U.S. GPO: 2000-460-705/30103

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P. O. Box 272400
Fort Collins, Colorado 80527-2400

OIPE AUG 13 2002 PATENT & TRADEMARK OFFICE

PATENT APPLICATION

ATTORNEY DOCKET NO. 10008102-1

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Eric A. PULSIPHER et al. **Confirmation No.:** #2

Application No.: 09/703,942 **Examiner:** Wellington Chin

Filing Date: 10/31/2000 **Group Art Unit:** 2664

Title: METHOD AND SYSTEM FOR IDENTIFYING AND PROCESSING CHANGES TO A NETWORK TOPOLOGY

RECEIVED
AUG 14 2002
Technology Center 2600

COMMISSIONER FOR PATENTS
Washington, D.C. 20231

INFORMATION DISCLOSURE STATEMENT

Sir:

This Information Disclosure Statement is submitted:

(X) under 37 CFR 1.97(b), or
(Within three months of filing national application; or date of entry of national application; or before mailing date of first office action on the merits; whichever occurs last)

( ) under 37 CFR 1.97(c) together with either a:
( ) Statement under 37 CFR 1.97(e), or
( ) a $180.00 fee under 37 CFR 1.17(p), or
(After the CFR 1.97 (b) time period, but before final action or notice of allowance, whichever occurs first)

( ) under 37 CFR 1.97 (d) together with a:
( ) Statement under 37 CFR 1.97(e), and
( ) a petition under 37 CFR 1.97(d)(2), and
( ) a $180.00 petition fee set forth in 37 CFR 1.17(p).
(Filed after final action or notice of allowance, whichever occurs first, but before payment of the issue fee)

Please charge to Deposit Account **08-2025** the sum of $0.00 . At any time during the pendency of this application, please charge any fees required or credit any overpayment to Deposit Account **08-2025** pursuant to 37 CFR 1.25.

(X) Applicant(s) submit herewith Form PTO 1449 - Information Disclosure Citation together with copies, of patents, publications or other information of which applicant(s) are aware, which applicant(s) believe(s) may be material to the examination of this application and for which there may be a duty to disclose in accordance with 37 CFR 1.56.

( ) A concise explanation of the relevance of foreign language patents, foreign language publications and other foreign language information listed on PTO Form 1449, as presently understood by the individuals(s) designated in 37 CFR 1.56 (c) most knowledgeable about the content is given on the attached sheet, or where a foreign language patent is cited in a search report or other action by a foreign patent office in a counterpart foreign application, an English language version of the search report or action which indicates the degree of relevance found by the foreign office is listed on form PTO 1449 and is enclosed herewith.

It is requested that the information disclosed herein be made of record in this application.

"Express Mail" label no.

Date of Deposit

I hereby certify that this is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to: Commissioner for Patents, Washington, D.C. 20231.

By ______________________

Typed Name:

Respectfully submitted,

Eric A. PULSIPHER et al.

By [signature]

Raymond Van Dyke
Attorney/Agent for Applicant(s)
Reg. No. **34,746**

Date: **August 13, 2002**

Telephone No.: **(202) 442-3000**

Rev 05/02 (IDSXML)

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 830 047 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
18.03.1998 Bulletin 1998/12

(51) Int Cl.6: H04Q 11/00, H04L 12/56

(21) Application number: 97440056.6

(22) Date of filing: 03.07.1997

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV RO SI

(30) Priority: 03.07.1996 US 675166

(71) Applicant: ALCATEL ALSTHOM COMPAGNIE GENERALE D'ELECTRICITE
75008 Paris (FR)

(72) Inventor: Seid, Howard A.
Fairfax, Virginia 22033-1251 (US)

(74) Representative: Pohl, Herbert, Dipl.-Ing. et al
Alcatel Alsthom,
Intellectual Property Department,
P.O. Box 30 09 29
70449 Stuttgart (DE)

(54) Connectivity matrix-based multi-cost routing

(57) Connectivity matrix-based multi-cost routing includes defining a generally additive operator which is able to add traditionally (arithmetic) additive cost factors and which takes into account cost factors which are not additive, the generally additive operator being defined such that distributive and communicative properties are applicable, and wherein the generally additive operator is applicable to connectivity matrix-based factors for determining the relative costs of paths within a network, particularly with respect to multi-cost factors. Connectivity matrix-based multi-cost routing is performed by first defining cost functions and establishing a criteria for prioritizing cost functions such that a composite multi-cost function includes the cost functions in the priority order defined by the criterion. A connectivity matrix is established including ordered n-tuples of cost factors corresponding to the priority established by the criterion, and a shortest path matrix determination is made by using the generally additive operator to apply the composite multi-cost function to the connectivity matrix. When links within a network support various functionality, a mask of a required functionality may be used to define a cost function for a given shortest path matrix determination. A correcting method is provided for a routing determination when, after a shortest path matrix determination, a routing choice is not provided which would otherwise satisfy a multi-cost requirement, the correcting method including the determination of a primary path and secondary paths between a source node and a destination node.

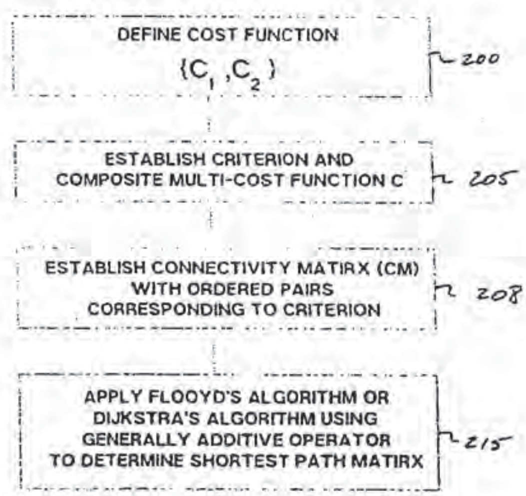


FIG. 2

Printed by Jouve, 75001 PARIS (FR)

Description

TECHNICAL FIELD

The present invention relates to routing, and more particularly, to connectivity matrix-based multi-cost routing.

BACKGROUND OF THE INVENTION

Several well-known network routing algorithms are based on the generation of path choices in each routing node. For example, in a packet switching (packet-based) network containing a plurality of packet switching nodes interconnected by respective links, routing algorithms are used to route a packet from a source node to a destination node over various nodes and links within the network. Examples of packet-based networks include: frame relay (FR) networks wherein a packet corresponds to a FR frame; a cell-switching network, e.g., an asynchronous transfer mode (ATM) network, wherein a packet corresponds to an ATM cell; etc.

For purposes of routing a packet within a network, if there are **n** routing nodes in the network, each node is given a unique Node Number (NN) such that 1<=NN<=**n**. The path choices are derived from the network node connectivity as specified in a nxn Connectivity Matrix (CM). The information stored in the i,j-th element of an nxn CM (CM[**i**,**j**]) is: c(**i**,**j**), the cost of a direct routing link between nodes **i** and **j**; ∞ if there is no direct routing link between node **i** and **j**; and ϕ (lowest cost connectivity) if node **i** is the same as node **j**.

A path between any two arbitrary nodes in the network is a sequence of nodes from the originating (source) node to the terminating (destination) node, where if node **j** directly follows node **i** in the sequence, there is a direct routing link between them. In order to select a "best" path, some figure of merit for any candidate path is derived. The figure of merit consists of the (additive) "costs" of each of the links between nodes derived from the CM. The link costs may include the (additive) "costs" associated with the nodes. Thus the cost of a path is just the sum of the link costs between adjacent nodes in the path sequence.

Regardless of the algorithm used to transform the network nodal connectivity to the "shortest path" between two nodes (e.g., Floyd's Algorithm, Dijkstra's Algorithm), the "cost" of such a path is obtainable by taking the sum of the link costs (perhaps including some additive nodal cost), either using Floyd's Algorithm or Dijkstra's Algorithm to make the required determination. As long as there is a single link cost factor used to determine the cost for the logical or physical links between node **i** and node **j**, the approach to determining the shortest path value is straightforward.

The situation is not as clear cut when there are two or more independent cost factors used to determine a figure of merit aggregate cost for a path to be used in evaluating the best or shortest path. It will be understood that when multiple costs are considered for evaluating a best or shortest path, the various cost considerations lead to different "shortest path" conclusions. For example, if hop count, i.e., the number of nodes traversed by a packet from source to destination node, is an overwhelming concern, one path within a network may be the shortest path. Alternatively, if delay is the major factor, i.e., the time it takes for a packet to travel from the source to the destination node, then another path within the network may be the best path. Still different conclusions may be reached if other factors, such as monetary cost, are taken into consideration.

When single cost functions are used as a means of obtaining figures of merit for various paths within the network, the concept of adding the costs of each link in the path, as discussed above, is the natural way to obtain the result. Known routing algorithms assume that costs add as the path is traversed. Common cost functions, used in path selection (e.g., "delay", "cost of transmission line facility", "hop count"), are additive. Individually, they are immediately adaptable to standard CM-based path-cost-determination algorithms. However, there are other cost-factors (e.g., "per cent available bandwidth", "ability to handle a specific protocol", "ability to handle a certain packet type") which are major determiners of path selection, but which are not additive. These types of cost factors cannot be treated by usual CM-based methods. The reason is that arithmetic addition is too limiting to describe the actions of many cost factors.

Therefore, a method for handling multi-cost factors in determining the relative costs of paths within a network is needed. Additionally, this method for handling multi-cost factors should also be capable of considering non-additive cost factors in the relative costs of paths determination.

SUMMARY OF THE INVENTION

Objects of the present invention include network routing capable of handling multi-cost factors in determining the relative costs of paths within a network.

Another object of the invention is to provide such a network routing which is also capable of considering non-additive cost factors in the relative costs of paths determination.

According to the present invention, connectivity matrix-based multi-cost routing includes defining a generally additive operator which is able to add traditionally (arithmetic) additive cost factors and which takes into account cost factors which are not additive, the generally additive operator being defined such that distributive and commutative properties are applicable, and wherein the generally additive operator is applicable to connectivity matrix-based factors for determining the relative costs of paths within a network, particularly with respect to multi-cost factors.

In further accord with the present invention, connectivity matrix-based multi-cost routing is performed by first defining cost functions and establishing a criterion for prioritizing cost functions such that a composite multi-cost function includes the cost functions in the priority order defined by the criterion. A connectivity matrix is established including ordered n-tuples of cost factors corresponding to the priority established by the criterion, and a shortest path matrix determination is made by using the generally additive operator to apply the composite multi-cost function to the connectivity matrix.

In further accord with the present invention, the shortest path matrix is determined by applying Floyd's Algorithm or Dijkstra's Algorithm to a connectivity matrix using the generally additive operator.

In still further accord with the present invention, when links within a network support various functionality, a mask of a required functionality may be used to define a cost function for a given shortest path matrix determination.

According still further to the present invention, a routing method is provided for a routing determination after a shortest path matrix determination, the routing method including the determination of a primary path(s) and secondary paths between a source node and a destination node.

According still further to the present invention, the determination of primary paths and secondary paths between source nodes and destination nodes is provided by a criterion cost determination and a composite constraint multi-cost determination from each node adjacent to the source node to the destination node and using the criterion cost and composite constraint multi-cost for a link-by-link least cost determination, each primary path being determined as the path(s) from the source node to the destination node(s) through the adjacent node having the lowest generalized sum of the link multi-cost from the source node to one of its adjacent nodes and the multi-cost of the "best" path(s) from that adjacent node(s) to the destination node. The remaining ordered link least-criterion-costs of paths, if any, are the secondary links or paths.

According still further to the present invention, the routing method further includes the steps of establishing a criterion **C**, for routing from a source node to a destination node; establishing a maximum multi-cost metric allowed for a successful routing; determining an ordered list of paths, including the primary path(s) and secondary paths from the source node to the destination node; and establishing link least-criterion-costs and link composite-constraint-costs for each of the primary and secondary paths from the source node to the destination node. If, for a given required routing from a source node to a destination node (path), the maximum metrics are such that the metrics are greater than or equal to the link least-criterion-costs for each individual cost component, then there is a path from the source node to the destination node which has multi-cost metrics no more than those required. If, for a given path between the source node and the destination node, the metrics are such that at least one metric is less than a corresponding link least-criterion-cost for each link, but there is at least one link wherein the metrics are greater than or equal to the link lease-constraint-costs for each individual cost function, then there may or may not be a path which achieves a satisfaction of the cost element while still satisfying the other metrics of the cost element. Finally, if the metrics are less than the link least-constraint-costs, for at least one individual cost component within each link, then there is no successful path for meeting the metrics from the source node to the destination node.

The foregoing and other objects, features and advantages of the present invention will become more apparent in light of the following detailed description of an exemplary embodiment thereof, as illustrated in the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a schematic diagram of a virtual private network utilizing connectivity matrix-based multi-cost routing in accordance with the invention;

Fig. 2 is a logic flow diagram of a shortest path determination for the virtual private network of Fig. 1 utilizing connectivity matrix-based multi-cost routing;

Fig. 3 is a schematic diagram of a second network example, including three cost functions, utilizing connectivity matrix-based multi-cost routing in accordance with the invention;

Fig. 4 is a schematic diagram of a third network example utilizing connectivity matrix-based multi-cost routing in accordance with the invention and illustrating the use of a Mask function (M) to define a required functionality;

Fig. 5 is a schematic diagram of a fourth network example utilizing connectivity matrix-based multi-cost routing in accordance with the invention and illustrating a correcting method to find a path meeting a given call criterion;

Fig. 6a is schematic diagram illustrating how different cost functions may achieve minimums on different links between different pairs of nodes;

Fig. 6b is schematic diagram illustrating how different cost functions may achieve minimums on different links between a the same two nodes;

Fig. 7 is a schematic diagram of a fifth network example illustrating the determination of a conceptual routing table from a source node (S) to a destination node (D); and

Fig. 8 is a schematic diagram illustrating that multiple cost metrics required by a call are not necessarily satisfied between two nodes even when individual call metrics are met.

DETAILED DESCRIPTION OF THE INVENTION

The present invention provides a method for handling multi-cost factors in the determination of the relative costs of paths within a network between a source node and a destination node. This method of handling multi-cost factors takes into consideration "non-additive" cost factors in the relative costs of paths determination. In the following description, the method of the invention for adding multi-costs to obtain a figure of merit is first presented. Thereafter, the method of routing table organization and path selection using such multi-costs is provided.

In order to consider multi-costs in a best path determination, a "generally additive operator" must be defined which is not only able to add traditionally (arithmetic) additive cost functions such as delay, monetary cost, and hop count, but which is also able to take into account cost factors which are not additive, such as "available bandwidth", "ability to handle traffic type", etc.

Let A be the range of values associated with a cost function, **C**. In accordance with the invention, a generally additive operator, ⊕, is defined as a mapping of $A \times A \rightarrow A$ with properties:

- $a \oplus b = b \oplus a$
- $(a \oplus b) \oplus c = a \oplus (b \oplus c)$

for all *a*, *b* and *c* in A.

The standard addition operator, +, on the set of real numbers satisfies this definition. But the ⊕ mapping can be satisfied by other operators as well. For example, the min and max functions and the Boolean *and* operation are just a few of the mappings which satisfy the stated properties in their appropriate domains of definition. The cost functions of "hop count", "delay" and "cost of transmission line facility" are generally additive functions under normal addition. The *min* function can be applied to the "percent available bandwidth" cost function and the Boolean *and* function to the "ability to handle traffic type" cost function to make them "generally additive."

Now that the generally additive operator ⊕ has been defined, connectivity matrix based single factor cost concepts can be extended to multi-cost functions. Let $\{\mathbf{C}_n\}$ be a set of N independent generally additive cost functions. (If there are two cost functions in the sequence that are functionally related, then one of those two can be removed from the sequence.) One way of dealing with the path information is to generate a sequence of connectivity matrixes $\{\mathbf{CM}_n\}$, in one-to-one correspondence to the sequence $\{\mathbf{C}_n\}$. Then, using traditional routing algorithms, N collections of path-cost information are obtained, each generated from $\mathbf{CM}_n$, 1<=n<=N. However, this collection is not correlated with the collections associated with the other $\mathbf{CM}_m$ matrixes, 1<=m<=N, where m≠n. In effect, the path which might have least cost for cost function $\mathbf{C}_n$ is an entirely different path than the one which minimizes the cost function $\mathbf{C}_m$. Using this method, there is no obvious way to ensure that the costs associated with all the $\{\mathbf{C}_n\}$ cost functions are within acceptable limits simultaneously.

In order to handle the $\{\mathbf{C}_n\}$ cost functions applied to the same paths simultaneously, it is observed that since each of the cost functions are independent, their minimum values will not in general be simultaneously achieved when applied to a path which minimizes one of them. Therefore, in accordance with the invention, the relative importance of the various cost functions must be prioritized. Let $\theta(n)$ be an ordering of the indexes such that $\{\mathbf{C}_{\theta(n)}\}$ is in priority order. Such an ordering will be called a *criterion.* There are n! possible criteria derivable from $\{\mathbf{C}_n\}$. If two cost functions are of equal importance, they must still be prioritized, making one artificially more important than the other. The n-tuple, $(\mathbf{C}_{\theta(1)}, \ldots, \mathbf{C}_{\theta(n)})$, is called the *composite multi-cost function,* denoted by $\tilde{\mathbf{C}}$. If two nodes in a network are connected by a link, L, the composite multi-cost function is applicable to the link. Let $\tilde{\mathbf{C}}(L)=(\mathbf{d}_{\theta(1)}, \ldots, \mathbf{d}_{\theta(n)})$ be the composite multi-cost n-tuple of the link, where $\mathbf{d}_{\theta(n)}$ is the value of $\mathbf{C}_{\theta(n)}$ over the link. Composite multi-cost n-tuples are generally additive using the operator n-tuple, $\tilde{\oplus} = (\oplus_{\theta(1)}, \ldots, \oplus_{\theta(n)})$, each element of the generally additive operator being applied to the corresponding cost function values.

If P is a path in the network from node **S** to node **D**, through intermediate nodes, **E, F, G,** then the value of $\tilde{\mathbf{C}}$ (P) is:

$$\tilde{\mathbf{C}}(S,E) \;\tilde{\oplus}\; \tilde{\mathbf{C}}(E,F) \;\tilde{\oplus}\; \tilde{\mathbf{C}}(F,G) \;\tilde{\oplus}\; \tilde{\mathbf{C}}(G,D)$$

where the $\tilde{\mathbf{C}}_{(S,E)}$, $\tilde{\mathbf{C}}_{(E,F)}$, $\tilde{\mathbf{C}}_{(F,G)}$ and $\tilde{\mathbf{C}}_{(G,D)}$ are the multi-cost n-tuples of the links found in the CM between adjacent nodes S to E, E to F, F to G, G to D, respectively. In similar manner, the composite multi-cost of any network path may be obtained by applying the generally additive operator to the inter-nodal multi-cost values as found in the CM for the network.

An ordering is defined for each of the cost function values which make up the composite multi-cost n-tuples. In order to find the "best" path in the network between two nodes S and D in a multi-cost environment, it is necessary to be able to compare multi-cost n-tuples. Let $(\mathbf{a}_1, ..., \mathbf{a}_n)$ and $(\mathbf{b}_1, ..., \mathbf{b}_n)$ be two multi-cost n-tuples. The composite multi-cost

$(\mathbf{a}_1, ..., \mathbf{a}_n)$ *is greater than* (>) $(\mathbf{b}_1, ..., \mathbf{b}_n)$ if:

(1) there is an M, 1<=M<=n, such that for 0<=i< M, either i is 0 or $\mathbf{a}_i = \mathbf{b}_i$, and $\mathbf{a}_M = \mathbf{b}_M$;

If: (2) $\mathbf{a}_i = \mathbf{b}_i$ for 1<=i<=n, then $(\mathbf{a}_1, ..., \mathbf{a}_n)$ *is equal to* (=) $(\mathbf{b}_1, ..., \mathbf{b}_n)$.

If neither of these two possibilities hold, then the composite multi-cost $(\mathbf{a}_1, ..., \mathbf{a}_n)$ is less than (<) $(\mathbf{b}_1, ..., \mathbf{b}_n)$. A path P1 from node S to node D is the *shortest path* if for every path P2 between these two nodes, $\tilde{\mathbf{C}}$(P1) <= $\tilde{\mathbf{C}}$(P2), where the length of a path is defined as the composite multi-cost n-tuple taking the generally additive sum of the composite multi-cost n-tuples of the individual links which make up the path.

Using the above definitions, single cost CM-based "shortest path algorithms" immediately extend to the multi-cost function environment.

The operation of the invention is best understood by example. Consider a virtual private network (VPN) as depicted in Fig. 1. Referring to Fig. 1, the bold lines exemplify links which can carry VPN traffic (VPNL) while the lighter lines exemplify links which cannot carry VPN traffic (L). Assume that the cost of paths are to be evaluated by two cost functions: (1) inability to carry VPN traffic (**V**); and (2) transmission cost (**T**). For "transmission cost" the normal addition operator will serve as the additive operator. For "inability to carry VPN traffic" there is a normal Boolean cost associated with each link: 1 if it cannot support VPN traffic and 0 if it can support VPN traffic. The or Boolean operator, (+), is chosen as the generally additive operator.

In order to make a shortest path or best path determination for the virtual private network of Fig. 1, the steps shown in the logic flow diagram of Fig. 2 are used. Referring to Fig. 2, in step 200, the cost functions $\{\mathbf{C}_1, \mathbf{C}_2\}$ to be applied to the virtual private network are defined. As discussed above, with respect to the virtual private network of Fig. 1, the cost functions of concern are the inability to carry VPN traffic (**V**) and the transmission costs (**T**).

Next, in step 205, the criterion θ(2) is established wherein the cost functions $\{\mathbf{C}_1, \mathbf{C}_2\}$ are prioritized. For example, with respect to Fig. 1, suppose that the "inability to carry VPN traffic" is the most important link characteristic. Then **V** will have a higher priority than **T**. Therefore, as described above, using the cost functions $\{\mathbf{C}_1, \mathbf{C}_2\}$ and the criterion θ (*2*), the composite multi-cost function, $\tilde{\mathbf{C}}$, is established.

After the criterion is established, a connectivity matrix for the network is established, in a step 208, having ordered pairs which correspond to the priority established by the criterion. For example, in the virtual private network of Fig. 1, the CM is a 6x6 matrix of ordered pairs, (v,t), where v is the link cost with respect to the **V** cost function and t is the link cost with respect to the **T** cost function. Using the link cost ordered pairs of Fig. 1, the CM is:

S 1	(ϕ, ϕ)	(0, 1)	(∞, ∞)	(∞, ∞)	(∞, ∞)	(1, 1)
O 2	(0, 1)	(ϕ, ϕ)	(0, 3)	(1, 1)	(∞, ∞)	(∞, ∞)
U 3	(∞, ∞)	(0, 3)	(ϕ, ϕ)	(0, 2)	(1, 1)	(∞, ∞)
R 4	(∞, ∞)	(1, 1)	(0, 2)	(ϕ, ϕ)	(1, 2)	(0, 3)
C 5	(∞, ∞)	(∞, ∞)	(1, 1)	(1, 2)	(ϕ, ϕ)	(1, 1)
E 6	(1, 1)	(∞, ∞)	(∞, ∞)	(0, 3)	(1, 1)	(ϕ, ϕ)
	1	2	3	4	5	6

DESTINATION

where (ϕ, ϕ) implies that a node is the shortest path to itself, and (∞, ∞) implies that there is no direct connectivity between the source and destination nodes.

Finally, in the step 215, the resulting shortest path matrix, obtained by applying Floyd's Algorithm to the CM, is:

		1	2	3	4	5	6
S	1	(φ, φ)	(0, 1)	(0, 4)	(0, 6)	(1, 5)	(0, 9)
O	2	(0, 1)	(φ, φ)	(0, 3)	(0, 5)	(1, 4)	(0, 8)
U	3	(0, 4)	(0, 3)	(φ, φ)	(0, 2)	(1, 1)	(0, 5)
R	4	(0, 6)	(0, 5)	(0, 2)	(φ, φ)	(1, 2)	(0, 3)
C	5	(1, 5)	(1, 4)	(1, 1)	(1, 2)	(φ, φ)	(1, 1)
E	6	(0, 9)	(0, 8)	(0, 5)	(0, 3)	(1, 1)	(φ, φ)

DESTINATION

The above shortest path matrix represents the shortest path from any source node to any destination node favoring VPN links over links not permitted to carry VPN traffic. From the standpoint of the VPN, this matrix represents the shortest path available between any source and destination nodes reachable by use of VPN links only. It is clear that certain source and destination nodes are not reachable within the VPN (e.g., node 3 cannot reach node 5). However, this matrix permits the accurate selection of paths which remain within the VPN between any source and destination connected through use of VPN links. It also permits the selection of a path to a non-VPN node which stays within the VPN until it reaches the closest node to the ultimate destination node reachable using only VPN links (e.g., the path from node 2 to node 5). Thus, traffic generated by subscribers of the VPN will use VPN resources as much as possible even when accessing network ports outside of its domain

As will be understood, the shortest path composite multi-cost value taken from the above shortest path matrix between a source and destination node pair is not necessarily minimal in the "transmission cost" cost function. However, given the chosen cost function priorities, the composite multi-cost function minimizes the cost function selected as being most important. For non-VPN traffic, this matrix does not in general yield the shortest path since all links (including those which carry VPN traffic) can be used in the path selection criteria.

Fig. 3 provides a second network example including three cost functions. Referring to Fig. 3, the network has its network elements connected by links which can support various services: X.25; Frame Relay; and ATM. Suppose that X.25 traffic should traverse X.25 links (X25L) whenever possible, using Frame Relay links (FRL) as an alternative of choice and ATM links (ATML) only when there is no other choice. In addition, suppose, that Frame Relay traffic cannot be carried by X.25 links and should use Frame Relay links in preference to ATM links wherever possible. ATM traffic may only use ATM links. Finally, in order that performance be within specification, no path is to take any more than 9 hops between source and destination. Within this selection scheme, suppose that delay is to be minimized, with hop count as the third priority selection criterion. Each trunk has an associated triple of numbers (a,b,c) where a is the rank, in inverse order, of the link type with respect to its usability, *b* is the delay in milliseconds, and c is the hop count. Considering the network with respect to ATM service, the link type is defined as follows: a = 0 for ATM; a = 1 for Frame Relay; and a = 2 for X.25.

The CM for a particular service is formulated as follows. First, if two nodes are connected to each other by one or more links, choose the link with the lowest 'a' value with respect to the specific service (e.g., for ATM service, choose the ATM link between nodes 1 and 2 and the Frame Relay link between nodes 1 and 5). Next, assign a triple (a,b,c) for each link selected. In order to easily identify how many hops of each type of link are in any path, a weighted hop count value is used for each type of line such that the resulting total weighted hops in a path can be decomposed, much as a decimal number, into its constituent hops. Since no path can have more than 9 hops, weighted hops of 1, 10 and 100 were chosen for ATM, Frame Relay and X.25 respectively. (In general, if there were a requirement that no path take more than **H** hops, then the weights **1**, **H+1** and $(H+1)^2$ may be used for ATM, Frame Relay and X.25 respectively).

Three CMs are used to describe the link connectivity with respect to each of the supported network services. For the ATM service topology, the CM is given by:

SOURCE	1	2	3	4	5
1	(ϕ,ϕ,ϕ)	(0,1,1)	(2,10,100)	(∞,∞,∞)	(1,3,10)
2	(0,1,1)	(ϕ,ϕ,ϕ)	(2,5,100)	(0,1,1)	(∞,∞,∞)
3	(2,10,100)	(2,5,100)	(ϕ,ϕ,ϕ)	(1,2,10)	(1,4,10)
4	(∞,∞,∞)	(0,1,1)	(1,2,10)	(ϕ,ϕ,ϕ)	(0,1,1)
5	(1,3,10)	(∞,∞,∞)	(1,4,10)	(0,1,1)	(ϕ,ϕ,ϕ)

DESTINATION

The CM is processed using Floyd's Algorithm on this CM using the triple of generally additive operators (max, +,+). This leads to an ATM service network 'shortest path' matrix (SPM) of:

SOURCE	1	2	3	4	5
1	(ϕ,ϕ,ϕ)	(0,1,1)	(1,4,12)	(0,2,2)	(0,3,3)
2	(0,1,1)	(ϕ,ϕ,ϕ)	(1,3,11)	(0,1,1)	(0,2,2)
3	(1,4,12)	(1,3,11)	(ϕ,ϕ,ϕ)	(1,2,10)	(1,3,11)
4	(0,2,2)	(0,1,1)	(1,2,10)	(ϕ,ϕ,ϕ)	(0,1,1)
5	(0,3,3)	(0,2,2)	(1,3,11)	(0,1,1)	(ϕ,ϕ,ϕ)

DESTINATION

The first element of each triple in the matrix represents the code of the least acceptable protocol used in the path for the service network. The second element represents the total delay associated with the links in the path. The third element is the weighted hop count of the path. Its value provides sufficient information to decompose the path into its constituent link types (e.g., a weighted hop count of 12 implies a unique decomposition of 1 hop of weight 10 and two hops of weight 1 which implies two ATM hops and one Frame Relay hop).

There are certain nodes that cannot be reached by a purely ATM service. For example, SPM [3,2] is (1,3,11). This means that path contains a Frame Relay link. Since this is the best path from node 3 to node 2, it follows that the ATM service is insufficient to handle any requirement for ATM traffic between these two nodes.

The ATM service network 'shortest path' matrix also indicates that there is always a path for X.25 traffic and Frame Relay traffic between any two nodes. Since there is no absolute prohibition from using ATM and/or Frame Relay links from carrying X.25 service, there is always X.25 connectivity between any two nodes. However, the objective is to carry X.25 traffic over pure X.25 links if at all possible. Thus, although this matrix is sufficient to handle, totally, the determination of the best path for ATM service, it is insufficient, by itself, for determining the 'best' paths for all services.

When this same network is considered with respect to the Frame Relay service, the most desirable links are those with Frame Relay protocol, so they are given the link protocol value (a) of 0. The next most desirable links are those carrying ATM. So such links have a protocol value of 1. Last, since X.25 links are not acceptable for Frame Relay traffic, these links are given a protocol value of 2.

The resultant FR connectivity matrix is given by:

SOURCE	1	2	3	4	5
1	(ϕ,ϕ,ϕ)	(0,3,10)	(2,10,100)	(∞,∞,∞)	(0,3,10)
2	(0,3,10)	(ϕ,ϕ,ϕ)	(2,5,100)	(1,1,1)	(∞,∞,∞)
3	(2,10,100)	(2,5,100)	(ϕ,ϕ,ϕ)	(0,2,10)	(0,4,10)
4	(∞,∞,∞)	(1,1,1)	(0,2,10)	(ϕ,ϕ,ϕ)	(1,1,1)
5	(0,3,10)	(∞,∞,∞)	(0,4,10)	(1,1,1)	(ϕ,ϕ,ϕ)

DESTINATION

Applying Floyd's algorithm to the FR connectivity matrix using the generally additive operator (*max*,+,+) obtains the following FR shortest path matrix.

SOURCE	1	2	3	4	5
1	(ϕ,ϕ,ϕ)	(0,3,10)	(0,7,20)	(0,9,30)	(0,3,10)
2	(0,3,10)	(ϕ,ϕ,ϕ)	(0,10,30)	(0,12,40)	(0,6,20)
3	(0,7,20)	(0,10,30)	(ϕ,ϕ,ϕ)	(0,2,10)	(0,4,10)
4	(0,9,30)	(0,12,40)	(0,2,10)	(ϕ,ϕ,ϕ)	(0,6,20)
5	(0,3,10)	(0,6,20)	(0,4,10)	(0,6,20)	(ϕ,ϕ,ϕ)

DESTINATION

The connectivity of the network is such that Frame Relay traffic follows paths using Frame Relay links. These paths are neither optimal in terms of delay nor hops. But for the Frame Relay traffic they represent the least delay Frame Relay paths within the network. Had there been several Frame Relay paths of equal delay between two nodes, then the hop count would have been a factor in determining the best path.

To handle the X.25 service aspects, the CM is generated from the X.25 protocol perspective. Here, X.25 links are the best to use, followed by Frame Relay and then by ATM. In this case, X.25 links are given a protocol value of 0, while Frame Relay links and ATM links have protocol values of 1 and 2, respectively. The appropriate connectivity matrix for the X.25 service network is:

SOURCE	1	2	3	4	5
1	(ϕ,ϕ,ϕ)	(1,3,10)	(0,10,100)	(∞,∞,∞)	(0,5,100)
2	(1,3,10)	(ϕ,ϕ,ϕ)	(0,5,100)	(2,1,1)	(∞,∞,∞)
3	(0,10,100)	(0,5,100)	(ϕ,ϕ,ϕ)	(1,2,10)	(0,5,100)
4	(∞,∞,∞)	(2,1,1)	(1,2,10)	(ϕ,ϕ,ϕ)	(0,4,100)
5	(0,5,100)	(∞,∞,∞)	(0,5,100)	(0,4,100)	(ϕ,ϕ,ϕ)

DESTINATION

Applying Floyd's algorithm, using the generally additive triple (*max*,+,+), obtains the X.25 service 'shortest path' matrix of:

SOURCE	1	2	3	4	5
1	(ϕ,ϕ,ϕ)	(0,15,200)	(0,10,100)	(0,9,200)	(0,5,100)
2	(0,15,200)	(ϕ,ϕ,ϕ)	(0,5,100)	(0,14,300)	(0,10,200)
3	(0,10,100)	(0,5,100)	(ϕ,ϕ,ϕ)	(0,9,200)	(0,5,100)
4	(0,9,200)	(0,14,300)	(0,9,200)	(ϕ,ϕ,ϕ)	(0,4,100)
5	(0,5,100)	(0,10,200)	(0,5,100)	(0,4,100)	(ϕ,ϕ,ϕ)

DESTINATION

The X.25 "shortest path" matrix indicates that all nodes can be reached by paths which use exclusively X.25 links. This is not to say that the delay, nor hop count are minimized by such paths. But the paths do minimize delay within the X.25 service network, in preference to nodal hops.

Fig. 4 is a third network example. Referring to Fig. 4, the network links support a collection of functions of importance to the subscribers of the network. A particular path in such a network is to be chosen according to the functions required by the call as far as possible while minimizing the hop count within the network. The network of Fig. 4 supports some or all of the following functionality: (a) security of data transmission (e.g., ability to handle classified data); (b) path diversity (e.g., the underlying bearer has multiple paths leading to high link reliability); (c) fiber transmission medium (e.g., for low Bit Error Rate); and (d) bandwidth-on-demand (e.g., for multi-media applications). Each link has an associated 4-tuple with 1s and 0s indicating the presence or absence of the functionality in the order indicated by the bullets (a) - (d).

In such a network, given a subset of the supported link-based functionality, it is reasonable to determine the best path capabilities available in the network as an aid to locating path deficiencies which must be corrected in order to provide the necessary network support. It may not be necessary that paths be found from a fixed source node to the other potential destination nodes which satisfy all the supported functions simultaneously. As an illustration of how these services are distributed, consider the determination of possible best connectivity for the subset of functions consisting simultaneously of 'path diversity' and 'bandwidth-on-demand'.
Additionally, the least hop routing is determined within the constraints that the selected paths are the best at meeting the desired functionality. To perform this task, the network is considered as being described by a 2-tuple cost function, (s,h), where s is a subset of the supported network functions and h is the hop count of the corresponding path. The generally additive operator on this cost 2-tuple is (*mand*,+), where *mand* refers to a 'masked Boolean *and*' and '+' is normal arithmetic addition. If (s_1,h_1) and (s_2,h_2) are 2-tuples, then $(s_1h_1)\ \tilde{\oplus}\ (s_2h_2)$ is (s_1 *mand* s_2, $h_1 + h_2$).

The following example illustrates the application of mand to a pair of Boolean strings using the generally additive operator. Let s_1 and s_2 be two Boolean strings of equal length and M be the mask of the same length of desired/required functionality associated with the Boolean strings. Then $s_1 = s_2$ with respect to M, if $s_1{*}M = s_2{*}M = (s_1 + s_2){*}M$ or $s_1{*}M \neq s_2{*}M \neq (s_1 + s_2){*}M$ (where "+" is the Boolean "or" function and "*" is the Boolean "and" function). Otherwise, we say either $s_1 < s_2$ or $s_2 < s_1$, meaning either $s_1{*}M = (s_1 + s_2){*}M$ or $s_2{*}M = (s_1 + s_2){*}M$, respectively. Then the term "less than or equal (≤)" is a well ordering on the masked Boolean strings, i.e., given any two Boolean strings of equal length and a mask M of the same length, then either the two strings are equal or one string is less than the other. If $s{*}M=0$, the $s \geq$ all other Boolean strings of equal length to that of the mask M.

We define $s_1 \oplus s_2$ with respect to mask M to be $s_1{*}s_2{*}M$, the *mand* of s_1 and s_2.

With the above example in mind, and returning to the example of Fig. 4, Let M be the mask of the required functionality (e.g., M is 0101 for this illustration). With these definitions, it is now possible to apply the multiple cost approach to determine the 'shortest paths' relative to the network with respect to a mask of functionality, M. The CM for this network is:

S	1	(φ,φ)	(0010,1)	(1111,1)	(∞,∞)	(∞,∞)	(∞,∞)
O	2	(0010,1)	(φ,φ)	(1011,1)	(1101,1)	(1010,1)	(0,∞)
U	3	(1111,1)	(1011,1)	(φ,φ)	(0010,1)	(∞,∞)	(1101,1)
R	4	(∞,∞)	(1101,1)	(0010,1)	(φ,φ)	(1101,1)	(0101,1)
C	5	(∞,∞)	(1010,1)	(∞,∞)	(1101,1)	(φ,φ)	(1101,1)
E	6	(∞,∞)	(∞,∞)	(1101,1)	(0101,1)	(1101,1)	(φ,φ)
		1	2	3	4	5	6

DESTINATION

where (∞,∞) implies the absence of direct node-to-node connectivity. The 'shortest path' matrix with respect to the mask-defined functionality is thus:

S	1	(φ,φ)	(0101,4)	(1111,1)	(0101,3)	(1101,3)	(1101,2)
O	2	(0101,4)	(φ,φ)	(0101,3)	(1101,1)	(1101,2)	(0101,2)
U	3	(1111,1)	(0101,3)	(φ,φ)	(0101,2)	(1101,2)	(1101,1)
R	4	(0101,3)	(1101,1)	(0101,2)	(φ,φ)	(1101,1)	(0101,1)
C	5	(1101,3)	(1101,2)	(1101,2)	(1101,1)	(φ,φ)	(1101,1)
E	6	(1101,2)	(0101,2)	(1101,1)	(0101,1)	(1101,1)	(φ,φ)
		1	2	3	4	5	6

DESTINATION

The 'shortest path' between any two nodes is seen to be the least hop path which includes the desired services. It is clear that there may be no path between some of the nodes for certain choices of the mask, M. Consider, for example, the path from node 1 to node 6 with respect to the functionality 'fiber transmission medium'. Although there are parts of paths which have the desired medium, there is always at least one link which does not possess a fiber transmission medium.

In single-cost shortest path algorithms, the idea of a minimal, least, or "shortest" path is intuitively clear. If the cost of a path is least compared to all other paths, it is the shortest path. It would be ideal if this same concept carried directly to a multiple cost shortest path algorithm. Unfortunately, the generalization rarely holds in practice. For, the generalization requires the simultaneous minimization of the individual costs of the multi-cost function, a condition much stronger than insisting on a single-cost obtaining its minimum. In common situations, additive cost functions comprising multi-cost criteria are completely unrelated, (e.g., hop count, monetary cost of transmission lines). It is quite possible that the minimization of one is unrelated to the minimum of another. When generally additive cost functions form part or all of a multi-cost function, there is even less predictability as to whether all the individual cost functions will achieve a minimum simultaneously.

With multiple cost components, the term "minimum cost" implies that the most significant cost function is minimal and that for this minimal value, the other cost functions are the smallest possible given the cost ordering. In effect, each component cost (excluding the most significant cost component) could achieve an individually lower value than is found in the "minimal cost" of the multi-cost function.

The "minimum cost" has a profound impact on the ability of Floyd's Algorithm or Dijkstra's Algorithm to provide sufficient information to allow a path to be found between two nodes in a multi-cost network. This concept can best be explained with reference to the network of Fig. 5. Referring to Fig. 5, the ordered triples represent the additive multi-cost functions, ($,d,h), where '$' represents monetary cost, 'd' denotes propagation delay and 'h' indicates hop count. Using the tri-cost criterion of "monetary cost is more important than propagation delay which is more important than hop count", a "minimal cost" is obtained from node S to node D of value (3,13,3) found by taking the path S-P-H-D. Conversely, considering the criterion of "propagation delay followed by monetary cost followed by hops", a "minimal cost" is obtained of value (10,7,7) via the path S-P-J-K-L-M-N-D. Note that the absolute minimum monetary cost and hop count cost is found in the path S-P-H-D where each has value 3. The absolute minimal delay path is S-P-J-K-L-

M-N-D with value 7. For this path, the monetary cost is 10 and the hop count is 7. There are four other possibilities for ordering the tri-cost functions. For example, there is the criterion of "delay followed by hops followed by monetary costs". In this case, the "minimal cost" path for this criterion is again S-P-J-K-L-M-N-D.

Note that if there are N independent cost functions comprising a multi-cost function, then there are N! possible multi-cost criteria. It becomes impractical to deal with all of these. Instead, a selected subset of the possible criteria may be considered as a basis for network routing. However, even if all N! possible criteria are used to calculate "shortest path" routes between two nodes, there is no guarantee that any of these will satisfy the requirements of routing a specific call requiring a specific combination of the multiple costs.

Using the network of Fig. 5, assume that it is desirable to route a call from S to D with a monetary cost of no more than 9, no more that 5 hops and no more than 9 units of delay. There is indeed such a path, namely S-B-C-F-G-D. If each of the possible six criteria are applied to the network, there are none whose corresponding shortest path from S to D ensures that the three metrics of the call can be satisfied simultaneously. The two criteria which minimize monetary cost first yield identical shortest path metrics of (3, 13, 3). The two criteria which minimize hop count first yield shortest path metrics of (3 13, 3) and (13, 8, 3).

Finally, the two criteria which minimize delay first result in the same shortest path metrics of (10, 7, 7). It follows that the delay metric is the problem. It does not achieve its minimum along the same path as the simultaneous minimums of hop count and monetary cost. The desired call metrics are distributed such that when the other two metrics can be achieved, the delay metric appears to be unachievable. Conversely, when a criterion is chosen such that the required delay metric is met, then the monetary cost and hop count metrics are too high.

The example of Fig. 5 illustrates the limitations of multi-cost routing choices and the associated path selection process. Even when paths exist in the network which satisfy the multi-cost requirements of a particular call, the multi-cost values associated with the "shortest paths" may mask this information so that there appears to be no possible path. Although there is no absolute guarantee that multi-cost routing information will ensure a path for a given call requiring conformance to specific multi-cost metrics, a correcting method in accordance with the invention improves the multi-cost information and gives the best chance for successfully finding a path.

For purposes of illustrating the correcting method of the invention, let CM(**C**) be the connectivity matrix of any network with respect to a given criterion, **C**, with ordered component cost functions $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_N$. Then CM(**C**) [**I**,**j**] has n-tuple, $(\mathbf{c}_1, \mathbf{c}_2 \ldots, \mathbf{c}_N)$, of values appropriate to a multi-cost link between adjacent nodes **I** and **j**. In case there are multiple links between two adjacent nodes, the one with the lowest multi-cost, according to **C** is selected for CM(**C**). The ***criterion cost*** $(E_1, E_2, \ldots, E_N)$, between a source node S and a destination node D with respect to the criterion **C** is just the value obtained by applying Floyd's Algorithm or Dijkstra's Algorithm to CM(**C**) for the "least cost" of a path between the two nodes with respect to **C**.

Now consider each individual cost function c_i, $1 \leq i \leq N$, which is a component of **C**. Apply Floyd's Algorithm or Dijkstra's Algorithm to CM(**C**) considering only the i-th member of each N-tuple, i.e., obtain the single-cost "least cost" of a path between a source node S and a destination node D with respect to $\mathbf{c}_i$ relative to criterion **C**. This cost will be called the i-th constraint cost, e_i, between the two nodes. The total set of the N constraint costs e_i, $1 \leq i \leq N$, can be used to form an N-tuple $(e_1, e_2, \ldots, e_N)$ defined as the *composite constraint multi-cost* between nodes S and node D.

Since the composite constraint multi-cost is composed of the individual least costs of each component without regard for the effects of the criterion **C**, it follows that $(e_1, e_2, \ldots, e_N) \leq (E_1, E_2 \ldots, E_N)$ for the two nodes, S and D. Since $\mathbf{c}_1$ is the most significant cost component of **C**, it follows that $E_1=e_1$. If all of the individual cost functions, $\mathbf{c}_i$, simultaneously attain their minimums along the same path from node S to node D, then $E_i=e_i$, $1 \leq i \leq N$. Any other situation will result in at least one value of i, $2 \leq i \leq N$, with $e_i < E_i$.

Considering the criterion, **C**, having **N** independent cost functions, $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_N$, if there is a link from node A to node B, the cost of that link may be low for $\mathbf{c}_i$, but high for $\mathbf{c}_j$. There may be other nodes, node F and node G, where the reverse is true. Alternatively, there may be multiple links between node A and node B. In this case, the choice of link for $\mathbf{c}_i$ may be different from the choice of link for $\mathbf{c}_j$ with the intent that the selected link has the minimum cost for the given cost function. This principle is illustrated in Figs. 6a and 6b for a pair of cost functions.

Referring to Figs. 6a and 6b, it is assumed that there are two cost functions $\mathbf{c}_1$ and $\mathbf{c}_2$ and the multi-costs of links are given as ordered pairs $(\mathbf{c}_1, \mathbf{c}_2)$. Fig. 6a illustrates the point that a path with a low cost value for $\mathbf{c}_1$ may be entirely different from a path with a low cost value for $\mathbf{c}_2$. Fig. 6b indicates that even when paths for each of the cost functions include the same nodes, they may require different links for achieving individual minimal costs. Therefore, there may not be a consistent choice of links which will simultaneously minimize $\mathbf{c}_1$ and $\mathbf{c}_2$.

Returning to the consideration of N independent cost functions, and generalizing the information illustrated by Fig. 5, it follows that each independent cost function, $\mathbf{c}_i$ $1 \leq i \leq N$, must have its own connectivity matrix CM(c_i), where the cost assigned to a connecting link between two nodes is chosen to have the minimal value for $\mathbf{c}_i$. It follows that CM($\mathbf{c}_i$) may differ significantly from CM($\mathbf{c}_j$) for some other cost function. Connectivity matrix CM($\mathbf{c}_i$) is used to find the absolute minimal cost path for the i-th cost function $\mathbf{c}_i$ between any two nodes in the network. Let ae_i be the minimal cost path with respect to the cost function $\mathbf{c}_i$. Then an *absolute constraint cost* $(ae_1, ae_2, \ldots, ae_N)$ is defined such that $ae_i \leq e_i \leq E_i$,

$1 \leq k \leq N$, for all criteria **C**, composed of the cost functions.

The criterion cost, absolute constraint cost and composite constraint cost can be used to define "primary" and "secondary" paths within the network. For example, let S be any node in a network. Let $A = \{A_k\}$, $1 \leq k \leq m$, be the set of all nodes of the network directly connected to node S by either single or multiple links. Let SCM(**C**) be CM(**C**) with all link connectivity from node S to A removed. In effect, SCM(**C**) isolates node S so that it cannot be part of any path from any $A_k \in A$ to any other destination node D (where node D is different from node S) in the network. Using SCM (**C**) and applying Flpyd's Algorithm (or Dijkstra's Algorithm) for each $A_k \in A$, a determination may be made of the criterion cost, $E_k=(E_1, E_2,..., E_N)_k$, and the composite constraint cost, $e_k=(e_1, e_2,..., e_N)_k$, from each source node $A_k \in A$ to every destination node D. The sets $\{E_k\}$ and $\{e_k\}$ form m n-tuples of multi-costs, each n-tuple associated with an adjacent node of node S (and not including S as described above).

To find the cost of a path from node S to node D, it is observed that any such path must transit through one of the adjacent nodes $A_k \in A$. A link "least-criterion-cost" ("composite-constraint-cost", "absolute-constraint-cost") of the path to node D through node A_k is defined as the sum of the multi-cost of the chosen link from node S to node A_k and the criterion cost (composite-constraint-cost, absolute-constraint-cost) of the path from A_k to the destination node D. A link least criterion-cost (composite-constraint-cost, absolute-constraint-cost) of a path from node S to destination node D through node A_k (according to criterion C) is defined as the sum of the cost of a multi-cost link from node S to node A_k and the criterion cost (composite-constraint-cost, absolute-constraint-cost) from A_k to node D. By considering all the different link least-criterion-costs from node S to each of the adjacent nodes A_k, $1 \leq k \leq m$, the resultant link least-criterion-costs from node S through node D can be ordered. If two or more such costs have the same value, then the corresponding link composite-constraint-costs are compared and used to determine the ordering. The link(s) with the lowest link least-criterion-costs are called the ***primary link(s).*** The corresponding adjacent node(s), A_k, is called the ***primary adjacent node(s).*** The remaining ordered link least-criterion-costs of paths, if any, are the ***secondary links.*** Throughout the remainder, it is assumed that a given node routes a call to one of its adjacent nodes. Once there, that node routes it in exactly the same manner, continuing until the destination node is reached. Thus, from the standpoint of a node, primary links are synonymous with *primary paths* and secondary links are synonymous with *secondary paths.* The set of primary paths and secondary paths associated with each destination node D forms the routing choices for the node S.

Using the above definition, a routing table can be defined for a path selection from a source node (S) to a destination node (D). Referring to the example of Fig. 7, let **C** be a criterion of m multi-cost functions $(\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_m)$. Let S be the node for which routing tables are being generated. Let D be any destination node in the network. The set $\{A_k\}$, 1<=**k**<=n, is the set of all network nodes that are link-connected (adjacent) to S.

Let CM(**C**) be the multi-cost connectivity matrix for the network. Let SCM(**C**) be the associated connectivity matrix at node S wherein all connectivity between node S and its adjacent nodes are removed (i.e., S is isolated from the connectivity in the network). SCM(**C**) is used to find the shortest path from each adjacent node A_k to the destination node D for criterion C. This least multi-cost path is denoted by $SP(A_k,D)$, 1<=**k**<=**n**. Let $CC(A_k,D)$ be the constraint cost of each path from A_k to D. It is noted that since SCM(**C**) has node S isolated from the rest of the network, $SP(A_kD)$ and $CC(A_k,D)$ do not involve any paths which go through S.

Suppose there are q(**k**) links from S to adjacent node A_k. Let $L_j(S,A_k)$, 1<**j**=<=q(**k**), be the link multi-cost of each of the links, *1(j)*. Let $R_{jk}(D)$ be the ordered triple $(S_{jk}, U_{jk}(D), B_{jk}(D))$, where $R_{jk}(D)$ is the j-th routing entry associated with the j-th link from S to A_k, S_{jk} is the link identifier of the j-th link from S to A_k, $U_{jk}(D) = L_j(S,A_k) \oplus SP(A_k,D)$ and $B_{jk}(D) = L_j(S,A_k) \oplus CC(A_k,D)$. It follows that $U_{jk}(D) \geq B_{jk}(D)$. The routing entries at node S for destination node D are $U_k U_j R_{jk}(D)$, 1<=**j**<=q(**k**) and 1<=**k**<=n. This set is ordered by comparing $U_{jk}(D)$ multi-cost values with respect to criterion **C**. If $U_{joko}(D) = U_{j1k1}(D)$ then ordering is accomplished by comparing $B_{joko}(D)$ and $B_{j1k1}(D)$. This ordered list is the conceptual routing table for destination node D in Node S.

Returning to the example network of Fig. 5, assume that this network is governed by the single criterion that *monetary cost is more important than delay which is more important than hop count.* Only the primary and secondary path information to node D with respect to node S are considered. The adjacent nodes to node S are node P and node B. Using Floyd's Algorithm, Dijkstra's Algorithm or inspection, the criterion cost $(\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3)$ from node P to node D is determined to be (2, 12, 2). The composite constraint cost $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ is (2, 6, 2). Note that the constraint cost element for delay is less than the criterion cost element for delay. This implies that at least the minimal delay constraint cost is achieved on a different path than that which best satisfies the criterion. The absolute constraint cost $(\mathbf{ae}_1, \mathbf{ae}_2, \mathbf{ae}_3)$ is (2, 6, 2) also. But note that the CMs used for obtaining the absolute minimums do not all coincide with the CM used for the selected criterion.

In a similar manner, the criterion cost $(\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3)$ from node B to node D is (5, 16, 3). The composite constraint cost $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ is (5, 7, 3). Since the delay constraint cost element is less than the delay criterion cost element, at least this constraint cost element was achieved on a different path than that which best satisfies the criterion. The absolute constraint cost $(\mathbf{ae}_1, \mathbf{ae}_2, \mathbf{ae}_3)$ from node B to node D is (5, 6, 3). Note that since the absolute constraint cost for delay is less than the composite constraint cost for the criterion, it must be achieved on a different path (either nodes, links between nodes, or both) from the path which minimizes the delay element for the constraint cost.

Adding the appropriate link costs from node S to node P or node B, the primary path cost is (3, 13, 3) with corresponding composite constraint and absolute constraint costs of (3, 7, 3) through the link to node P. Additionally, the secondary path cost is (6, 17, 4) with corresponding composite constraint cost (6, 8, 4) and absolute constraint cost of (6, 7, 4) through the link to node B.

Suppose it is desirable to route a call from node S to node D with the requirements that its monetary cost be no more than 9, its delay be no more than 9 and its hop count be no more than 5. This is formulated as a multi-cost call requirement of (9, 9, 5). If this allowed cost is compared with the primary path value of (3, 13, 3), it fails to meet the appropriate condition to route the call because the delay element of the primary path is too large. Note, however, the corresponding constraint cost is strictly less than the cost required by the call. As will be seen in the following, this implies that there may be a sub-optimal path from node S to node D through the link to node P which satisfies the cost requirements of the call.

Similarly, if the cost requirement of the call is compared against the secondary path value of (6, 17, 4) it also fails to meet the appropriate condition for routing the call since the delay element of the criterion cost is too large. If the cost requirement of the call is compared against the corresponding composite constraint cost, it is seen that the constraint cost is less than the call requirement cost. It follows that there may be some sub-optimal path from node S to node D through the link to node B which satisfies the cost requirement of the call. As discussed above, there is indeed such a path through node B. There is no path through node P even though it appears to be the primary (and therefore "best") choice.

The above defined criterion costs, composite constraint costs and absolute constraint costs can be used for path selection. Assume that a call is to be routed by a node S in the network to an ultimate destination node D according to some criterion, C. Suppose the corresponding maximal multi-cost metrics allowed for successful routing of the call are given by $M=(M_1, M_2, \ldots, M_i, \ldots, M_N)$. When node S tries to select a path to the next node toward the destination, it must use its ordered primary and secondary path information for node D with respect to the criterion, **C**, to find a suitable candidate link.

Let R_D be the set of criterion-cost-ordered links in node S to be used in path selection to node D. Let Z be the number of entries in R_D. Finally let E_z and e_z be the link least-criterion-cost and link composite-constraint-costs respectively for the z-th entry of R_D, $1 \leq z \leq Z$.

If for some $1 \leq z \leq Z$, the maximal call matrices are such that $M \geq (E_z)_i$ for each individual cost component, then there is a path from node S to node D which has multi-cost metrics no more than those required by the call. It follows that there is at least one path, using the link corresponding to E_z, able to support the call within the desired metrics.

Suppose that for each $1 \leq z \leq Z$, the call metrics are such that $M_i < (E_z)_i$ for at least one individual cost component, but there is at least one index z_0, with $1 \leq z_0 \leq Z$, such that $M \geq (e_{z0})_i$ for each individual cost function. Then there is at least one cost element of the call which **may or may not** have a path for which it can achieve the satisfaction of that cost element while still satisfying the other metrics of the other cost elements. Since the individual cost elements could achieve their individual constraint values along different paths, there is no guarantee that the call routed down the corresponding link will have a successful completion. As a consequence, if paths for which $M \geq (E_{z0})_i$ do not exist or are not successful, then node S should select a link such that $M \geq (e_z)_i$, for each individual cost function.

Finally, if for each $1 \leq z \leq Z$ the condition $M_i < (e_z)_i$, for at least one individual cost component holds, then there is likely to be no successful path for the call from node S to node D. For, e_z represents the lowest cost path, *relative to criterion* **C**, for the corresponding link. Therefore a call with smaller metrics has very low chance of completion using that link while preserving the topology of the network with respect to **C**. The absolute constraint costs may be such that $ae_z \leq M_i < (e_z)_i$ for at least one cost component. In this case, there is a remote possibility that a path exists which satisfies the metrics of the call. However, this situation is unlikely, at best, for two reasons. First, the CM topology used for the absolute constraint costs must have differed (in terms of link costs between nodes) from the topology used for the criterion **C**. Second, the path selection algorithm must allocate all the multi-costs associated with a given link while only the single cost component of a link, which minimizes a given cost component, is used in calculating the absolute constraint cost.

Consequently, attempting to route a call, based on absolute constraint costs, may on rare occasions produce a successful path selection. But much more typically, such path selection attempts will be fruitless with the additional undesirable affect of ineffectively using network resources. Therefore if, for all potential paths, a call has one of its metrics less than the corresponding constraint cost relative to a criterion, it is preferable to consider a "best effort" path which minimizes the most important cost component (for example) rather than to consider a path selections based on the absolute constraint cost metrics.

Special considerations must be taken when only two cost functions are considered for path selection. Suppose that $\mathbf{c}_1$ and $\mathbf{c}_2$ are two cost components comprising a multi-cost function. Then only two criteria are possible, either $\mathbf{c}_1$ is more important than $\mathbf{c}_2$, or the reverse is true. In this case, if **C** is one of the criteria, then the criterion and composite constraint costs with respect to **C** serve as a bound for routing a call within the limits of the criterion just as described above. The criterion cost with respect to **C** minimizes the "more important" of the two cost components with respect to **C**. The criterion cost of **C'**, the remaining criterion, effectively gives the minimum of the second cost component. It can be used as the means of achieving the "best effort" routing if the metrics of the given call fail to yield a route because of failure in the "less important metric" to be within the bounds of criterion **C**. The use of these criterion cost metrics may introduce a potential path which satisfies the "less important metric", while satisfying the metrics (sub-optimally) for the "more important metric".

It must be emphasized that this technique still does not ensure a path can be found. Fig. 8 illustrates such a situation. Here the ordered pairs represent the two costs. If criterion **C** is to consider the first cost to be more important, then the criterion cost will be (6, 13) from node A to node B. The criterion cost for **C'** is seen to be (10, 11). From these two criterion costs, it is inferred that the absolute constraint cost is (6, 11). But there is no path that achieves these cost metrics. In fact, if a call is to be routed from node A to node B with associated cost metrics of (7, 12), there is no path capable of routing the call. Node A might try to route such a call using a "best effort" service. Depending on the policy of the path selection in node I, either the (5, 12) or the (9, 10) link could be chosen to satisfy at least one of the cost metrics.

It should also be observed that this methodology does not, in general, extend to multi-cost functions of more than two cost components.

Routing choices can be effectively used in the path selection process. When there is a call request at a source network node, either there is an explicit multi-cost required or there is an implicit default multi-cost. In any event, it can be assumed that call set-up requests always contain a (maximum) multi-cost value, M, which must be satisfied to allow successful call completion. It can be further assumed that the source node knows the identity of the destination network node of the call and that each intermediate node participating in the path selection is aware of the total maximum allowed cost for the call request and of the multi-cost accumulated so far in the path selection process.

Suppose for example that {N(k)}, **k**=1,...,m, is the sequence of consecutive nodes in a path from a source node S = N(1) to a destination node D = N(**m**). The intermediate nodes are to be determined so as to satisfy the multi-cost constraints, **M**, on the service request.

If Q(**k-1**) is denied to be the accumulated multi-cost of the path from the source node S to node N(**k**), and P(**k**) is defined to be a path from node N(**k**) toward node D with multi-cost T(**k**), then each node N(**k**) selects a path **P(k)** predicated on the rules that:

- M is greater of equal to Q(**k-1**) $\tilde{\oplus}$ T(**k**); and
- if P'(**k**) is any other possible choice, then: (a) if it is the rule to select a path which is the "best fit", then a path should be selected such that Q(**k-1**) $\tilde{\oplus}$ T'(**k**) is less than or equal to Q(**k-1**) $\tilde{\oplus}$ T(**k**) (b) if it is the rule to select the "best possible" path, then a path should be selected such that Q(**k-1**) $\tilde{\oplus}$ T'(**k**) is greater than or equal to Q(**k-1**) $\tilde{\oplus}$ T(**k**)

Where the "best fit" path is the path which provides performance closest to the requirements of the call and the "best possible" path is defined as the path which provides the "best" performance possible for the call, as described in greater detail below.

If Q(*0*) is defined to be the 0 multi-cost value, then for **k** = 2,...,m, the value Q(**k-1**) = Q(**k-2**) $\tilde{\oplus}$ $L[i_{(k-1)}, j_{(k)}]$, where $L[i_{(k-1)}, j_{(k)}]$ is the multi-cost of the link selected from node N(**k-1**) to node N(**k**). The value of T(**k**) = $L[i_{(k)}, j_{(k+1)}]$ $\tilde{\oplus}$ W $[i_{(k+1)}, j_D]$, where $W[i_{(k+1)}, j_D]$ is the multi-cost value at node N(**k**) from the selected adjacent node N(**k+1**) to the destination D as determined from the routing choice information.

For best fit routing, in order for node N(**k**) to select a link to node N(**k+1**), the criterion costs of the routing choices to destination D should be scanned from **worst to best**. The link corresponding to the first entry encountered, if any, which satisfies the condition that the remaining maximum multi-cost of the call (i.e. M - Q(**k-1**) is greater or equal to the criterion cost, for each individual component cost, should be used to route to node N(**k+1**). If there is no entry whose criterion cost satisfies this condition, then the routing choices should again be scanned from **worst to best** to select the first entry encountered such that each individual cost component of M - Q(**k-1**) is greater or equal to each individual cost component of the composite constraint cost. During the search, if at any time an entry is encountered with $M_1 < e_1$, where e_1 is the first component of the composite constraint cost for the entry, then there is no path possible for the call and the search should be terminated. If a routing choice meeting the search condition is found, the path corresponding to this choice is a candidate for a possibly successful call completion. If the call routing attempt, using this choice, is unsuccessful, the search for additional possible choices should continue.

The effect of this best fit algorithm is to select a path which most closely matches the multi-cost requirements of the call request. There are numerous benefits of this path selection approach. First, an effective application of network resources is achieved. Call requests are evaluated against available potential path choices so that selection uses the least required resources to satisfy the cost function wherever possible. In this way, requests requiring premium performance can be reserved for those calls which are prepared to pay for this capability. Those call requests which can tolerate lower performance will use those resources as the criteria of choice. Only when lower performance resources are unavailable will higher performance resources be used.

For best possible routing, the minimum cost path available meeting the call requirements is selected. This is the intuitive method of path selection. The criterion costs of the routing choices to the destination are scanned from best to worst using the premium path first. This method has the advantage of allowing intermediate nodes within the path to elect sub-optimal paths with still having a high degree of certainty that the criterion will be met. However, this method may use resources for routing which are much better than required. If there is a later routing request with more stringent requirements, there may not be premium resources available to meet the requirements of the request.

No matter whether best possible or best fit routing is used, the path selection approach preserves path multi-cost information without explicit path knowledge. The cumulative cost information together with adjacency and local shortest path multi-cost information is all that is needed to permit allowable path evaluation at a node. Finally, this path selection approach provides ease of alternate path selection. If at some point along the path for call establishment, the most desirable selection of the next node is unobtainable, then either an alternate choice can be found (according to the algorithmic next best choice) or the path selection can be returned to the previous node by merely subtracting multi-costs locally known to the node.

Although the invention has been described and illustrated with respect to exemplary embodiments thereof, the foregoing and various other changes, omissions and additions may be made therein and thereto without departing from the spirit and scope of the present invention.

Claims

1. A method of network routing between nodes in the network, including the steps of:

   defining cost functions to be taken into consideration when making routing choices;
   prioritizing each of said cost functions with respect to one another to thereby provide a composite multi-cost function including said cost functions in a priority order;
   defining a connectivity matrix for the network including ordered n-tuples having n elements, each n-tuple representing a multi-cost directly between each pair of nodes in the network, and each one of said elements of each n-tuple representing a cost function value of each one of said cost functions directly between each said pair of nodes in the network, wherein n is the number of said cost functions, and wherein said ordered n-tuples include each of said cost functions ordered in said priority order established in said step of prioritizing;
   defining a generally additive operator such that distributive and commutative properties are applicable to a generally additive sum of said cost function values;
   defining an ordering between a pair of said n-tuples by comparing said cost function value of respective elements;
   deriving a shortest path matrix by applying said composite multi-cost function to said connectivity matrix using said generally additive operator and said ordering, said shortest path matrix including summation ordered n-tuples having multi-costs equal to the generally additive sum of said cost function values over a shortest path between each pair of nodes.

2. The method of claim 1, further including the steps of:

   determining a multi-cost requirement indicative of the multi-cost of said cost functions required to satisfy a routing request between a pair of nodes;
   comparing said multi-cost requirement to said summation ordered n-tuples between said pair of nodes; and
   determining that a path exists for routing between said pair of nodes if said summation ordered n-tuples between said pair of nodes satisfies said multi-cost requirement.

3. The method according to claim 1, wherein

   the nodes are interconnected by links;
   each node and/or link may or may not support functions generally available to the network;
   said step of defining cost functions includes defining a cost function which is a Boolean representation of functionality supported by the network between each said pair of nodes; and
   each n-tuple of said connectivity matrix includes an element indicative of the functionality, in said ordering of functionality, available directly between each said pair of nodes.

4. The method of claim 3, wherein

said step of prioritizing each of said cost functions includes defining one of said cost functions as a mask of a desired functionality of network capability, said desired functionality being indicative of a desired functionality between a pair of nodes;
said step of defining a generally additive operator includes assigning an operator MAND for said Boolean representation; and
said step of deriving said shortest path matrix includes deriving said summation n-tuples such that each summation n-tuple includes an element indicative fo the fulfillment of said desired functionality available over said shortest path between each said pair of nodes.

5. The method of claim 1, wherein said step of deriving a shortest path matrix includes the step of taking the generally additive sum of said cost functions over a shortest path between each pair of nodes using Floyd's Algorithm.

6. The method of claim 1, wherein said step of deriving a shortest path matrix includes the step of taking the generally additive sum of said cost functions over a shortest path between each pair of nodes using Dijkstra's Algorithm.

7. The method of claim 1, further including the steps of:

determining a multi-cost requirement indicative of the multi-cost required to satisfy a routing request between a pair of nodes;
determining at least one primary path and any secondary paths between said pair of nodes which are used to satisfy said multi-cost requirement.

8. The method of claim 7, wherein said step of determining at least one primary path and any secondary paths includes the steps of:

defining said pair of nodes to include a source node and a destination node;
determining each adjacent node in the network directly connect to said source node;
defining a source-isolated connectivity matrix including ordered n-tuples representing the multi-cost directly between each pair of nodes in the network excluding said source node;
deriving a criterion cost as a shortest path between each said adjacent node and said destination node with respect to said composite multi-cost function by applying said composite multi-cost function to said source-isolated connectivity matrix using said generally additive operator and said ordering;
deriving a composite constraint cost as a shortest path between each adjacent node and said destination node with respect to each individual cost function by applying each individual cost function to said source-isolated connectivity matrix using said generally additive operator and said ordering of said individual cost function;
defining link least-criterion-costs between said source node and said destination node, each being the generally additive sum of said cost functions of each respective link from said source node to a respective one of said adjacent nodes and said criterion cost from said respective one of said adjacent nodes to said destination node;
defining link composite-constraint-costs between said source node and said destination node, each being the generally additive sum of said cost functions of each said respective link from said source node to said respective one of said adjacent nodes and said composite constraint cost from said respective one of said adjacent nodes to said destination node;
ordering said link least-criterion-costs;
defining said at least one primary path as a route between said source node and said destination node having said link least-criterion-costs with the smallest multi-cost; and
defining said secondary paths as the remaining routes between said source node and said destination node.

9. The method according to claim 8, wherein each said secondary path is prioritized in inverse order with respect to the multi-cost of said link least-criterion-costs associated with each said secondary path.

10. The method according to claim 8, wherein said step of ordering said link least-criterion-costs includes determining said order by comparing associated link composite-constraint-costs if said link least-criterion-costs for two paths are the same.

11. The method according to claim 8, further including the step of comparing said multi-cost requirement to said link

least-criterion-costs, and if said multi-cost requirement is greater than or equal to said link least-criterion-costs for each individual cost component, determining that there is a path from said source node to said destination node which satisfies said multi-cost requirement.

12. The method according to claim 11, wherein said step of comparing said multi-cost requirement to said link lease-criterion-costs further includes, if at least one element of said multi-cost requirement is less than a corresponding one of said link least-criterion-costs for each path, but there is at least one path wherein said multi-cost requirement is greater than or equal to said one of said link composite-constraint-costs for each individual cost function, then there may or may not be a path which satisfies said multi-cost requirement for said at least one element while still satisfying all other elements of the multi-cost requirement.

13. The method according to claim 12, wherein said step of comparing said multi-cost requirement to said link least-criterion-costs further includes, if said multi-cost requirement is less than one of said link composite-constraint-costs for at least one individual element for each path, then there is no successful path for satisfying said multi-cost requirement between said source node and said destination node.

14. The method of claim 9, wherein said step of routing includes routing over the lowest priority secondary path which satisfies said multi-cost requirement.

15. The method according to claim 1, wherein

the nodes are interconnected by links;
each node and/or link may or may not support one or more service types generally available to the network;
said step of defining cost functions includes defining a cost function which is an ordering of said service types with respect to each one of said service types between each said pair of nodes; and
said step of defining a connectivity matrix includes defining a plurality of connectivity matrices, each corresponding to a desired one of said service types, each n-tuple of said connectivity matrix including an element indicative of service type available between each said pair of nodes.

16. The method according to claim 15, wherein

said step of defining a cost function which is an ordering of said service types with respect to each one of said service types includes assigning a value the magnitude of which is directly related to the desirability of said service types with respect to said one service type;
said step of defining a generally additive operator includes assigning an operator MAX for said service type cost function; and
said step of deriving said shortest path matrix includes deriving said summation n-tuples such that each summation n-tuple includes an element indicative of the most desirable one of said service types with respect to said one of said service types available over said shortest path between each said pair of nodes.

17. The method of claim 16, wherein:

said step of defining cost functions further includes defining a cost function which is indicative of the number of links traversed between a pair of nodes;
said step of defining a connectivity matrix includes defining a plurality of connectivity matrices, each corresponding to one of said service types, where m is the number of service types, each n-tuple of said connectivity matrix including an element indicative of the number of links traversed with respect to each service type between each said pair of nodes, where each of said links traversed is assigned a weighted service-related value of 1, H+1, $(H+1)^2$,..., $(H+1)^{m-1}$, with respect to said ordering of said service types, where H is the maximum allowed number of links traversed of all types between a source node and a destination node.

18. The method according to claim 1, wherein

the nodes are interconnected by links;
each node and/or link may or may not be part of a virtual private network;
said step of defining cost functions includes defining a cost function which is indicative of the ability to route traffic associated with said virtual private network; and
each n-tuple of said connectivity matrix includes an element indicative of the ability to route traffic associated

with said virtual private network directly between each said pair of nodes.

19. The method of claim 18, wherein said step of deriving said shortest path matrix includes deriving said summation n-tuples such that each summation n-tuple includes an element indicative of the ability to route traffic associated with said virtual private network over said shortest path between each said pair of nodes.

20. The method of claim 9, wherein said step of routing includes routing over said at least one primary path or a highest priority secondary path which satisfies said multi-cost requirement.

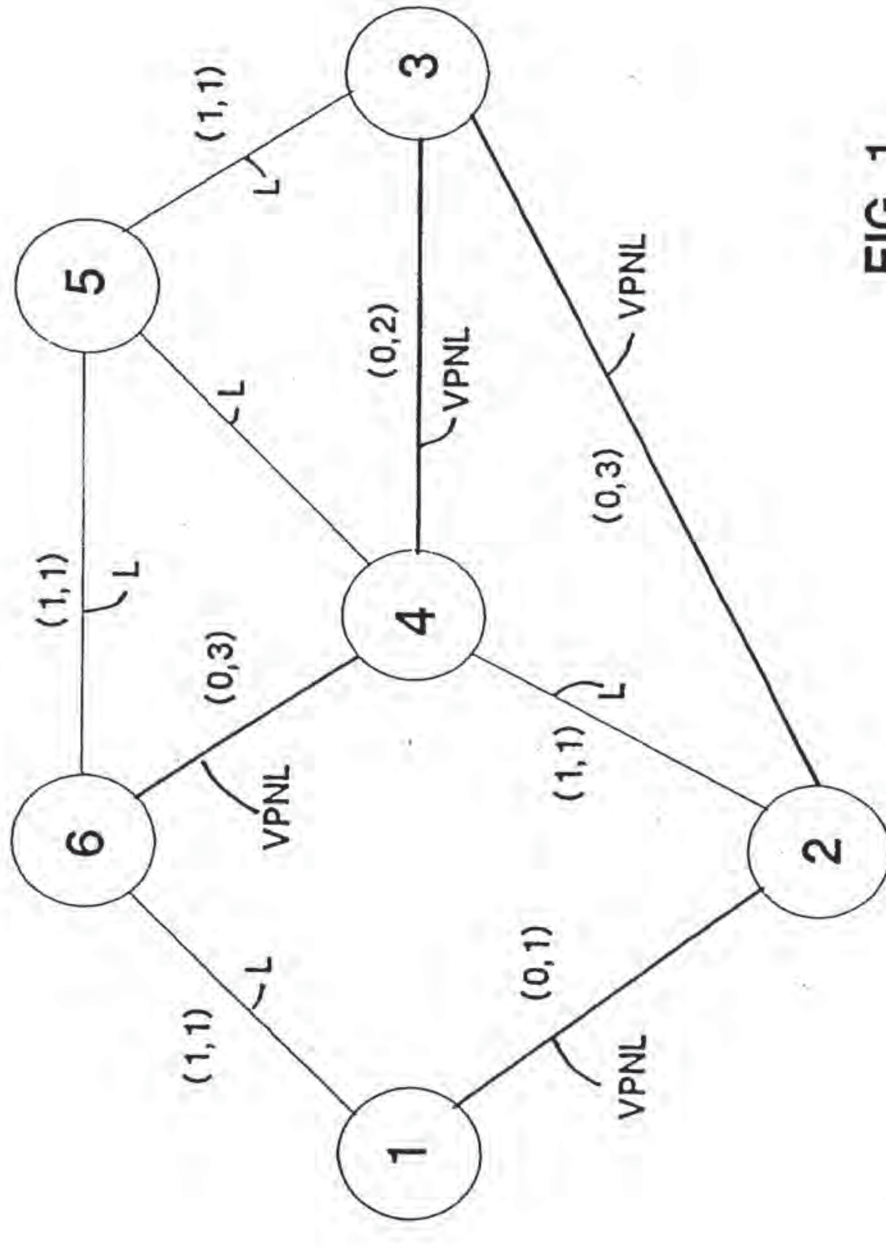


FIG. 1

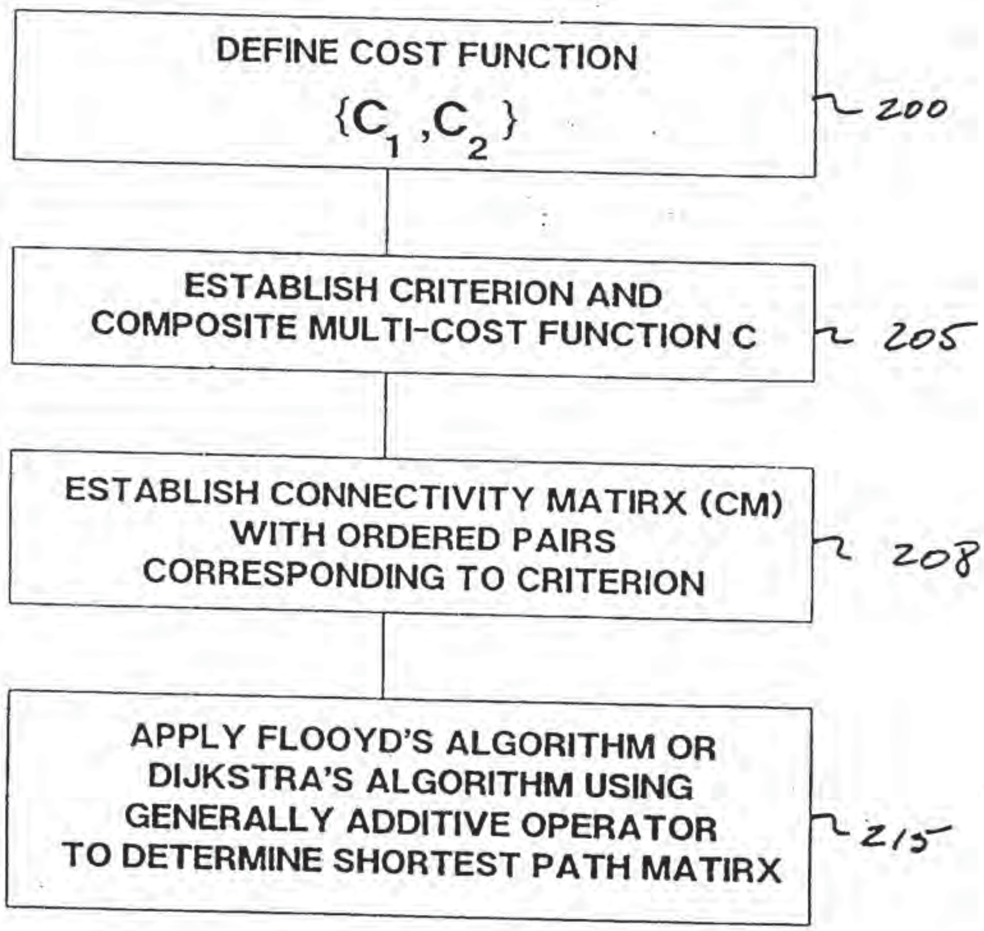

DEFINE COST FUNCTION
{C1 ,C2 }
200
ESTABLISH CRITERION AND
COMPOSITE MULTI-COST FUNCTION C
205
ESTABLISH CONNECTIVITY MATIRX (CM)
WITH ORDERED PAIRS
CORRESPONDING TO CRITERION
208
APPLY FLOOYD'S ALGORITHM OR
DIJKSTRA'S ALGORITHM USING
GENERALLY ADDITIVE OPERATOR
TO DETERMINE SHORTEST PATH MATIRX
215

FIG. 2

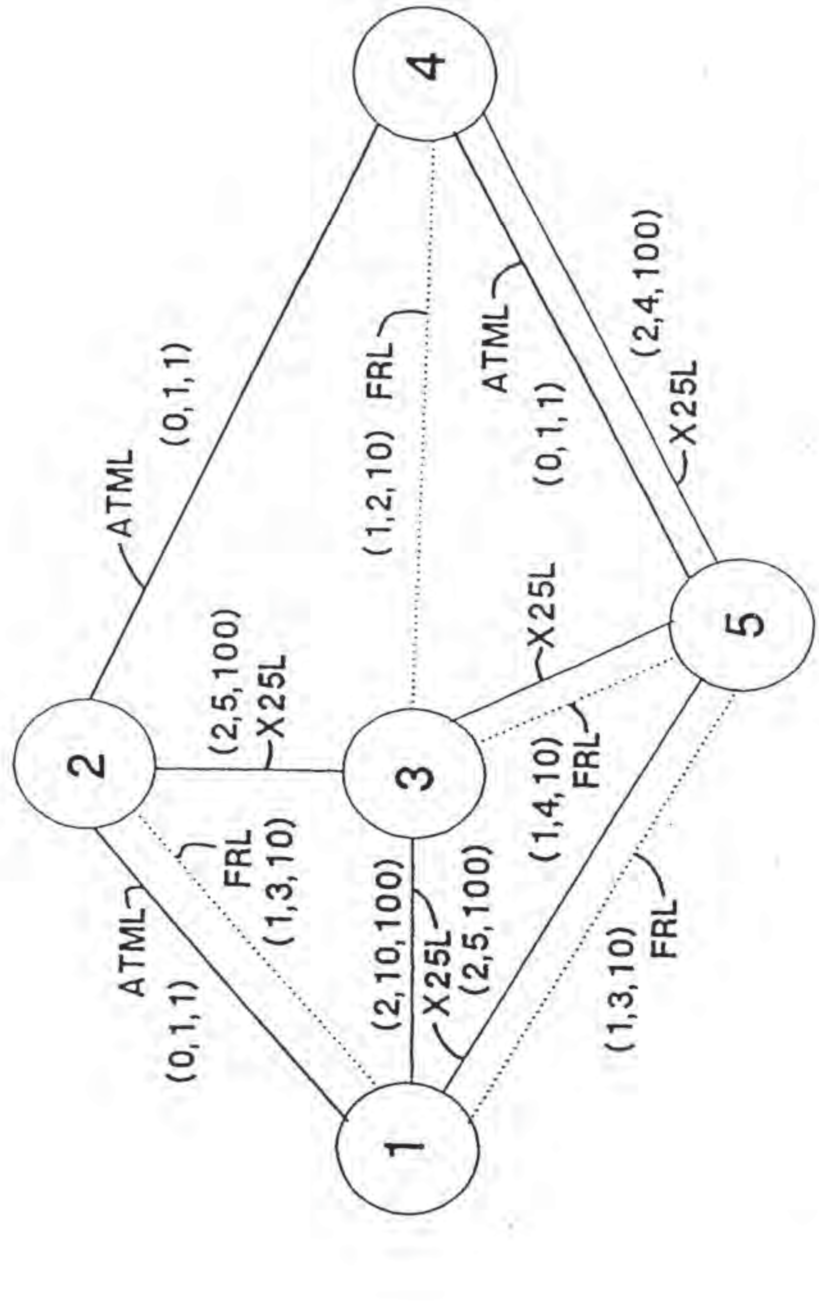

1
2
3
4
5
ATML
(0,1,1)
ATML
(0,1,1)
FRL
(1,3,10)
(2,5,100)
X25L
(2,10,100)
X25L
(2,5,100)
(1,2,10)
FRL
ATML
(0,1,1)
(2,4,100)
X25L
X25L
(1,4,10)
FRL
(1,3,10)
FRL

FIG. 3

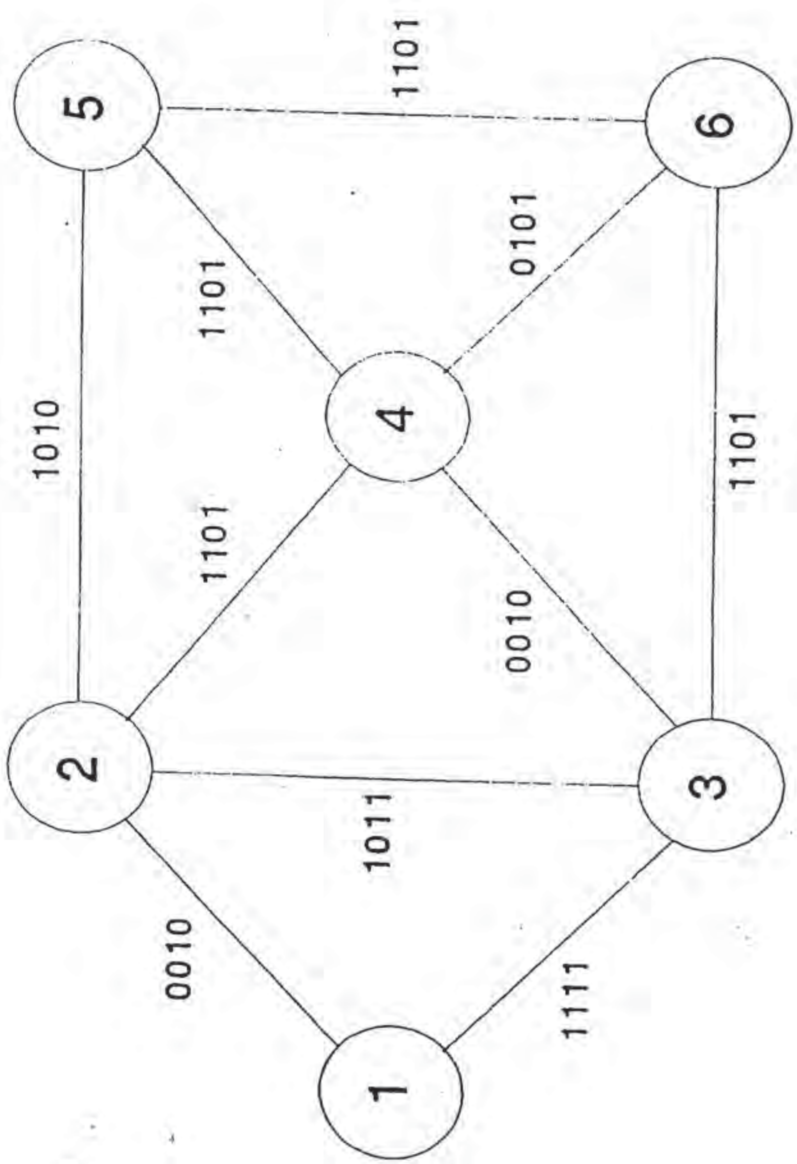


FIG. 4

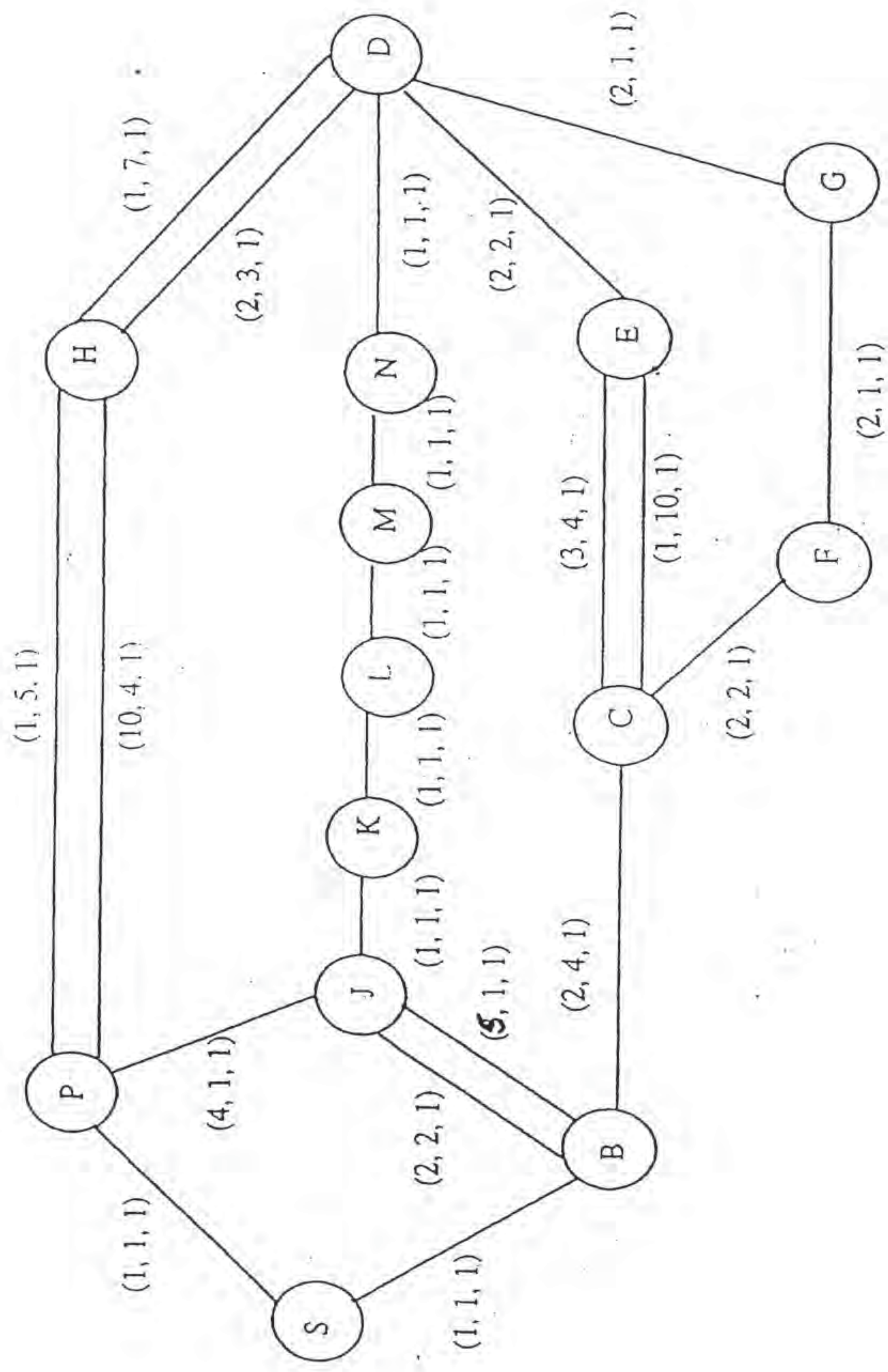

D
H
N
M
L
K
J
P
S
B
C
E
F
G
(1, 7, 1)
(2, 3, 1)
(1, 1, 1)
(2, 2, 1)
(2, 1, 1)
(1, 5, 1)
(10, 4, 1)
(1, 1, 1)
(1, 1, 1)
(3, 4, 1)
(1, 10, 1)
(2, 1, 1)
(1, 1, 1)
(2, 2, 1)
(1, 1, 1)
(5, 1, 1)
(2, 4, 1)
(4, 1, 1)
(2, 2, 1)
(1, 1, 1)
(1, 1, 1)

FIG. 5

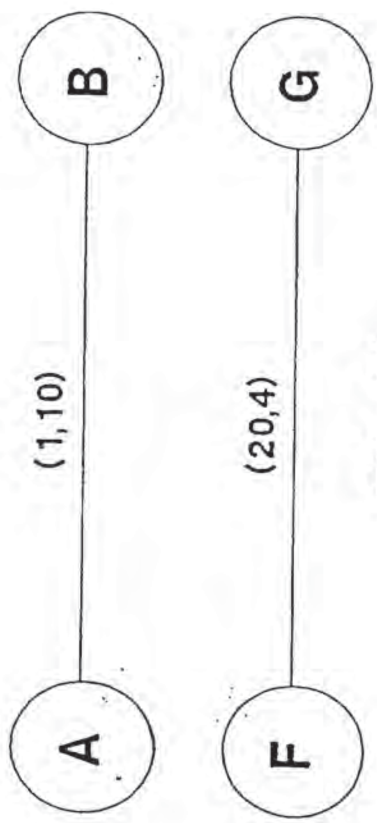


FIG. 6A

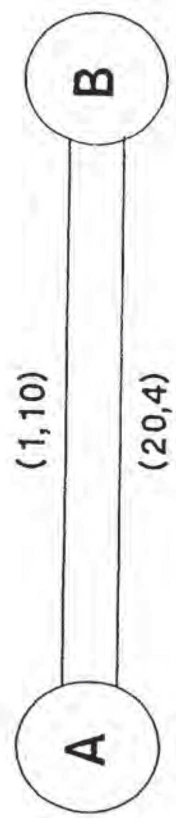


FIG. 6B

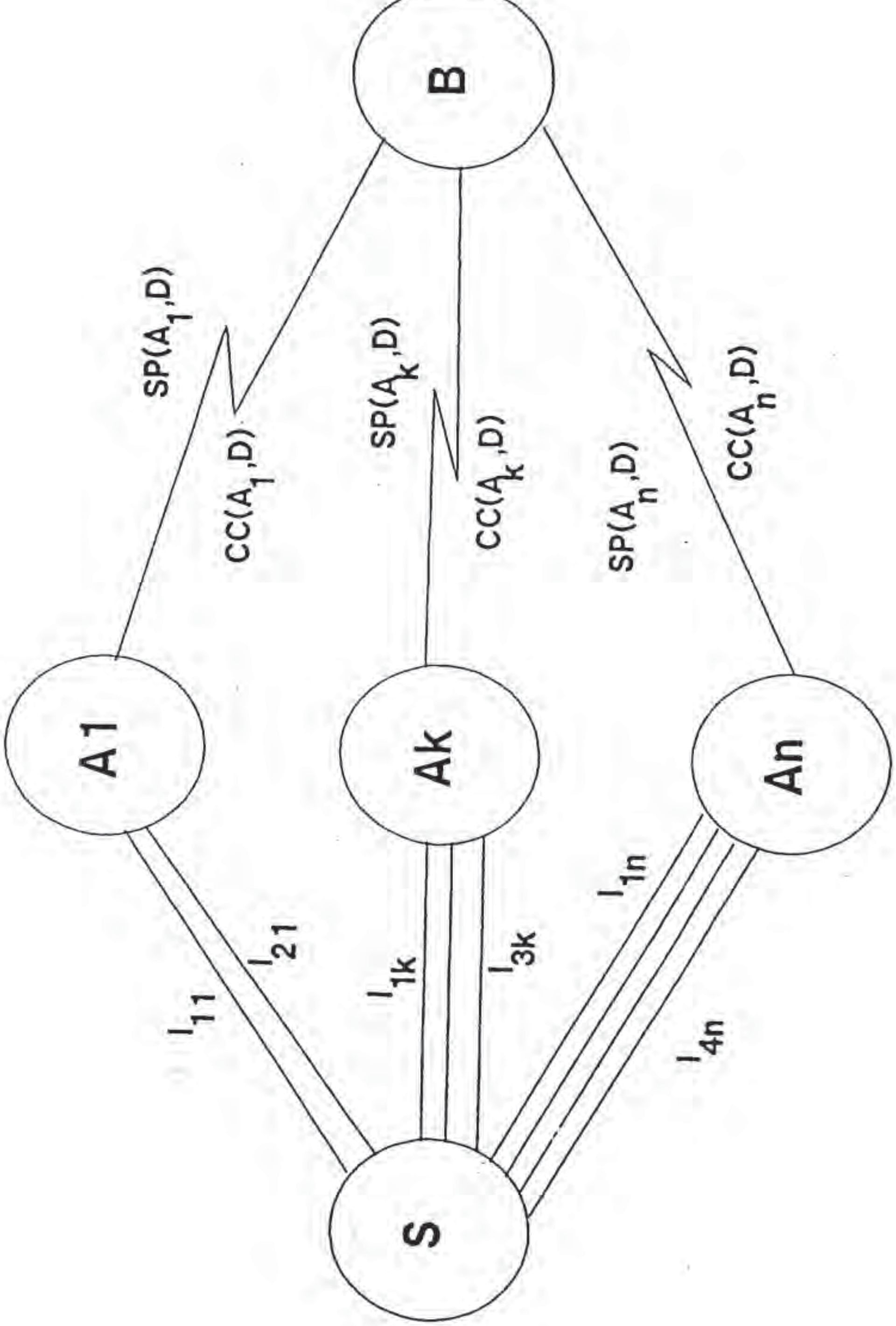

B
SP(A1,D)
CC(A1,D)
SP(Ak,D)
CC(Ak,D)
SP(An,D)
CC(An,D)
A1
Ak
An
l11
l21
l1k
l3k
l1n
l4n
S

FIG. 7

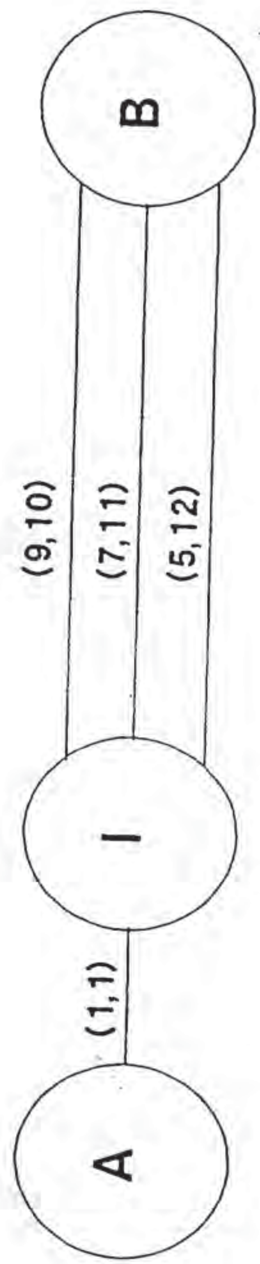


FIG. 8

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 830 047 A3

(12) EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
22.09.1999 Bulletin 1999/38

(43) Date of publication A2:
18.03.1998 Bulletin 1998/12

(21) Application number: 97440056.6

(22) Date of filing: 03.07.1997

(51) Int Cl.6: H04Q 11/00, H04L 12/56, H04Q 3/66

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV RO SI

(30) Priority: 03.07.1996 US 675166

(71) Applicant: ALCATEL
75008 Paris (FR)

(72) Inventor: Seid, Howard A.
Fairfax, Virginia 22033-1251 (US)

(74) Representative:
Knecht, Ulrich Karl, Dipl.-Ing. et al
Alcatel
Intellectual Property Department, Stuttgart
Postfach 30 09 29
70449 Stuttgart (DE)

(54) Connectivity matrix-based multi-cost routing

(57) Connectivity matrix-based multi-cost routing includes defining a generally additive operator which is able to add traditionally (arithmetic) additive cost factors and which takes into account cost factors which are not additive, the generally additive operator being defined such that distributive and communicative properties are applicable, and wherein the generally additive operator is applicable to connectivity matrix-based factors for determining the relative costs of paths within a network, particularly with respect to multi-cost factors. Connectivity matrix-based multi-cost routing is performed by first defining cost functions and establishing a criteria for prioritizing cost functions such that a composite multi-cost function includes the cost functions in the priority order defined by the criterion. A connectivity matrix is established including ordered n-tuples of cost factors corresponding to the priority established by the criterion, and a shortest path matrix determination is made by using the generally additive operator to apply the composite multi-cost function to the connectivity matrix. When links within a network support various functionality, a mask of a required functionality may be used to define a cost function for a given shortest path matrix determination. A correcting method is provided for a routing determination when, after a shortest path matrix determination, a routing choice is not provided which would otherwise satisfy a multi-cost requirement, the correcting method including the determination of a primary path and secondary paths between a source node and a destination node.

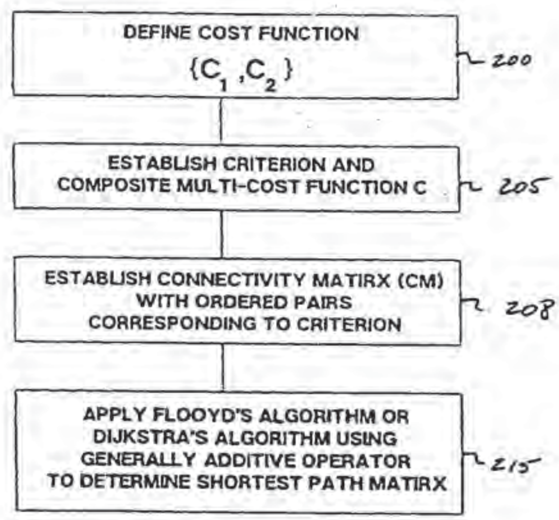


FIG. 2

EP 0 830 047 A3

Printed by Jouve, 75001 PARIS (FR)

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 97 44 0056

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
X	WANG Z ET AL: "BANDWIDTH-DELAY BASED ROUTING ALGORITHMS" IEEE PROCEEDINGS OF THE GLOBAL TELECOMMUNICATIONS CONFERENCE, SINGAPORE, 14-16 NOVEMBER 1995, pages 2129-2133, XP000633662 * the whole document *	1,2,5,6	H04Q11/00 H04L12/56 H04Q3/66
A		3,4, 15-19	
X	WO 95 18498 A (IBM) 6 July 1995 (1995-07-06) * page 4, line 18 - page 6, line 13 * * page 7, line 10 - page 9, line 2 *	1,5,6	
A		2-4, 15-19	
A	LEE W C ET AL: "ROUTING SUBJECT TO QUALIFY OF SERVICE CONSTRAINTS IN INTEGRATED COMMUNICATION NETWORKS" IEEE NETWORK, vol. 9 (1995), July/August, no. 4, pages 46-55, XP000526591 * page 47, line 37 - line 60 *	1,2, 7-14,20	

TECHNICAL FIELDS SEARCHED (Int.Cl.6)
H04L
H04Q

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	29 July 1999	Vercauteren, S

CATEGORY OF CITED DOCUMENTS
X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 97 44 0056

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

29-07-1999

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9518498 A	06-07-1995	DE 69325398 D EP 0736237 A JP 9504671 T	22-07-1999 09-10-1996 06-05-1997

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

PCT
WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F	A2	(11) International Publication Number:	WO 98/44400
		(43) International Publication Date:	8 October 1998 (08.10.98)

(21) International Application Number: PCT/US98/05881

(22) International Filing Date: 26 March 1998 (26.03.98)

(30) Priority Data:
08/831,063 1 April 1997 (01.04.97) US

(71) Applicant (*for all designated States except US*): SONY TRANS COM INC. [US/US]; 1833 Alton Avenue, Irvine, CA 92606 (US).

(72) Inventor; and
(75) Inventor/Applicant (*for US only*): FUKUI, Toshiharu [JP/US]; 266 Baywood Drive, Newport Beach, CA 92660 (US).

(74) Agents: HAVERSTOCK, Thomas, B. et al.; Haverstock & Owens LLP, Suite 420, 260 Sheridan Avenue, Palo Alto, CA 94306 (US).

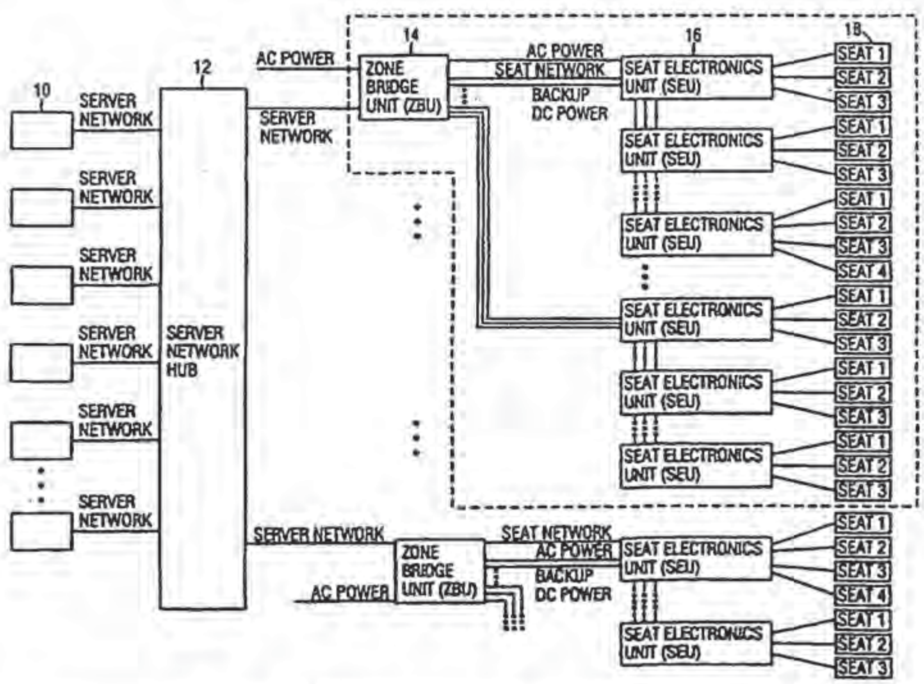
(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: AUTOMATIC CONFIGURATION SYSTEM FOR MAPPING NODE ADDRESSES WITHIN A BUS STRUCTURE TO THEIR PHYSCAL LOCATION

(57) Abstract

An automatic configuration system maps a device address of each node coupled to a bus structure to a network protocol address corresponding to the physical location of the node. A configuration database is built which includes the network protocol address of each node and its corresponding device address. A wiring database, including each device position within the network, is maintained within a network server. Preferably, the network is an IEEE 1394 serial bus network. A topology map, including the device addresses of the nodes within the network and their relationship to each other, is generated during a self-ID sequence and is maintained by the network server. This information is then compared to the wiring database by the network server in order to build a restored topology map including the corresponding device address for each device at each position. A DHCP database is then generated which includes each device address and its corresponding network protocol address. The network protocol address corresponds to the position of the respective device. Using the DHCP database, the network protocol address is maintained for the devices at each position, in order to monitor the positions of the devices and communications from the devices.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

AUTOMATIC CONFIGURATION SYSTEM FOR MAPPING NODE ADDRESSES WITHIN A BUS STRUCTURE TO THEIR PHYSICAL LOCATION

FIELD OF THE INVENTION:

The present invention relates to the field of configuration of nodes within a bus structure. More particularly, the present invention relates to the field of automatic configuration of node addresses within a bus structure.

BACKGROUND OF THE INVENTION:

The IEEE standard, "P1394 Standard For A High Performance Serial Bus," Draft 8.0v2, July 7, 1995, is an international standard for implementing an inexpensive high-speed serial bus architecture which supports both asynchronous and isochronous format data transfers. Isochronous data transfers are real-time transfers which take place such that the time intervals between significant instances have the same duration at both the transmitting and receiving applications. Each packet of data transferred isochronously is transferred in its own time period. The IEEE 1394 standard bus architecture provides multiple channels for isochronous data transfer between applications. A six bit channel number is broadcast with the data to ensure reception by the appropriate application. This allows multiple applications to simultaneously transmit isochronous data across the bus structure. Asynchronous transfers are traditional data transfer operations which take place as soon as possible and transfer an amount of data from a source to a destination.

The IEEE 1394 standard provides a high-speed serial bus for interconnecting digital devices thereby providing a universal I/O connection. The IEEE 1394 standard defines a digital interface for the applications thereby eliminating the need for an application to convert digital data to analog data before it is transmitted across the bus. Correspondingly, a receiving application will receive digital data from the bus, not analog data, and will therefore not be required to convert analog data to digital data. The cable required by the IEEE 1394 standard is very thin in size compared to other bulkier cables used to connect such devices. Devices can be added and removed from an IEEE 1394 bus while the bus is active. If a device is so added or removed the bus will then automatically reconfigure itself for transmitting data between the then existing nodes. A node is considered a logical entity with a unique address on the bus structure.

Each node on the IEEE 1394 bus structure has a 16 bit node ID. The node ID is the address that is used for data transmission on the data link layer. This allows address space for potentially up to 64K nodes on the bus structure. The node ID is divided into two smaller fields: the higher order 10 bits specify a bus ID and the lower order 6 bits specify a physical ID. The bus ID is assigned by a root node and the physical ID is assigned during a self identify sequence upon reset of the bus. Each physical ID field is unique in a single IEEE 1394 bus, but the physical ID field is not a fixed value for each node itself. The physical ID field is fixed for the position of the node. If a device is moved from one position in the IEEE 1394 bus to another position within the same IEEE 1394 bus, the device will have a different node ID because its physical ID will have a different value when in the new position.

Within each of the bus ID and physical ID fields a value of all logical "1"s is reserved for special purposes. Accordingly, this addressing scheme provides for up to 1023 busses, each with 63 independently addressable nodes. Each IEEE 1394 compatible device includes a node unique ID which is a 64 bit number saved within a configuration read-only memory (ROM) of the device. The node unique ID is permanent for each device and does not depend on the position of the device within an IEEE 1394 bus. The node unique ID is not used for addressing of data transmissions on the data link layer.

IEEE 1394 serial bus and other complex networks, although necessary in many environments, are very complex and difficult to maintain. It is desirable in such networks to have the ability to automatically configure and maintain a mapping of the devices within the network and their corresponding physical positions. For example, within an in-flight entertainment system implemented on an aircraft, it is likely that the airline responsible for the aircraft will periodically reconfigure the layout of the cabin or remove and replace seats and equipment within the cabin. In such instances, it is desirable for the in-flight entertainment system to have the ability to automatically map the devices to their corresponding physical position within the aircraft. Without this automatic configuration ability, this map or database of the devices to their physical position would have to be manually maintained and updated after installation of the system and when devices within the system are replaced or reconfigured. This type of manual method is labor intensive and will be subject to human error. Having the ability for the network to automatically map the devices to their physical position will reduce the possibility of this human error, allow flexibility for platform configuration changes and minimize the downtime of the network.

Including a database within the network which automatically maps devices to their corresponding physical location will also make it easy to complete device specific or user specific billing, easy to block service or types of services to particular devices and easy to personalize service to specific devices and users. Automatically maintaining such a configuration database also makes it easy to display the network configuration and topology of the devices for the use of both attendant and maintenance personnel.

SUMMARY OF THE INVENTION:

An automatic configuration system maps a device address of each node coupled to a bus structure to a network protocol address corresponding to the physical location of the node. A configuration database is built which includes the network protocol address of each node and its corresponding device address.

The preferred embodiment of the automatic configuration system is implemented within an in-flight entertainment system. Passengers access this in-flight entertainment system, from their seats, using passenger sets of seat peripherals. The passenger sets are controlled by seat CPUs within a seat electronic unit. The seat CPUs are coupled together within a zone and to a zone bridge unit forming a bus structure, with each seat CPU forming a node on the bus structure. Preferably, this bus structure is an IEEE 1394 serial bus network. The zone bridge unit controls and routes communications between the seat CPUs and the headend of the system.

A wiring database, including each seat position within the aircraft, is maintained within a system manager unit. A topology map, including the device addresses of the nodes within the zone and their relationship to each other, is generated during a self-ID sequence and is maintained for each zone by the zone bridge unit. This information is then compared to the wiring database by the system manager unit in order to build a restored topology map which includes the corresponding device address for each device at each seat position. A dynamic host configuration protocol (DHCP) database is then generated which includes each device address and its corresponding network protocol address. The network protocol address corresponds to the seat position of the respective device. Using this DHCP database, each seat CPU can then determine the network protocol address for the seat position where the seat CPU has been installed. This network protocol address is obtained by the seat CPUs from the DHCP database at boot time.

In an alternate embodiment of the present invention, this DHCP database is then combined with a passenger list to build a passenger database. The passenger database is updated for each flight to also include the appropriate passenger list and each passenger's assigned seat. Personalized messages can then be sent to each passenger and usage of the system by each passenger can also be tracked by the system. This passenger database can also be displayed for use by both attendant and maintenance personnel in order to configure the network for specific use patterns and maintain and monitor the network.

<u>BRIEF DESCRIPTION OF THE DRAWINGS</u>:

Figure 1 illustrates a block diagram of an overall in-flight entertainment system in which the preferred embodiment of the present invention is implemented.

Figure 2 illustrates a block diagram of the components within a zone of an aircraft.

Figure 3 illustrates a flow diagram showing the flow and the steps included in generating a configuration database of the present invention.

Figure 4 illustrates an example of the zone of Figure 2 after a self-ID sequence has been run.

Figure 5 illustrates a topology link list for the zone of Figure 2.

Figure 6 illustrates a topology link list for the zone of Figure 2, including MAC addresses for the devices within the zone.

Figure 7 illustrates a wiring database for the zone of Figure 2.

Figure 8 illustrates a restored topology map for the zone of Figure 2.

Figure 9 illustrates a restored topology link list for the zone of Figure 2, including the IP address for the devices within the zone.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT:

The preferred embodiment of the present invention is implemented within an in-flight entertainment system, as taught in U.S. Patent Application Serial Number 08/714,772, filed on September 16, 1996, and entitled "Combined Digital Audio/Video On Demand And Broadcast Distribution System," which is hereby incorporated by reference. A block diagram of this overall in-flight entertainment system is illustrated in Figure 1. Within a headend of this system, one or more servers 10 are coupled to a server network hub 12. The headend of this system includes a data server, a media controller, one or more media servers, a system manager unit and an attendant control panel. Coupled to the server network hub 12 are one or more zone bridge units 14. The server network hub 12 is preferably an ATM switch which routes communications between the servers 10 and the zone bridge units 14. Each zone bridge unit 14 is coupled to one or more seat electronics units 16. Seat central processing units (CPUs) within the seat electronics units 16 are coupled to control passenger control sets of seat peripherals 18. Each seat electronics unit 16 preferably includes a number of seat CPUs for controlling a number of passenger control sets of seat peripherals 18. The zone bridge units 14 and the seat electronics units 16 are preferably coupled together by an IEEE 1394 serial bus network. Each seat CPU provides a node on the bus structure and controls local applications at the passenger control sets 18. Through the passenger control sets 18, a passenger has access to one or more services such as audio and video on demand, video games, gambling, telephone service and information services. Each passenger control set of seat peripherals 18 includes a set of peripherals which allow the passenger to access the features and services available through the system. Preferably, each passenger control set 18 includes a seat video display, a passenger control unit and a passenger control handset, which are coupled for sending communications to and receiving communications from the corresponding seat CPU. The passenger control units are mechanically stored within the seats of the aircraft, such that when a seat is replaced, the passenger control unit is also replaced. The seat electronics units can also be replaced or reconfigured during maintenance or reconfiguration of the aircraft.

An automatic configuration system according to the present invention is used to map a device or media access control (MAC) address of each passenger control unit within an aircraft to the physical location or seat position of the passenger control unit within the aircraft. Using this automatic configuration system, a configuration database is built and stored within the system manager unit. The configuration database maps the MAC address to a network protocol address. The system manager unit in connection with a dynamic host configuration protocol (DHCP) server within the headend of the system is used to build this configuration database. Using this DHCP database, each seat CPU can then determine the network protocol address for the seat position where the seat CPU has been installed. At boot time, each seat CPU then obtains its corresponding network protocol address using this configuration database.

Preferably, each seat CPU has a unique MAC address which is saved within a ROM inside the seat CPU corresponding to the passenger control unit. Alternatively, the unique MAC address is saved within a memory device inside any other appropriate piece of equipment, including the passenger control set of seat peripherals 18. In a further alternate embodiment, the unique MAC address is saved within any other appropriate memory device including an EEPROM, EPROM or flash memory device. For communications over the network between the headend and the passenger control sets, a transmission control protocol (TCP) / Internet Protocol (IP) or network protocol address is preferably used. When building the configuration database, the system manager unit maps each seat CPU, using its MAC address, to a corresponding IP address. The IP address is mapped to a seat location or position within the aircraft and is used to identify the passenger control units for communications using TCP/IP.

A block diagram of the components within a zone of an aircraft is illustrated in Figure 2. A zone bridge unit 20 includes an IEEE 1394 network interface 22 through which communications over an IEEE 1394 serial bus network 26 are routed. A number of seat electronics units 24 are coupled to the network interface 22 by the bus structure 26. Each seat electronics unit 24 includes a number of seat CPUs 28, each coupled to a corresponding passenger control unit 29. Each seat CPU 28 includes an IEEE 1394 network interface 27 through which communications over the IEEE 1394 serial bus network 26, between the seat CPUs 28 and the zone bridge unit 20, are routed. Each seat CPU 28 also includes a unique MAC address. If a seat CPU 28 and corresponding passenger control unit 29 are replaced, the new seat CPU 28 will have a different MAC address. The zone bridge unit 20 is coupled to an ATM switch 12 (Figure 1) for controlling and routing communications between the seat CPUs 28 and the headend of the in-flight entertainment system. Each passenger control unit 29 corresponds to a seat position within the aircraft.

A flow diagram showing the flow and the steps included in generating a configuration database of the present invention is illustrated in Figure 3. A wiring database 30 is generated when the system is designed and incorporated into an aircraft. Once the system is implemented within the aircraft, the wiring database for each zone is loaded within the system manager unit. The wiring database includes a wiring map of the seat numbers within each of the zones and their relationship to each other and to the zone bridge unit 20. The wiring database also includes the correlation between the seat position and the port of the seat CPU 28. The seat CPU 28 corresponds to a passenger control unit 29 which corresponds to a seat position. The wiring database for the zone illustrated within Figure 2 includes seat numbers 1A-1C, 2A-2C and 3A-3C.

Each zone bridge unit 20 builds a topology map 32 using the self-ID sequence of the IEEE 1394 serial bus protocol. This self-ID sequence determines the physical topology of the nodes on the IEEE 1394 serial bus network due to their port connections. From each port it is determined if there is a connection and if the connection is to a child or parent node. During this self-ID sequence, all of the physical connections are assigned a direction pointing towards the root node. The zone bridge unit 20 is preferably configured as the root node within each zone. The direction is set by labeling each connected port as either a parent port or a child port. A parent port is a port connected to a node closer to the root than the reporting node. A child port is a port connected to a node further from the root than the reporting node. From these relationships, the zone bridge unit 20, as the root node, can determine the physical topology of the devices connected to the IEEE 1394 serial bus network within its zone.

Once the zone bridge unit 20 builds the topology map 32, it then obtains the MAC addresses 34 from each of the seat CPUs 28 within the zone and includes the MAC addresses 34 within the topology map 32. Once built, this topology map 32 includes the physical topology of the devices connected to the IEEE 1394 serial bus network within the zone and the corresponding MAC address of those devices.

This topology map 32 is then compared to the wiring database 30 and a restored topology map 38 is generated which includes each seat position and a corresponding MAC address of the seat CPU 28 at that seat position. The restored topology map 38 for each zone is generated by the system manager unit. A host database 36 includes a host name and a corresponding IP address. The host name is the name assigned to the respective device corresponding to the IP address. For example, in the preferred embodiment the host name for the seat CPUs is a combination of the word "SEAT" and the position of the seat; e.g. "SEAT1A." The host database is also generated when the system is designed and incorporated into an aircraft. The host database is also stored within the system manager unit. The host database 36 is then merged with the restored topology map 38 to generate a DHCP database 40 which maps the IP address of each seat position to the MAC address of the seat CPU 28 at that seat position.

In order to maintain the DHCP database, the system manager unit will preferably perform this configuration method when triggered under maintenance mode, when the seat configuration is changed or a seat CPU 28 is replaced. Alternatively, as should be apparent to those skilled in the art, the DHCP database can be automatically updated on a periodic basis or after a predetermined event. Within the preferred embodiment of the present invention, the DHCP database is only updated when manually triggered or initiated during maintenance of the aircraft, when a seat electronics unit is replaced or an airplane is reconfigured.

The zone illustrated in Figure 2 will be used as an example to describe the methods used to generate the configuration database of the preferred embodiment of the present invention. Each seat CPU 28 within the zone includes a unique MAC address. During the self-ID sequence, the zone bridge unit 20 generates the topology map 32 and assigns a physical ID to each seat CPU 28. An example of the zone after the self-ID sequence has been run is illustrated in Figure 4. As illustrated in Figure 4, the seat CPUs 28 have all been assigned a physical ID.

After the self-ID sequence is run, the zone bridge unit 20 generates the topology map 32, which is illustrated below, within Table 1.

Length	CRC
node_count=10	self_id_count=10

Self ID Packet #0: NNP
Self ID Packet #1: NCP
Self ID Packet #2: NNP
Self ID Packet #3: NCP
Self ID Packet #4: NNP
Self ID Packet #5: NCP
Self ID Packet #6: CPN
Self ID Packet #7: CPC
Self ID Packet #8: CPC
Self ID Packet #9: NNC

Table 1: Zone Topology Map

This topology map is built and stored by the zone bridge unit 20. The node_count value is equal to the number of nodes within the zone. In this example, the node_count value is equal to ten, including the nine seat CPUs 28 and the zone bridge unit 20. The self_id_count value is equal to the number of self-ID packets that are received by the zone bridge unit 20 during the self-ID sequence. In this example, the self_id_count value is equal to ten, because all of the nodes within the zone are operational and reported their self-ID packets to the zone bridge unit 20. If a node is not operational when the self-ID sequence is run, the physical layer of the node will still send a self-ID packet to the zone bridge unit 20. The physical layer of the node will not send a self-ID packet to the zone bridge unit 20 only in extreme circumstances, such as if the node transceiver is broken or if the cable is bad.

The topology map, illustrated in Table 1, includes the self-ID packets from each of the nodes within the zone. Each IEEE 1394 self-ID packet includes more information than is shown in Table 1. Only the information relevant to the operation of the present invention is shown within the self-ID packets listed in Table 1. The information shown within the topology map of Table 1 includes the port status of each operational node within the zone. The self-ID packets within Table 1 each include a number which corresponds to the physical ID of the node which sent the particular self-ID packet. The last three letters within a self-ID packet correspond to the port connections of the corresponding node: if the port is not connected to another port, a value N for that port is reported; if the port is connected to a child node, a value C for that port is reported; and if the port is connected to a parent node, a value P for that port is reported. The last three letters within the self-ID packets are arranged to correspond to the order of the ports within each node, as numbered within the Figure 4. In this example, each node includes three ports.

The self-ID packet for the node having a physical ID equal to zero, which corresponds to the seat position 1C, includes port status information equal to NNP. As illustrated in Figure 4, ports 1 and 2 of this seat CPU are not connected and port 3 is connected to a parent port. The remaining self-ID packets within Table 1, represent the port connections of the corresponding seat CPUs within the zone illustrated in Figure 4.

Once the zone bridge unit 20 generates the topology map of Table 1, the system manager unit then creates a topology link list as illustrated in Figure 5. The system manager unit retrieves the topology map from the zone bridge unit 20 and creates the topology link list by processing the self-ID packets from the zone bridge unit 20. The system manager unit processes the last received self-ID packet, which has a physical ID equal to 9 and corresponds to the zone bridge unit 20, and then processes the remaining self-ID packets in the reverse order from which they were received until the first received self-ID packet, which has a physical ID equal to 0 and corresponds to the seat CPU at the seat position 1C is processed. By processing the self-ID packets in this order, the system manager unit will generate the topology link list illustrated in Figure 5.

The system manager unit then reads the MAC addresses of each of the nodes within the zone and adds the MAC addresses to the topology link list. The MAC addresses are read by the system manager unit through the zone bridge unit. The topology link list including the MAC addresses is illustrated in Figure 6.

The wiring database 30 stored within the system manager unit, is illustrated in Figure 7. This wiring database 30 includes the component position for the zone bridge unit and the seats within the zone, with a map of the design for those components. As described above, this wiring database 30 is prepared by the system designer and is entered into the system manager unit when the aircraft is designed and the system is implemented.

The system manager unit compares the topology link list 32, including the MAC addresses, with the wiring database 30 to obtain the restored topology map 38, as illustrated in Figure 8. This restored topology map includes the information from the topology link list and the corresponding position for each node, including the seats and the zone bridge unit. The restored topology map 38 therefore will include the physical ID for each node, the representative port information for the ports of each node, the MAC address and the corresponding position within the aircraft for each node.

The host database 36 for the zone of Figure 4, is illustrated below within Table 2.

HOST DATABASE	
IP ADDRESS	HOST NAME
10.0.0.8	BRIDGE1
10.0.0.10	SEAT1A
10.0.0.11	SEAT1B
10.0.0.12	SEAT1C
10.0.0.13	SEAT2A
10.0.0.14	SEAT2B
10.0.0.15	SEAT2C
10.0.0.16	SEAT3A
10.0.0.17	SEAT3B
10.0.0.18	SEAT3C

Table 2: Host Database

The host database illustrated within Table 2 is prepared by the system designer and entered into the system manager unit when the system is designed and implemented on an aircraft. The host database is independent of the wiring database. The host database combines the host name or position of the node with the IP or network protocol address of that position. This IP address is therefore consistent for the position within the aircraft, independent of the actual device or component that is located at that position.

After the restored topology map 38 is generated, the system manager unit combines the restored topology map 38 and the host database 36 to generate the DHCP database 40. The system manager unit first generates a restored topology link list which includes the IP address, as illustrated in Figure 9. From this restored topology link list, the system manager unit then generates the DHCP database 40, which is illustrated in Table 3 below and includes only the IP address and the corresponding MAC address for each node within the zone.

DHCP DATABASE	
IP ADDRESS	MAC ADDRESS
10.0.0.8	08:0046:00:aa:03
10.0.0.10	08:0046:00:12:03
10.0.0.11	08:0046:00:12:f3
10.0.0.12	08:0046:00:12:35
10.0.0.13	08:0046:00:12:28
10.0.0.14	08:0046:00:12:97
10.0.0.15	08:0046:00:12:ac
10.0.0.16	08:0046:00:12:a1
10.0.0.17	08:0046:00:12:b3
10.0.0.18	08:0046:00:12:59

Table 3: DHCP Database

This DHCP database is then utilized to coordinate the proper IP address with the corresponding component's MAC address. The IP address corresponds to a particular seat position within the aircraft within the host database of Table 2. Accordingly, in order to access a particular seat CPU at a specific seat position, the system manager unit uses the IP address corresponding to that seat position to determine the appropriate corresponding MAC address for the seat CPU at that seat position. In this same manner, usage and other appropriate information can be tracked to a seat position using the corresponding MAC address of the communicating device.

When a seat which includes a passenger control unit is replaced, the system will perform the automatic configuration method of Figure 3 to update the configuration database with the new MAC address of the seat CPU corresponding to the IP address at the seat position. When a seat electronics unit is removed or replaced within the IEEE 1394 serial bus network, the system manager unit will be instructed to automatically update the configuration database by performing the method described above as part of the maintenance routine. During this update, the zone bridge unit will have to run the self-ID sequence to obtain the topology map and MAC addresses for the components within the zone.

In an alternate embodiment of the present invention, application software will combine the DHCP database with a passenger list for each flight, thereby allowing the application to address a specific passenger through a passenger set of seat peripherals at a specific seat within the aircraft. This passenger list for each flight can be provided at an attendant control station or via any other appropriate links to the in-flight entertainment system. Greetings, messages and available features can then be tailored specifically to each passenger. In this alternate embodiment, the passenger list and corresponding seat position for each flight are loaded into the system from the airline's database. This passenger information includes the passenger's name, the seat to which the passenger is assigned and the passenger's previous use patterns on the in-flight entertainment system. Thus a passenger can be advised for example regarding a list of in-flight movies previously seen, music listened to, and the like. Using this passenger database, the system is able to send personalized communications to the passenger by addressing the communications to the network protocol address corresponding to the seat CPU controlling the passenger control unit at the passenger's seat. Using the passenger database, the system is also able to block service or types of services to particular seat CPUs or particular passengers. When receiving communications from the seat CPU, the system can also determine which passenger is requesting which information, using the passenger database. A passenger's activity on the in-flight entertainment system is thereby tracked so that available features on future flights can be tailored to that passenger's previous requests. Using this passenger database also maximizes the ease of billing for services provided to a passenger. It is important to the airlines operating the system that the billing for services provided is easy and transparent. The passenger database can also be displayed at an attendant control station or a maintenance station. This display makes it easy for maintenance and attendant personnel to configure the network for specific use patterns, such as blocking service or types of services to particular devices, and to maintain and monitor the network.

In the manner as described above, a configuration database is automatically maintained. The configuration database includes each IP address, MAC address and corresponding seat position. A wiring database is entered into the system manager unit once the wiring of the aircraft has been designed. The wiring database includes the wiring pattern of each seat position on the aircraft. This wiring database is then compared to a topology map which includes the MAC addresses of each component and the relationship of the components within the zone is received from each zone bridge unit for generating a restored topology map. The restored topology map is compared to a host database, which includes an IP or network protocol address corresponding to each seat position, to generate a DHCP database, which maps each IP address to a corresponding MAC address. Using this DHCP database, each seat CPU can then determine at boot time the network protocol address for the seat position where the seat CPU has been installed.

Without the automatic configuration system of the present invention, a configuration database would have to be manually maintained and updated after installation of the system and when any passenger control unit within an aircraft is replaced. This type of manual method is labor intensive and will be subject to human error.

In an alternative embodiment, the seat position may be saved on each seat CPU. In this alternate embodiment, the seat position is manually entered at each passenger set of seat peripherals, after the seat electronics unit is connected to the IEEE 1394 bus structure. The system manager unit will then query each seat CPU, over the IEEE 1394 bus structure, to compile the configuration database.

The present invention has been described in terms of specific embodiments incorporating details to facilitate the understanding of principles of construction and operation of the invention. Such reference herein to specific embodiments and details thereof is not intended to limit the scope of the claims appended hereto. It will be apparent to those skilled in the art that modifications may be made in the embodiment chosen for illustration without departing from the spirit and scope of the invention. For example, while an ATM network and IEEE 1394 serial bus network have been chosen for use within the preferred embodiment, it will be apparent that the automatic configuration system and method of the present invention can be used within other types of networks in other types of environments including but not limited to home, office, computer center, school and hospital. It will also be apparent that the entertainment system of the preferred embodiment of the present invention can also be readily adapted for operation in environments other than aircraft, including but not limited to other transportation modes, e.g. train, bus, ferry and cruise ship. It will be apparent that the automatic configuration system of the present invention can be easily adapted for use in other types of networks and in other environments than those specifically described herein.

CLAIMS

We Claim:

1. An apparatus for automatically mapping device addresses within a bus structure to corresponding physical locations of devices, comprising:
   a. a plurality of devices coupled together to form a bus structure;
   b. a network interface coupled to the bus structure for controlling communications to and from the devices and for maintaining a topology map representing device addresses of each of the devices; and
   c. a system controller coupled to the network interface for monitoring operation of the devices, wherein the system controller maintains a system wiring database including wiring diagram and physical locations of devices and further wherein the system controller compares the wiring database to the topology map and forms a restored topology map by matching each device address to a corresponding physical location of a device.

2. The apparatus as claimed in claim 1 further comprising a display for displaying the restored topology map.

3. The apparatus as claimed in claim 1 wherein each of the plurality of devices has a corresponding unique device address and further wherein when an existing device is replaced by a new device at a physical location, the device address of the new device is different than the device address of the existing device.

4. The apparatus as claimed in claim 3 wherein the bus structure is an IEEE 1394 serial bus network.

5. The apparatus as claimed in claim 4 wherein the network interface automatically generates the topology map during self-ID sequence of IEEE 1394 protocol.

6. The apparatus as claimed in claim 1 wherein the apparatus is part of an in-flight entertainment system for use on board an aircraft and each device is a passenger control unit through which a passenger has access to the in-flight entertainment system.

7. The apparatus as claimed in claim 1 wherein the system controller further maintains a host database representing network protocol addresses of the physical locations and compares the host database to the restored topology map to form a link list, listing each device address and corresponding network protocol address.

8. The apparatus as claimed in claim 6 wherein the restored topology map is updated for each flight of the aircraft to also include passenger information thereby forming a passenger configuration database.

9. The apparatus as claimed in claim 8 further comprising a display for displaying the passenger configuration database.

10. A method of automatically mapping device addresses to physical locations of devices within a bus structure corresponding to the device addresses, comprising the steps of:

a. maintaining a wiring database including wiring diagram and physical locations of devices;

b. maintaining a topology map representing device addresses of devices coupled to the bus structure; and

c. comparing the wiring database to the topology map and matching device addresses of devices coupled to the bus structure to physical locations of devices.

11. The method as claimed in claim 10 wherein each device address is unique and further wherein when an existing device is replaced by a new device, the device address of the new device is different than the device address of the existing device.

12. The method as claimed in claim 10 further comprising the step of forming a configuration database including physical locations and corresponding device addresses.

13. The method as claimed in claim 12 further comprising the step of matching the device addresses to network protocol addresses corresponding to physical locations of devices.

14. The method as claimed in claim 10 wherein the bus structure is an IEEE 1394 serial bus network.

15. The method as claimed in claim 14 wherein the topology map is automatically generated during a self-ID sequence of IEEE 1394 protocol.

16. The method as claimed in claim 10 wherein the method is implemented within an in-flight entertainment system for use on board an aircraft.

17. The method as claimed in claim 16 wherein the devices are seat CPUs coupled to control passenger sets of seat peripherals through which passengers have access to the in-flight entertainment system.

18. The method as claimed in claim 17 wherein the wiring database is entered into and maintained by a system manager unit within a headend of the system.

19. The method as claimed in claim 18 wherein the topology map is automatically generated for a zone within the aircraft by a zone bridge unit coupled to one or more seat electronics units, wherein each seat electronics unit includes one or more seat CPUs.

20. The method as claimed in claim 19 wherein the method of automatically mapping is performed only when manually triggered during a maintenance cycle.

21. The method as claimed in claim 20 further comprising the step of updating the configuration database to include passenger information for each flight of the aircraft.

22. A method of automatically mapping device addresses to network protocol addresses of device locations within a bus structure, comprising the steps of:

a. maintaining a wiring database in a first control computer including wiring diagram and physical locations of devices;

b. maintaining a topology map in a second control computer representing device addresses of devices coupled to the bus structure;

c. generating a restored topology map by combining the wiring database with the topology map to map device addresses to a corresponding physical location;

d. maintaining a host database in the first control computer representing network protocol addresses of the physical locations of devices; and

e. generating a network database by combining the host database and the restored topology map, wherein the network database includes network protocol addresses and corresponding device addresses.

23. The method as claimed in claim 22 wherein the restored topology map is generated within the first control computer.

24. The method as claimed in claim 22 wherein the network database is generated within the first control computer.

25. The method as claimed in claim 22 wherein each device address is unique and when an existing device is replaced by a new device, the device address of the new device is different than the device address of the existing device.

26. The method as claimed in claim 25 wherein the bus structure is an IEEE 1394 serial bus network.

27. The method as claimed in claim 26 wherein the topology map is automatically generated during a self-ID sequence of IEEE 1394 protocol.

28. The method as claimed in claim 22 wherein the method is implemented within an in-flight entertainment system for use on board an aircraft.

29. The method as claimed in claim 28 wherein the devices are seat CPUs coupled to control passenger sets of seat peripherals through which passengers access the in-flight entertainment system.

30. The method as claimed in claim 29 wherein the wiring database is entered into and maintained by a system manager unit within a headend of the system.

31. The method as claimed in claim 30 wherein the topology map is automatically generated for a zone within the aircraft by a zone bridge unit coupled to one or more seat electronics units, wherein each seat electronics unit includes one or more seat CPUs.

32. The method as claimed in claim 31 wherein the method of automatically mapping is performed only when manually triggered during a maintenance cycle.

33. The method as claimed in claim 32 further comprising the step of updating the configuration database to include passenger information for each flight of the aircraft.

34. The method as claimed in claim 33 further comprising the step of displaying a selected one of the network database and the configuration database.

35. An in-flight entertainment system comprising:

a. one or more zones each including:

i. a plurality of passenger control units, through which a passenger accesses the in-flight entertainment system;

ii. a plurality of seat electronics units each including at least one seat CPU coupled to each other to form a bus structure, wherein each seat CPU is coupled to control a passenger set of seat peripherals, each seat CPU having a unique device address;

iii. a control unit coupled to the bus structure for controlling communications to and from the seat CPUs and for maintaining a topology map representing device addresses of each of the seat CPUs; and

b. a system manager unit within a headend of the system, coupled to the control units for comparing the topology map to a wiring database, including a wiring diagram and physical locations of the seat CPUs, and forming a configuration database including device addresses and corresponding physical locations of seat CPUs.

36. The in-flight entertainment system as claimed in claim 35 wherein the physical locations are seat positions within the aircraft.

37. The in-flight entertainment system as claimed in claim 36 wherein the configuration database is updated for each flight of the aircraft to also include passenger information.

38. The in-flight entertainment system as claimed in claim 37 wherein the bus structure is an IEEE 1394 serial bus network.

39. The in-flight entertainment system as claimed in claim 38 wherein the topology map is automatically generated during a self-ID sequence of IEEE 1394 protocol.

40. The in-flight entertainment system as claimed in claim 39 wherein the control unit is a zone bridge unit.

41. An automatically configurable in-flight entertainment system comprising:

a. a plurality of passenger seat units, each configured as a passenger entertainment system;

b. an IEEE 1394 network for coupling the passenger seat units together;

c. a system manager unit coupled to the IEEE 1394 network for generating a topological map to correlate each of the passenger seat units to a location on the IEEE 1394 network and for generating a replacement topological map after a passenger seat unit is either moved, repaired and replaced; and

d. a data input device coupled to the system manager configured to receive a collection of data regarding a passenger, one collection for each passenger, and for correlating each of the collection of data to a passenger seat unit.

42. The automatically configurable in-flight entertainment system as claimed in claim 41 wherein the system manager unit is coupled to the IEEE 1394 network through an ATM switch.

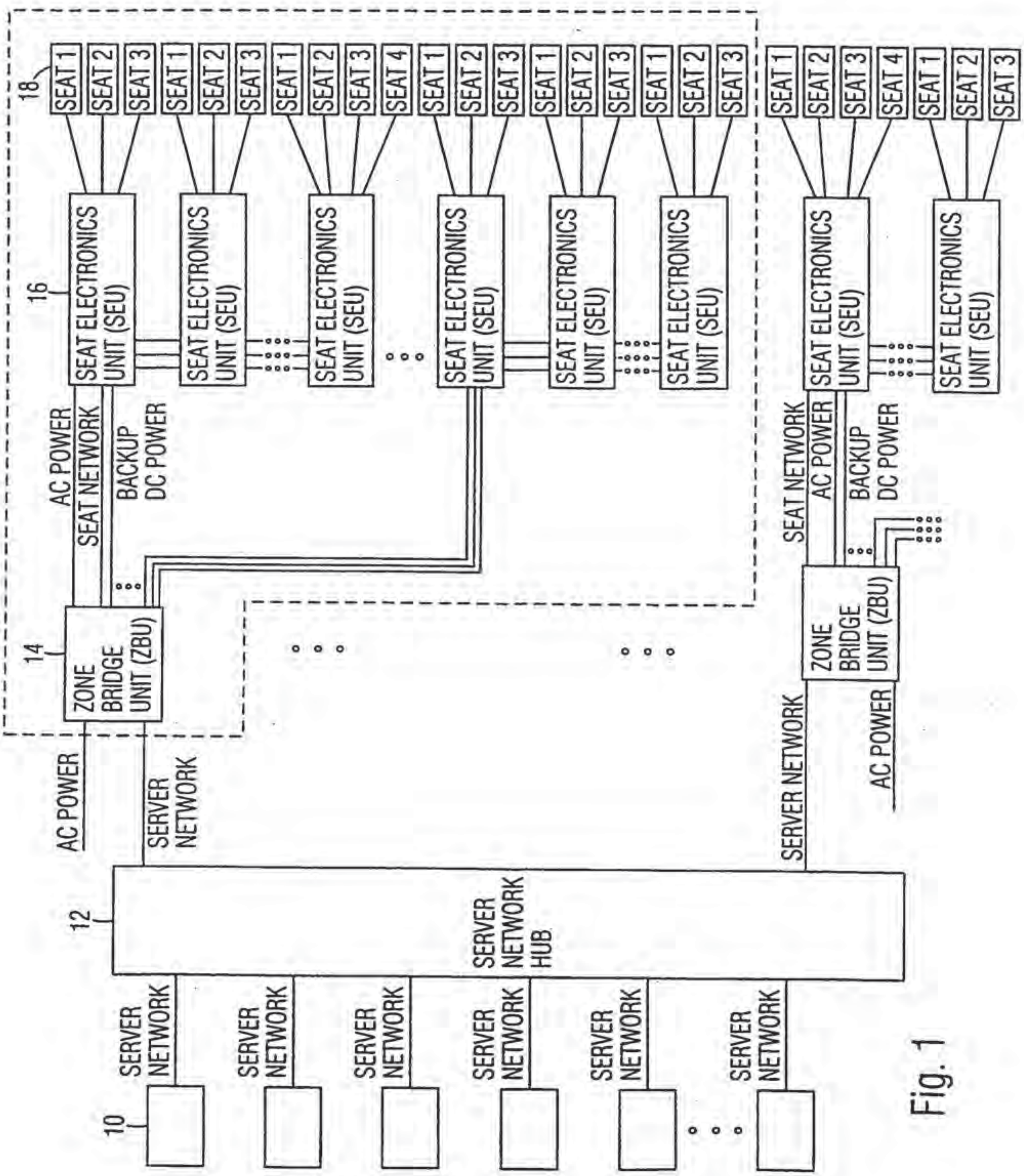

10
SERVER NETWORK
12
SERVER NETWORK HUB
AC POWER
SERVER NETWORK
14
ZONE BRIDGE UNIT (ZBU)
AC POWER
SEAT NETWORK
BACKUP DC POWER
16
SEAT ELECTRONICS UNIT (SEU)
18
SEAT 1
SEAT 2
SEAT 3
SEAT 4

Fig. 1

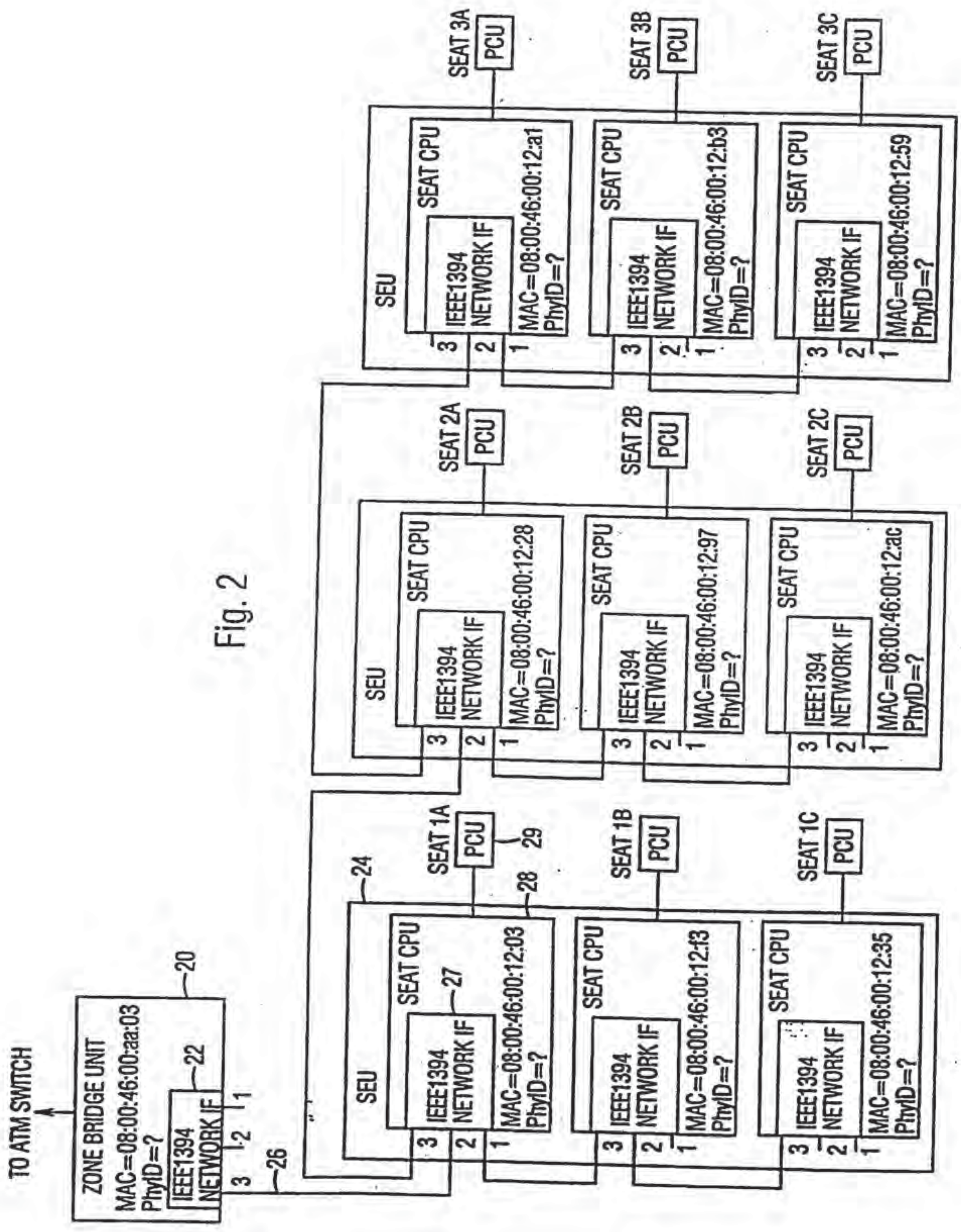


Fig. 2

FIG. 2

SUBSTITUTE SHEET (RULE 26)

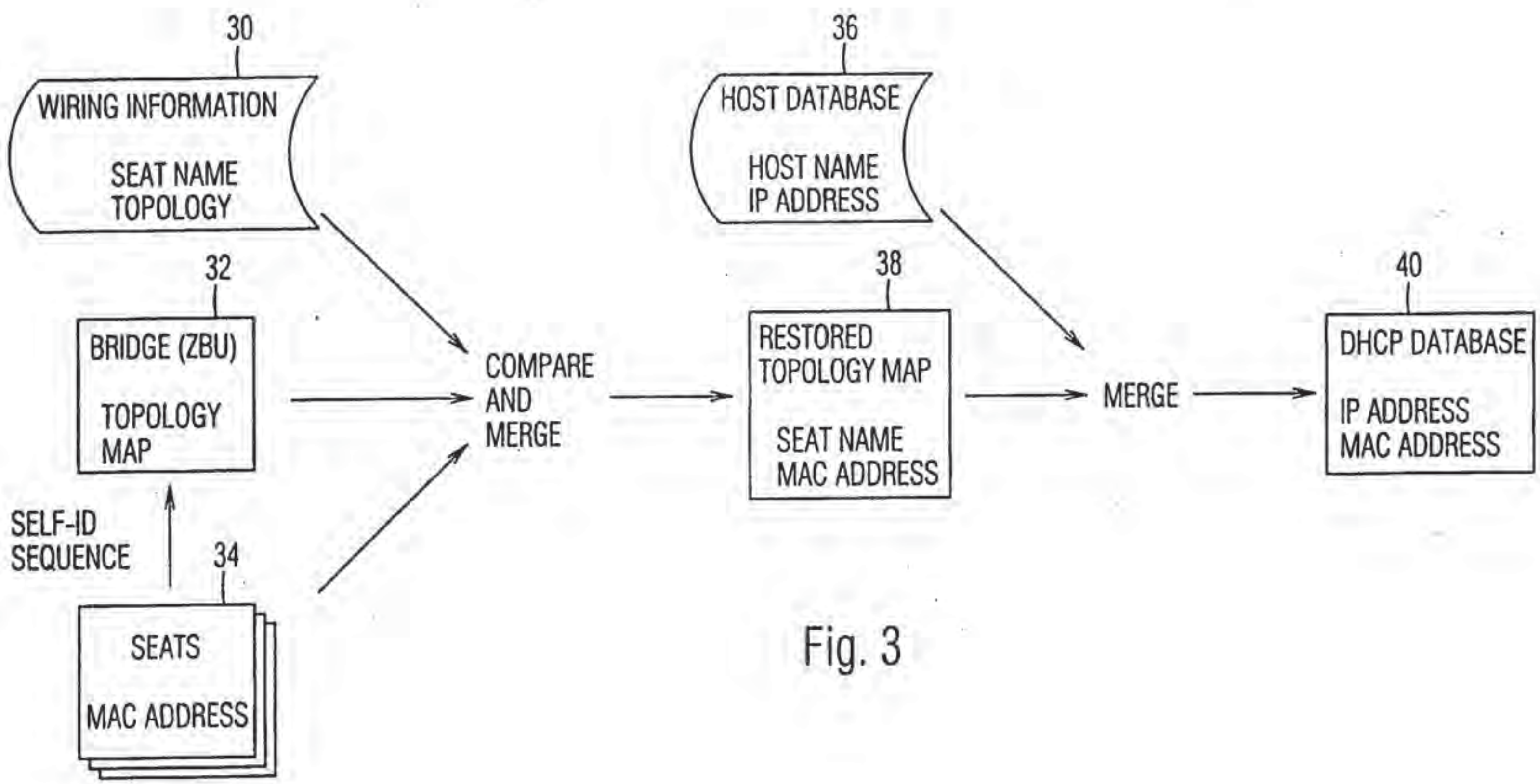

30
WIRING INFORMATION
SEAT NAME
TOPOLOGY
32
BRIDGE (ZBU)
TOPOLOGY
MAP
SELF-ID
SEQUENCE
34
SEATS
MAC ADDRESS
COMPARE
AND
MERGE
36
HOST DATABASE
HOST NAME
IP ADDRESS
38
RESTORED
TOPOLOGY MAP
SEAT NAME
MAC ADDRESS
MERGE
40
DHCP DATABASE
IP ADDRESS
MAC ADDRESS

Fig. 3

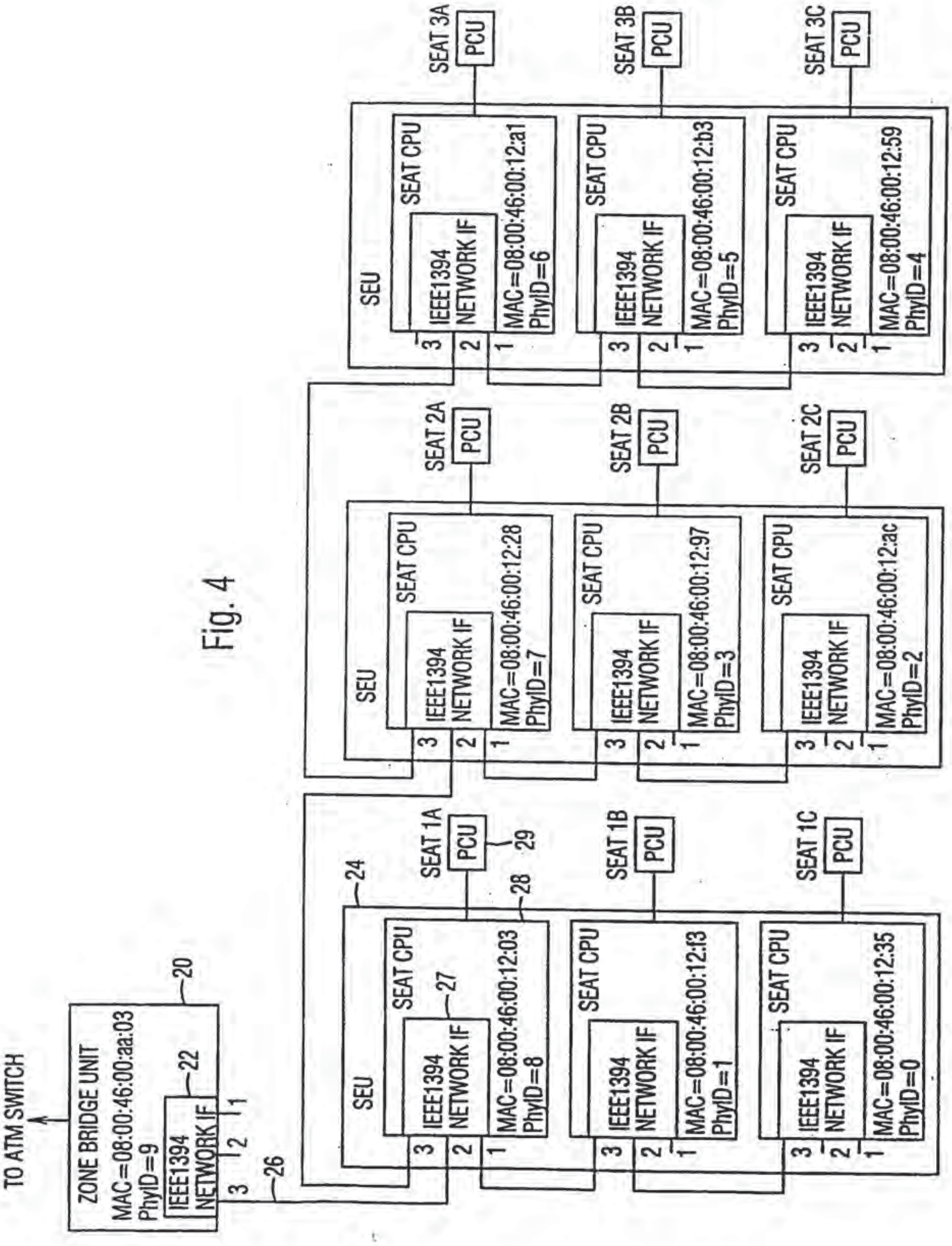


Fig. 4

FIG. 4
SUBSTITUTE SHEET (RULE 26)

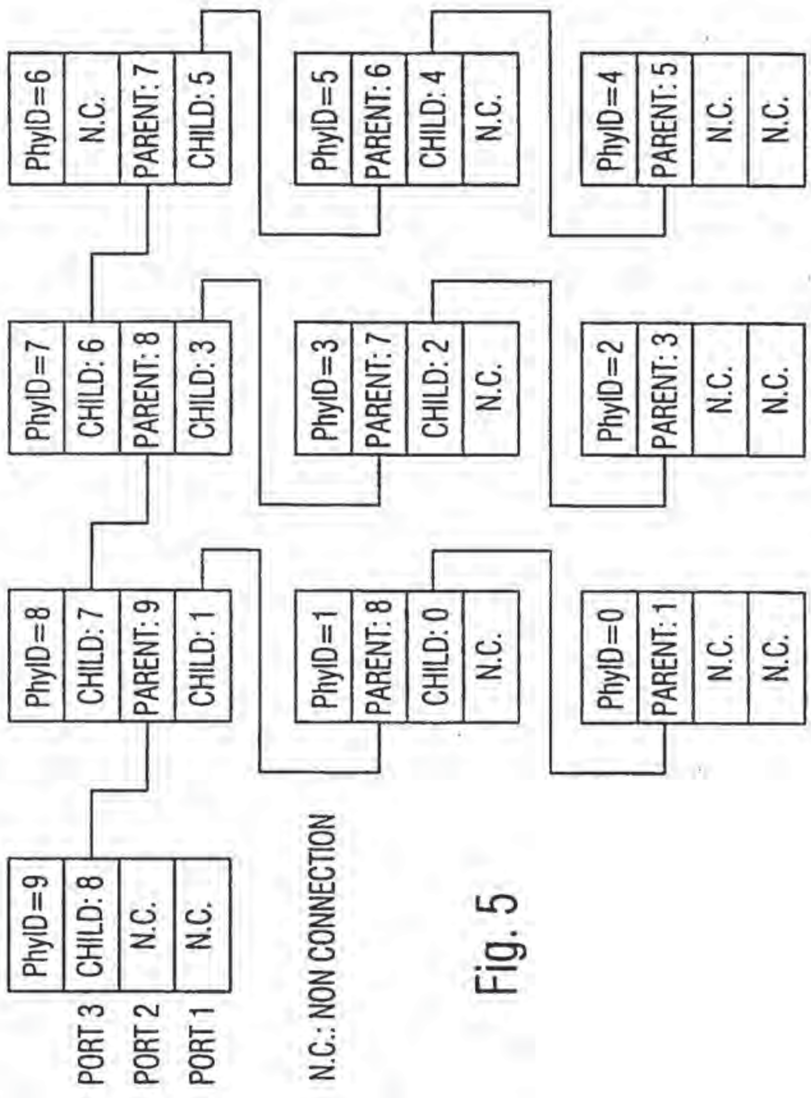

PhyID=9
CHILD: 8
N.C.
N.C.
PORT 3
PORT 2
PORT 1
PhyID=8
CHILD: 7
PARENT: 9
CHILD: 1
PhyID=7
CHILD: 6
PARENT: 8
CHILD: 3
PhyID=6
N.C.
PARENT: 7
CHILD: 5
PhyID=1
PARENT: 8
CHILD: 0
N.C.
PhyID=3
PARENT: 7
CHILD: 2
N.C.
PhyID=5
PARENT: 6
CHILD: 4
N.C.
PhyID=0
PARENT: 1
N.C.
N.C.
PhyID=2
PARENT: 3
N.C.
N.C.
PhyID=4
PARENT: 5
N.C.
N.C.
N.C.: NON CONNECTION

Fig. 5

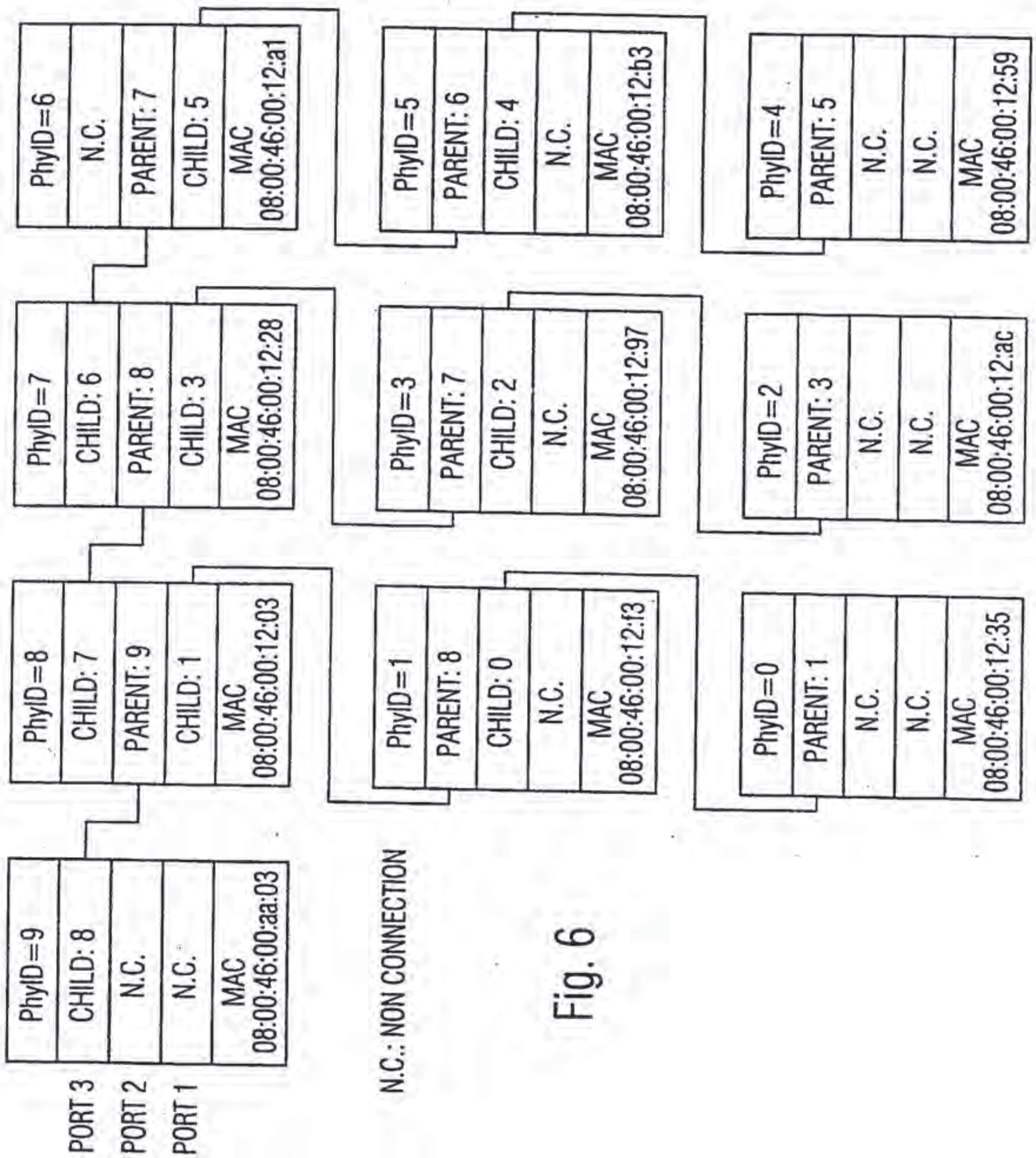

PORT 3
PORT 2
PORT 1
PhyID=9
CHILD: 8
N.C.
N.C.
MAC 08:00:46:00:aa:03
PhyID=8
CHILD: 7
PARENT: 9
CHILD: 1
MAC 08:00:46:00:12:03
PhyID=7
CHILD: 6
PARENT: 8
CHILD: 3
MAC 08:00:46:00:12:28
PhyID=6
N.C.
PARENT: 7
CHILD: 5
MAC 08:00:46:00:12:a1
PhyID=1
PARENT: 8
CHILD: 0
N.C.
MAC 08:00:46:00:12:f3
PhyID=3
PARENT: 7
CHILD: 2
N.C.
MAC 08:00:46:00:12:97
PhyID=5
PARENT: 6
CHILD: 4
N.C.
MAC 08:00:46:00:12:b3
PhyID=0
PARENT: 1
N.C.
N.C.
MAC 08:00:46:00:12:35
PhyID=2
PARENT: 3
N.C.
N.C.
MAC 08:00:46:00:12:ac
PhyID=4
PARENT: 5
N.C.
N.C.
MAC 08:00:46:00:12:59
N.C.: NON CONNECTION

Fig. 6

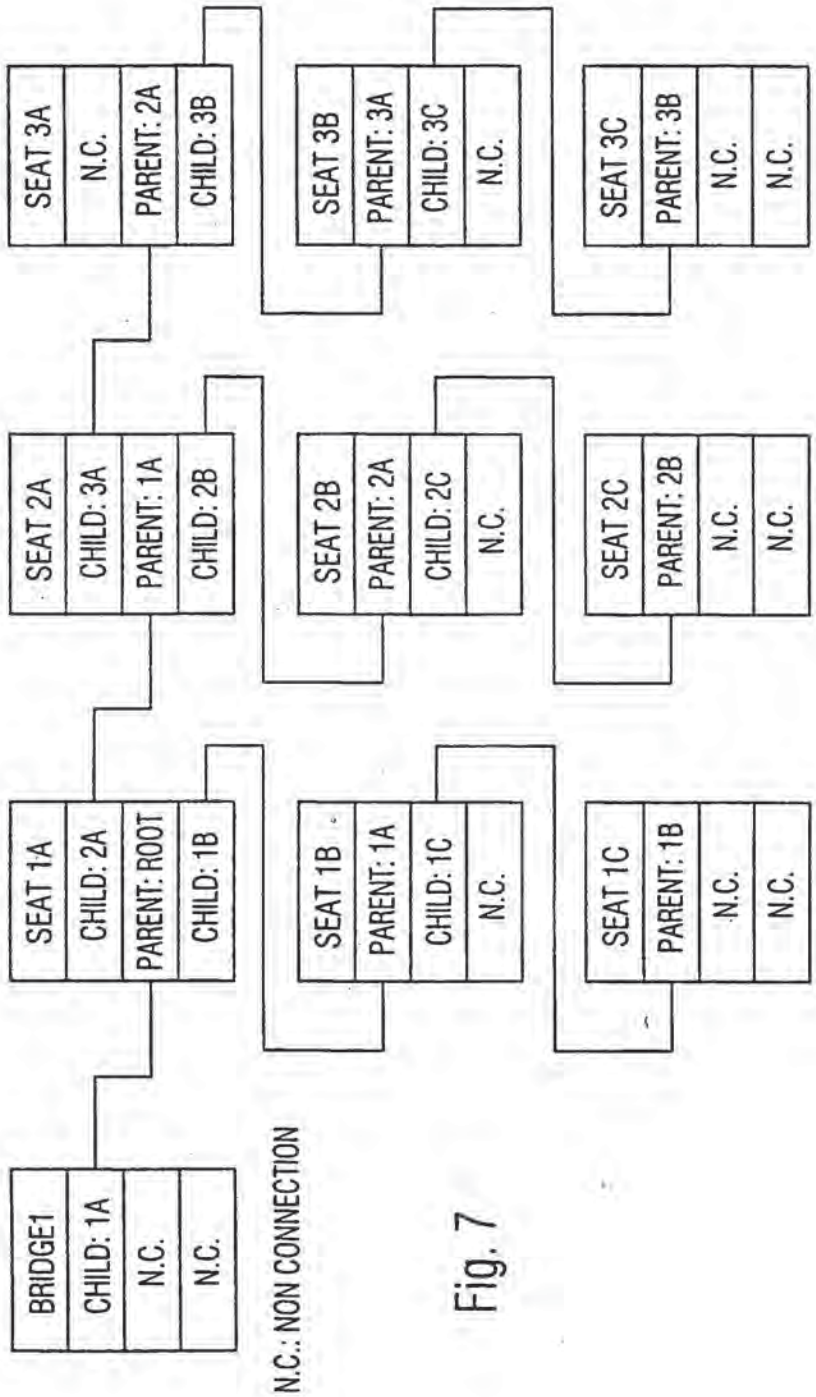

SEAT 3A
N.C.
PARENT: 2A
CHILD: 3B
SEAT 3B
PARENT: 3A
CHILD: 3C
N.C.
SEAT 3C
PARENT: 3B
N.C.
N.C.
SEAT 2A
CHILD: 3A
PARENT: 1A
CHILD: 2B
SEAT 2B
PARENT: 2A
CHILD: 2C
N.C.
SEAT 2C
PARENT: 2B
N.C.
N.C.
SEAT 1A
CHILD: 2A
PARENT: ROOT
CHILD: 1B
SEAT 1B
PARENT: 1A
CHILD: 1C
N.C.
SEAT 1C
PARENT: 1B
N.C.
N.C.
BRIDGE1
CHILD: 1A
N.C.
N.C.
N.C.: NON CONNECTION

Fig. 7

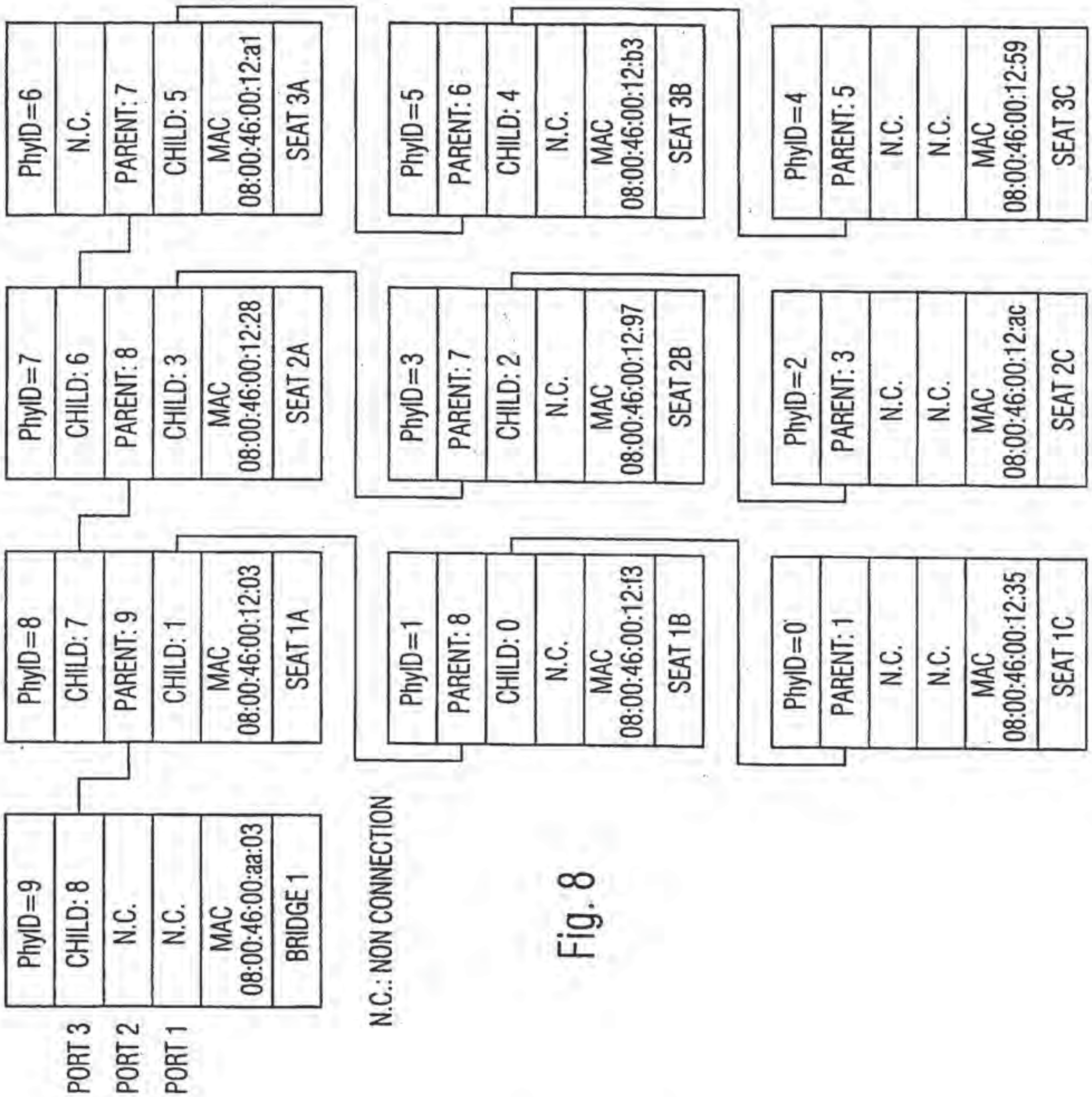

PORT 3
PORT 2
PORT 1
PhyID=9 | CHILD: 8 | N.C. | N.C. | MAC 08:00:46:00:aa:03 | BRIDGE 1
PhyID=8 | CHILD: 7 | PARENT: 9 | CHILD: 1 | MAC 08:00:46:00:12:03 | SEAT 1A
PhyID=7 | CHILD: 6 | PARENT: 8 | CHILD: 3 | MAC 08:00:46:00:12:28 | SEAT 2A
PhyID=6 | N.C. | PARENT: 7 | CHILD: 5 | MAC 08:00:46:00:12:a1 | SEAT 3A
PhyID=1 | PARENT: 8 | CHILD: 0 | N.C. | MAC 08:00:46:00:12:f3 | SEAT 1B
PhyID=3 | PARENT: 7 | CHILD: 2 | N.C. | MAC 08:00:46:00:12:97 | SEAT 2B
PhyID=5 | PARENT: 6 | CHILD: 4 | N.C. | MAC 08:00:46:00:12:b3 | SEAT 3B
PhyID=0 | PARENT: 1 | N.C. | N.C. | MAC 08:00:46:00:12:35 | SEAT 1C
PhyID=2 | PARENT: 3 | N.C. | N.C. | MAC 08:00:46:00:12:ac | SEAT 2C
PhyID=4 | PARENT: 5 | N.C. | N.C. | MAC 08:00:46:00:12:59 | SEAT 3C
N.C.: NON CONNECTION

Fig. 8

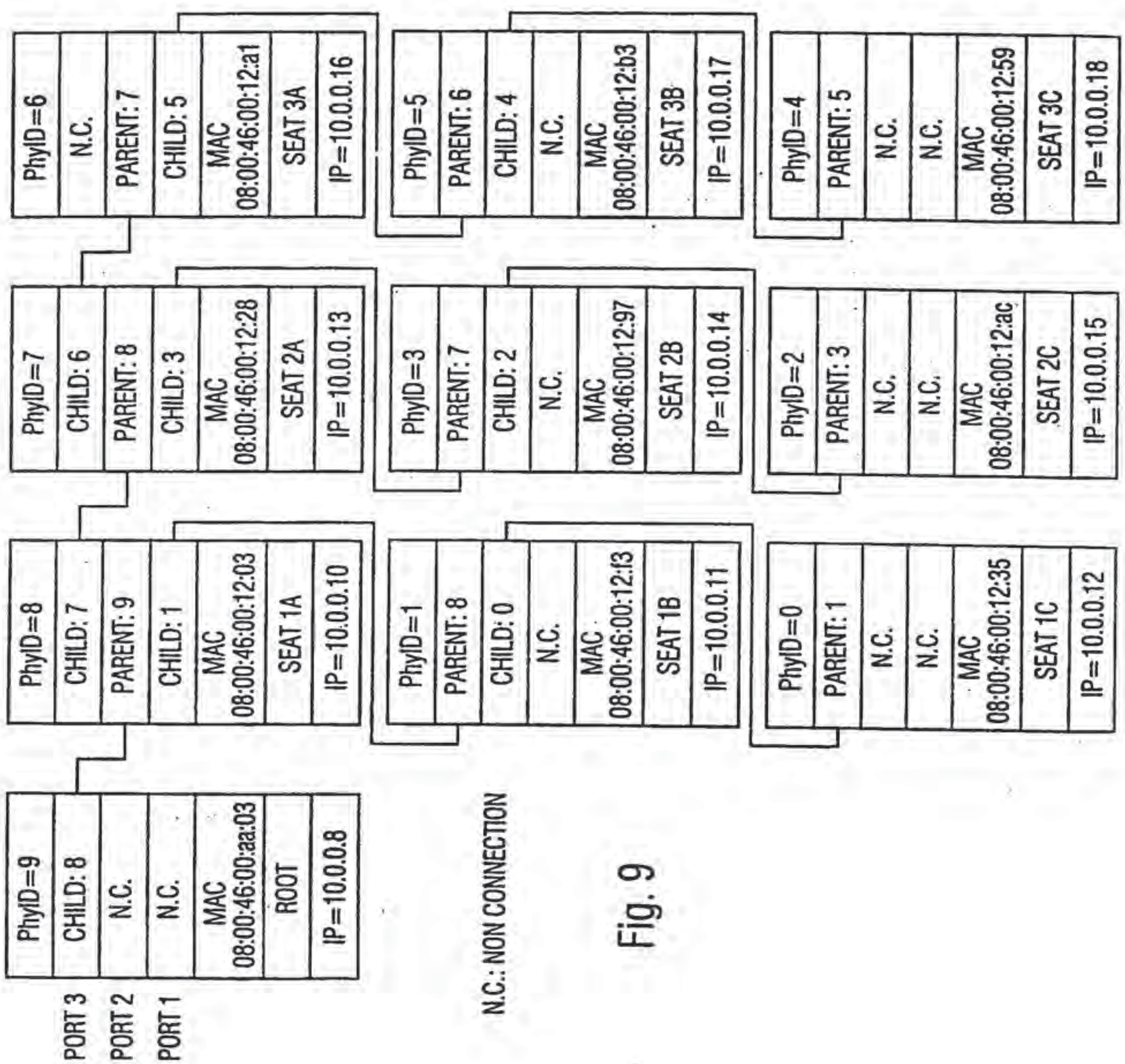

PORT 3
PORT 2
PORT 1
PhyID=9
CHILD: 8
N.C.
N.C.
MAC 08:00:46:00:aa:03
ROOT
IP=10.0.0.8
PhyID=8
CHILD: 7
PARENT: 9
CHILD: 1
MAC 08:00:46:00:12:03
SEAT 1A
IP=10.0.0.10
PhyID=7
CHILD: 6
PARENT: 8
CHILD: 3
MAC 08:00:46:00:12:28
SEAT 2A
IP=10.0.0.13
PhyID=6
N.C.
PARENT: 7
CHILD: 5
MAC 08:00:46:00:12:a1
SEAT 3A
IP=10.0.0.16
PhyID=1
PARENT: 8
CHILD: 0
N.C.
MAC 08:00:46:00:12:f3
SEAT 1B
IP=10.0.0.11
PhyID=3
PARENT: 7
CHILD: 2
N.C.
MAC 08:00:46:00:12:97
SEAT 2B
IP=10.0.0.14
PhyID=5
PARENT: 6
CHILD: 4
N.C.
MAC 08:00:46:00:12:b3
SEAT 3B
IP=10.0.0.17
PhyID=0
PARENT: 1
N.C.
N.C.
MAC 08:00:46:00:12:35
SEAT 1C
IP=10.0.0.12
PhyID=2
PARENT: 3
N.C.
N.C.
MAC 08:00:46:00:12:ac
SEAT 2C
IP=10.0.0.15
PhyID=4
PARENT: 5
N.C.
N.C.
MAC 08:00:46:00:12:59
SEAT 3C
IP=10.0.0.18
N.C.: NON CONNECTION

Fig. 9

FIG. 9

THIS PAGE BLANK (USPTO)

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
G06F 13/00

A3

(11) International Publication Number: WO 98/44400

(43) International Publication Date: 8 October 1998 (08.10.98)

(21) International Application Number: PCT/US98/05881

(22) International Filing Date: 26 March 1998 (26.03.98)

(30) Priority Data:
08/831,063 1 April 1997 (01.04.97) US

(71) Applicant *(for all designated States except US)*: SONY TRANS COM INC. [US/US]; 1833 Alton Avenue, Irvine, CA 92606 (US).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: FUKUI, Toshiharu [JP/US]; 266 Baywood Drive, Newport Beach, CA 92660 (US).

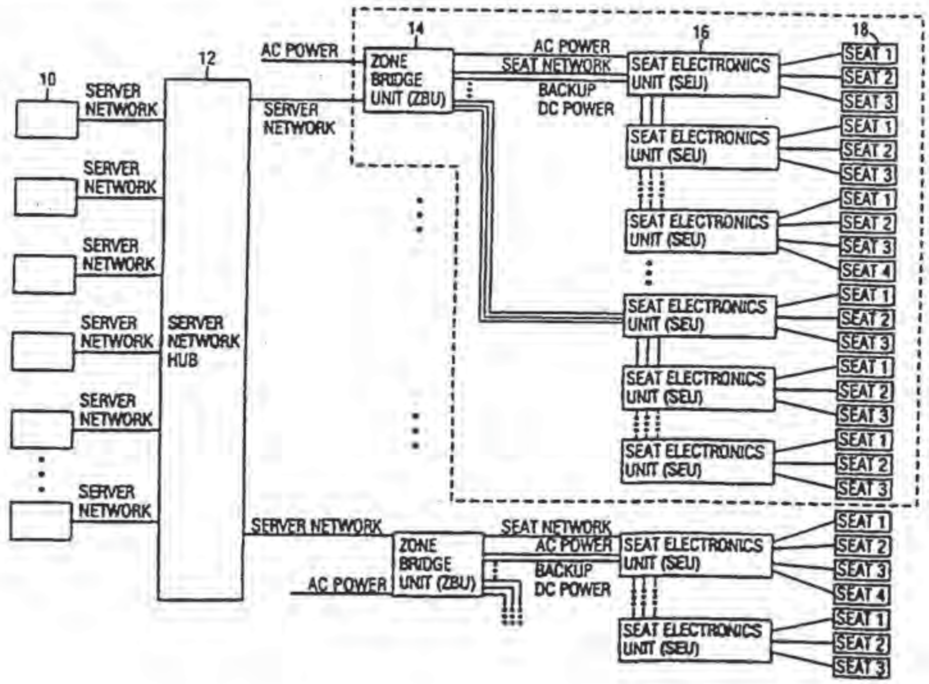
(74) Agents: HAVERSTOCK, Thomas, B. et al.; Haverstock & Owens LLP, Suite 420, 260 Sheridan Avenue, Palo Alto, CA 94306 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(88) Date of publication of the international search report: 30 December 1998 (30.12.98)

(54) Title: AUTOMATIC CONFIGURATION SYSTEM FOR MAPPING NODE ADDRESSES WITHIN A BUS STRUCTURE TO THEIR PHYSCAL LOCATION



(57) Abstract

An automatic configuration system maps a device address of each node (18) coupled to a bus structure (26) to a network protocol address corresponding to the physical location of the node.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US98/05881

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :GO6F 13/00
US CL :370/254; 395/200.54
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 370/254, 257; 395/200.54

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A, P	US 5737319 A (CROSLIN ET AL.) 7 APRIL 1998	1-42
A, E	US 5734824 A (CHOI ET AL.) 31 MARCH 1998.	1-42
A, E	US 5729685 A (CHATWANI ET AL.).	1-42
A, E	US 5727157 A (ORR ET AL.).	1-42
A, E	US 5706440 A (COMPLIMENT ET AL.).	1-42
A, E	US 5684959 A (BHAT ET AL.).	1-42
A, E	US 5664107 A (CHATWANI ET AL.).	1-42

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

25 JUNE 1998

Date of mailing of the international search report

25 SEP 1998

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

LANCE LEONARD BARRY, ESQ

Telephone No. (703) 305-9600

Form PCT/ISA/210 (second sheet)(July 1992)*

This Page Blank (uspto)

- EPODOC / EPO

PN - JP2000076209 A 20000314
PD - 2000-03-14
PR - SG19980002903 19980806
OPD - 1998-08-06
TI - NETWORK TOPOLOGY BASED ON THREE DEGREE NODE AND ROUTING ALGORITHM
IN - PEKU YU TAN
PA - MATSUSHITA ELECTRIC IND CO LTD
IC - G06F15/173 ; H04L12/56

- WPI / DERWENT

TI - Network topology designing method in computer network, involves interconnecting adjacent 3 link nodes to form network topology of specific structure, using mathematical standard based on digital display of nodes
PR - SG19980002903 19980806
PN - SG75867 A1 20001024 DW200060 H04L12/46 000pp
\- JP2000076209 A 20000314 DW200024 G06F15/173 010pp
PA - (MATU) MATSUSHITA DENKI SANGYO KK
\- (MATU) MATSUSHITA ELECTRIC IND CO LTD
IC - G06F15/173 ;H04L12/46 ;H04L12/56 ;H04L12/64
IN - TAN P Y
AB - JP2000076209 NOVELTY - A node in network, is converted to a 2D network topology consisting of 3 link nodes. Address of 3 link node is generated based on a 2 table radix n scheme, in order to secure intrinsic address of node in a specific network. Then, the adjacent 3 link nodes are interconnected, to form network topology of specific structure, using mathematical standard based on digital display of 3 link nodes.
\- DETAILED DESCRIPTION - An INDEPENDENT CLAIM is also included for routing algorithm.
\- USE - For designing network topology in computer network.
\- ADVANTAGE - Distance of the network topology which includes a node 3 degree is improved and inter-arrival time of data between the route designated capability of a node and source node and address node is raised.
\- DESCRIPTION OF DRAWING(S) - The figure shows the connection of double link between adjacent nodes.
\- (Dwg.1/10)
OPD - 1998-08-06
AN - 2000-278065 [24]

- PAJ / JPO

PN - JP2000076209 A 20000314
PD - 2000-03-14
AP - JP19990222352 19990805
IN - PEKU YU TAN
PA - MATSUSHITA ELECTRIC IND CO LTD
TI - NETWORK TOPOLOGY BASED ON THREE DEGREE NODE AND ROUTING ALGORITHM
AB - PROBLEM TO BE SOLVED: To reduce the distance of a network topology including three degree nodes by reducing the complexity of a routing algorithm for the network topology.
\- SOLUTION: The configuration of a network topology can be divided into two main steps, that is, a naming step and a connecting step. In the naming step, a node in an unconstructed network is mapped into any of three link nodes which are erected or reversed and oriented. An address is generated by using a naming scheme based on a 2-tuple cardinal number (n) scheme to insure the inherent address of each of the three link nodes in a specified network by using a numerical equation. The respective three link nodes are interconnected with three adjacent link

This Page Blank (uspto)

nodes to form a network topology of a specific structure by performing connecting through point-to-point connection using the mathematical reference of the three respective link nodes.
I - G06F15/173 ;H04L12/56

This Page Blank (uspto)

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報**（Ａ）

(11)特許出願公開番号
特開2000－76209
（P2000－76209A）
(43)公開日　平成12年3月14日(2000.3.14)

(51)Int.Cl.⁷	識別記号	ＦＩ	テーマコード゛（参考）
Ｇ０６Ｆ　15/173		Ｇ０６Ｆ　15/173	Ｘ
Ｈ０４Ｌ　12/56		Ｈ０４Ｌ　11/20	１０２Ｄ

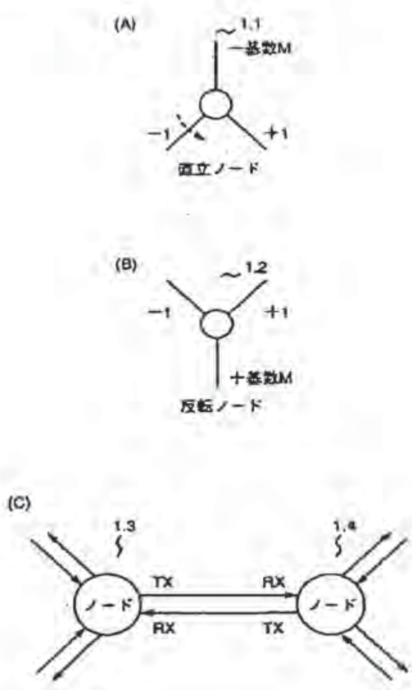
審査請求　未請求　請求項の数７　ＯＬ　（全 10 頁）

(21)出願番号　特願平11－222352
(22)出願日　平成11年8月5日(1999.8.5)
(31)優先権主張番号　９８０２９０３－６
(32)優先日　平成10年8月6日(1998.8.6)
(33)優先権主張国　シンガポール（ＳＧ）

(71)出願人　000005821
松下電器産業株式会社
大阪府門真市大字門真1006番地
(72)発明者　ペク　ユー・タン
シンガポール534415シンガポール、タイ・セン・アベニュー、ブロック1022、04－3530番、タイ・セン・インダストリアル・エステイト、パナソニック・シンガポール研究所株式会社内
(74)代理人　100062144
弁理士　青山　葆　（外1名）

(54)【発明の名称】　3度ノードを基礎とするネットワークトポロジー及びルーティングアルゴリズム

(57)【要約】
【課題】　3度ノードを含むネットワークトポロジーの距離を改善する。
【解決手段】　既存の網状ネットワークにおけるノードを3リンクノードから成る二次元ネットワークトポロジーに変換し、2タプル基数ｎスキームに基づくネーミングスキームを使用して各3リンクノードについて固有のアドレスを生成し、3リンクノードの数値表示に基づく数学的基準を使用して各3リンクノードと隣接する3リンクノードとを相互接続し、これによって新たなネットワークトポロジーを形成する。



【特許請求の範囲】

【請求項1】　帯域幅を増加させ、網状に接続されたネットワークにおけるネットワークノード間の距離を減少させる方法であって、

a．ネットワークサブシステムの代表である網状に接続されたネットワークにおけるノードを3リンクノードから成る二次元的ネットワークトポロジーに変換するステップと、

b．数式を使用して特定ネットワークにおける各3リンクノードの固有アドレスを確保するため、2タプル基数nスキームに基づくネーミングスキームを使用して3リンクノードのアドレスを生成するステップと、

c．各3リンクノードの数字表示に基づいて数学的基準を使用する二地点間接続を介して連結を行うことによって、各3リンクノードを隣接する3リンクノードと相互接続し、特定構造のネットワークトポロジーを形成するステップとからなる方法。

【請求項2】　ネットワークにおける各3リンクノードのアドレスの生成方法は、2タプル基数n数であるX（MST，LST）を基礎とし、MSTは最も有意なタプルであり，LSTは最も有意でないタプルであり，特定ネットワークにおける各3リンクノードに対して2タプル数を生成する数式は特定ネットワークの列数nと行数mを基礎とする請求項1記載の方法。

【請求項3】　数理的基準は、直立及び反転ノードの3リンクの数値表示に基づいてネットワーク内の任意の異なる2ノード間の接続を決定し、2つのノード間のリンクの正しい接続は、直立及び反転ノード双方の3リンクの各々の数学的基準に基づく和をゼロにする請求項1記載の方法。

【請求項4】　網状に接続された3リンクノードで組織されるネットワークにおいて、データメッセージをソースノードから宛先ノードまで経路指定するための最短経路を発見する方法であって、ネットワークの各3リンクノードに総称的ルーティングアルゴリズムを導入するステップを含み、前記総称的ルーティングアルゴリズムはソースノードから横方向ノードを介して宛先ノードに至る経路が最短となるようにデータメッセージを現行ノードから隣接ノードへと出力するためのリンクを決定し、データメッセージが送信される各ノードにおいて総称的ルーティングアルゴリズムを実行するステップと、総称的ルーティングアルゴリズムによって決定されたリンクを介してデータメッセージを隣接ノードに出力するステップと，データメッセージが宛先ノードに到着するまで、前記実行ステップと前記出力ステップとを繰り返すステップとを含み、これにより、ソースノードから宛先ノードまでの最短経路を介するデータメッセージの送信を有効化する方法。

【請求項5】　総称的ルーティングアルゴリズムは、

a．現行ノード及び宛先ノード間の行距離及び列距離を2タプル基数nで表示されたこれらのノードのアドレスに基づいて計算するステップと、

b．経路指定されるデータメッセージの入力ポイントを決定するステップと、

c．ノードのタイプと現行ノード及び宛先ノード間の計算された最短距離とに基づいてデータメッセージの出力を決定するステップとを含む請求項4記載の方法。

【請求項6】　3リンクノードを使用してネットワークトポロジーを構成する方法であって，

a．ネットワーク内の合計ノード数に基づいて、構成されるネットワークの行数及び列数を決定するステップと、

b．ネットワークの全てのノードについてアドレスを生成するステップと、

c．数学的基準を使用してネットワーク内のノードの3リンクを相互接続するステップとを含み、前記数学的基準は直立及び反転ノードの3つのリンクの数値表示に基づいてネットワークにおける任意の異なる2ノード間の接続を決定し、2ノード間のリンクの正しい接続は、直立及び反転ノードの3リンクの各々の数学的基準に基づく和をゼロにする方法。

【請求項7】　ネットワーク構成の最小拡張増分を減少させるために、既存のネットワークに2行または2列のノードを追加して既存のネットワークを新たなネットワークに再形成し、これにより任意の2ノード間の距離の縮小及びネットワークのスケーラビィリィティの改善を可能にするネットワークの位相的方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ノードの経路指定能力及びソースノードと宛先ノードとの間のデータの到着間時間を向上させるために、トポロジーにおけるネットワークノードを配置する方法に関する。本発明による方法は、ネットワークプロトコルとは独立したものであり，異なるコンピュータネットワークにおけるコンピュータ間の遅延を改善するように既存のネットワークシステムを構築し相互接続するために使用することができる。

【0002】

【従来の技術】高速通信に対するニーズは従来、新たなネットワークトポロジーの設計に際して、及びバックボーンネットワーク上での効率的なデータ経路指定のためのルーティングアルゴリズムに依存するネットワークトポロジーにとって無視できないものである。

【0003】バックボーンネットワークの場合、ネットワークゲートウェイを効率的な方法で相互接続することの必要性は，端末間遅延に関わる良好なネットワーク性能にとって必須である。相互接続されたネットワークにおけるゲートウェイ数が増大するにつれて、ネットワークゲートウェイを相互接続するために選択されるネット

ワークトポロジーもまた、異なるゲートウェイのコンピュータ端末間のリアルタイムサービスを提供する上での決定因子となり得る。

【0004】より多くのネットワークノード数の相互接続に使用されるネットワークトポロジーもまた、スイッチング技術及びトポロジーに使用される方法を決定する。スイッチング技術の使用を介したネットワークトポロジーは、低コスト、高信頼性でスケーラブルな連結性及びサービスを配信する能力がある。本発明はまた、ファイバーチャンネルスイッチング技術に対応できるユーザによって定義されるネットワークトポロジーの1つとしても使用可能である。規格のファイバーチャンネル族は、標準化委員会T11の責任であり、T11は、「情報技術規格全国委員会（NCITS）」に属する一委員会である。

【0005】既に存在し、国際規格の一部として実行されているネットワークトポロジーには、各々リング及びバストポロジーを基礎とするネットワークであるトークンリング及びイーサネット等、多数がある。相互接続ネットワークトポロジーは、静的タイプと動的タイプの2つの主要クラスに大別することができる。静的ネットワークトポロジーの大部分は、各々が2つ以上のリンクを有するノードを必要とする。

【0006】ネットワークトポロジーの性能を測る際には、ネットワークの直径が極めて重要である。ネットワークトポロジーの直径は、ネットワークにおける全ての可能ノード間の最短経路を表す1つの尺度である。ネットワークの直径は、ソースノードから宛先までメッセージを経路指定するためのリンク数を決定する。ネットワークトポロジーの直径は最短の可能経路を決定し、ルーティングアルゴリズムは、ソースノードから宛先までメッセージを経路指定することができる。その他、重要な2つのパラメータには、ネットワークノードの度数及びネットワークに対するノードの最小拡張増分が含まれる。ネットワークノードの度数は、ネットワークノードが保有しなければならないリンク数の一尺度である。最小拡張増分は、ネットワークに加えられる最小ノード数である。

【0007】3度ノードを使用する現時点で周知のネットワークトポロジーは、立方接続されたサイクルネットワークである。これは、ネットワークサイズが大きくなるにつれて大きい最小拡張増分を保有する。ネットワーク内のノード数が増加すると、ネットワークの直径はかなり大きくなる。このネットワークトポロジーにおける最も有害な要素の1つは、これが複雑なルーティングアルゴリズムを必要としていることにあり、当該アルゴリズムもまたネットワークトポロジーにおけるノード数に依存している。各ネットワークノードに於いて総称的アルゴリズムを使用することは不可能になるため、ネットワークトポロジーは、ネットワークサイズの増大に伴って変更を余儀なくされるアルゴリズムを使用する。

【0008】既存のルーティングアルゴリズム及びネットワークトポロジーの多くは、ネットワークプロトコルのタイプに依存している。これらの制限事項は、コスト及び技術的困難さの両面で既に禁制であるとされている遺産ネットワークシステムに関する新たなルーティングアルゴリズム及びネットワークトポロジーの実行を必要としている。ルーティングアルゴリズムの中には、ソースと宛先との間のノードへと至る横断経路に関する情報を必要とするものがある。追加的な経路指定情報は、既に乏しい帯域幅資源にとっての負担となる。

【0009】

【発明の趣旨】本発明の1つの目的は、3度ノードを含むネットワークトポロジーの距離を改善することにある。本発明の他の目的は、網目状のネットワークトポロジーのためのノードの最小拡張増分を改善することにある。

【0010】本発明のさらなる目的は、ネットワークトポロジーのためのルーティングアルゴリズムの複雑さを低減することにより、特定サイズのネットワークトポロジーにおける3度ノードの各々について低コストルータの実行を有効化することにある。

【0011】本発明のさらに他の目的は、バックボーンネットワークアーキテクチャとして機能する能力のある高度にスケーラブルなネットワークトポロジーを提供して、ネットワークにおける任意の2ノード間の高速データ配信を可能にすることにある。

【0012】本発明のさらなる目的の1つは、ネットワークトポロジー及びルーティングアルゴリズムの何れにも依存しない2地点間で接続されたネットワークのためのネットワークプロトコルを提供することにある。本発明のさらに他の目的の1つは、ソースノードから中間の横断ノードに至る経路指定情報の伝送をなくして経路指定管理における帯域幅ユーティリティを改善することにある。

【0013】これらの目的を達成するため、本発明によれば、帯域幅を増大し、網目接続されたネットワークにおけるネットワークノード間の距離を低減する、以下のステップを含む方法が提供されている。即ち、

a） ルータ、ブリッジ、ネットワーク交換用サブシステム等のネットワークサブシステムの代表である網目接続されたネットワークにおけるノードを、3リンクノードで構成される二次元ネットワークトポロジーに転換するステップと、

b） 数式を使用して特定ネットワークにおける3リンクノードの各々の固有アドレスを確保するために、2タプル基数−nスキームに基づくネーミングスキームを使用して3リンクノードのためのアドレスを生成するステップと、

c） 各3リンクノードの数値表示に基づいて、数学的基

準を使用して各3リンクノードと2地点間接続を介して連結を行う隣接する3リンクノードとを相互接続させ、特定形状のネットワークトポロジーを形成するステップとからなる方法である。

【0014】また、本発明にによれば、以下の手段を含むシステムもまた提供されている。即ち、ネットワークサブシステムを構成し、当該ネットワークサブシステムを3リンクノードに基づくネットワークトポロジーに相互接続するための手段と、反転及び直立して配位された3リンクノードを使用して各ネットワークのサブシステムを構成するための手段と、3リンクノードの各々の固有アドレスまたは名称を生成するための手段と、ネットワークのサブシステムを構成してネットワークトポロジーに相互接続するための手段と、ルーティングアルゴリズムを使用してネットワークトポロジーにおけるソースノードと宛先ノードとの間でデータメッセージを経路指定し、ソースノードから宛先ノードまでの最短経路を選択するための手段とで構成している。

【0015】相互接続されるブリッジ、ルータ及びゲートウェイ等のネットワークサブシステムは、3リンクのネットワークノードによって表示される。これらのサブシステムの各々は、3つの二重リンクによって隣接するサブシステムとの相互接続を実行され、これにより、3リンクノードがネットワークサブシステムの代表にされる。これらのノードは各々、直立または反転されて配向される。3リンクノード間の相互接続は、数式に基づいて実行される。相互に接続された各ノードには、2タプル基数－n数値システムに基づいて固有のアドレスまたはネーミング規則が与えられる。ノードのネーミングに採用されるアドレス指定スキームは、ネットワーク構成における任意の2ノード間の最短経路を見い出すためのルーティングアルゴリズムによって使用される。

【0016】

【発明の実施の形態】本発明は、例示的実施形態に関する以下の詳細な説明を参照し、添付の図面に関連してこれを読み取れば最も良く理解されるであろう。図1の(a)及び(b)には各々、直立した3リンクノード1．1及び反転された3リンクノード1．2が示されている。3リンクノードの各リンクには、図のように固有の数字による表示(数値)が割り当てられている。ノードの各リンクへの数字表示の割当ては、直立及び反転されたノードの各リンクに割り当てられた数字表示の合計がゼロであれば、直立ノード1．1及び反転ノード1．2間の正しい接続が達成されるように決定される。

【0017】図1の(c)には、2つのノード1．3及び1．4間の完全二重リンクの接続が示されている。

【0018】各ノードのアドレス指定またはネーミング規則は図2に示すように以下の数式によって説明される。

［1］　$X(MST_M, LST_N)=X((y/n)\%m, y\%n)$

ここで、$y=0, 1, 2, 3, \ldots (m\times n)-1$

MST及びLSTは、有効タプルの最大、最小値を表し、%は作動率、mはネットワークにおける行数、nはネットワークにおける列数、yは＋veの整数、(y／n)は整数叙法を表している。

【0019】互いに隣接するノードの2つのリンクの接続は、以下のような基準に従って達成される。

［2］　直立配向された3リンクノード

X(x2, x1)．＋veリンクは、X(x2, [(x1+1)%n]．－veリンクに接続される。X(x2, x1)．－veリンクは、X(x2, [(x1－1)%n]．＋veリンクに接続される。X(x2, x1)．－ve基数Mリンクは、X([(x2－1)%m], x1)．－ve基数Mリンクに接続される。

【0020】［3］　反転配向された3リンクノード

X(x2, x1)．＋veリンクは、X(x2, [(x1+1)%n]．－veリンクに接続される。X(x2, x1)．－veリンクは、X(x2, [(x1－1)%n]．＋veリンクに接続される。X(x2, x1)．＋ve基数Mリンクは、X([(x2+1)%m], x1)．－ve基数Mリンクに接続される。

【0021】図3から図5を参照して、3リンクノードを基礎とするネットワークトポロジーの構成は、図3から図5のフローチャートが示すアルゴリズムに基づいて実行することができる。

【0022】ネットワークトポロジーの構成は、2つの主要ステップ、即ちネーミングステップと接続ステップとに分割することができる。ネーミングステップでは、構築されていないネットワークにおけるノードが、直立してまたは反転されて配向された3リンクノードの何れかにマップされる。マップされたネットワークトポロジーにおける3リンクノードは、全て名称を付される。このネーミング工程は、ステップ2．1、2．2、2．3、2．4及び他の付番されていない諸ステップで構成される。

【0023】ステップ2．1では、所望のネットワークを構成するための行数mと列数nとを割り当てることによって、ネットワークパラメータが初期化される。さらに、ノード計数Node_Cntがゼロにリセットされる。次にステップ2．2で、固有の名称または名称の代表の生成が、上述の式［1］に基づいて、ネットワーク内に包含される全ての3リンクノードについて実行される。

【0024】1つの3リンクノードに固有のノードアドレスが割り当てられると、ノード計数が偶数であるか否かについて判断される。ノード計数が偶数であれば、ステップ2．3において、固有のノードアドレスを有する3リンクノードが直立して配向される3リンクノードとして選定される。ノード計数が奇数であれば、ステップ2．4において、該当する3リンクノードは反転されて配向されるノードとして選定される。ネットワークに包含さ

れる全ての3リンクノードについて、ノードアドレスの生成及び直立または反転配向の何れかの3リンクノードへの割当てが完了すると、ノードカウンタNode_Cnt、行カウンタm_Cnt及び列カウンタn_Cntを各々ゼロにリセットした後、工程は3リンクノードを接続する第2のステップに進む。

【0025】図3から図5を参照すると、ネットワーク構成の第2ステップは、構成されるネットワークにおける3リンクノードの各々に対して、3リンクの送受信ポートの接続を提供する。この接続動作は、［2］及び［3］に引用された基準に基づいて行われる。

【0026】ステップ2．6では、2つの3リンクノード間のリンク（＋veリンク）の受信ポートと送信ポートが、［2］及び［3］の基準に基づいて接続される。またステップ2．7では、2つの3リンクノード間のリンク（－veリンク）の受信ポートと送信ポートとの接続が、基準［2］及び［3］に従って達成される。さらに、ステップ2．8では、直立して配向されたノードのリンク（－ve基数Mリンク）に対して、2つの3リンクノード間で受信ポート及び送信ポートが基準［2］及び［3］に従って接続され、ステップ2．9では、反転されて配向されたノードのリンク（＋ve基数Mリンク）に対して、受信ポート及び送信ポートが基準［2］及び［3］に従って接続される。3リンクノードの全てについて上述のステップを繰り返すことにより、図2が示すようなネットワークトポロジーが構成される。

【0027】図7から図10までは、ネットワークトポロジーにおける任意の2つの3リンクノード間でメッセージを経路指定するために各3リンクノード内で実行されるルーティングアルゴリズムを示すフローチャートである。以後、3リンクノードを簡単にノードと称する。

【0028】このルーティングアルゴリズムは、構成されるネットワークトポロジーにおけるソースノード及び宛先ノード間の最短経路を保証するものである。当該アルゴリズムは、ネットワークトポロジーのサイズを知るためにm及びnの値を必要としている。さらに、最短の可能経路を探索するために、ソースノード及び宛先ノードのアドレスも必要である。

【0029】ステップ3．1では、ネットワーク内の仮想行の数が計算される。「仮想行」は、アドレスフィールドX（MST_M、LST_N）と同一のMST_M値を有する直立配向ノードまたは反転配向ノードの何れかのみで構成される行として定義される。

【0030】次のステップ3．2では、3．2のボックス内に表記された式に従ってソースノード及び宛先ノード間の列距離が計算される。列距離は、アドレスフィールドX（MST_M、LST_N）におけるソースノード及び宛先ノードのLST_N値を使用して計算される。正方向の列距離（n_pos-dist）は、ソースノードの＋veリンクの送信ポートから宛先ノードに至るまでにメッセージが横断しなければならない横方向ノードの距離を与える。横方向ノードは、メッセージが宛先ノードへ送信されるまでに通過するノードである。負方向の列距離（n_neg-dist）は、ソースノードの－veリンクの送信ポートから宛先ノードに至るまでにメッセージが横断しなければならない横方向ノードのリンク数を与える。n_pos-distとn_neg-distの単位はどちらも同じである。

【0031】ステップ3．3では、ステップ3．1で計算された仮想行を使用して仮想行距離が計算される。正方向の仮想行距離（m_pos-dist）は、ソースノードの＋ve基数Mリンクの送信ポートから宛先ノードに至るまでにメッセージが横断しなければならない横方向ノードの距離である。負方向の仮想行距離（m_neg-dist）は、ソースノードの＋ve基数Mリンクの送信ポートから宛先ノードに至るまでにメッセージが横断しなければならない横方向ノードの距離である。これらの2つの行距離の計算は、ステップ3．3のボックス内に表記された条件に従って行われる。

【0032】ステップ3．2及び3．3において列距離及び行距離が計算された後は、メッセージの経路指定が水平方向及び垂直方向で決定される。

【0033】ステップ3．5及び3．4では、メッセージ経路指定の水平方向への可能方法が決定される。n_pos-dist＜n_neg-dist（ステップ3．5）であれば、＋veリンクが現行ノードのメッセージ経路指定方向n_directとして選択され、n_pos-dist＞n_neg-dist（ステップ3．4）であれば、－veリンクがn_directとして選択される。第3の可能性は、n_pos-dist＝n_neg-distの場合のn_direct＝LEVELである。「LEVEL」は、水平方向へのメッセージ経路指定が不要であることを意味している。この場合は、現行ノードの配向故に、現行ノードが垂直方向への経路指定のためのリンクを保有していないことが必要である。

【0034】ステップ3．6及び3．7では、メッセージ経路指定の垂直方向への3つの可能方法が決定される。m（宛先）／2＝m（ソース）／2（ステップ3．6）であれば、m_directはLEVELに設定される。m_pos-dist＜m_neg-distであれば、m_directは、現行ノードの＋ve基数Mリンクを指示するDOWNに設定される。m_pos-dist＞m_neg-distであれば、m_directは、現行ノードの－ve基数Mリンクを指示するUPに設定される。「LEVEL」は、宛先ノードが現行ノードと同一行にあることを示しており、ルーティングアルゴリズムがメッセージの入力ポートをメッセージの出力ポートとして選択することを防止する。m_directがDOWNのときは、メッセージの経路指定として＋ve基数Mリンクが選定され、m_directがUPのときは－ve基数Mリンクが選定される。

【0035】ステップ3．8では、経路指定されるメッセージの原点を指示する「エントリポイント」に依存して

3.9、3.10、3.11、3.12の4ケースに分類するために、メッセージの「エントリポイント」がチェックされる。

【0036】エントリポイント＝ホストノードであれば、m_directは、3．9のボックス内に定義されている条件に従って設定される。エントリポイント＝－ｖｅリンクであれば、m_direct及び／またはn_directは、3.10のボックス内に定義されている条件に従って設定される。エントリポイント＝＋ｖｅリンクであれば、m_direct及び／またはn_directは、3.11のボックス内に定義されている条件に従って設定される。エントリポイントが－ｖｅまたは＋ｖｅ基数Ｍリンクのどちらでもよい場合は、m_directはステップ3.12でＬＥＶＥＬに設定される。

【0037】3.9乃至3.12の4ステップのうちの1つにおいて決定される変数m_direct及びn_directは、各々、垂直及び水平方向における最終的な方向ポインタである。ステップ3.13は、3.13のボックス内に定義された条件に従って代替の経路指定手順を実行する。ステップ3.14では、最終的な経路指定出力リンクが決定される。

【0038】ソースノード及び横方向ノードは、宛先ノードに到達するまでにルーティングアルゴリズムを1回実行している。ルーティングアルゴリズムは、宛先アドレスが現行ノードのアドレスと同じになれば、処理を終了する。この場合は、現行ノードが宛先ノードである。ルーティングアルゴリズムは、横方向ノードの場合は2つの可能リンクのうちの1つから、またソースノードであれば3つの可能リンクのうちの1つからメッセージを経路指定する。このルーティングアルゴリズムは、ソースノードと宛先ノードとの間のメッセージの最短経路を発見することができる。

【0039】ネットワークの性能基準は、以下のように与えられる。即ち、ネットワーク内の2ノード間の最小距離は、ネットワークの直径に反映される。

直径＝Ｍ／２＋Ｎ／２　Ｎ＞Ｍの場合

直径＝Ｍ　　　　　　　Ｎ≦Ｍの場合

但し、Ｍはネットワークの行数であり、Ｎは列数である。

【0040】既存のネットワークトポロジーに追加することが可能なノードの最小増分数は、最小拡張増分に反映される。ネットワークトポロジーのこの特性は、ネットワークトポロジーのスケーラビリティの標識を提供する。既存のネットワークへのノードの追加には、行追加及び列追加の２方法がある。

事例１：行追加

$N_{新} = M_{旧} + 2$

追加ノード、$N = 2 \times N_{旧}$

事例２：列追加

$N_{新} = N_{旧} + 2$

追加ノード、$N = 2 \times M_{旧}$

【0041】3リンクノードを使用するネットワークトポロジーのためのルーティングアルゴリズムの優位点は、以下のように要約される。

１．各ノードまたは中央ノード毎にルーティングスキームを保持する必要がない。

２．アドレスまたは横方向ノード名の代表等の横方向ノードの経路指定情報を横方向ノード及び宛先ノードに送る必要がない。これによってメッセージを経路指定する間の帯域幅ユーティリティが削減される。

３．ルーティングアルゴリズムに必要な計算及びメモリリソースは最小であり、効率的な実行を達成することができる。これは主として、ルーティングタスクがネットワーク内に接続されたノード間に配分されるネットワークトポロジーのアキテクチャに起因している。各ノードは3つの可能リンクからの経路指定データのみを担当し、ネットワーク内のノードによって実行される分配された経路指定努力は、理論上の最短経路の決定を保証する。

４．１つのノードがネットワーク全体の経路指定を行う際に高度な計算性能を必要としない。

５．ルーティングアルゴリズムの分散型特性故に、3リンクノードのエミュレーションが容易に実行可能である。このエミュレーションは、既存のネットワークに追加されるノード数が最小拡張増分より少ない場合に必要である。物理的ノードは、仮想ノードを最小の計算及びメモリリソースによって容易に実行することができる。

【0042】本発明の好適な実施形態に関連して添付図面を参照しながら本発明について詳細に説明してきたが、当業者には様々な変更及び修正が明白である点は留意されなければならない。こうした変更及び修正は、添付の請求項によって定義された本発明の範囲を逸脱することなく、その範囲内に含まれるものとして理解されなければならない。

【図面の簡単な説明】

【図１】　（ａ）と（ｂ）とは、各々直立して配位されたノード及び反転されて配位されたノードを示したものであり、（ｃ）は隣接するノード間の完全二重リンクの接続を示したものである。

【図２】　3リンクノードで構成されたネットワークトポロジーを示したものである。

【図３】　本発明に一致するネットワークトポロジー構成を示すフローチャートである。

【図４】　本発明に一致するネットワークトポロジー構成を示すフローチャートである。

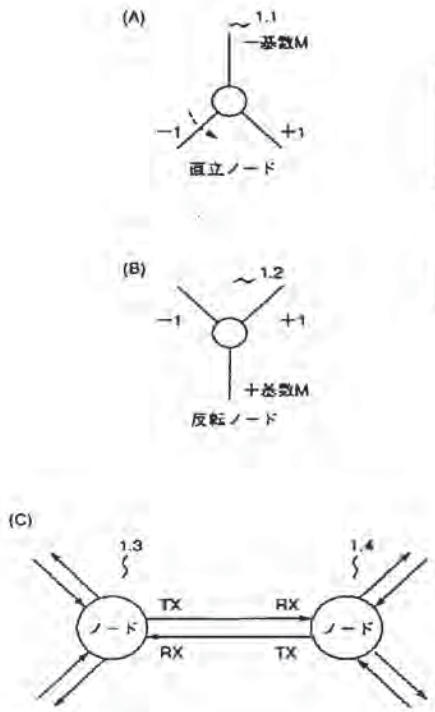
【図５】　本発明に一致するネットワークトポロジー構成を示すフローチャートである。

【図６】　本発明に一致するネットワークトポロジー構成を示すフローチャートである。

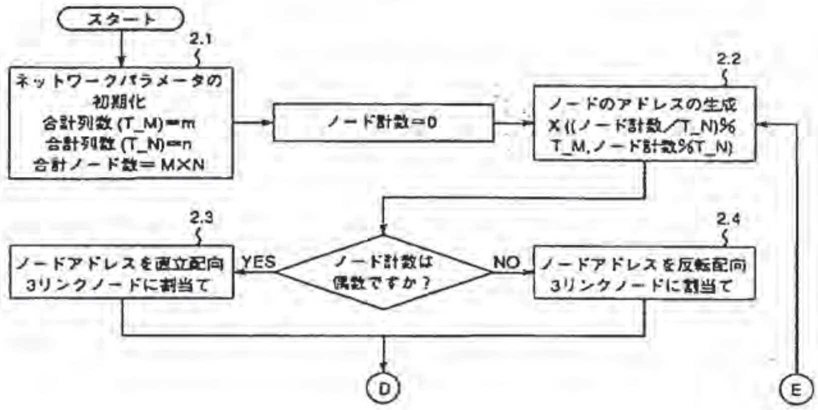
【図７】　本発明に一致する最短経路のルーティングアルゴリズムを示すフローチャートである。

【図8】 本発明に一致する最短経路のルーティングアルゴリズムを示すフローチャートである。
【図9】 本発明に一致する最短経路のルーティングアルゴリズムを示すフローチャートである。
【図10】 本発明に一致する最短経路のルーティングアルゴリズムを示すフローチャートである。
【符号の説明】
1．1・・・直立した3リンクノード
1．2・・・反転された3リンクノード
1．3，1．4・・・ノード

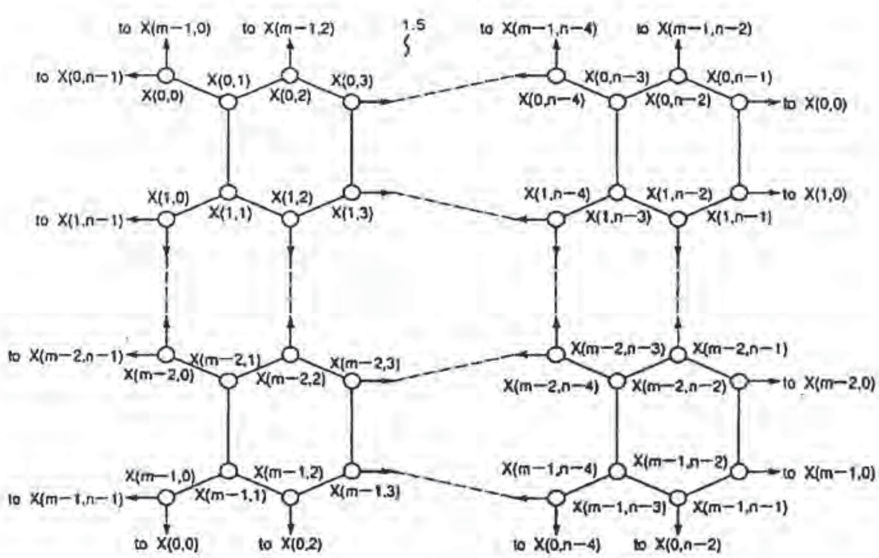
【図1】



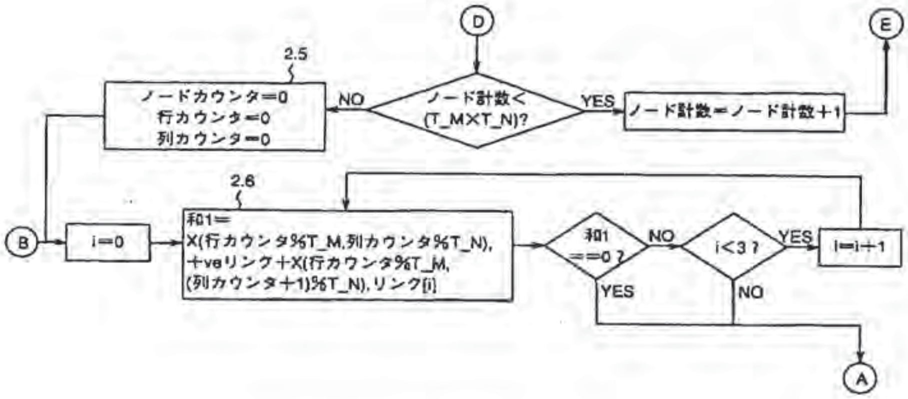
【図3】



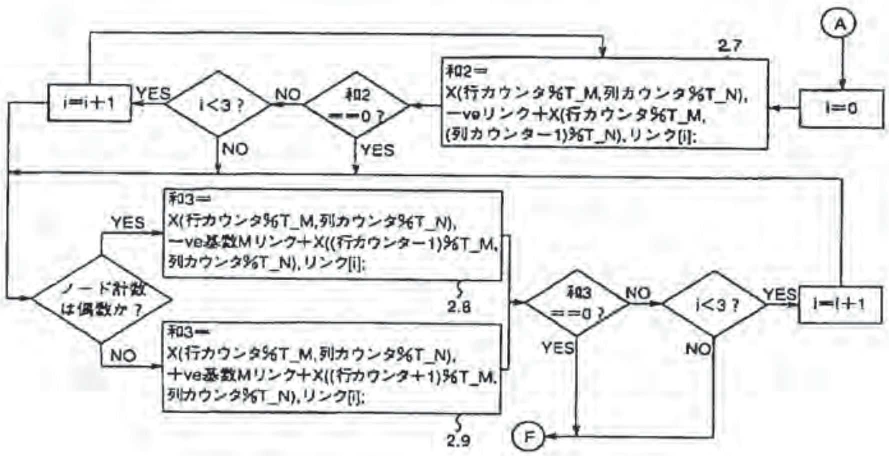
【図2】



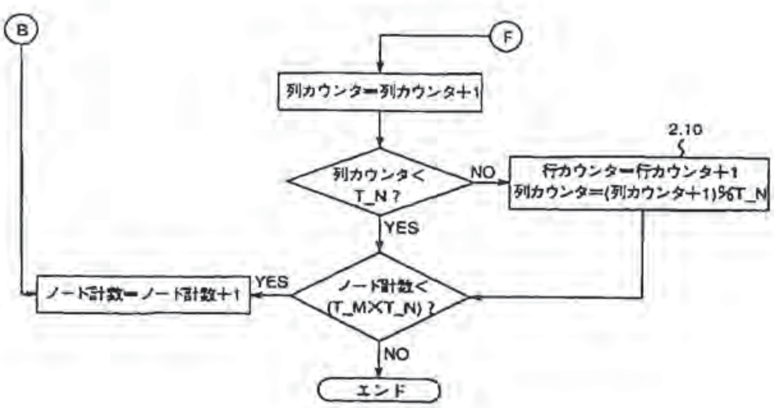
【図4】


D
E
2.5
ノードカウンタ=0
行カウンタ=0
列カウンタ=0
NO
ノード計数<
(T_M×T_N)?
YES
ノード計数=ノード計数+1
2.6
B
i=0
和1=
X(行カウンタ%T_M,列カウンタ%T_N),
+veリンク+X(行カウンタ%T_M,
(列カウンタ+1)%T_N),リンク[i]
和1
==0 ?
NO
i<3 ?
YES
i=i+1
YES
NO
A

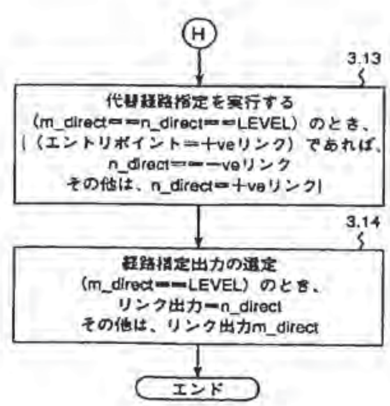
【図5】


A
2.7
和2=
X(行カウンタ%T_M,列カウンタ%T_N),
−veリンク+X(行カウンタ%T_M,
(列カウンタ−1)%T_N),リンク[i];
i=0
i=i+1
YES
i<3 ?
NO
和2
==0 ?
NO
YES
和3=
X(行カウンタ%T_M,列カウンタ%T_N),
−ve基数Mリンク+X((行カウンタ−1)%T_M,
列カウンタ%T_N),リンク[i];
YES
ノード計数
は偶数か ?
NO
2.8
和3=
X(行カウンタ%T_M,列カウンタ%T_N),
+ve基数Mリンク+X((行カウンタ+1)%T_M,
列カウンタ%T_N),リンク[i];
2.9
和3
==0 ?
NO
i<3 ?
YES
i=i+1
YES
NO
F

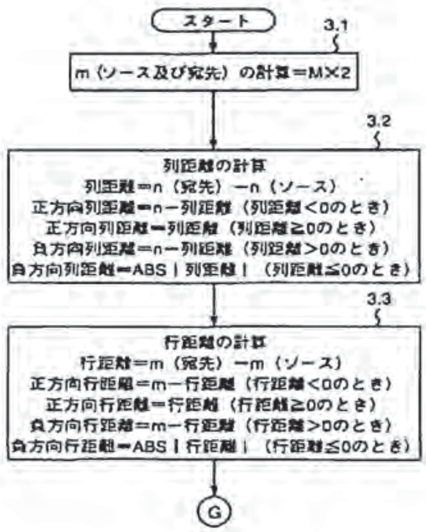
【図6】


B
F
列カウンタ=列カウンタ+1
2.10
列カウンタ<
T_N ?
NO
行カウンタ=行カウンタ+1
列カウンタ=(列カウンタ+1)%T_N
YES
ノード計数=ノード計数+1
YES
ノード計数<
(T_M×T_N) ?
NO
エンド

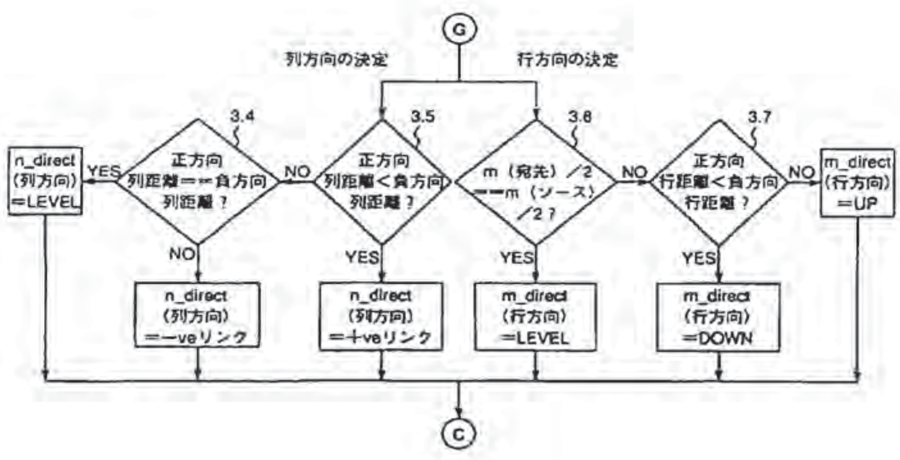
【図10】


H
3.13
代替経路指定を実行する
(m_direct==n_direct==LEVEL) のとき、
|(エントリポイント=+veリンク) であれば、
n_direct=−veリンク
その他は、n_direct=+veリンク|
3.14
経路指定出力の選定
(m_direct==LEVEL) のとき、
リンク出力=n_direct
その他は、リンク出力m_direct
エンド

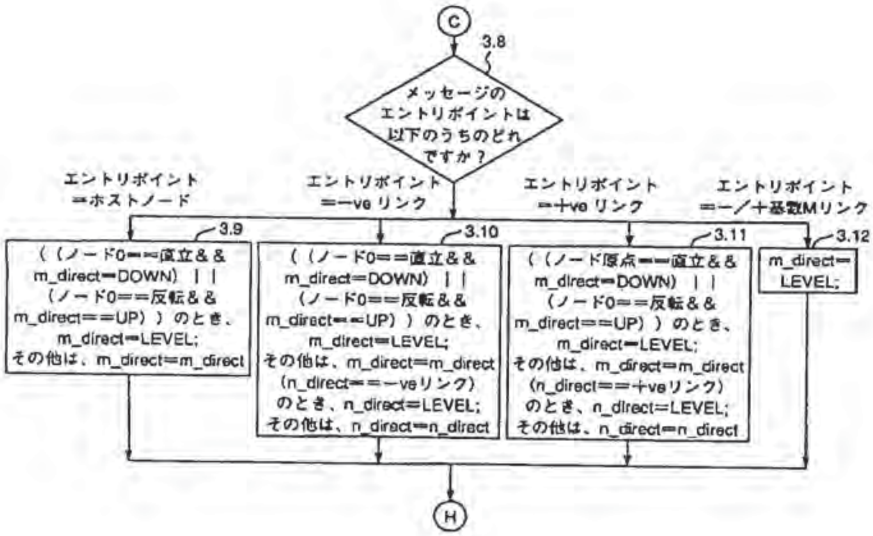
【図7】


スタート
3.1
m（ソース及び宛先）の計算＝M×2
3.2
列距離の計算
列距離＝n（宛先）−n（ソース）
正方向列距離＝n−列距離（列距離＜0のとき）
正方向列距離＝列距離（列距離≧0のとき）
負方向列距離＝n−列距離（列距離＞0のとき）
負方向列距離＝ABS｜列距離｜（列距離≦0のとき）
3.3
行距離の計算
行距離＝m（宛先）−m（ソース）
正方向行距離＝m−行距離（行距離＜0のとき）
正方向行距離＝行距離（行距離≧0のとき）
負方向行距離＝m−行距離（行距離＞0のとき）
負方向行距離＝ABS｜行距離｜（行距離≦0のとき）
G

【図8】


G
列方向の決定
行方向の決定
3.4
正方向列距離＝＝負方向列距離？
YES
n_direct（列方向）＝LEVEL
NO
n_direct（列方向）＝−veリンク
3.5
正方向列距離＜負方向列距離？
YES
n_direct（列方向）＝＋veリンク
3.6
m（宛先）／2＝＝m（ソース）／2？
YES
m_direct（行方向）＝LEVEL
NO
3.7
正方向行距離＜負方向行距離？
YES
m_direct（行方向）＝DOWN
NO
m_direct（行方向）＝UP
C

【図9】


C
3.8
メッセージのエントリポイントは以下のうちのどれですか？
エントリポイント＝ホストノード
エントリポイント＝＝−ve リンク
エントリポイント＝＋ve リンク
エントリポイント＝−／＋基数Mリンク
3.9
((ノード0＝＝直立＆＆m_direct＝DOWN) || (ノード0＝＝反転＆＆m_direct＝＝UP)) のとき、m_direct＝LEVEL; その他は、m_direct＝m_direct
3.10
((ノード0＝＝直立＆＆m_direct＝DOWN) || (ノード0＝＝反転＆＆m_direct＝＝UP)) のとき、m_direct＝LEVEL; その他は、m_direct＝m_direct (n_direct＝＝−veリンク) のとき、n_direct＝LEVEL; その他は、n_direct＝n_direct
3.11
((ノード原点＝＝直立＆＆m_direct＝DOWN) || (ノード0＝＝反転＆＆m_direct＝＝UP)) のとき、m_direct＝LEVEL; その他は、m_direct＝m_direct (n_direct＝＝＋veリンク) のとき、n_direct＝LEVEL; その他は、n_direct＝n_direct
3.12
m_direct＝LEVEL;
H

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/703,942	10/31/2000	Eric A. Pulsipher	10008102-1	4047

22879 7590 07/06/2004

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
SCHULTZ, WILLIAM C	
ART UNIT	**PAPER NUMBER**
2664	13

DATE MAILED: 07/06/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Office Action Summary	Application No.	Applicant(s)	
	09/703,942	PULSIPHER ET AL.	
	Examiner	Art Unit	
	William C. Schultz	2664	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *31 October 2000*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *31 October 2000* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *8/13/2002*.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

Information Disclosure Statement

The information disclosure statement (IDS) submitted on 8/13/2002 is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statement has been considered by the examiner.

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Claims 1-7,9,15-20 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention.

"converting", one skilled in the art knows that conversion is changing something from one thing into another thing. The specification fails to disclose what the topology was before conversion into a tuple. The topology had to be stored as some kind of data representation before conversion into a tuple. Since, there is no disclosure of what a topology was before conversion, the specification fails to enable.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 8,10,12-14 are rejected under 35 U.S.C. 102(b) as being anticipated by George et al. [U.S. Pat. 4,644,532].

A note on claim interpretation: the term tuple is defined twice in spec. First on page 5 , line 25 then on page 6, line 7. Page 6 contains a broader definition of a tuple as "any collection of assorted data", this could be thought of as a database record. No limit on size is presented in the spec., big or small, with respect to what the tuple holds, some examples have been presented of what a tuple could contain but still nothing limits a tuple from being just one item of data about a link or path or arc, as would be understood by one skilled in the art of network topology mapping to be equivalents of each other.

"converting": The examiner will take the meaning of "converting" to mean building, see above 112 1st. Claim 8 recites apparatus type limitations and since the applicant can label any box anything no specific meaning is construed into the limitation's function by merely it's labelling.

Regarding claim 8, George et al. discloses all the following subject matter: A system for mapping a network topology by identifying changes between an existing topology and a new topology, based on changes to data tuples that represent nodal connections comprising:

a topology database that stores an existing topology of a network; and **(fig. 2, part 105; col. 5, lines 49-51)** a topology converter **(fig. 2, part 103)** connected to the topology database **(fig. 2 part 105)** that receives new tuples that represent new nodal connections;**(col. 5, lines 60-63; col. 6, lines 55-68)** and compares the new tuples with the existing topology to identify changes in the network.**(col. 12, lines 54-66)**

regarding claim 10, George et al. further discloses the topology converter updates the topology database with a new topology based on the new tuples. **(col. 12, lines 56-58 – the time stamp and sequence number is data in a tuple)**

regarding claim 12, George et al. further discloses the topology converter identifies duplicate tuples that appear both in the list of existing tuples and in the new tuples, and maintains a current status of the topology for these tuples. **(col. 12, lines 58-62)**

regarding claims 13,14, George et al. further discloses the topology converter searches for a host of a new singly-heard host link tuple or a new multi-heard host link tuple in the list of existing tuples. **(fig. 1 – disclosed in the figure are what appears to be singly-heard hosts, since the limitation is in the alternative this citation meets the claimed limitation; col. 12, lines 54-62 – searching is inherent because the sequence number is compared, the sequence number could not have been compared if it was not searched for in the database)**

further regarding claim 14, the conflict tuple being searched for is the tuple that has a change associated with it. As already disclosed in the rejection for claim 13, the topology monitor searches for the"tuple" that matches the message it got a message for.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claim 11 is rejected under 35 U.S.C. 103(a) as being unpatentable over George et al. [U.S. Pat. 4,644,532] as applied to claim 8 above, and further in view of Stevenson et al. [U.S. Pat. 5,023,873].

Regarding claim 11, George et al. discloses link failures as events for causing topology changes that needs updating in the topology database. The link goes down for some reason. George et al. fails to disclose that the link failure could be resolved by swapping the ports and the host appears on the network again but with a different port.

Stevenson et al. discloses that swapping ports in communication systems to correct for a link failure was well known at the time of invention. **(col. 12, lines 8-12, lines 26-34)**

It would have been obvious for one skilled in the art at the time of invention to include in George et al. the ability to identify that a port has been swapped and that causes a topology change update to occur.

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Diebboll et al. [U.S. Pat. 5,886,643] - The reference has all the featured limitations of the rejected independent claims and could have been used in a second 102 rejection but George et al. is older.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to William C. Schultz whose telephone number is 703-305-2367. The examiner can normally be reached on M-F(7-4)(first bi-week) M-Th(7-4)(second bi-week).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wellington Chin can be reached on 703-305-4366. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

William Schultz

WELLINGTON CHIN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2600

Notice of References Cited	Application/Control No. 09/703,942	Applicant(s)/Patent Under Reexamination PULSIPHER ET AL.	
	Examiner William C. Schultz	Art Unit 2664	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-4,644,532	02-1987	George et al.	370/255
	B	US-5,023,873	06-1991	Stevenson et al.	714/4
	C	US-5,740,346	04-1998	Wicki et al.	714/22
	D	US-6,295,541	09-2001	Bodnar et al.	707/203
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

OIPE AUG 1 3 2002 PATENT & TRADEMARK OFFICE

PATENT APPLICATION
Sheet 1 of 1

FORM PTO-1449 LIST OF PATENTS AND PUBLICATIONS FOR APPLICANT'S INFORMATION DISCLOSURE STATEMENT (Use several sheets if necessary)	ATTY. DOCKET NO. 10008102-1	APPLICATION NO. 09/703,942	CONFIRMATION NO.
	APPLICANT Eric A. PULSIPHER et al.		
	FILING DATE 10/31/2000	GROUP	

REFERENCE DESIGNATION **U.S. PATENT DOCUMENTS**

EXAMINER INITIAL		DOCUMENT NUMBER	PUBLICATION DATE	NAME	Pages, Columns, Lines Where Relevant Passages or Figures Appear
WCS	1A	5,886,643	3/23/99	Diebboll, et al.	
	1B				
	1C				
	1D				
	1E				
	1F				
	1G				
	1H				
	1I				
	1J				
	1K				

RECEIVED
AUG 1 4 2002
Technology Center 2600

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	PUBLICATION DATE	NAME OF PATENTEE OR APPLICANT	Pages/Columns/Lines Where Relevant Passages/Figures Appear	Check if Translation attached
WCS	1L	EP0830047A2	3/18/98	Howard A. SEID		
WCS	1M	WO98/44400	10/8/98	Toshiharu FUKUI		
WCS	1N	JP2000-76209	3/14/00	Peku Yu TAN		
	1O					
	1P					

OTHER REFERENCES (including Author, Title, Date, Pertinent Pages, etc.)

	1Q	GB Search Report dated 5/16/2002 issued in connection with corresponding GB application no. 0125507.4
	1R	
	1S	

EXAMINER	DATE CONSIDERED
WCS	6/21/04

Rev 05/02 (PTO1449)

10-06-04

2664
41

Serial No. 09/703,942

Attorney Docket No.: HP10008102-1 **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Pulsipher, et al.	Examiner:	Schultz, William C.
Serial No.:	09/703,942	Art Unit:	2664
Filed:	10/31/2000		
For:	METHOD AND SYSTEM FOR IDENTIFYING AND PROCESSING CHANGES TO A NETWORK TOPOLOGY		

Box Non-Fee Amendment
Commissioner of Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDMENT AND REQUEST FOR RECONSIDERATION

Sir:

In response to the July 6, 2004, Office Action, Applicant hereby submits the following Amendment and Request for Reconsideration. Please amend the above-identified application as follows:

Amendments to the Claims are reflected in the listing of claims which begins on page 2 of this paper.

Remarks/Arguments begin on page 5 of this paper.

I hereby certify that the document is being deposited with the United States Postal Service as express mail post office to addressee in an envelope addressed to: Mail Stop Non-Fee Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on:

10-4-2004 ER 264911265 US
(Date of Deposit and Express Mail No.)

Shane P. Coleman
(Name)

Shane P. Coleman
Signature

Express Mail Label No.: # ER 264911265 US

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Pulsipher, et al.

Application No.: 09/703,942

Filed: October 31, 2000

For: METHOD AND SYSTEM FOR IDENTIFYING AND PROCESSING CHANGES TOA NETWORK TOPOLOGY

Examiner: Schultz, William C.

Art Unit: 2664

CERTIFICATE OF MAILING BY EXPRESS MAIL

Box Non-Fee Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

The undersigned hereby certifies that the following documents:

1. Amendment and Request for Reconsideration (7 pages);
2. Certification of Mailing by Express Mail (1 page); and
3. Return Card.

relating to the above application, were deposited as "Express Mail," Mailing Label No. ER 264911265 US with the United States Postal Service, addressed to Box Non-Fee Amendment, Commissioner of Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on this 4th day of October, 2004.

Date: October 4, 2004

Shane Coleman
Mailer
Dorsey & Whitney LLP

[signature]
Signature of Mailer

This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS

1. (currently amended) In a network having interconnected nodes with data tuples that represent nodal connections, a method for mapping a network topology by identifying changes between an existing topology and a new topology, the method comprising:

creating a plurality of tuples for a topology of a network, wherein the tuples represent nodal connections of the network and wherein each of the tuples comprises a host identifier, interface information, and a port specification;

~~converting an existing topology into a list of existing tuples that represent existing nodal connections;~~

receiving new tuples that represent new nodal connections; and

comparing the list of existing tuples with the new tuples to identify changes to the topology.

2. (original) The method of claim 1, further comprising updating a topology database with a new topology.

3. (original) The method of claim 1, further comprising taking action on the changes to the topology.

4. (canceled)

5. (original) The method of claim 1, wherein the step of comparing comprises identifying duplicate tuples that appear both in the list of existing tuples and in the new tuples, and maintaining a current status of the topology for these tuples.

6. (original) The method of claim 1, wherein the step of comparing comprises identifying a swapped port condition on a connector.

7. (original) The method of claim 1, wherein the step of comparing comprises searching for a host of a new singly-heard host link tuple or a new multi-heard host link tuple in the list of existing tuples.

8. (currently amended) A system for mapping a network topology by identifying changes between an existing topology and a new topology, based on changes to data tuples that represent nodal connections comprising:

a topology database that stores an existing topology of a network using tuples, wherein each tuple includes a host identifier, interface information, and a port specification for a node in the network; and

a topology converter connected to the topology database that receives new tuples that represent new nodal connections,~~;~~ and compares the new tuples with the existing topology to identify changes in the network by comparing the host identifiers, the interface information, and the port specifications.

9. (currently amended) The system of claim 8, wherein the topology converter creates the tuples for the topology of the network. ~~converts the existing topology into a list of existing tuples that represent existing nodal connections.~~

10. (original) The system of claim 8, wherein the topology converter updates the topology database with a new topology based on the new tuples.

11. (original) The system of claim 8, wherein the topology converter attempts to identify swapped ports on connectors.

12. (original) The system of claim 8, wherein the topology converter identifies duplicate tuples that appear both in the list of existing tuples and in the new tuples, and maintains a current status of the topology for these tuples.

13. (original) The system of claim 8, wherein the topology converter searches for a host of a new singly-heard host link tuple or a new multi-heard host link tuple in the list of existing tuples.

14. (original) The system of claim 8, wherein the topology converter searches for a connector of a new conflict links tuple in the list of existing tuples.

15. (currently amended) A computer-readable medium having computer-executable instructions for performing a method for mapping a network topology by

identifying changes between an existing topology and a new topology in a network having a interconnected nodes, the method comprising:

creating a plurality of tuples for a topology of a network, wherein the tuples represent nodal connections of the network and wherein each of the tuples comprises a host identifier, interface information, and a port specification;

~~converting an existing topology into a list of existing tuples that represent existing nodal connections;~~

receiving new tuples that represent new nodal connections;

comparing the list of existing tuples with the new tuples to identify changes to the topology; and

updating a topology database with a new topology based on the comparing.

16. (currently amended) The medium ~~method~~ of claim 15, wherein a topology converter receives the new tuples from a connection calculator that calculates connections between nodes.

17. (currently amended) The medium ~~method~~ of claim 15, wherein the step of comparing comprises identifying duplicate tuples that appear both in the list of existing tuples and in the new tuples, and maintaining a current status of the topology for these tuples.

18. (currently amended) The medium ~~method~~ of claim 15, wherein the step of comparing comprises identifying a swapped port condition on a connector.

19. (currently amended) The medium ~~method~~ of claim 15, wherein the step of comparing comprises searching for a host of a new singly-heard host link tuple or a new multi-heard host link tuple in the list of existing tuples.

20. (currently amended) The medium ~~method~~ of claim 15, wherein the step of comparing comprises searching for a connector of a new conflict links tuple in the list of existing tuples.

REMARKS

Claims 1-3 and 5-20 are pending. Claims 1-20 are rejected. Claims 1-7, 9, and 15-20 are rejected under 35 U.S.C. § 112, ¶ 1 as lacking enablement. Claims 8, 10, and 12-14 are rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 4,644,532 to George, et al. (hereinafter "George"). Claim 11 is rejected under 35 U.S.C. § 103(a) as being unpatentable over George in view of U.S. Patent No. 5,023,873 to Stevenson, et al. (hereinafter "Stevenson"). By this amendment, claims 1, 8, 9, and 15-20 are amended. Claim 4 is canceled. No new claims are added. No new matter is added. Reconsideration and allowance is requested.

Claim 1 is amended to recite the step of "creating a plurality of tuples for a topology of a network" in place of the current step of "converting…." Claim 1 is rejected under section 112 for lacking enablement for the term "converting." Applicant respectively disagrees with the examiner's assertion that the term "converting" lacks enablement. The converting process is described with respect to Figures 18a-b, also referred to as the "morph topo" phase of the process. However, the above amendments replace the term "converting" in favor of the step of "creating" to more particularly claim the invention. Support for this amendment is found, for example, on pages 16-17 of the specification and in Figures 17 and 18a-b. As amended, claim 1 overcomes the section 112 rejection. Reconsideration is requested.

Claim 1 is further amended to define the tuples as comprising "a host identifier, interface information, and a port specification." As amended, claim 1 overcomes the cited references. The cited references do not teach or suggest a tuple containing these items. As amended, claim 1 is allowable. Claims 2-3, and 5-7 depend from claim 1 and for this reason and the other limitations they recite, are also allowable.

Claim 4 is hereby canceled without prejudice or disclaimer.

Claim 8 has been amended to recite that the tuples comprise "a host identifier, interface information, and a port specification" and that the system compares new tuples with the existing topology by "comparing the host identifiers, the interface information, and the port specifications." The cited references do not teach or suggest these particular elements of the tuple. Nor do the cited references teach or suggest comparing these elements to identify changes to the network topology. In contrast, George monitors data in the topology database by monitoring the weights of the data and by comparing time references for the data. As amended, claim 8 is allowable. Claims 9-14 depend from claim 8 and for this reasons and the other limitations they recite are also allowable. Reconsideration is requested.

Claim 15 is amended to recite that the method performs the step of "creating a plurality of tuples for a topology of a network" in place of the current step of "converting…." Claim 15 is rejected under section 112 for lacking enablement for the term "converting." Applicant respectively disagrees with the examiner's assertion that the term "converting" lacks enablement. The converting process is described with respect to Figures 18a-b, also referred to as the "morph topo" phase of the process. However, the above amendments replace the term "converting" in favor of the step of "creating" to more particularly claim the invention. Support for this amendment is found, for example, on pages 16-17 of the specification and in Figures 17 and 18a-b. As amended, claim 15 overcomes the section 112 rejection. Reconsideration is requested.

Claim 15 is further amended to define the tuples as comprising "a host identifier, interface information, and a port specification." As amended, claim 15 overcomes the cited references. The cited references do not teach or suggest a tuple containing these items. Claim 15 is further amended to clarify that the "updating" step is based on the "comparing" step. As amended, claim 15 is allowable. Claims 16-20 depend from claim 15 and for this reason and the other limitations they recite, are also allowable.

Claims 16-20 are also amended for reasons unrelated to patentability to make clear that claims 16-20 are dependent claims to a medium rather than a method. As amended, claims 16-20 are allowable.

In view of the above amendments and remarks, Applicant respectfully asserts that the application is in condition for allowance. The above amendments should generate no additional fees. However, if any such fees are necessary, the Office is hereby authorized to charge any fee deficiency associated with this communication to Deposit Account 04-1420. Prompt reexamination and allowance of claims 1-3, and 5-20 is respectfully requested.

Respectfully submitted,

Date: **October 4, 2004**

Shane P. Coleman, Reg. No. 44,623
DORSEY & WHITNEY LLP
125 Bank Street, Suite 600
Missoula, MT 59802-4407
Tel. (406) 721-6025
Fax (406) 543-0863

Refine Search

Search Results -

Terms	Documents
5297138.uref. and ((identification or ID) same interface$ same port$)	4

Database:

US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search:

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Thursday, February 17, 2005 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=ADJ		
L29	5297138.uref. and ((identification or ID) same interface$ same port$)	4	L29
L28	5297138.uref. and L3	0	L28
L27	5297138.pn. or 5450408.pn.	2	L27
L26	(compar$6 or updat$6 or creat$6) and L25	4	L26
L25	((identification or ID) same interface$ same port$) and (L23 or L21)	5	L25
L24	L23 and L3	0	L24
L23	6405248.uref. or 5886643.uref. or 5727157.uref.	43	L23
L22	L21 and L3	0	L22
L21	6405248.pn. or 5886643.pn. or 5727157.pn.	3	L21
L20	6563835.pn. and L3	0	L20
L19	(compar$6 or updat$6 or creat$6) same L14	3	L19
L18	L17 same L3	0	L18
L17	(topology near4 (table$ or (database$ or (data near1 base$))))	889	L17

L16	(compar$6 or updat$6 or creat$6) same L3	6	L16
L15	L14 same (compar$6 or updat$6 or creat$6)	3	L15
L14	(table$ or (database$ or (data near1 base$))) same L3	25	L14
L13	L3 and L12	0	L13
L12	4644532.uref. and ((identification or ID) same interface$ same port$)	11	L12
L11	((identification or ID) same interface$ same port$) and L2	0	L11
L10	((identification or ID) same interface$ same port$) and L3	46	L10
L9	((identification or ID) same interface$ same port$) and L1	0	L9
L8	((identification or ID or number) same interface$ same port$) and L1	1	L8
L7	((identification or ID or number) same interface$ same port$) and (L3 or L1)	98	L7
L6	L3 and L1	0	L6
L5	(host near1 (identification or ID or number) same port$) and L2	0	L5
L4	L3 and L2	0	L4
L3	(host near1 (identification or ID or number) same interface$ same port$)	97	L3
L2	5023873.uref.	29	L2
L1	5023873.pn.	1	L1

END OF SEARCH HISTORY

Refine Search

Search Results -

Terms	Documents
((database$ or (data near1 base$) or table$) same interface$) and 6405248.pn.	1

Database:

- US Pre-Grant Publication Full-Text Database
- US Patents Full-Text Database
- US OCR Full-Text Database
- EPO Abstracts Database
- JPO Abstracts Database
- Derwent World Patents Index
- IBM Technical Disclosure Bulletins

Search:

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Friday, February 18, 2005 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=ADJ		
L25	((database$ or (data near1 base$) or table$) same interface$) and 6405248.pn.	1	L25
L24	6405248.pn. and interface$	1	L24
L23	L22 and ((identifier or id or number) same interface$ same port$)	0	L23
L22	L20 and ((compar$6 or updat$6) near1 (database$ or (data near1 base$) or table$))	5	L22
L21	L20 and L12	0	L21
L20	5732086.uref.	23	L20
L19	((identifier or id or number) same interface$ same port$ same (database$ or (data near1 base$) or table$)) and (4827411.uref. or 5049873.uref. or 5101348.uref.)	7	L19
L18	L17 and (4827411.uref. or 5049873.uref. or 5101348.uref.)	122	L18
L17	(database$ or (data near1 base$) or table$) same L2	174481	L17

L16	L15 and L12	11	L16
L15	L2 and (4827411.uref. or 5049873.uref. or 5101348.uref.)	231	L15
L14	L12 and l6	11	L14
L13	L12 and L4	11	L13
L12	(compar$6 or updat$6) near1 (topology near4 (database$ or (data near1 base$) or table$))	124	L12
L11	L10 and (4827411.uref. or 5049873.uref. or 5101348.uref.)	1	L11
L10	((host or client or node or device$) near1 (identifier or id or number)) same interface$ same port$	765	L10
L9	(tuple$ same L1) and (4827411.uref. or 5049873.uref. or 5101348.uref.)	1	L9
L8	tuple$ and (4827411.uref. or 5049873.uref. or 5101348.uref.)	14	L8
L7	L6 and L1	44	L7
L6	L4 and (topology near4 (database$ or (data near1 base$) or table$))	44	L6
L5	L1 and (4827411.uref. or 5049873.uref. or 5101348.uref.) and (database$ or (data near1 base$) or table$) and topology	81	L5
L4	L1 and (4827411.uref. or 5049873.uref. or 5101348.uref.)	219	L4
L3	L2 and (4827411.pn. or 5049873.pn. or 5101348.pn.)	1	L3
L2	(identifier or id or number) same interface$ or port$	2458262	L2
L1	((host or client or node or device$) near1 (identifier or id or number)) same interface$ or port$	2442325	L1

END OF SEARCH HISTORY

HmN

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/703,942	10/31/2000	Eric A. Pulsipher	10008102-1	4047

22879 7590 02/23/2005

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
HO, CHUONG T	
ART UNIT	**PAPER NUMBER**
2664	

DATE MAILED: 02/23/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Office Action Summary	**Application No.**	**Applicant(s)**	
	09/703,942	PULSIPHER ET AL.	
	Examiner	**Art Unit**	
	CHUONG T HO	2664	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 October 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-3 and 5-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-3 and 5-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

1. The amendment filed 10/13/04 have been entered and made of record.

2. Applicant's amendment filed 10/13/04 with respect to claims 1-20 have been considered but are moot in view of the new ground(s) of rejection.

3. Claims 1-3, 5-20 are pending.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-3, 5-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Liang et al. (U.S.Patent No. 5,732,086) in view of Wood (U.S.Patent No. 6,405,248 B1).

In the claim 1, see figure 3, Liang et al. discloses topology table 26 (tuples) which comprises: node identifier (host identifier), link port (port sepecification) (see col. 5, lines 7-15, lines 21-25, col. 6, lines 5-9, col. 7, lines 20-23, col. 14-16); comprising:

Creating a plurality of tuples (topology table 26, see figure 3) for a topology of a network, wherein the tuples represent nodal connections of the network and wherein each of the tuples (topology tables 26) comprises a host identifier, a port specification (see col. 5, lines 7-15, lines 21-25, col. 6, lines 5-9, col. 7, lines 20-23, col. 14-16);

receiving new tuples (host identifier, port specification) that represent new nodal connections (see col. 7, lines 20-23);

comparing the list of existing tuples (host identifier, port specification) with the new tuples to identify changes to the topology (see col. 7, lines 20-23).

However, Liang et al. is silent to disclosing tuples comprising interface information.

Wood discloses creating an accurate topology map of a given network by: obtaining a list of managed network device; identifying link port and node port....device interface information (see abstract); comprising:

Each of the tuples (topology map) comprises interface information, a port specification, host identifier (see abstract, table 1, col. 6, lines 10-20).

Liang and Wood are directed to routing packets using IP protocol. Wood recognizes that there may be conflicting source address. Hence, it would have been obvious to use the interface information of Wood in Liang as it would have resolved source address conflicts when routing using the table (tuples) of both Liang and Wood.

5. In the claim 8, see figure 3, Liang et al. discloses topology table 26 (tuples) which comprises: node identifier (host identifier), link port (port sepecification) (see col. 5, lines 7-15, lines 21-25, col. 6, lines 5-9, col. 7, lines 20-23, col. 14-16); comprising:

A topology database that stores an existing topology of a network using tuples (topology table 26, see figure 3) , wherein each tuple includes a host identifier, port specification for a node in the network (see col. 5, lines 7-15, lines 21-25, col. 6, lines 5-9, col. 7, lines 20-23, col. 14-16);

A topology converter connected to the topology database that receives new tuples that represent new nodal connections, and compare the new tuples with the existing topology to identify changes in the network by comparing the host identifier, port specification (see col. 7, lines 20-23);

However, Liang et al. is silent to disclosing tuples comprising interface information.

Wood discloses creating an accurate topology map of a given network by: obtaining a list of managed network device; identifying link port and node port....device interface information (see abstract); comprising:

Each of the tuples (topology map) comprises interface information, a port specification, host identifier (see abstract, table 1, col. 6, lines 10-20).

Liang and Wood are directed to routing packets using IP protocol. Wood recognizes that there may be conflicting source address. Hence, it would have been obvious to use the interface information of Wood in Liang as it would have resolved source address conflicts when routing using the table (tuples) of both Liang and Wood.

6. In the claim 15, see figure 3, Liang et al. discloses topology table 26 (tuples) which comprises: node identifier (host identifier), link port (port sepecification) (see col. 5, lines 7-15, lines 21-25, col. 6, lines 5-9, col. 7, lines 20-23, col. 14-16); comprising:

Creating a plurality of tuples (topology table 26, see figure 3) for a topology of a network, wherein the tuples represent nodal connections of the network and wherein each of the tuples (topology tables 26) comprises a host identifier, a port specification (see col. 5, lines 7-15, lines 21-25, col. 6, lines 5-9, col. 7, lines 20-23, col. 14-16);

receiving new tuples (host identifier, port specification) that represent new nodal connections (see col. 7, lines 20-23);

comparing the list of existing tuples (host identifier, port specification) with the new tuples to identify changes to the topology (see col. 7, lines 20-23).

However, Liang et al. is silent to disclosing tuples comprising interface information.

Wood discloses creating an accurate topology map of a given network by: obtaining a list of managed network device; identifying link port and node port....device interface information (see abstract); comprising:

Each of the tuples (topology map) comprises interface information, a port specification, host identifier (see abstract, table 1, col. 6, lines 10-20).

Liang and Wood are directed to routing packets using IP protocol. Wood recognizes that there may be conflicting source address. Hence, it would have been obvious to use the interface information of Wood in Liang as it would have resolved source address conflicts when routing using the table (tuples) of both Liang and Wood.

7. In the claims 2, 10, Liang et al. discloses updating a topology database with a new topology (see col. 7, lines 20-23).

8. In the claim 3, Liang et al. discloses taking action on the changes to the topology (see col. 7, lines 20-23).

9. In the claims 5, 6, 11, 12, 18, Liang et al. discloses the step of comparing comprises identifying duplicate tuples that appear both in the list of existing tuples and in the new tuples, and maintaining a current status of the topology for these tuples (see col. 7, lines 20-23).

10. In the claims 7, 13, 14, 19, Liang et al. discloses the step of comparing comprises searching for a host of a new singly-heard host link tuple or a new multi-heard host link tuple in the list of existing tuples (see col. 7, lines 20-23).

11. In the claim 9, Wood discloses wherein the topology converter creates the tuples for the topology of the network (see abstract, table 1, col. 6, lines 10-20).

12. In the claims 17, 16, 20, Liang et al. discloses wherein the step of comparing comprises identifying duplicate tuples that appear both in the list of existing tuples and in the new tuples, and maintaining a current status of the topology of these tuples (see col. 7, lines 20-23).

13. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Conclusion

14. Any inquiry concerning this communication or earlier communications from the examiner should be directed to CHUONG T HO whose telephone number is (571) 272-3133. The examiner can normally be reached on 8:00 am to 4:00 pm.

The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

02/16/05

Notice of References Cited

Application/Control No.	Applicant(s)/Patent Under Reexamination
09/703,942	PULSIPHER ET AL.
Examiner	Art Unit
CHUONG T HO	2664

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-6,405,248	06-2002	Wood, Michael	709/223
	B	US-6,108,702	08-2000	Wood, Michael	709/224
	C	US-5,732,086	03-1998	Liang et al.	370/410
	D	US-5,729,685	03-1998	Chatwani et al.	709/224
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Index of Claims	Application No. 09/703,942	Applicant(s) PULSIPHER ET AL.
	Examiner CHUONG T HO	Art Unit 2664

√	Rejected	–	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Claim Final	Claim Original	Date 02/16/05	Claim Final	Claim Original	Date	Claim Final	Claim Original	Date
	(1)	√		51			101	
	2	√		52			102	
	3	√		53			103	
	~~4~~			54			104	
	5	√		55			105	
	6	√		56			106	
	7	√		57			107	
	(8)	√		58			108	
	9	√		59			109	
	10	√		60			110	
	11	√		61			111	
	12	√		62			112	
	13	√		63			113	
	14	√		64			114	
	(15)	√		65			115	
	16	√		66			116	
	17	√		67			117	
	18	√		68			118	
	19	√		69			119	
	20	√		70			120	
	21			71			121	
	22			72			122	
	23			73			123	
	24			74			124	
	25			75			125	
	26			76			126	
	27			77			127	
	28			78			128	
	29			79			129	
	30			80			130	
	31			81			131	
	32			82			132	
	33			83			133	
	34			84			134	
	35			85			135	
	36			86			136	
	37			87			137	
	38			88			138	
	39			89			139	
	40			90			140	
	41			91			141	
	42			92			142	
	43			93			143	
	44			94			144	
	45			95			145	
	46			96			146	
	47			97			147	
	48			98			148	
	49			99			149	
	50			100			150	

Search Notes	Application No. 09/703,942	Applicant(s) PULSIPHER ET AL.
	Examiner CHUONG T HO	Art Unit 2664

updated search

SEARCHED

Class	Subclass	Date	Examiner
370	254 ✓	02/16/05	CH
370	410	↓	↓
370	403 ✓		
370	402 ✓		
709	224		
714	717 ✓		
714	4		

INTERFERENCE SEARCHED

Class	Subclass	Date	Examiner

SEARCH NOTES (INCLUDING SEARCH STRATEGY)

	DATE	EXMR
STN EAST WEST IEEE	02/16/05 ↓	CH ↓
Consulting with Wellington Chin SPE AU 2664	02/18/05	CH

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

BIBDATASHEET

Bib Data Sheet

CONFIRMATION NO. 4047

SERIAL NUMBER	FILING DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
09/703,942	10/31/2000 RULE	370	2664	10008102-1

APPLICANTS

Eric A. Pulsipher, Ft Collins, CO;

Joseph R. Hunt, Loveland, CO;

** CONTINUING DATA ************************** NO CH

** FOREIGN APPLICATIONS ******************** NO CH

IF REQUIRED, FOREIGN FILING LICENSE GRANTED
** 02/05/2001

Foreign Priority claimed / 35 USC 119 (a-d) conditions met / Verified and Acknowledged	STATE OR COUNTRY	SHEETS DRAWING	TOTAL CLAIMS	INDEPENDENT CLAIMS
Foreign Priority claimed ☐ yes ☑ no 35 USC 119 (a-d) conditions met ☐ yes ☑ no ☐ Met after Allowance Verified and Acknowledged CH — Examiner's Signature, Initials	CO	26	20	3

ADDRESS
022879
HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS , CO
80527-2400

TITLE
Method and system for identifying and processing changes to a network topology

FILING FEE RECEIVED	FEES: Authority has been given in Paper No. ________ to charge/credit DEPOSIT ACCOUNT No. ________ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue)

710		☐ Other ____________ ☐ Credit

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:

Pulsipher, *et al.*

Serial No.: **09/703,942**

Filed: **October 31, 2000**

Confirmation No.: **4047**

Group Art Unit: **2664**

Examiner: **Ho, Chuong T.**

HP Docket No. **10008102-1**
TKHR Docket No. **050836-1530**

For: **METHOD AND SYSTEM FOR IDENTIFYING AND PROCESSING CHANGES TO A NETWORK TOPOLOGY**

RESPONSE TO FINAL OFFICE ACTION

Mail Stop: RCE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Sir:

The outstanding final Office Action mailed *February 23, 2005* (Part of Paper No. 9) has been carefully considered. In response thereto, please enter the following amendments in which claims 1-2, 5-9 and 15-20 are amended. Claims 1-3 and 5-20 are now pending in the present application. Reconsideration and allowance of the application and presently pending claims, as amended, are respectfully requested.

AUTHORIZATION TO DEBIT ACCOUNT

It is believed that no extensions of time or fees for net addition of claims are required, beyond those which may otherwise be provided for in documents accompanying this paper. However, in the event that additional extensions of time are necessary to allow consideration of this paper, such extensions are hereby petitioned under 37 C.F.R. § 1.136(a), and any fees required therefor (including fees for net addition of claims) are hereby authorized to be charged to deposit account no. 08-2025.

REQUEST FOR CONTINUED EXAMINATION

In accordance with 37 U.S.C. 1.114, a Request For Continued Examination (RCE) is filed concurrently with this Response To The Final Office Action so that the Office Action dated February 23, 2005 (Part of Paper No. 9) is effectively made non-final. Under 37 U.S.C. 1.114, the effect of the RCE, which makes the instant Office Action non-final, is to cause examination of the instant application to remain open. Accordingly, amendments submitted herein are to be entered as a matter of right, and *each* claim is entitled to continued examination, *particularly with respect to the responses provided herein.*

AMENDMENTS TO THE CLAIMS

Please amend the claims as indicated hereafter. [Use ~~strikethrough~~ for deleted matter (or double square brackets "[[]]" if the strikethrough is not easily perceivable, *i.e.*, "4" or a punctuation mark) and underlined for added matter.]

1. (Currently amended) In a network having interconnected nodes with data tuples that represent nodal connections, a method for mapping a network topology by identifying changes between an existing topology and a new topology, the method comprising:

creating a list of existing tuples from an existing topology representing nodal connections of a network at a prior time;

creating a new list of a plurality of tuples for a topology of [[a]] the network at a current time, wherein the new list of tuples represent nodal connections of the network at the current time, and wherein each of the tuples comprises a host identifier, interface information, and a port specification;

receiving new tuples list that represent new nodal connections; and

comparing the list of existing tuples with the new tuples list to identify changes to the topology.

2. (Currently amended) The method of claim 1, further comprising updating a topology database with a new topology corresponding to the list of existing tuples modified by the changes to the topology.

3. (Original) The method of claim 1, further comprising taking action on the changes to the topology.

4. (Canceled)

5. (Currently amended) The method of claim 1, wherein the step of ~~comparing~~ creating a new list of tuples comprises identifying duplicate tuples that appear both in the list of existing tuples and in the new tuples, and maintaining a current status of the topology for these tuples.

6. (Currently amended) The method of claim 1, wherein the step of ~~comparing~~ creating a new list of tuples comprises identifying a swapped port condition on a connector.

7. (Currently amended) The method of claim 1, wherein the step of ~~comparing~~ creating a new list of tuples comprises searching for a host of a new singly-heard host link tuple or a new multi-heard host link tuple in the list of existing tuples.

8. (Currently amended) A system for mapping a network topology by identifying changes between an existing topology and a new topology, based on changes to data tuples that represent nodal connections comprising:

a topology database that stores an existing topology of a network using tuples, wherein each tuple includes a host identifier, interface information, and a port specification for a node in the network from the existing topology representing nodal connections of the network at a prior time; and

a topology converter connected to the topology database the receives new tuples that represent new nodal connections for a topology of the network at a current time, ~~and~~ compares the new tuples with the existing ~~topology~~ existing tuples to identify changes in the network by comparing the host identifiers, the interface information, and the port specifications, and determines differences between the new tuples with the existing tuples representing nodal connections of the network at the prior time.

9. (Currently amended) The system of claim 8, wherein the topology converter creates the new tuples for the topology of the network.

10. (Original) The system of claim 8, wherein the topology converter updates the topology database with a new topology based on the new tuples.

11. (Original) The system of claim 8, wherein the topology converter attempts to identify swapped ports on connectors.

12. (Original) The system of claim 8, wherein the topology converter identifies duplicate tuples that appear both in the list of existing tuples and in the new tuples, and maintains a current status of the topology for these tuples.

13. (Original) The system of claim 8, wherein the topology converter searches for a host of a new singly-heard host link tuple or a new multi-heard host link tuple in the list of existing tuples.

14. (Original) The system of claim 8, wherein the topology converter searches for a connector of a new conflict links tuple in the list of existing tuples.

15. (Currently amended) A computer-readable medium having computer-executable instructions for performing a method for mapping a network topology by identifying changes between an existing topology and a new topology in a network having a interconnected nodes, the method comprising:

creating a list of existing tuples from an existing topology representing nodal connections of a network at a prior time;

creating a new list of a plurality of tuples for a topology of [[a]] the network at a current time, wherein the new list of tuples represent nodal connections of the network at the current time and wherein each of the tuples comprises a host identifier, interface information, and a port specification;

receiving new tuples list that represent new nodal connections;

comparing the list of existing tuples with the new tuples list to identify changes to the topology; and

updating a topology database with a new topology based on the comparing.

16. (Currently amended) The medium of claim 15, wherein a topology converter receives the new tuples list from a connection calculator that calculates connections between nodes.

17. (Currently amended) The medium of claim 15, wherein the step of ~~comparing~~ creating the new tuples list comprises identifying duplicate tuples that

appear both in the list of existing tuples and in the new tuples list, and maintaining a current status of the topology for these duplicate tuples.

18. (Currently amended) The medium of claim 15, wherein the step of ~~comparing~~ creating the new tuples list comprises identifying a swapped port condition on a connector.

19. (Currently amended) The medium of claim 15, wherein the step of ~~comparing~~ creating the new tuples list comprises searching for a host of a new singly-heard host link tuple or a new multi-heard link tuple in the list of existing tuples.

20. (Currently amended) The medium of claim 15, wherein the steps of ~~comparing~~ creating the new tuples list comprises searching for a connector of a new conflict links tuple in the list of existing tuples.

REMARKS

This is a full and timely response to the outstanding final Office Action mailed February 23, 2005. Reconsideration and allowance of the application and presently pending claims 1-3 and 5-20, as amended, are respectfully requested.

1. Present Status of Patent Application

Upon entry of the amendments in this response, claims 1-3 and 5-20 remain pending in the present application. More specifically, claims 1-2, 5-9 and 15-20 are directly amended. These amendments are specifically described hereinafter. It is believed that the foregoing amendments and additions add no new matter to the present application.

In accordance with 37 U.S.C. 1.114, a Request For Continued Examination is filed concurrently with this Response To The Final Office Action so that the Office Action mailed October 26, 2000 (Paper No. 8) is effectively made non-final.

2. Response to Rejection of Claims 1-3 and 5-20 Under 35 U.S.C. §103

In the Office Action, claims 1-3 and 5-20 stand rejected under 35 U.S.C. §103(a) as allegedly being unpatentable over *Liang et al.* (U.S. Patent 5,732,086), hereinafter *Liang*, in view of *Wood* (U.S. Patent 6,405,248). It is well-established at law that, for a proper rejection of a claim under 35 U.S.C. §103 as being obvious based upon a combination of references, the cited combination of references must disclose, teach, or suggest, either implicitly or explicitly, all elements/features/steps of the claim at issue. See, *e.g.*, *In Re Dow Chemical*, 5 U.S.P.Q.2d 1529, 1531 (Fed. Cir. 1988), and *In re Keller*, 208 U.S.P.Q.2d 871, 881 (C.C.P.A. 1981).

a. Independent Claims 1, 8 and 15

Applicants respectfully submit that independent claims 1, 8 and 15, as amended, are allowable for at least the reason that the proposed combination of *Liang* in view of *Wood* does not disclose, teach, or suggest at least the feature of "creating a list of existing tuples from an existing topology representing nodal connections of a network at a prior time" as recited in claims 1 and 15. Similarly, the proposed combination of *Liang* in view of *Wood* does not disclose, teach, or suggest at least a feature that "determines differences between the new tuples with the existing tuples representing nodal connections of the network at the prior time" as recited in claim 8.

Liang is apparently limited to, at most, a system where "the originating node constructs and stores a topology table entry which includes data from received ACK messages." (Abstract). That is, *Liang* is limited to generating the network topology at a current time. Thus, *Liang* fails to disclose, teach or suggest every element of the Applicants' claimed invention.

Wood also fails to disclose, teach or suggest at least the above recited features of independent claims 1, 8 and 15. As alleged in the Office Action, *Wood* is limited to teaching, at most, "creating an accurate topology map of a given network by: obtaining a list of managed network device; identifying link port and node port ... device interface information (see abstract)" (Office Action at page4). Thus, *Wood* fails to disclose, teach or suggest every element of the Applicants' claimed invention.

Accordingly, the proposed combination of *Liang* in view of *Wood* does not teach at least the claimed limitations of "creating a list of existing tuples from an existing topology representing nodal connections of a network at a prior time" as recited in claims 1 and 15, or a feature that "determines differences between the new tuples with the existing tuples representing nodal connections of the network at the prior time" as recited in claim 8. Therefore, a prima facie case establishing an obviousness rejection by *Liang* in view of *Wood* has not been made. Thus, claims 1, 8 and 15 are not obvious under proposed combination of *Liang* in view of *Wood*, and the rejection should be withdrawn.

Because independent claim 1 is allowable over the cited art of record, dependent claims 2-3 and 5-7 (which depend from independent claim 1) are allowable as a matter of law for at least the reason that dependent claims 2-3 and 5-7 contain all limitations of independent claim 1. Similarly, because independent claims 8 and 15 are allowable over the cited art of record, claims 9-14 and 16-20 (which depend from independent claims 8 and 15, respectively) are allowable as a matter of law for at least the reason that dependent claims 9-14 and 16-20 contain all limitations of their respective independent base claim. See, *e.g.*, *In re Fine*, 837 F.2d 1071 (Fed. Cir. 1988). Accordingly, the rejection to these claims should be withdrawn.

3. Observations Regarding *George*

In the Office Action mailed July 6, 2004, the claims were rejected under 35 U.S.C. §102 under an allegation that the claims were anticipated by *George et al.*, (U.S. Patent 4,644,532), hereinafter *George*. Applicants amended independent claims 1, 8 and 15 to distinguish the claims from *George*. These distinguishing amendments were sufficient to distinguish the claims, as shown by the new rejections under 35 U.S.C. §103(a) which allege all claims are unpatentable over *Liang* in view of *Wood*.

Applicants respectfully note that *George* is limited to a system that "upon determining changes in network status, such network status changes are communicated to the adjacent control nodes" (Abstract). *George* states that "the purpose for maintaining a topology data base in the control node and *continuous* identification of an ownership session ..." (Col. 14, lines 27-29, emphasis added). That is, *George* is *continuously* updating its topology database since "changes in resources adjacent to the NC are recorded locally by the NC, and ... it forwards the cause of its report (new resource, failed resource, or change in characteristic) and its local time stamp as a sequence identifier" (Col. 8, lines 35-44).

Applicants respectfully refer the Examiner to MPEP §2143.02, entitled "THE PROPOSED MODIFICATIONS CANNOT CHANGE THE PRINCIPLE OF OPERATION OF A REFERENCE." The MPEP states that "if the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims *prima facie* obvious."

If *George* is modified by *Liang*, *Wood* or another reference to arrive at embodiments of the present invention, then the principle of operation of *George* will change because, first, *George* will no longer be updating continuously, and second, because the NCs will not be the devices initiating changes in the network topology. Because the principle of operation of *George* after modification by *Liang*, *Wood* or another reference to arrive at embodiments of the present invention would be changed, a *prima facie* of obviousness cannot be established under any such possible scenario. Accordingly, the a rejection of the claims under 35 U.S.C. §103(a) cannot be properly established under a scenario wherein *George* is modified by *Liang*, *Wood* or another reference.

CONCLUSION

In light of the foregoing amendments and for at least the reasons set forth above, Applicants respectfully submit that all objections and/or rejections have been traversed, rendered moot, and/or accommodated, and that the now pending claims 1-3 and 5-20 are in condition for allowance. Favorable reconsideration and allowance of the present application and all pending claims are hereby courteously requested. If, in the opinion of the Examiner, a telephonic conference would expedite the examination of this matter, the Examiner is invited to call the undersigned agent at (770) 933-9500.

Respectfully submitted,

Raymond W. Armentrout
Reg. No. 45,866

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P. O. Box 272400
Fort Collins, Colorado 80527-2400

OIPE JC101 APR 1 1 2005 PATENT & TRADEMARK OFFICE

Rce 2664 2fw

PATENT APPLICATION

ATTORNEY DOCKET NO. 10008102-1

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Pulsipher, et al. **Confirmation No.:** 4047

Application No.: 09/703,942 **Examiner:** Ho, Chuong T.

Filing Date: 10/31/2000 **Group Art Unit:** 2664

Title: METHOD AND SYSTEM FOR IDENTIFYING AND PROCESSING CHANGES TO A NETWORK TOPOLOGY

Mail Stop RCE
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

REQUEST FOR CONTINUED EXAMINATION (RCE) 37 CFR 1.114

Subsection (b) of 35 U.S.C. 132, effective on May 29, 2000, provides for continued examination of an utiltity or plant application filed on or after June 8, 1995.
See The American Inventors Protection Act of 1999 (AIPA).

Sir:

This is a Request for Continued Examination (RCE) under CFR 1.114 of the above-identified applicaiton.

NOTE: *37 CFR 1.114 is effective on May 20, 2000. If the above- application was filed prior to May 29, 2000, applicant may wish to consider filing a continued prosecution application (CPA) under CFR 1.53(d) (PTO/SB/29) instead of a RCE to be eligible for the patent term adjustment provisions of the AIPA. See Changes to Application Examination and Provisional Application Practice, Interim Rule, 65 Fed. Reg. 14865 (Mar. 20, 2000), 1233 off. Gaz. Pat. Office 47 (Apr. 11, 2000), which established RCE practice.*

Submission under 37 CFR 1.114

( ) Previously submitted:
- ( ) Consider the amendment(s)/reply under 37 CFR 1.116 previously filed on ______________.
  (Any unentered amendment(s) referred to above will be entered).
- ( ) Consider the arguments in the Appeal Brief or Reply Brief previously filed on ______________.
- ( ) Other ______________.

(X) Enclosed:
- **(X)** Amendment/Reply
- ( ) Affidavit(s)/Declarations(s)
- ( ) Information Disclosure Statement (IDS)
- ( ) Other ______________.

Miscellaneous

- ( ) Suspension of action is requested under 37 CFR 1.103(c) for a period of ________ months.
  The fee for this Suspension is (37 CFR 1.17(i)) $130.00
- ( ) Other ______________.

04/12/2005 AWONDAF1 00000164 082025 09703942

01 FC:1801 790.00 DA

(X) RCE filing fee $790.00

( ) A Petition for Extension of Time

( )	one month	$120.00
( )	two months	$450.00
( )	three months	$1020.00
( )	four months	$1590.00

Please charge to Deposit Account **08-2025** the sum of **$790.00**. At any time during the pendency of this application, please charge any fees required or credit any overpayment to Deposit Account **08-2025** pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees.

(X) A duplicate copy of this transmittal letter is enclosed.

(X) I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313-1450.
Date of Deposit: 04-08-05

OR

( ) I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number
on ______________ ______________

Number of pages:

Typed Name: **Marianne Boland**

Signature: *Marianne Boland*

Respectfully submitted,

Pulsipher, et al.

By *[signature]*

Raymond W. Armentrout
Attorney/Agent for Applicant(s)
Reg. No. **45,866**

Date: **04-08-05**

Telephone No.: **(770) 933-9500**

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P. O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 10008102-1

OIPE JC101 APR 1 1 2005 PATENT & TRADEMARK OFFICE

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Pulsipher, et al. **Confirmation No.:** 4047

Application No.: 09/703,942 **Examiner:** Ho, Chuong T.

Filing Date: 10/31/2000 **Group Art Unit:** 2664

Title: METHOD AND SYSTEM FOR IDENTIFYING AND PROCESSING CHANGES TO A NETWORK TOPOLOGY

Mail Stop RCE
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL LETTER FOR RESPONSE/AMENDMENT

Sir:

Transmitted herewith is/are the following in the above-identified application:

- (X) Response/Amendment
- ( ) New fee as calculated below
- (X) No additional fee
- (X) Other: Request For Continued Examination (fee $ 790.00)
- ( ) Petition to extend time to respond
- ( ) Supplemental Declaration

CLAIMS AS AMENDED BY OTHER THAN A SMALL ENTITY						
(1) FOR	(2) CLAIMS REMAINING AFTER AMENDMENT	(3) NUMBER EXTRA	(4) HIGHEST NUMBER PREVIOUSLY PAID FOR	(5) PRESENT EXTRA	(6) RATE	(7) ADDITIONAL FEES
TOTAL CLAIMS	19	MINUS	20	= 0	X $50	$ 0
INDEP. CLAIMS	3	MINUS	3	= 0	X $200	$ 0
[ ] FIRST PRESENTATION OF A MULTIPLE DEPENDENT CLAIM					+ $360	$ 0
EXTENSION FEE	1ST MONTH $120.00 [ ]	2ND MONTH $450.00 [ ]	3RD MONTH $1020.00 [ ]	4TH MONTH $1590.00 [ ]		$ 0
					OTHER FEES	$ 790
					TOTAL ADDITIONAL FEE FOR THIS AMENDMENT	$ 790

Charge $ 790 to Deposit Account 08-2025. At any time during the pendency of this application, please charge any fees required or credit any overpayment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees. A duplicate copy of this sheet is enclosed.

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313-1450.

Date of Deposit: **04-08-05**

Typed Name: **Marianne Boland**

Signature: Marianne Boland

Respectfully submitted,

Pulsipher, et al.

By Raymond W. Armentrout

Raymond W. Armentrout

Attorney/Agent for Applicant(s)
Reg. No. **45,866**

Date: **04-08-05**

Telephone No.: **(770) 933-9500**

Rev 12/04 (TransAmd)

- Attach as First Page to Transmitted Papers -

Application or Docket Number 09/703,942

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2000

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	20	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	20 minus 20=	* 0
INDEPENDENT CLAIMS	3 minus 3 =	* 0
MULTIPLE DEPENDENT CLAIM PRESENT		☐

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE ☐ OR OTHER THAN SMALL ENTITY

RATE	FEE	OR	RATE	FEE
BASIC FEE	355.00	OR	BASIC FEE	710.00
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL		OR	TOTAL	

CLAIMS AS AMENDED - PART II

SMALL ENTITY OR OTHER THAN SMALL ENTITY

AMENDMENT A

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 19	Minus	** 20	= 0
Independent	* 3	Minus	*** 3	= 0
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE	OR	RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

4.11.5

AMENDMENT B

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 19	Minus	** 20	= 0
Independent	* 3	Minus	*** 3	= 0
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE	OR	RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE	OR	RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

FORM PTO-875
(Rev. 8/00)

Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE
*U.S. GPO: 2000-460-705/30103

Refine Search

Search Results -

Terms	Documents
(host near1 (ID or identifier$)) and interface$ and port$ and L19	1

Database:

US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search:

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Tuesday, June 14, 2005 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=ADJ		
L20	(host near1 (ID or identifier$)) and interface$ and port$ and L19	1	L20
L19	L18 and L1	14	L19
L18	updat$6 same (database$ or data-base$ or (data near1 base$) or table$ or list$)	51663	L18
L17	L16 and (host$ same interface$ same port$)	3	L17
L16	updat$6 same L15	73	L16
L15	topology near1 network$6 near1 (database$ or data-base$ or (data near1 base$) or table$ or list$)	175	L15
	DB=PGPB,USPT; PLUR=YES; OP=ADJ		
L14	L13 and L5	1	L14
L13	updat$ same L12	95	L13
L12	topology near1 network$6 near1 (database$ or data-base$ or (data near1 base$) or table$ or list$)	278	L12

L11	(host$ same interface$ same port$) and L1	1	L11
L10	(host$ same interface$ same port$) and L4	16480	L10
L9	(host$ same interface$ same port$) and L2	44	L9
L8	(host$ same interface$ same port$) same L2	1	L8
L7	L3 and L5	0	L7
L6	L2 same L5	0	L6
L5	(host near1 (ID or identifier$)) same interface$ same port$	65	L5
L4	(host near1 (ID or identifer$)) same interface$ or port$	3122675	L4
L3	L1 and L2	3	L3
L2	(updat$6 or chang$6) same (topology near1 (database$ or data-base$ or (data near1 base$) or table$ or list$))	416	L2
L1	5732086.uref.	24	L1

END OF SEARCH HISTORY

Refine Search

Search Results -

Terms	Documents
compar$6 same L63	5

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search:

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Thursday, June 16, 2005 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=ADJ		
L65	compar$6 same L63	5	L65
L64	L59 and L63	3	L64
L63	(list$ or table$ or database$ or (data near1 base$) or tuple$) same L62	61	L63
L62	((host or node) near1 (ID$ or identifier$)) same interface$ same port$	241	L62
L61	creat$6 same compar$6 same L50	4	L61
L60	L58 same L59	6	L60
L59	creat$6 same compar$6 same (list$ or table$ or database$ or (data near1 base$) or tuple$)	8341	L59
L58	(network$6 near1 topology) same (list$ or table$ or database$ or (data near1 base$) or tuple$)	1131	L58
L57	creat$6 near4 L50	51	L57
L56	compar$6 and L55	4	L56
L55	L52 and L53	4	L55

L54	L53 and L53	9708	L54
L53	(old or exist$6 or previous) near1 (list$ or table$ or database$ or (data near1 base$) or tuple$)	9708	L53
L52	new near1 L50	14	L52
L51	(creat$6 or compar$6) same L50	140	L51
L50	topology near1 (list$ or table$ or database$ or (data near1 base$) or tuple$)	509	L50
L49	topology near4 (chang$6 or updat$6 or map$6)	2356	L49
L48	L45 and L44	6	L48
L47	L44 same ((host or node) near1 (ID or identifier$)) same interface$ same port$	0	L47
L46	L45 same L44	1	L46
L45	topology near1 (chang$6 or updat$6)	1081	L45
L44	compar$6 same L43	765	L44
L43	new near1 (list$ or table$ or database$ or (data near1 base$) or tuple$)	7986	L43
L42	(list$ or table$ or database$ or (data near1 base$) or tuple$)	3623936	L42
L41	(topology near4 chang$6) and l39	2	L41
L40	(list$ or table$ or database$ or (data near1 base$) or tuple$) and L39	15	L40
L39	6108702.uref.	17	L39
L38	5751965.pn.	1	L38
L37	5751965.pn.	1	L37
L36	5754532.pn.	1	L36
L35	5793362.pn.	1	L35
L34	5809286.pn.	1	L34
L33	5812750.pn.	1	L33
L32	5822305.pn.	1	L32
L31	5825772.pn.	1	L31
L30	5905859.pn.	1	L30
L29	6108702.pn.	1	L29
L28	6108702.pn.	1	L28
L27	6167052.pn.	1	L27
L26	6205147.pn.	1	L26
L25	L24 and 6405248.pn.	0	L25
L24	compar$6 near4 (list$ or table$ or database$ or (data near1 base$) or tuple$)	88833	L24
L23	L22 and 6405248.pn.	1	L23
L22	(list$ or table$ or database$ or (data near1 base$) or tuple$) same compar$6	313893	L22
L21	compar$6 and 6405248.pn.	1	L21
L20	topology same L19	4	L20
L19	L17 same L18	1230	L19
L18	new near4 (list$ or table$ or database$ or (data near1 base$) or tuple$)	32608	L18

L17	(existing or old or previous) near1 (list$ or table$ or database$ or (data near1 base$) or tuple$)	6679	L17
L16	(existing or old or previous) near1 (list$ or table$ or database$ or (data near1 base$))	6642	L16
L15	(5732086.pn. or 5732086.uref.) and L11	3	L15
L14	(5732086.pn. or 5732086.uref.) and L2	1	L14
L13	L12 and (L1 or L4)	0	L13
L12	compar$6 same L11	1018	L12
L11	creat$6 near1 (list$ or table$ or database$ or (data near1 base$))	13280	L11
L10	L4 and L9	7	L10
L9	(new near1 (topology or tuple$)) or L2	8242	L9
L8	L4 and L7	0	L8
L7	new near1 tuple$	124	L7
L6	L4 and L2	5	L6
L5	L4 and new	34	L5
L4	5886643.uref. or 5727157.uref.	42	L4
L3	L1 and L2	0	L3
L2	new near1 (list$ or table$ or database$ or (data near1 base$))	7882	L2
L1	5886643.pn. or 5727157.pn.	2	L1

END OF SEARCH HISTORY

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/703,942	10/31/2000	Eric A. Pulsipher	10008102-1	4047

22879 7590 06/28/2005

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
HO, CHUONG T	
ART UNIT	**PAPER NUMBER**
2664	

DATE MAILED: 06/28/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Office Action Summary	Application No.	Applicant(s)	
	09/703,942	PULSIPHER ET AL.	
	Examiner	Art Unit	
	CHUONG T. HO	2664	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11 April 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-3 and 5-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) *1-3,5-7 and 15-20* is/are allowed.

6)☒ Claim(s) *8 and 10-14* is/are rejected.

7)☒ Claim(s) *9* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

1. The amendment filed 04/11/05 have been entered and made of record.

2. Applicant's amendment filed 04/11/05 with respect to claims 1-3, 5-20 have been considered but are moot in view of the new ground(s) of rejection.

Response to Arguments

3. Applicant's arguments filed 04/11/05 have been fully considered but they are not persuasive.

In the page 7, lines 26-28, the applicant alleged "the proposed combination of Liang (U.S.Patent No. 5,732,086) in view of Wood (U.S.Patent No. 6,405,248) does not disclose, teach, or suggest at least the features of "creating a list of existing tuples from an existing topology representing nodal connections of a network at a prior time".

The Applicant's argument is not persuasive.

Liang (U.S.Patent No. 5,732,086) discloses "creating (updating) a list of existing tuples from an existing topology (its topology database) representing nodal connections of a network at a prior time" (see abstract, the originating node constructs and stores a topology table entry which includes data from received ACK messages. Each entry includes a node identifier , an originating node link identifier and a neighbor node identifier from which an ACK message was received and a neighbor node link identifier for the link) (see col. 5, lines 55-57, An UPDATE message includes the same first five fields listed above for the ACK message as well as a field which contains the entire "topology table row" for the node originating an UPDATE message) (see col. 8, lines 13-15, the "Adjusting" state means that the node is receiving topology update information from other nodes and is merging the changes into its topology table) (see col. 7, lines

20-24, to the extent that other nodes receive duplicate information, that information is discarded and the row data is updated accordingly. Each topology table can further updated during run time through use of the Update message).

Wood (U.S.Patent No. 6,405,248) discloses "creating a list of existing tuples from an existing topology (topology database) representing nodal connections of a network at a prior time" (see abstract, creating an accurate topology map of a given network by: obtaining a list of managed network device....Filters may then be utilized on the source address tables in order to provide more accurate topology results. Connections between nodes are also resolved by utilizing sorting methods).

Therefore, "the proposed combination of Liang (U.S.Patent No. 5,732,086) in view of Wood (U.S.Patent No. 6,405,248) disclose, teach, or suggest at least the features of "creating a list of existing tuples from an existing topology representing nodal connections of a network at a prior time"

4. Claims 1-3, 5-20 are pending.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 8, 10-14 are rejected under 35 U.S.C. 103(a) as being unpatentable over Liang et al. (U.S.Patent No. 5,732,086) in view of Wood (U.S.Patent No. 6,405,248 B1).

In the claim 8, see figure 3, Liang et al. discloses topology table 26 (tuples) which comprises: node identifier (host identifier), link port (port sepecification) (see col. 5, lines 7-15, lines 21-25, col. 6, lines 5-9, col. 7, lines 20-23, col. 14-16); comprising:

A topology database that stores an existing topology of a network using tuples (topology table 26, see figure 3) , wherein each tuple includes a host identifier, port specification for a node in the network from the existing topology representing nodal connections of the network at a prior time (see col. 5, lines 7-15, lines 21-25, col. 6, lines 5-9, col. 7, lines 20-23, col. 14-16);

A topology converter connected to the topology database that receives new tuples that represent new nodal connections, and compare the new tuples with the existing topology to identify changes in the network by comparing the host identifier, port specification, and determines differences between the new tuples with the existing tuples representing nodal connections of the network at the prior time (see col. 7, lines 20-23) (see col. 8, lines 13-16);

However, Liang et al. is silent to disclosing tuples comprising interface information.

Wood discloses creating an accurate topology map of a given network by: obtaining a list of managed network device; identifying link port and node port....device interface information (see abstract); comprising:

Each of the tuples (topology map) comprises interface information, a port specification, host identifier (see abstract, table 1, col. 6, lines 10-20).

Liang and Wood are directed to routing packets using IP protocol. Wood recognizes that there may be conflicting source address. Hence, it would have been obvious to use the interface information of Wood in Liang as it would have resolved source address conflicts when routing using the table (tuples) of both Liang and Wood.

6. In the claim 10, Liang et al. discloses updating a topology database with a new topology (see col. 7, lines 20-23).

7. In the claims 11, 12, Liang et al. discloses the step of comparing comprises identifying duplicate tuples that appear both in the list of existing tuples and in the new tuples, and maintaining a current status of the topology for these tuples (see col. 7, lines 20-23).

8. In the claims 13, 14, Liang et al. discloses the step of comparing comprises searching for a host of a new singly-heard host link tuple or a new multi-heard host link tuple in the list of existing tuples (see col. 7, lines 20-23).

Allowable Subject Matter

9. Claims 1-3, 5-7, 15-20 are allowed.

Claim 9 is objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

The following is an examiner's statement of reasons for allowance: the prior art (5732086, 6405248, 4644532, 5727157, 6160796) of record does not appear to teach or render obvious the claimed limitations in combinations with the specific added limitations, as recited from independent claims 1, 15: "creating a new list of a plurality of tuples for a topology of the network at a current time, wherein the new list of tuples represent nodal connections of the network at the current time".

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to CHUONG T HO whose telephone number is (571) 272-3133. The examiner can normally be reached on 8:00 am to 4:00 pm.

The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

06/16/05

WELLINGTON CHIN
SUPERVISORY PATENT EXAMINER

Notice of References Cited	Application/Control No. 09/703,942	Applicant(s)/Patent Under Reexamination PULSIPHER ET AL.	
	Examiner CHUONG T. HO	Art Unit 2664	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-5,732,086	03-1998	Liang et al.	370/410
	B	US-6,405,248	06-2002	Wood, Michael	709/223
	C	US-4,644,532	02-1987	George et al.	370/255
	D	US-5,727,157	03-1998	Orr et al.	709/224
	E	US-6,160,796	12-2000	Zou, Feng (Frank)	370/257
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Index of Claims

Application/Control No.	Applicant(s)/Patent under Reexamination
09/703,942	PULSIPHER ET AL.
Examiner	**Art Unit**
CHUONG T. HO	2664

√	Rejected	–	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	+	Restricted	I	Interference	O	Objected

Final	Original	06/16/05	Final	Original	Date	Final	Original	Date
	(1)	=		51			101	
	2	=		52			102	
	3	=		53			103	
	~~4~~			54			104	
	5			55			105	
	6			56			106	
	7			57			107	
	(8)	√		58			108	
	9	O		59			109	
	10	√		60			110	
	11	√		61			111	
	12	√		62			112	
	13	√		63			113	
	14	√		64			114	
	(15)	=		65			115	
	16	=		66			116	
	17	=		67			117	
	18	=		68			118	
	19	=		69			119	
	20	=		70			120	
	21			71			121	
	22			72			122	
	23			73			123	
	24			74			124	
	25			75			125	
	26			76			126	
	27			77			127	
	28			78			128	
	29			79			129	
	30			80			130	
	31			81			131	
	32			82			132	
	33			83			133	
	34			84			134	
	35			85			135	
	36			86			136	
	37			87			137	
	38			88			138	
	39			89			139	
	40			90			140	
	41			91			141	
	42			92			142	
	43			93			143	
	44			94			144	
	45			95			145	
	46			96			146	
	47			97			147	
	48			98			148	
	49			99			149	
	50			100			150	

Search Notes	Application/Control No. 09/703,942	Applicant(s)/Patent under Reexamination PULSIPHER ET AL.	
	Examiner CHUONG T. HO	Art Unit 2664	

updated search

SEARCHED

Class	Subclass	Date	Examiner
370	254 ✓	06/16/05	CH
370	410 ✓	↓	↓
370	403 ✓		
370	402 ✓		
709	224 ✓		
714	717		
714	4		
709	223 ✓		
709	238 ✓		
370	350 ✓		
370	255		
370	400		
340	825.52		
370	254 ✓	↓	↓

INTERFERENCE SEARCHED

Class	Subclass	Date	Examiner

SEARCH NOTES (INCLUDING SEARCH STRATEGY)

	DATE	EXMR
STN EAST WEST IEEE	06/08/05 ↓	CH ↓
Consulting with Wellington Chin SPE AU 2664	06/16/05	CH

U.S. Patent and Trademark Office

Part of Paper No. 11

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P. O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 10008102-1

2664

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

OIPE IAP52 SEP 0 8 2005 PATENT & TRADEMARK OFFICE

Inventor(s): Pulsipher, et al.

Application No.: 09/703,942

Filing Date: 10/31/00

Confirmation No.: 4047

Examiner: Chuong T. Ho

Group Art Unit: 2664

Title: Method and System for Identifying and Processing Changes to a Network Topology

Mail Stop Amendment
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL LETTER FOR RESPONSE/AMENDMENT

Sir:

Transmitted herewith is/are the following in the above-identified application:

(X) Response/Amendment
( ) New fee as calculated below
(X) No additional fee
( ) Other: ____________________ (fee $__________)
( ) Petition to extend time to respond
( ) Supplemental Declaration

CLAIMS AS AMENDED BY OTHER THAN A SMALL ENTITY

(1) FOR	(2) CLAIMS REMAINING AFTER AMENDMENT	(3) NUMBER EXTRA	(4) HIGHEST NUMBER PREVIOUSLY PAID FOR	(5) PRESENT EXTRA	(6) RATE	(7) ADDITIONAL FEES
TOTAL CLAIMS	18	MINUS	20	= 0	× $50	$ 0
INDEP. CLAIMS	3	MINUS	3	= 0	× $200	$ 0
[ ] FIRST PRESENTATION OF A MULTIPLE DEPENDENT CLAIM					+ $360	$ 0
EXTENSION FEE	1ST MONTH $120.00	2ND MONTH $450.00	3RD MONTH $1020.00	4TH MONTH $1590.00		$ 0
					OTHER FEES	$
					TOTAL ADDITIONAL FEE FOR THIS AMENDMENT	$ 0

Charge $ 0 to Deposit Account 08-2025. At any time during the pendency of this application, please charge any fees required or credit any overpayment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees. A duplicate copy of this sheet is enclosed.

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313-1450.

Date of Deposit: **8/31/05**

Typed Name: **Stephanie Riley**

Signature: *Stephanie Riley*

Respectfully submitted,

Pulsipher, et al.

By *[signature]*

M. Paul Qualey, Jr.

Attorney/Agent for Applicant(s)
Reg. No. **43,024**

Date: **8/31/05**

Telephone No.: **(770) 933-9500**

Rev 12/04 (TransAmd)

- Attach as First Page to Transmitted Papers -

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:

Pulsipher, *et al.*

Serial No.: **09/703,942**

Filed: **October 31, 2000**

Confirmation No.: **4047**

Group Art Unit: **2664**

Examiner: **Ho, Chuong T.**

HP Docket No. **10008102-1**
TKHR Docket No. **050836-1530**

For: **METHOD AND SYSTEM FOR IDENTIFYING AND PROCESSING CHANGES TO A NETWORK TOPOLOGY**

RESPONSE TO OFFICE ACTION

Mail Stop: Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Sir:

The outstanding non-final Office Action mailed June 28, 2005 has been carefully considered. In response thereto, please enter the following amendments an remarks

AUTHORIZATION TO DEBIT ACCOUNT

It is believed that no extensions of time or fees for net addition of claims are required, beyond those which may otherwise be provided for in documents accompanying this paper. However, in the event that additional extensions of time are necessary to allow consideration of this paper, such extensions are hereby petitioned under 37 C.F.R. § 1.136(a), and any fees required therefor (including fees for net addition of claims) are hereby authorized to be charged to deposit account no. 08-2025.

AMENDMENTS TO THE CLAIMS

Please amend the claims as indicated hereafter, wherein ~~strikethrough~~ indicates deleted matter and underlined indicates added matter:

1. (Previously Presented) In a network having interconnected nodes with data tuples that represent nodal connections, a method for mapping a network topology by identifying changes between an existing topology and a new topology, the method comprising:

creating a list of existing tuples from an existing topology representing nodal connections of a network at a prior time;

creating a new list of a plurality of tuples for a topology of the network at a current time, wherein the new list of tuples represent nodal connections of the network at the current time, and wherein each of the tuples comprises a host identifier, interface information, and a port specification;

receiving new tuples list that represent new nodal connections; and

comparing the list of existing tuples with the new tuples list to identify changes to the topology.

2. (Previously Presented) The method of claim 1, further comprising updating a topology database with a new topology corresponding to the list of existing tuples modified by the changes to the topology.

3. (Original) The method of claim 1, further comprising taking action on the changes to the topology.

4. (Canceled)

5. (Previously Presented) The method of claim 1, wherein the step of creating a new list of tuples comprises identifying duplicate tuples that appear both in the list of existing tuples and in the new tuples, and maintaining a current status of the topology for these tuples.

6. (Previously Presented) The method of claim 1, wherein the step of creating a new list of tuples comprises identifying a swapped port condition on a connector.

7. (Previously Presented) The method of claim 1, wherein the step of creating a new list of tuples comprises searching for a host of a new singly-heard host link tuple or a new multi-heard host link tuple in the list of existing tuples.

8. (Currently Amended) A system for mapping a network topology by identifying changes between an existing topology and a new topology, based on changes to data tuples that represent nodal connections comprising:

a topology database that stores an existing topology of a network using tuples, wherein each tuple includes a host identifier, interface information, and a port specification for a node in the network from the existing topology representing nodal connections of the network at a prior time; and

a topology converter connected to the topology database the receives new tuples that represent new nodal connections for a topology of the network at a current time, compares the new tuples with the ~~existing~~ existing tuples to identify changes in the network by comparing the host identifiers, the interface information, and the port specifications, and determines differences between the new tuples with the existing tuples representing nodal connections of the network at the prior time,

wherein the topology converter creates the new tuples for the topology of the network.

9. (Canceled)

10. (Original) The system of claim 8, wherein the topology converter updates the topology database with a new topology based on the new tuples.

11. (Original) The system of claim 8, wherein the topology converter attempts to identify swapped ports on connectors.

12. (Original) The system of claim 8, wherein the topology converter identifies duplicate tuples that appear both in the list of existing tuples and in the new tuples, and maintains a current status of the topology for these tuples.

13. (Original) The system of claim 8, wherein the topology converter searches for a host of a new singly-heard host link tuple or a new multi-heard host link tuple in the list of existing tuples.

14. (Original) The system of claim 8, wherein the topology converter searches for a connector of a new conflict links tuple in the list of existing tuples.

15. (Previously Presented) A computer-readable medium having computer-executable instructions for performing a method for mapping a network topology by identifying changes between an existing topology and a new topology in a network having a interconnected nodes, the method comprising:

creating a list of existing tuples from an existing topology representing nodal connections of a network at a prior time;

creating a new list of a plurality of tuples for a topology of the network at a current time, wherein the new list of tuples represent nodal connections of the network at the current time and wherein each of the tuples comprises a host identifier, interface information, and a port specification;

receiving new tuples list that represent new nodal connections;

comparing the list of existing tuples with the new tuples list to identify changes to the topology; and

updating a topology database with a new topology based on the comparing.

16. (Previously Presented) The medium of claim 15, wherein a topology converter receives the new tuples list from a connection calculator that calculates connections between nodes.

17. (Previously Presented) The medium of claim 15, wherein the step of creating the new tuples list comprises identifying duplicate tuples that appear both in the list of existing tuples and in the new tuples list, and maintaining a current status of the topology for these duplicate tuples.

18. (Previously Presented) The medium of claim 15, wherein the step of creating the new tuples list comprises identifying a swapped port condition on a connector.

19. (Previously Presented) The medium of claim 15, wherein the step of creating the new tuples list comprises searching for a host of a new singly-heard host link tuple or a new multi-heard link tuple in the list of existing tuples.

20. (Previously Presented) The medium of claim 15, wherein the steps of creating the new tuples list comprises searching for a connector of a new conflict links tuple in the list of existing tuples.

REMARKS

This is a full and timely response to the outstanding final Office Action mailed June 28, 2005. Upon entry of the amendments in this response, claims 1 – 3, 5 – 8 and 10 - 20 remain pending. In particular, Applicants have amended claim 8, and have canceled claim 9 without prejudice, waiver, or disclaimer. Applicants have canceled claim 9 merely to reduce the number of disputed issues and to facilitate early allowance and issuance of other claims in the present application. Applicants reserve the right to pursue the subject matter of this canceled claim in a continuing application, if Applicants so choose, and do not intend to dedicate the canceled subject matter to the public. Reconsideration and allowance of the application and presently pending claims are respectfully requested.

Indication of Allowable Subject Matter

The Office Action indicates that claims 1 – 3, 5 – 7 and 15 – 20 are allowed. The Office Action also indicates that claim 9 is objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims. As set forth above, Applicants have amended claim 8 to include the limitations previously recited in claim 9. Therefore, Applicant respectfully asserts that claim 8 and its dependent claims 10 – 14 are in condition for allowance.

Rejections Under 35 U.S.C. §103

The Office Action indicates that claims 8 and 10 - 14 stand rejected under 35 U.S.C. 103(a) as being unpatentable over *Liang* in view of *Wood*. As set forth above, Applicants have amended claim 8 to include the limitations previously recited in claim 9, the allowability of which is set forth in the Office Action. Therefore, Applicant respectfully asserts that claim 8 is in condition for allowance.

Since claims 10 - 14 are dependent claims that incorporate all the features/limitations of claim 8, Applicants respectfully assert that these claims also are in condition for allowance. Additionally, these claims recite other features/limitations that can serve as an independent basis for patentability.

Cited Art Made of Record

The cited art made of record has been considered, but is not believed to affect the patentability of the presently pending claims.

CONCLUSION

In light of the foregoing amendments and for at least the reasons set forth above, Applicants respectfully submit that all objections and/or rejections have been traversed, rendered moot, and/or accommodated, and that the pending claims are in condition for allowance. Favorable reconsideration and allowance of the present application and all pending claims are hereby courteously requested. If, in the opinion of the Examiner, a telephonic conference would expedite the examination of this matter, the Examiner is invited to call the undersigned agent at (770) 933-9500.

Respectfully submitted,

M. Paul Qualey, Jr.
Reg. No. 43,024

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2000

Application or Docket Number: 09/703,942

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	20	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	20 minus 20=	* 0
INDEPENDENT CLAIMS	3 minus 3 =	* 0
MULTIPLE DEPENDENT CLAIM PRESENT		☐

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE ☐ OR OTHER THAN SMALL ENTITY

RATE	FEE		RATE	FEE
BASIC FEE	355.00	OR	BASIC FEE	710.00
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL		OR	TOTAL	

CLAIMS AS AMENDED - PART II

AMENDMENT A

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 19	Minus	** 20	= 0
Independent	* 3	Minus	*** 3	= 0
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

SMALL ENTITY OR OTHER THAN SMALL ENTITY

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 19	Minus	** 20	= 0
Independent	* 3	Minus	*** 3	= 0
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

9-21-05

AMENDMENT C

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 18	Minus	** 20	=
Independent	* 3	Minus	*** 3	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X40=		OR	X80=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

FORM PTO-875 (Rev. 8/00)

Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE
*U.S. GPO: 2000-460-702/00103

BEST AVAILABLE COPY

Refine Search

Search Results -

Terms	Documents
L16 and L2	0

Database:

US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search:

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Friday, September 23, 2005 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=PGPB,USPT; PLUR=YES; OP=ADJ		
L17	L16 and L2	0	L17
L16	L1 and (L12 or L13 or L14 or L15)	11	L16
L15	714/4.ccls.	1349	L15
L14	714/717.ccls.	86	L14
L13	709/223-224.ccls.	7438	L13
L12	370/254.ccls. or 370/402-403.ccls. or 370/351.ccls.	2482	L12
L11	L9 and L10	1	L11
L10	(new near1 (tuple$ or list$ or database$ or table$))	14210	L10
L9	L8 and L5	4	L9
L8	topology near1 (table$ or database$ or list$)	889	L8
L7	L2 and L5	0	L7
L6	L1 and L5	0	L6
L5	5732086.uref. or 6405248.uref.	33	L5
L4	L2 same L1	0	L4

L3	L1 same L2	0	L3
L2	(nodal near1 (connect$6 or connection)) same topology	15	L2
L1	creat$6 near4 (new near1 (tuple$ or list$))	564	L1

END OF SEARCH HISTORY

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

022879 7590 09/30/2005

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
HO, CHUONG T	
ART UNIT	**PAPER NUMBER**
2664	

DATE MAILED: 09/30/2005

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/703,942	10/31/2000	Eric A. Pulsipher	10008102-1	4047

TITLE OF INVENTION: METHOD AND SYSTEM FOR IDENTIFYING AND PROCESSING CHANGES TO A NETWORK TOPOLOGY

APPLN. TYPE	SMALL ENTITY	ISSUE FEE	PUBLICATION FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1400	$0	$1400	12/30/2005

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATEN <u>PROSECUTION ON THE MERITS IS CLOSED</u>. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHT THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPO PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN <u>THREE MONTHS</u> FROM TH MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. <u>TH STATUTORY PERIOD CANNOT BE EXTENDED</u>. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOV REFLECTS A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE APPLIED IN THIS APPLICATION. THE PTOL-85B (O AN EQUIVALENT) MUST BE RETURNED WITHIN THIS PERIOD EVEN IF NO FEE IS DUE OR THE APPLICATION WIL BE REGARDED AS ABANDONED.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.

B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

A. Pay TOTAL FEE(S) DUE shown above, or

B. If applicant claimed SMALL ENTITY status before, or is n claiming SMALL ENTITY status, check box 5a on Part B - Fee Transmittal and pay the PUBLICATION FEE (if required) and 1 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL should be completed and returned to the United States Patent and Trademark Office (USPTO) w your ISSUE FEE and PUBLICATION FEE (if required). Even if the fee(s) have already been paid, Part B - Fee(s) Transmittal should completed and returned. If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should completed and an extra copy of the form should be submitted.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PTOL 85 (Rev 07/05) Approved for use through 04/30/2007

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

or Fax (571) 273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed wh appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

022879 7590 09/30/2005

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

Note: A certificate of mailing can only be used for domestic mailings of Fee(s) Transmittal. This certificate cannot be used for any other accompany papers. Each additional paper, such as an assignment or formal drawing, m have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the Un States Postal Service with sufficient postage for first class mail in an envel addressed to the Mail Stop ISSUE FEE address above, or being facsim transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's na

(Signat

(D

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/703,942	10/31/2000	Eric A. Pulsipher	10008102-1	4047

TITLE OF INVENTION: METHOD AND SYSTEM FOR IDENTIFYING AND PROCESSING CHANGES TO A NETWORK TOPOLOGY

APPLN. TYPE	SMALL ENTITY	ISSUE FEE	PUBLICATION FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1400	$0	$1400	12/30/2005

EXAMINER	ART UNIT	CLASS-SUBCLASS
HO, CHUONG T	2664	370-254000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
   - ☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
   - ☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
   (1) the names of up to 3 registered patent attorneys or agents OR, alternatively,
   (2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.
   1 ____________
   2 ____________
   3 ____________

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE

(B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ☐ Individual ☐ Corporation or other private group entity ☐ Governm

4a. The following fee(s) are enclosed:
- ☐ Issue Fee
- ☐ Publication Fee (No small entity discount permitted)
- ☐ Advance Order - # of Copies ____________

4b. Payment of Fee(s):
- ☐ A check in the amount of the fee(s) is enclosed.
- ☐ Payment by credit card. Form PTO-2038 is attached.
- ☐ The Director is hereby authorized by charge the required fee(s), or credit any overpayment Deposit Account Number ____________ (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)
- ☐ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.
- ☐ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

The Director of the USPTO is requested to apply the Issue Fee and Publication Fee (if any) or to re-apply any previously paid issue fee to the application identified above.
NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other part interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature ____________ Date ____________

Typed or printed name ____________ Registration No. ____________

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/703,942	10/31/2000	Eric A. Pulsipher	10008102-1	4047

022879 7590 09/30/2005

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
HO, CHUONG T	
ART UNIT	**PAPER NUMBER**
2664	

DATE MAILED: 09/30/2005

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 937 day(s). If the issue fee is paid on the date that is three months after t mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a ha months) after the mailing date of this notice, the Patent Term Adjustment will be 937 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date th determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retriev (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office Patent Legal Administration at (571) 272-7702. Questions relating to issue and publication fee payments should directed to the Customer Service Center of the Office of Patent Publication at (703) 305-8283.

PTOL 85 (Rev 07/05) Approved for use through 04/30/2007

Notice of Allowability	Application No.	Applicant(s)	
	09/703,942	PULSIPHER ET AL.	
	Examiner	Art Unit	
	CHUONG T. HO	2664	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to 09/06/05.
2. ☒ The allowed claim(s) is/are 1-8,10-20 renumbered 1-19.
3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the:
   1. ☐ Certified copies of the priority documents have been received.
   2. ☐ Certified copies of the priority documents have been received in Application No. _____.
   3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.
5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
   1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☒ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☐ Information Disclosure Statements (PTO-1449 or PTO/SB/08), Paper No./Mail Date _____
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application (PTO-152)
6. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____.
7. ☐ Examiner's Amendment/Comment
8. ☐ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

WELLINGTON CHIN
[illegible]RVISORY PATENT EXAMIN[illegible]

Notice of References Cited	Application/Control No. 09/703,942	Applicant(s)/Patent Under Reexamination PULSIPHER ET AL.	
	Examiner CHUONG T. HO	Art Unit 2664	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-6,885,644	04-2005	Knop et al.	370/254
*	B	US-6,791,948	09-2004	Desnoyers et al.	370/254
*	C	US-6,636,981	10-2003	Barnett et al.	714/4
*	D	US-6,347,336	02-2002	Song et al.	709/223
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

Bib Data Sheet

CONFIRMATION NO. 4047

SERIAL NUMBER	FILING DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
09/703,942	10/31/2000 RULE	370	2664	10008102-1

APPLICANTS

Eric A. Pulsipher, Ft Collins, CO;

Joseph R. Hunt, Loveland, CO;

** CONTINUING DATA ************************* NO CH

** FOREIGN APPLICATIONS ******************** NO CH

IF REQUIRED, FOREIGN FILING LICENSE GRANTED
** 02/05/2001

	STATE OR COUNTRY	SHEETS DRAWING	TOTAL CLAIMS	INDEPENDENT CLAIMS
Foreign Priority claimed ☐ yes ☑ no 35 USC 119 (a-d) conditions met ☐ yes ☑ no ☐ Met after Allowance Verified and Acknowledged CH Examiner's Signature Initials	CO	26	20	3

ADDRESS
022879
HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS , CO
80527-2400

TITLE
Method and system for identifying and processing changes to a network topology

FILING FEE RECEIVED	FEES: Authority has been given in Paper	
710	No. ________ to charge/credit DEPOSIT ACCOUNT No. ________ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue)

		☐ Other ____________
		☐ Credit

Issue Classification	Application/Control No.	Applicant(s)/Patent under Reexamination	
	09/703,942	PULSIPHER ET AL.	
	Examiner	Art Unit	
	CHUONG T. HO	2664	

ISSUE CLASSIFICATION

ORIGINAL						CROSS REFERENCE(S)							
CLASS	SUBCLASS					CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)						
370	254					370	403	402	351				
INTERNATIONAL CLASSIFICATION						714	717						
H	0	4	L	12/28		709	223	224					
				/									
				/									
				/									
				/									

CHUONG HO 09/22/05
(Assistant Examiner) (Date)

(Legal Instruments Examiner) 9-29-05 (Date)

WELLINGTON CHIN
SUPERVISORY PATENT EXAMINER
(Primary Examiner) 9/28/05 (Date)

Total Claims Allowed: 19

O.G. Print Claim(s)	O.G. Print Fig.
1	7

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
1	(1)		31		61		91		121		151		181
2	2		32		62		92		122		152		182
3	3		33		63		93		123		153		183
4	4		34		64		94		124		154		184
5	5		35		65		95		125		155		185
6	6		36		66		96		126		156		186
7	7		37		67		97		127		157		187
8	(8)		38		68		98		128		158		188
—	~~9~~		39		69		99		129		159		189
9	10		40		70		100		130		160		190
10	11		41		71		101		131		161		191
11	12		42		72		102		132		162		192
12	13		43		73		103		133		163		193
13	14		44		74		104		134		164		194
14	(15)		45		75		105		135		165		195
15	16		46		76		106		136		166		196
16	17		47		77		107		137		167		197
17	18		48		78		108		138		168		198
18	19		49		79		109		139		169		199
19	20		50		80		110		140		170		200
	21		51		81		111		141		171		201
	22		52		82		112		142		172		202
	23		53		83		113		143		173		203
	24		54		84		114		144		174		204
	25		55		85		115		145		175		205
	26		56		86		116		146		176		206
	27		57		87		117		147		177		207
	28		58		88		118		148		178		208
	29		59		89		119		149		179		209
	30		60		90		120		150		180		210

Index of Claims	Application/Control No. 09/703,942	Applicant(s)/Patent under Reexamination PULSIPHER ET AL.
	Examiner CHUONG T. HO	Art Unit 2664

√	Rejected	–	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Claim Final	Claim Original	Date 09/24/05	Claim Final	Claim Original	Date	Claim Final	Claim Original	Date
1	(1)	=		51			101	
2	2	=		52			102	
3	3	=		53			103	
4	4	=		54			104	
5	5	=		55			105	
6	6	=		56			106	
7	7	=		57			107	
8	(8)	=		58			108	
	~~9~~			59			109	
9	10	=		60			110	
10	11	=		61			111	
11	12	=		62			112	
12	13	=		63			113	
13	14	=		64			114	
14	(15)	=		65			115	
15	16	=		66			116	
16	17	=		67			117	
17	18	=		68			118	
18	19	=		69			119	
19	20	=		70			120	
	21			71			121	
	22			72			122	
	23			73			123	
	24			74			124	
	25			75			125	
	26			76			126	
	27			77			127	
	28			78			128	
	29			79			129	
	30			80			130	
	31			81			131	
	32			82			132	
	33			83			133	
	34			84			134	
	35			85			135	
	36			86			136	
	37			87			137	
	38			88			138	
	39			89			139	
	40			90			140	
	41			91			141	
	42			92			142	
	43			93			143	
	44			94			144	
	45			95			145	
	46			96			146	
	47			97			147	
	48			98			148	
	49			99			149	
	50			100			150	

Search Notes	Application/Control No.	Applicant(s)/Patent under Reexamination	
	09/703,942	PULSIPHER ET AL.	
	Examiner	Art Unit	
	CHUONG T. HO	2664	

updated search

SEARCHED			
Class	Subclass	Date	Examiner
370 370 370	254✓ 410 403✓	09/22/05	CH
370	402✓	↓	↓
709	224✓	↓	↓
714	717✓	↓	↓
714	4	↓	↓
370	252	↓	↓
370	351✓	↓	↓
709	223✓	↓	↓
715	735	↓	↓

INTERFERENCE SEARCHED			
Class	Subclass	Date	Examiner
370	254	09/22/05	CH
370	403	↓	↓
370	402	↓	↓
714 709 709	717 223 224	↓	↓
370	351	↓	↓

SEARCH NOTES (INCLUDING SEARCH STRATEGY)	DATE	EXMR
WEST (Text search, Class search) EAST STN	09/22/05	CH
IEEE	↓	↓

OIPE IAP
NOV 1 8 2005
PATENT & TRADEMARK OFFICE

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571) 273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed wh appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

022879 7590 09/30/2005

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

Note: A certificate of mailing can only be used for domestic mailings of Fee(s) Transmittal. This certificate cannot be used for any other accompany papers. Each additional paper, such as an assignment or formal drawing, m have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the Un States Postal Service with sufficient postage for first class mail in an envel addressed to the Mail Stop ISSUE FEE address above, or being facsim transmitted to the USPTO (571) 273-2885, on the date indicated below.

Cathi Christensen (Depositor's na
Cathi Christensen (Signat
11-18-2005 (D

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/703.942	10/31/2000	Eric A. Pulsipher	10008102-1	4047

TITLE OF INVENTION: METHOD AND SYSTEM FOR IDENTIFYING AND PROCESSING CHANGES TO A NETWORK TOPOLOGY

APPLN. TYPE	SMALL ENTITY	ISSUE FEE	PUBLICATION FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1400	$0	$1400	12/30/2005

EXAMINER	ART UNIT	CLASS-SUBCLASS
HO, CHUONG T	2664	370-254000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
   - ☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
   - ☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. Use of a Customer Number is required.

2. For printing on the patent front page, list
(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,
(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.
1 ____
2 ____
3 ____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE: Hewlett-Packard Development Company, L.P.

(B) RESIDENCE: (CITY and STATE OR COUNTRY): Houston, Texas

Please check the appropriate assignee category or categories (will not be printed on the patent): ☐ Individual ☒ Corporation or other private group entity ☐ Governm

4a. The following fee(s) are enclosed:
- ☒ Issue Fee
- ☐ Publication Fee (No small entity discount permitted)
- ☐ Advance Order - # of Copies ____

4b. Payment of Fee(s):
- ☐ A check in the amount of the fee(s) is enclosed.
- ☐ Payment by credit card. Form PTO-2038 is attached.
- ☒ The Director is hereby authorized to charge the required fee(s), or credit any overpayment Deposit Account Number 08-2025 (enclose an extra copy of this form).

5. Change in Entity Status (from status indicated above)
- ☐ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.
- ☐ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

The Director of the USPTO is requested to apply the Issue Fee and Publication Fee (if any) or to re-apply any previously paid issue fee to the application identified above.
NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other part interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature John C. Moran Date 11/17/2005

Typed or printed name John C. Moran Registration No. 30,782

11/21/2005 TBESHAH2 00000080 082025 09703942

01 FC:1501 1400.00 DA

BEST AVAILABLE COPY

● PRINTER RUSH ●
(PTO ASSISTANCE)

Application : 09/703942 **Examiner :** Ho **GAU :** 2664

From: NPB **Location:** (IDC) FMF FDC **Date:** 12-1-05

Tracking #: 09703942 Week Date: 10-3-05

DOC CODE	DOC DATE	MISCELLANEOUS
☐ 1449		☐ Continuing Data
☐ IDS		☐ Foreign Priority
☑ CLM	9-6-05	☐ Document Legibility
☑ IIFW	9-30-05	☐ Fees
☐ SRFW		☐ Other
☐ DRW		
☐ OATH		
☐ 312		
☐ SPEC		

[RUSH] **MESSAGE:** Original claim 4 canceled in claimset but present in index of claims.

Thank You, NPB

[XRUSH] **RESPONSE:**

INITIALS:

NOTE: This form will be included as part of the official USPTO record, with the Response document coded as XRUSH.

REV 10/04

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/703,942	10/31/2000	Eric A. Pulsipher	10008102-1	4047

22879 7590 12/27/2005

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
HO, CHUONG T	

ART UNIT	PAPER NUMBER
2664	

DATE MAILED: 12/27/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

AK

SUPPLEMENTAL *Notice of Allowability*	**Application No.** 09/703,942	**Applicant(s)** PULSIPHER ET AL.	
	Examiner CHUONG T. HO	**Art Unit** 2664	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *09/06/05*.

2. ☒ The allowed claim(s) is/are *1-3, 5-8,10-20 renumbered 1-18*.

Ajit Patel
Primary Examiner

3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the:
   1. ☐ Certified copies of the priority documents have been received.
   2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
   3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
   1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☐ Information Disclosure Statements (PTO-1449 or PTO/SB/08), Paper No./Mail Date _____
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application (PTO-152)
6. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____.
7. ☐ Examiner's Amendment/Comment
8. ☐ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

● PRINTER RUSH ●
(PTO ASSISTANCE)

Application : 09/703942 **Examiner :** Ho **GAU :** 2664

From: NGB **Location:** (IDC) FMF FDC **Date:** 12-1-05

Tracking #: 6pm 09703942 Week Date: 10-3-05

DOC CODE	DOC DATE	MISCELLANEOUS
☐ 1449		☐ Continuing Data
☐ IDS		☐ Foreign Priority
☑ ~~CLM~~ C	9-6-05	☐ Document Legibility
☑ ~~IIFW~~ C	9-30-05	☐ Fees
☐ SRFW		☐ Other
☐ DRW		
☐ OATH		
☐ 312		
☐ SPEC		

[RUSH] **MESSAGE:** Original claim 4 canceled in claimset but present in index of claims.

Thank You,
NGB

[XRUSH] **RESPONSE:** Completed in index of claims
Above errors has been corrected.

INITIALS: CH

NOTE: This form will be included as part of the official USPTO record, with the Response document coded as XRUSH.
REV 10/04

Issue Classification	Application/Control No.	Applicant(s)/Patent under Reexamination
	09/703,942	PULSIPHER ET AL.
	Examiner	Art Unit
	CHUONG T. HO	2664

ISSUE CLASSIFICATION

ORIGINAL		CROSS REFERENCE(S)							
CLASS	SUBCLASS	CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)						
370	254	370	403	402	351				
INTERNATIONAL CLASSIFICATION		714	717						
H04L	12/28	709	223	224					
	/								
	/								
	/								
	/								

CHUONG HO 09/22/05
(Assistant Examiner) (Date)

[illegible] 9-29-05
(Legal Instruments Examiner) (Date)

WELLINGTON CHIN
SUPERVISORY PATENT EXAMINER
(Primary Examiner) 9/28/05 (Date)

Total Claims Allowed: ~~19~~ 18

IIFW sent

O.G. Print Claim(s)	O.G. Print Fig.
1	7

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

see index

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
1	1		31		61		91		121		151		181
2	2		32		62		92		122		152		182
3	3		33		63		93		123		153		183
4	4		34		64		94		124		154		184
5	5		35		65		95		125		155		185
6	6		36		66		96		126		156		186
7	7		37		67		97		127		157		187
8	8		38		68		98		128		158		188
—	9		39		69		99		129		159		189
9	10		40		70		100		130		160		190
10	11		41		71		101		131		161		191
11	12		42		72		102		132		162		192
12	13		43		73		103		133		163		193
13	14		44		74		104		134		164		194
14	15		45		75		105		135		165		195
15	16		46		76		106		136		166		196
16	17		47		77		107		137		167		197
17	18		48		78		108		138		168		198
18	19		49		79		109		139		169		199
19	20		50		80		110		140		170		200
	21		51		81		111		141		171		201
	22		52		82		112		142		172		202
	23		53		83		113		143		173		203
	24		54		84		114		144		174		204
	25		55		85		115		145		175		205
	26		56		86		116		146		176		206
	27		57		87		117		147		177		207
	28		58		88		118		148		178		208
	29		59		89		119		149		179		209
	30		60		90		120		150		180		210

U.S. Patent and Trademark Office Part of Paper No. 14

Index of Claims	Application/Control No. 09/703,942	Applicant(s)/Patent under Reexamination PULSIPHER ET AL.
	Examiner CHUONG T. HO	Art Unit 2664

√	Rejected	–	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

(per query return)

Claim Final	Claim Original	Date 06/16/05
1	1	=
2	2	=
3	3	=
	~~4~~	
4	5	
5	6	
6	7	
7	8	✓
/	9	O
8	10	✓
9	11	✓
10	12	✓
11	13	✓
12	14	✓
13	15	=
14	16	=
15	17	=
16	18	=
17	19	=
18	20	=
	21	
	22	
	23	
	24	
	25	
	26	
	27	
	28	
	29	
	30	
	31	
	32	
	33	
	34	
	35	
	36	
	37	
	38	
	39	
	40	
	41	
	42	
	43	
	44	
	45	
	46	
	47	
	48	
	49	
	50	

Claim Final	Claim Original	Date
	51	
	52	
	53	
	54	
	55	
	56	
	57	
	58	
	59	
	60	
	61	
	62	
	63	
	64	
	65	
	66	
	67	
	68	
	69	
	70	
	71	
	72	
	73	
	74	
	75	
	76	
	77	
	78	
	79	
	80	
	81	
	82	
	83	
	84	
	85	
	86	
	87	
	88	
	89	
	90	
	91	
	92	
	93	
	94	
	95	
	96	
	97	
	98	
	99	
	100	

Claim Final	Claim Original	Date
	101	
	102	
	103	
	104	
	105	
	106	
	107	
	108	
	109	
	110	
	111	
	112	
	113	
	114	
	115	
	116	
	117	
	118	
	119	
	120	
	121	
	122	
	123	
	124	
	125	
	126	
	127	
	128	
	129	
	130	
	131	
	132	
	133	
	134	
	135	
	136	
	137	
	138	
	139	
	140	
	141	
	142	
	143	
	144	
	145	
	146	
	147	
	148	
	149	
	150	

U.S. Patent and Trademark Office   Part of Paper No. 11

AO 120 (Rev. 08/10)

TO: Mail Stop 8 Director of the U.S. Patent and Trademark Office P.O. Box 1450 Alexandria, VA 22313-1450	REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court Northern District of California on the following

☐ Trademarks or ☑ Patents. (☐ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED	U.S. DISTRICT COURT Northern District of California
PLAINTIFF Hewlett-Packard Company 3000 Hanover Street, Palo Alto, California 94304		DEFENDANT ServiceNow, Inc. 3260 Jay Street, Santa Clara, California 95054

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 See Complaint		
2 7,925,981		
3 7,945,860		
4 7,890,802		
5 7,610,512		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY ☐ Amendment ☐ Answer ☐ Cross Bill ☐ Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 8,224,683		
2 6,321,229		
3 7,392,300		
4 7,027,411		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

tutorials, demonstrations, and "best practice methodology"; and offers technical support, consulting services, and assistance to its customers, who directly infringe the '802 Patent.

39. The infringing products were especially made or adapted for use in infringement of the '802 Patent, and are not staple articles or commodities of commerce suitable for substantial non-infringing use.

40. HP has sustained damages as a direct and proximate result of ServiceNow's infringement of the '802 Patent.

41. HP will suffer and is suffering irreparable harm from ServiceNow's infringement the '802 Patent. HP has no adequate remedy at law and is entitled to an injunction against ServiceNow's continuing infringement of the '802 Patent. Unless enjoined, ServiceNow will continue its infringing conduct.

Count 4

(Infringement of the '512 Patent)

42. HP realleges and incorporates herein by reference the allegations in paragraphs 1-41 above.

43. ServiceNow has infringed, and continues to infringe, directly and indirectly through contributory and/or induced infringement, one or more claims of the '512 Patent by making, using, selling, and offering to sell one or more of the products identified in this Complaint, including ServiceNow Service Automation Platform, ServiceNow Incident Management, and/or ServiceNow Orchestration.

44. ServiceNow had actual notice of its infringement of the '512 Patent no later than the filing date of this Complaint. Despite ServiceNow's actual notice of infringement, ServiceNow continues to make, use, sell, and/or offer to sell infringing products with the knowledge or willful blindness that its conduct will induce ServiceNow's customers to infringe the '512 Patent.

45. ServiceNow induces others to infringe the '512 Patent, in violation of 35 U.S.C. § 271, by assisting, facilitating, and encouraging others to perform acts or construct products known by ServiceNow to infringe the '512 Patent. ServiceNow advertises and promotes the

infringing products; offers a "ServiceNow Wiki" page with detailed product information, tutorials, demonstrations, and "best practice methodology"; and offers technical support, consulting services, and assistance to its customers, who directly infringe the '512 Patent.

46. The infringing products were especially made or adapted for use in infringement of the '512 Patent, and are not staple articles or commodities of commerce suitable for substantial non-infringing use.

47. HP has sustained damages as a direct and proximate result of ServiceNow's infringement of the '512 Patent.

48. HP will suffer and is suffering irreparable harm from ServiceNow's infringement the '512 Patent. HP has no adequate remedy at law and is entitled to an injunction against ServiceNow's continuing infringement of the '512 Patent. Unless enjoined, ServiceNow will continue its infringing conduct.

Count 5

(Infringement of the '683 Patent)

49. HP realleges and incorporates herein by reference the allegations in paragraphs 1-48 above.

50. ServiceNow has infringed, and continues to infringe, directly and indirectly through contributory and/or induced infringement, one or more claims of the '683 Patent by making, using, selling, and offering to sell one or more of the products identified in this Complaint, including ServiceNow Service Automation Platform and/or ServiceNow Incident Management.

51. ServiceNow had actual notice of its infringement of the '683 Patent no later than the filing date of this Complaint. Despite ServiceNow's actual notice of infringement, ServiceNow continues to make, use, sell, and/or offer to sell infringing products with the knowledge or willful blindness that its conduct will induce ServiceNow's customers to infringe the '683 Patent.

52. ServiceNow induces others to infringe the '683 Patent, in violation of 35 U.S.C. § 271, by assisting, facilitating, and encouraging others to perform acts or construct products

known by ServiceNow to infringe the '683 Patent. ServiceNow advertises and promotes the infringing products; offers a "ServiceNow Wiki" page with detailed product information, tutorials, demonstrations, and "best practice methodology"; and offers technical support, consulting services, and assistance to its customers, who directly infringe the '683 Patent.

53. The infringing products were especially made or adapted for use in infringement of the '683 Patent, and are not staple articles or commodities of commerce suitable for substantial non-infringing use.

54. HP has sustained damages as a direct and proximate result of ServiceNow's infringement of the '683 Patent.

55. HP will suffer and is suffering irreparable harm from ServiceNow's infringement the '683 Patent. HP has no adequate remedy at law and is entitled to an injunction against ServiceNow's continuing infringement of the '683 Patent. Unless enjoined, ServiceNow will continue its infringing conduct.

Count 6

(Infringement of the '229 Patent)

56. HP realleges and incorporates herein by reference the allegations in paragraphs 1-55 above.

57. ServiceNow has infringed, and continues to infringe, directly and indirectly through contributory and/or induced infringement, one or more claims of the '229 Patent by making, using, selling, and offering to sell one or more of the products identified in this Complaint, including ServiceNow Service Automation Platform, ServiceNow Business Services Management Map, and/or ServiceNow Configuration Management Database.

58. ServiceNow had actual notice of its infringement of the '229 Patent no later than the filing date of this Complaint. Despite ServiceNow's actual notice of infringement, ServiceNow continues to make, use, sell, and/or offer to sell infringing products with the knowledge or willful blindness that its conduct will induce ServiceNow's customers to infringe the '229 Patent.

59. ServiceNow induces others to infringe the '229 Patent, in violation of 35 U.S.C. § 271, by assisting, facilitating, and encouraging others to perform acts or construct products known by ServiceNow to infringe the '229 Patent. ServiceNow advertises and promotes the infringing products; offers a "ServiceNow Wiki" page with detailed product information, tutorials, demonstrations, and "best practice methodology"; and offers technical support, consulting services, and assistance to its customers, who directly infringe the '229 Patent.

60. The infringing products were especially made or adapted for use in infringement of the '229 Patent, and are not staple articles or commodities of commerce suitable for substantial non-infringing use.

61. HP has sustained damages as a direct and proximate result of ServiceNow's infringement of the '229 Patent.

62. HP will suffer and is suffering irreparable harm from ServiceNow's infringement the '229 Patent. HP has no adequate remedy at law and is entitled to an injunction against ServiceNow's continuing infringement of the '229 Patent. Unless enjoined, ServiceNow will continue its infringing conduct.

Count 7

(Infringement of the '300 Patent)

63. HP realleges and incorporates herein by reference the allegations in paragraphs 1-62 above.

64. ServiceNow has infringed, and continues to infringe, directly and indirectly through contributory and/or induced infringement, one or more claims of the '300 Patent by making, using, selling, and offering to sell one or more of the products identified in this Complaint, including ServiceNow Service Automation Platform, ServiceNow Configuration Management Database, ServiceNow Business Services Management Map, and/or ServiceNow Discovery.

65. ServiceNow had actual notice of its infringement of the '300 Patent no later than the filing date of this Complaint. Despite ServiceNow's actual notice of infringement, ServiceNow continues to make, use, sell, and/or offer to sell infringing products with the

knowledge or willful blindness that its conduct will induce ServiceNow's customers to infringe the '300 Patent.

66. ServiceNow induces others to infringe the '300 Patent, in violation of 35 U.S.C. § 271, by assisting, facilitating, and encouraging others to perform acts or construct products known by ServiceNow to infringe the '300 Patent. ServiceNow advertises and promotes the infringing products; offers a "ServiceNow Wiki" page with detailed product information, tutorials, demonstrations, and "best practice methodology"; and offers technical support, consulting services, and assistance to its customers, who directly infringe the '300 Patent.

67. The infringing products were especially made or adapted for use in infringement of the '300 Patent, and are not staple articles or commodities of commerce suitable for substantial non-infringing use.

68. HP has sustained damages as a direct and proximate result of ServiceNow's infringement of the '300 Patent.

69. HP will suffer and is suffering irreparable harm from ServiceNow's infringement the '300 Patent. HP has no adequate remedy at law and is entitled to an injunction against ServiceNow's continuing infringement of the '300 Patent. Unless enjoined, ServiceNow will continue its infringing conduct.

Count 8

(Infringement of the '411 Patent)

70. HP realleges and incorporates herein by reference the allegations in paragraphs 1-69 above.

71. ServiceNow has infringed, and continues to infringe, directly and indirectly through contributory and/or induced infringement, one or more claims of the '411 Patent by making, using, selling, and offering to sell one or more of the products identified in this Complaint, including ServiceNow Service Automation Platform, ServiceNow Configuration Management Database, ServiceNow Baseline CMDB plugin, and/or ServiceNow Discovery.

72. ServiceNow had actual notice of its infringement of the '411 Patent no later than the filing date of this Complaint. Despite ServiceNow's actual notice of infringement,

ServiceNow continues to make, use, sell, and/or offer to sell infringing products with the knowledge or willful blindness that its conduct will induce ServiceNow's customers to infringe the '411 Patent.

73. ServiceNow induces others to infringe the '411 Patent, in violation of 35 U.S.C. § 271, by assisting, facilitating, and encouraging others to perform acts or construct products known by ServiceNow to infringe the '411 Patent. ServiceNow advertises and promotes the infringing products; offers a "ServiceNow Wiki" page with detailed product information, tutorials, demonstrations, and "best practice methodology"; and offers technical support, consulting services, and assistance to its customers, who directly infringe the '411 Patent.

74. The infringing products were especially made or adapted for use in infringement of the '411 Patent, and are not staple articles or commodities of commerce suitable for substantial non-infringing use.

75. HP has sustained damages as a direct and proximate result of ServiceNow's infringement of the '411 Patent.

76. HP will suffer and is suffering irreparable harm from ServiceNow's infringement the '411 Patent. HP has no adequate remedy at law and is entitled to an injunction against ServiceNow's continuing infringement of the '411 Patent. Unless enjoined, ServiceNow will continue its infringing conduct.

JURY DEMAND

Pursuant to Federal Rule of Civil Procedure 38, Plaintiff HP demands a jury trial on all issues triable by a jury.

PRAYER FOR RELIEF

WHEREFORE, HP prays for relief as follows:

1. A judgment that ServiceNow has directly infringed one or more claims of each of HP's asserted patents.

2. A judgment that ServiceNow is actively inducing and/or contributing to the infringement of one or more claims of each of the asserted patents.

3. A judgment awarding HP all damages adequate to compensate for ServiceNow's infringement of HP's asserted patents, including lost profits, and in no event less than a reasonable royalty for ServiceNow's acts of infringement, including all pre-judgment and post judgment interest at the maximum rate permitted by law.

4. An order permanently enjoining ServiceNow and its officers, agents, directors, servants, employees, affiliates, representatives, attorneys, and any others acting in privity or in concert with them, and their parents, subsidiaries, divisions, successors and assigns, from directly or indirectly infringing the asserted patents.

5. For such other and further relief as may be proper.

Dated: February 6, 2014

WILMER CUTLER PICKERING
HALE AND DORR LLP

/s/ Mark D. Flanagan

Mark D. Flanagan (SBN 130303)
(mark.flanagan@wilmerhale.com)
WILMER CUTLER PICKERING
HALE AND DORR LLP
950 Page Mill Road
Palo Alto, CA 94304
Telephone: (650) 858-6000
Facsimile: (650) 858-6100

Attorneys for Plaintiff
Hewlett-Packard Co.

MARK D. FLANAGAN (SBN 130303)
mark.flanagan@wilmerhale.com
MARK D. SELWYN (SBN 244180)
mark.selwyn@wilmerhale.com
JOSEPH F. HAAG (SBN 248749)
joseph.haag@wilmerhale.com
EVELYN C. MAK (SBN 258086)
evelyn.mak@wilmerhale.com
WILMER CUTLER PICKERING
HALE AND DORR LLP
950 PAGE MILL ROAD
PALO ALTO, CALIFORNIA 94304
Telephone: (650) 858-6000
Facsimile: (650) 858-6100

Attorneys for Plaintiff
HEWLETT-PACKARD COMPANY

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

HEWLETT-PACKARD COMPANY, a Delaware corporation, **Plaintiff,** **v.** **SERVICENOW, INC., a Delaware Corporation,** **Defendant.**	**Case No.** **COMPLAINT FOR PATENT INFRINGEMENT** **DEMAND FOR JURY TRIAL**

Plaintiff Hewlett-Packard Company ("HP") complains and alleges as follows against Defendant ServiceNow, Inc. ("ServiceNow").

THE PARTIES

1. Plaintiff HP is a Delaware corporation having a principal place of business at 3000 Hanover Street, Palo Alto, California 94304.

2. Founded in 1939 in a Palo Alto garage by college friends William Hewlett and David Packard, HP is today among the largest and most innovative technology companies in the world, serving customers in more than 170 countries with products ranging from software, personal computing, printing and imaging to IT infrastructure and digital entertainment. In the last decade alone, HP has invested more than 20 billion dollars in research and development, and has been awarded thousands of patents for its innovations by the U.S. Patent and Trademark Office.

3. One of the fields in which HP has been a pioneer and industry leader is Information Technology Service Management, or ITSM. The proliferation of sophisticated and expensive IT networks of computers, software, and associated devices and services has made effective management of IT resources a mission-critical need for businesses and organizations of virtually any size. ITSM software provides the tools necessary to do just that. Demonstrating HP's long-time leadership in ITSM, and ITSM software in particular, HP has led the development of the recognized industry framework of best practices for ITSM promulgated by the Information Technology Infrastructure Library (ITIL), including by authoring significant portions of past and current versions of ITIL. As a result of its innovations in ITSM, HP and the companies it has acquired collectively have been awarded numerous patents relating to managing and operating an IT infrastructure, including ITSM-specific patents.

4. Defendant ServiceNow is a Delaware corporation having a principal place of business at 3260 Jay Street, Santa Clara, California 95054. ServiceNow maintains numerous offices around the world and is doing business in this judicial district.

JURISDICTION

5. This Court has subject matter jurisdiction under 28 U.S.C. §§ 1331 and 1338.

6. This Court has personal jurisdiction over ServiceNow because ServiceNow regularly does business in this judicial district and/or has infringed or caused infringement in California and in this district. Upon information and belief, ServiceNow derives significant revenue from the sale of infringing products within this district, and knows its actions will have consequences within this district.

VENUE

7. Venue is proper in this Court under 28 U.S.C. §§ 1391(b) and (c) because ServiceNow transacts business in this district, acts of infringement have been committed in this district, and ServiceNow is subject to personal jurisdiction in this district. In addition, venue is proper because HP's principal place of business is in this district, and HP has suffered harm in this district.

INTRADISTRICT ASSIGNMENT

8. This Complaint includes an Intellectual Property Action for Patent Infringement, which is an excepted category under Civil Local Rule 3-2(c). Consequently, this action is assigned on a district-wide basis.

BACKGROUND

HP's History of Innovation

9. HP is a pioneer in ITSM software and is a leading supplier of hardware and software information technology management solutions for businesses throughout the United States and the world. HP's software offerings include a suite of software solutions for IT management, including application lifecycle management, automation and orchestration of IT operations, ITSM, security intelligence and risk management, hardware asset management and software asset management.

10. HP introduced its first IT software products, known as the OpenView products, in the early 1990s and added service management capabilities to its OpenView products in 1996. HP has continually improved the technology through both its own research and development and

acquisitions of some of the most innovative companies in the field. HP's continued innovation in the field has allowed it to introduce new ITSM solutions for its customers and has resulted in HP being awarded patents on a continual basis that recognize that innovation.

ServiceNow

11. ServiceNow is a direct competitor of HP in the ITSM software market. ServiceNow's products infringe numerous HP patents stemming from its technology investments, including at least those identified herein. As a result of ServiceNow's infringement of HP's patents, HP has suffered and continues to suffer irreparable harm, as well as damages in the form of lost profits and a reasonable royalty for ServiceNow's infringement of those patents. Consequently, HP seeks a permanent injunction prohibiting the continued infringement of HP's patents by ServiceNow's products, as well as compensatory damages.

The Patents-In-Suit

12. HP is the owner by assignment of all right, title, and interest in and to United States Patent No. 7,925,981 ("the '981 Patent"), entitled "Systems and Methods for Managing Web Services Via a Framework of Interfaces." The '981 Patent was issued on April 12, 2011, from U.S. Patent Application No. 10/438,716, filed May 14, 2003. A true and correct copy of the '981 Patent is attached as Exhibit 1.

13. HP is the owner by assignment of all right, title, and interest in and to United States Patent No. 7,945,860 ("the '860 Patent"), entitled "Systems and Methods for Managing Conversations Between Information Technology Resources." The '860 Patent was issued on May 17, 2011, from U.S. Patent Application No. 10/438,576, filed May 14, 2003. A true and correct copy of the '860 Patent is attached as Exhibit 2.

14. HP is the owner by assignment of all right, title, and interest in and to United States Patent No. 7,890,802 ("the '802 Patent"), entitled "Systems and Method for Automated and Assisted Resolution of IT Incidents." The '802 Patent was issued on February 15, 2011, from U.S. Patent Application No. 12/543,387, filed August 18, 2009. A true and correct copy of the '802 Patent is attached as Exhibit 3.

15. HP is the owner by assignment of all right, title, and interest in and to United States Patent No. 7,610,512 ("the '512 Patent"), entitled "Systems and Method for Automated and Assisted Resolution of IT Incidents." The '512 Patent was issued on October 27, 2009, from U.S. Patent Application No. 11/327,745, filed January 6, 2006. A true and correct copy of the '512 Patent is attached as Exhibit 4.

16. HP is the owner by assignment of all right, title, and interest in and to United States Patent No. 8,224,683 ("the '683 Patent"), entitled "Information Technology Service Request Level of Service Monitor." The '683 Patent was issued on July 17, 2012, from U.S. Patent Application No. 10/615,054, filed July 8, 2003. A true and correct copy of the '683 Patent is attached as Exhibit 5.

17. HP is the owner by assignment of all right, title, and interest in and to United States Patent No. 6,321,229 ("the '229 Patent"), entitled "Method and Apparatus for Using an Information Model to Organize an Information Repository into a Hierarchy of Information." The '229 Patent was issued on November 20, 2001, from U.S. Patent Application No. 09/258,576, filed February 26, 1999. A true and correct copy of the '229 Patent is attached as Exhibit 6.

18. HP is the owner by assignment of all right, title, and interest in and to United States Patent No. 7,392,300 ("the '300 Patent"), entitled "Method and System for Modelling a Communications Network." The '300 Patent was issued on June 24, 2008, from U.S. Patent Application No. 10/753,841, filed January 8, 2004. A true and correct copy of the '300 Patent is attached as Exhibit 7.

19. HP is the owner by assignment of all right, title, and interest in and to United States Patent No. 7,027,411 ("the '411 Patent"), entitled "Method and System for Identifying and Processing Changes to a Network Topology." The '411 Patent was issued on April 11, 2006, from U.S. Patent Application No. 09/703,942, filed October 31, 2000. A true and correct copy of the '411 Patent is attached as Exhibit 8.

ServiceNow's Infringing Products

20. ServiceNow has infringed, and continues to infringe, directly and indirectly through contributory and/or induced infringement, the asserted patents by making, using, selling, and offering to sell one or more of the products identified in this Complaint, including the ServiceNow Service Automation Platform; ServiceNow Incident Management; ServiceNow Configuration Management Database (CMDB); ServiceNow Business Services Management Map, ServiceNow Baseline CMDB plugin; ServiceNow Discovery; and ServiceNow Orchestration (formerly known as ServiceNow Runbook Automation).

CAUSES OF ACTION

Count 1

(Infringement of the '981 Patent)

21. HP realleges and incorporates herein by reference the allegations in paragraphs 1-20 above.

22. ServiceNow has infringed, and continues to infringe, directly and indirectly through contributory and/or induced infringement, one or more claims of the '981 Patent by making, using, selling, and offering to sell one or more of the products identified in this Complaint, including ServiceNow Service Automation Platform.

23. ServiceNow had actual notice of its infringement of the '981 Patent no later than the filing date of this Complaint. Despite ServiceNow's actual notice of infringement, ServiceNow continues to make, use, sell, and/or offer to sell infringing products with the knowledge or willful blindness that its conduct will induce ServiceNow's customers to infringe the '981 Patent.

24. ServiceNow induces others to infringe the '981 Patent, in violation of 35 U.S.C. § 271, by assisting, facilitating, and encouraging others to perform acts or construct products known by ServiceNow to infringe the '981 Patent. ServiceNow advertises and promotes the infringing products; offers a "ServiceNow Wiki" page with detailed product information, tutorials, demonstrations, and "best practice methodology"; and offers technical support, consulting services, and assistance to its customers, who directly infringe the '981 Patent.

25. The infringing products were especially made or adapted for use in infringement of the '981 Patent, and are not staple articles or commodities of commerce suitable for substantial non-infringing use.

26. HP has sustained damages as a direct and proximate result of ServiceNow's infringement of the '981 Patent.

27. HP will suffer and is suffering irreparable harm from ServiceNow's infringement the '981 Patent. HP has no adequate remedy at law and is entitled to an injunction against ServiceNow's continuing infringement of the '981 Patent. Unless enjoined, ServiceNow will continue its infringing conduct.

Count 2

(Infringement of the '860 Patent)

28. HP realleges and incorporates herein by reference the allegations in paragraphs 1-27 above.

29. ServiceNow has infringed, and continues to infringe, directly and indirectly through contributory and/or induced infringement, one or more claims of the '860 Patent by making, using, selling, and offering to sell one or more of the products identified in this Complaint, including ServiceNow Service Automation Platform.

30. ServiceNow had actual notice of its infringement of the '860 Patent no later than the filing date of this Complaint. Despite ServiceNow's actual notice of infringement, ServiceNow continues to make, use, sell, and/or offer to sell infringing products with the knowledge or willful blindness that its conduct will induce ServiceNow's customers to infringe the '860 Patent.

31. ServiceNow induces others to infringe the '860 Patent, in violation of 35 U.S.C. § 271, by assisting, facilitating, and encouraging others to perform acts or construct products known by ServiceNow to infringe the '860 Patent. ServiceNow advertises and promotes the infringing products; offers a "ServiceNow Wiki" page with detailed product information, tutorials, demonstrations, and "best practice methodology"; and offers technical support, consulting services, and assistance to its customers, who directly infringe the '860 Patent.

32. The infringing products were especially made or adapted for use in infringement of the '860 Patent, and are not staple articles or commodities of commerce suitable for substantial non-infringing use.

33. HP has sustained damages as a direct and proximate result of ServiceNow's infringement of the '860 Patent.

34. HP will suffer and is suffering irreparable harm from ServiceNow's infringement the '860 Patent. HP has no adequate remedy at law and is entitled to an injunction against ServiceNow's continuing infringement of the '860 Patent. Unless enjoined, ServiceNow will continue its infringing conduct.

Count 3

(Infringement of the '802 Patent)

35. HP realleges and incorporates herein by reference the allegations in paragraphs 1-34 above.

36. ServiceNow has infringed, and continues to infringe, directly and indirectly through contributory and/or induced infringement, one or more claims of the '802 Patent by making, using, selling, and offering to sell one or more of the products identified in this Complaint, including ServiceNow Service Automation Platform and/or ServiceNow Orchestration.

37. ServiceNow had actual notice of its infringement of the '802 Patent no later than the filing date of this Complaint. Despite ServiceNow's actual notice of infringement, ServiceNow continues to make, use, sell, and/or offer to sell infringing products with the knowledge or willful blindness that its conduct will induce ServiceNow's customers to infringe the '802 Patent.

38. ServiceNow induces others to infringe the '802 Patent, in violation of 35 U.S.C. § 271, by assisting, facilitating, and encouraging others to perform acts or construct products known by ServiceNow to infringe the '802 Patent. ServiceNow advertises and promotes the infringing products; offers a "ServiceNow Wiki" page with detailed product information,

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	PATENT NUMBER	GROUP ART UNIT	FILE WRAPPER LOCATION
09/703,942	7027411	2616	9200

OC000000074019410

Correspondence Address/Fee Address Change

The following fields have been set to Customer Number 56436 on 03/16/2015

- **Correspondence Address**
- **Maintenance Fee Address**

The address of record for Customer Number 56436 is:

56436
HP Company
Intellectual Property Administration
3404 E. Harmony Road
Mail Stop 35
Fort Collins, CO 80528