

jc490 U.S. PTO  
02/15/00

A

**NEW UNITED STATES UTILITY PATENT APPLICATION**  
under 37 C.F.R. 1.53(b)

Page 1

Atty. Docket No. 00479.85672

Commissioner of Patents  
Box Patent Applications  
Washington, D.C. 20231

jc135 U.S. PTO  
09/504783  
02/15/00

Enclosed herewith is a new patent application and the following papers:

First Named Inventor (or application identifier): Edmund Colby Munger, et al.

Title of Invention: Improvements To An Agile Network Protocol For Secure Communications With Assured System Availability

- 1.  Specification 84 pages (including specification, claims, abstract) / 71 claims (9 independent)
- 2.  Declaration/Power of Attorney is:
  - attached in the regular manner.
  - NOT included, but deferred under 37 C.F.R. § 1.53(f).
- 3.  35 Distinct sheets of  Formal  Informal Drawings
- 4.  Preliminary Amendment.
- 5.  Information Disclosure Statement
  - Form 1449
  - A copy of each cited prior art reference
- 6.  Assignment with Cover Sheet.
- 7.  Priority is hereby claimed under 35 U.S.C. § 119(e) and §120 based upon the following application(s):

Country	Application Number	Date of Filing (day, month, year)
US	60/106,261	10/30/98
US	60/137,704	6/7/99
US	09/429,643	10/29/99

- 8.  Priority document(s).
- 9.  Statement Claiming Small Entity Status.
- 10.  Microfiche Computer Program (Appendix).

**NEW UNITED STATES UTILITY PATENT APPLICATION**  
under 37 C.F.R. 1.53(b)

Page 2

Atty. Docket No. 00479.85672

11. Calculation of Fees:

FEES FOR	EXCESS CLAIMS	FEE	AMOUNT DUE
Basic Filing Fee (37 C.F.R. § 1.16(a))			\$690.00
Total Claims in Excess of 20 (37 C.F.R. § 1.16(c))	51	18.00	\$918.00
Independent Claims in Excess of 3 (37 C.F.R. § 1.16(b))	6	78.00	\$468.00
Multiple Dependent Claims (37 C.F.R. § 1.16(d))	0	260.00	\$0.00
Subtotal - Filing Fee Due			\$2,076.00
	REDUCE BY (%)   (\$)		
Reduction by 50%, if Small Entity (37 C.F.R. §§ 1.9, 1.27, 1.28)	0		\$0.00
<b>TOTAL FILING FEE DUE</b>			\$2,076.00
Assignment Recordation Fee (if applicable) (37 C.F.R. § 1.21(h))	1	40.00	\$40.00
<b>GRAND TOTAL DUE</b>			<b>\$2,116.00</b>

12. PAYMENT is:

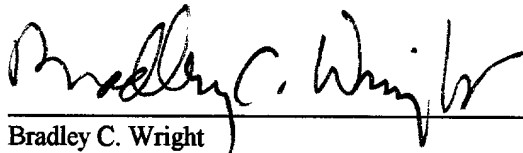
- included in the amount of the GRAND TOTAL by our enclosed check. A general authorization under 37 C.F.R. § 1.25(b), second sentence, is hereby given to credit or debit our Deposit Account No. 19-0733 for the instant filing and for any other fees during the pendency of this application under 37 C.F.R. §§ 1.16, 1.17 and 1.18.
- not included, but deferred under 37 C.F.R. § 1.53(f).

13. All correspondence for the attached application should be directed to:

Banner & Witcoff, Ltd.  
1001 G Street, N.W.  
Washington, D. C. 20001-4597  
Telephone: (202) 508-9100  
Facsimile: (202) 508-9299

14. Other: \_\_\_\_\_

Date: February 15, 2000

By:   
\_\_\_\_\_  
Bradley C. Wright  
Reg. No. 38,061

BCW/pp

**IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL  
FOR SECURE COMMUNICATIONS  
WITH ASSURED SYSTEM AVAILABILITY**

5

**CROSS-REFERENCE TO RELATED APPLICATION**

This application claims priority from and is a continuation-in-part of previously filed U.S. application serial number 09/429,643, filed on October 29, 1999. The subject matter of that application, which is bodily incorporated herein, derives from provisional U.S. application numbers 60/106,261 (filed October 30, 1998) and 60/137,704 (filed June 7, 1999).

10

**BACKGROUND OF THE INVENTION**

A tremendous variety of methods have been proposed and implemented to provide security and anonymity for communications over the Internet. The variety stems, in part, from the different needs of different Internet users. A basic heuristic framework to aid in discussing these different security techniques is illustrated in FIG. 1. Two terminals, an originating terminal 100 and a destination terminal 110 are in communication over the Internet. It is desired for the communications to be secure, that is, immune to eavesdropping. For example, terminal 100 may transmit secret information to terminal 110 over the Internet 107. Also, it may be desired to prevent an eavesdropper from discovering that terminal 100 is in communication with terminal 110. For example, if terminal 100 is a user and terminal 110 hosts a web site, terminal 100's user may not want anyone in the intervening networks to know what web sites he is "visiting." Anonymity would thus be an issue, for example, for companies that want to keep their market research interests private and thus would prefer to prevent outsiders from knowing which web-sites or other Internet resources they are "visiting." These two security issues may be called data security and anonymity, respectively.

15

20

25

Data security is usually tackled using some form of data encryption. An encryption key 48 is known at both the originating and terminating terminals 100 and 110. The keys may be private and public at the originating and destination terminals 100 and 110, respectively or they

may be symmetrical keys (the same key is used by both parties to encrypt and decrypt). Many encryption methods are known and usable in this context.

To hide traffic from a local administrator or ISP, a user can employ a local proxy server in communicating over an encrypted channel with an outside proxy such that the local administrator or ISP only sees the encrypted traffic. Proxy servers prevent destination servers from determining the identities of the originating clients. This system employs an intermediate server interposed between client and destination server. The destination server sees only the Internet Protocol (IP) address of the proxy server and not the originating client. The target server only sees the address of the outside proxy. This scheme relies on a trusted outside proxy server.

Also, proxy schemes are vulnerable to traffic analysis methods of determining identities of transmitters and receivers. Another important limitation of proxy servers is that the server knows the identities of both calling and called parties. In many instances, an originating terminal, such as terminal A, would prefer to keep its identity concealed from the proxy, for example, if the proxy server is provided by an Internet service provider (ISP).

To defeat traffic analysis, a scheme called Chaum's mixes employs a proxy server that transmits and receives fixed length messages, including dummy messages. Multiple originating terminals are connected through a mix (a server) to multiple target servers. It is difficult to tell which of the originating terminals are communicating to which of the connected target servers, and the dummy messages confuse eavesdroppers' efforts to detect communicating pairs by analyzing traffic. A drawback is that there is a risk that the mix server could be compromised. One way to deal with this risk is to spread the trust among multiple mixes. If one mix is compromised, the identities of the originating and target terminals may remain concealed. This strategy requires a number of alternative mixes so that the intermediate servers interposed between the originating and target terminals are not determinable except by compromising more than one mix. The strategy wraps the message with multiple layers of encrypted addresses. The first mix in a sequence can decrypt only the outer layer of the message to reveal the next destination mix in sequence. The second mix can decrypt the message to reveal the next mix and so on. The target server receives the message and, optionally, a multi-layer encrypted payload



containing return information to send data back in the same fashion. The only way to defeat such a mix scheme is to collude among mixes. If the packets are all fixed-length and intermixed with dummy packets, there is no way to do any kind of traffic analysis.

Still another anonymity technique, called 'crowds,' protects the identity of the originating terminal from the intermediate proxies by providing that originating terminals belong to groups of proxies called crowds. The crowd proxies are interposed between originating and target terminals. Each proxy through which the message is sent is randomly chosen by an upstream proxy. Each intermediate proxy can send the message either to another randomly chosen proxy in the "crowd" or to the destination. Thus, even crowd members cannot determine if a preceding proxy is the originator of the message or if it was simply passed from another proxy.

ZKS (Zero-Knowledge Systems) Anonymous IP Protocol allows users to select up to any of five different pseudonyms, while desktop software encrypts outgoing traffic and wraps it in User Datagram Protocol (UDP) packets. The first server in a 2+-hop system gets the UDP packets, strips off one layer of encryption to add another, then sends the traffic to the next server, which strips off yet another layer of encryption and adds a new one. The user is permitted to control the number of hops. At the final server, traffic is decrypted with an untraceable IP address. The technique is called onion-routing. This method can be defeated using traffic analysis. For a simple example, bursts of packets from a user during low-duty periods can reveal the identities of sender and receiver.

Firewalls attempt to protect LANs from unauthorized access and hostile exploitation or damage to computers connected to the LAN. Firewalls provide a server through which all access to the LAN must pass. Firewalls are centralized systems that require administrative overhead to maintain. They can be compromised by virtual-machine applications ("applets"). They instill a false sense of security that leads to security breaches for example by users sending sensitive information to servers outside the firewall or encouraging use of modems to sidestep the firewall security. Firewalls are not useful for distributed systems such as business travelers, extranets, small teams, etc.

### SUMMARY OF THE INVENTION

A secure mechanism for communicating over the internet, including a protocol referred to as the Tunneled Agile Routing Protocol (TARP), uses a unique two-layer encryption format and special TARP routers. TARP routers are similar in function to regular IP routers. Each TARP router has one or more IP addresses and uses normal IP protocol to send IP packet messages (“packets” or “datagrams”). The IP packets exchanged between TARP terminals via TARP routers are actually encrypted packets whose true destination address is concealed except to TARP routers and servers. The normal or “clear” or “outside” IP header attached to TARP IP packets contains only the address of a next hop router or destination server. That is, instead of indicating a final destination in the destination field of the IP header, the TARP packet’s IP header always points to a next-hop in a series of TARP router hops, or to the final destination. This means there is no overt indication from an intercepted TARP packet of the true destination of the TARP packet since the destination could always be next-hop TARP router as well as the final destination.

Each TARP packet’s true destination is concealed behind a layer of encryption generated using a link key. The link key is the encryption key used for encrypted communication between the hops intervening between an originating TARP terminal and a destination TARP terminal. Each TARP router can remove the outer layer of encryption to reveal the destination router for each TARP packet. To identify the link key needed to decrypt the outer layer of encryption of a TARP packet, a receiving TARP or routing terminal may identify the transmitting terminal by the sender/receiver IP numbers in the cleartext IP header.

Once the outer layer of encryption is removed, the TARP router determines the final destination. Each TARP packet undergoes a minimum number of hops to help foil traffic analysis. The hops may be chosen at random or by a fixed value. As a result, each TARP packet may make random trips among a number of geographically disparate routers before reaching its destination. Each trip is highly likely to be different for each packet composing a given message because each trip is independently randomly determined. This feature is called *agile routing*. The fact that different packets take different routes provides distinct advantages by making it difficult

for an interloper to obtain all the packets forming an entire multi-packet message. The associated advantages have to do with the inner layer of encryption discussed below. Agile routing is combined with another feature that furthers this purpose; a feature that ensures that any message is broken into multiple packets.

5           The IP address of a TARP router can be changed, a feature called *IP agility*. Each TARP router, independently or under direction from another TARP terminal or router, can change its IP address. A separate, unchangeable identifier or address is also defined. This address, called the TARP address, is known only to TARP routers and terminals and may be correlated at any time by a TARP router or a TARP terminal using a Lookup Table (LUT). When a TARP router or  
10          terminal changes its IP address, it updates the other TARP routers and terminals which in turn update their respective LUTs.

          The message payload is hidden behind an inner layer of encryption in the TARP packet that can only be unlocked using a session key. The session key is not available to any of the intervening TARP routers. The session key is used to decrypt the payloads of the TARP packets  
15          permitting the data stream to be reconstructed.

          Communication may be made private using link and session keys, which in turn may be shared and used according to any desired method. For example, public/private keys or symmetric keys may be used.

          To transmit a data stream, a TARP originating terminal constructs a series of TARP  
20          packets from a series of IP packets generated by a network (IP) layer process. (Note that the terms “network layer,” “data link layer,” “application layer,” etc. used in this specification correspond to the Open Systems Interconnection (OSI) network terminology.) The payloads of these packets are assembled into a block and chain-block encrypted using the session key. This assumes, of course, that all the IP packets are destined for the same TARP terminal. The block is  
25          then interleaved and the interleaved encrypted block is broken into a series of payloads, one for each TARP packet to be generated. Special TARP headers  $IP_T$  are then added to each payload using the IP headers from the data stream packets. The TARP headers can be identical to normal IP headers or customized in some way. They should contain a formula or data for deinterleaving

the data at the destination TARP terminal, a time-to-live (TTL) parameter to indicate the number of hops still to be executed, a data type identifier which indicates whether the payload contains, for example, TCP or UDP data, the sender's TARP address, the destination TARP address, and an indicator as to whether the packet contains real or decoy data or a formula for filtering out  
5 decoy data if decoy data is spread in some way through the TARP payload data.

Note that although chain-block encryption is discussed here with reference to the session key, any encryption method may be used. Preferably, as in chain block encryption, a method should be used that makes unauthorized decryption difficult without an entire result of the encryption process. Thus, by separating the encrypted block among multiple packets and making  
10 it difficult for an interloper to obtain access to all of such packets, the contents of the communications are provided an extra layer of security.

Decoy or dummy data can be added to a stream to help foil traffic analysis by reducing the peak-to-average network load. It may be desirable to provide the TARP process with an ability to respond to the time of day or other criteria to generate more decoy data during low  
15 traffic periods so that communication bursts at one point in the Internet cannot be tied to communication bursts at another point to reveal the communicating endpoints.

Dummy data also helps to break the data into a larger number of inconspicuously-sized packets permitting the interleave window size to be increased while maintaining a reasonable size for each packet. (The packet size can be a single standard size or selected from a fixed range  
20 of sizes.) One primary reason for desiring for each message to be broken into multiple packets is apparent if a chain block encryption scheme is used to form the first encryption layer prior to interleaving. A single block encryption may be applied to portion, or entirety, of a message, and that portion or entirety then interleaved into a number of separate packets. Considering the agile IP routing of the packets, and the attendant difficulty of reconstructing an entire sequence of  
25 packets to form a single block-encrypted message element, decoy packets can significantly increase the difficulty of reconstructing an entire data stream.

The above scheme may be implemented entirely by processes operating between the data link layer and the network layer of each server or terminal participating in the TARP system.

Because the encryption system described above is insertable between the data link and network layers, the processes involved in supporting the encrypted communication may be completely transparent to processes at the IP (network) layer and above. The TARP processes may also be completely transparent to the data link layer processes as well. Thus, no operations at or above the Network layer, or at or below the data link layer, are affected by the insertion of the TARP stack. This provides additional security to all processes at or above the network layer, since the difficulty of unauthorized penetration of the network layer (by, for example, a hacker) is increased substantially. Even newly developed servers running at the session layer leave all processes below the session layer vulnerable to attack. Note that in this architecture, security is distributed. That is, notebook computers used by executives on the road, for example, can communicate over the Internet without any compromise in security.

IP address changes made by TARP terminals and routers can be done at regular intervals, at random intervals, or upon detection of "attacks." The variation of IP addresses hinders traffic analysis that might reveal which computers are communicating, and also provides a degree of immunity from attack. The level of immunity from attack is roughly proportional to the rate at which the IP address of the host is changing.

As mentioned, IP addresses may be changed in response to attacks. An attack may be revealed, for example, by a regular series of messages indicating that a router is being probed in some way. Upon detection of an attack, the TARP layer process may respond to this event by changing its IP address. In addition, it may create a subprocess that maintains the original IP address and continues interacting with the attacker in some manner.

Decoy packets may be generated by each TARP terminal on some basis determined by an algorithm. For example, the algorithm may be a random one which calls for the generation of a packet on a random basis when the terminal is idle. Alternatively, the algorithm may be responsive to time of day or detection of low traffic to generate more decoy packets during low traffic times. Note that packets are preferably generated in groups, rather than one by one, the groups being sized to simulate real messages. In addition, so that decoy packets may be inserted in normal TARP message streams, the background loop may have a latch that makes it more

likely to insert decoy packets when a message stream is being received. Alternatively, if a large number of decoy packets is received along with regular TARP packets, the algorithm may increase the rate of dropping of decoy packets rather than forwarding them. The result of dropping and generating decoy packets in this way is to make the apparent incoming message size different from the apparent outgoing message size to help foil traffic analysis.

In various other embodiments of the invention, a scalable version of the system may be constructed in which a plurality of IP addresses are preassigned to each pair of communicating nodes in the network. Each pair of nodes agrees upon an algorithm for “hopping” between IP addresses (both sending and receiving), such that an eavesdropper sees apparently continuously random IP address pairs (source and destination) for packets transmitted between the pair. Overlapping or “reusable” IP addresses may be allocated to different users on the same subnet, since each node merely verifies that a particular packet includes a valid source/destination pair from the agreed-upon algorithm. Source/destination pairs are preferably not reused between any two nodes during any given end-to-end session, though limited IP block sizes or lengthy sessions might require it.

Further improvements described in this continuation-in-part application include: (1) a load balancer that distributes packets across different transmission paths according to transmission path quality; (2) a DNS proxy server that transparently creates a virtual private network in response to a domain name inquiry; (3) a large-to-small link bandwidth management feature that prevents denial-of-service attacks at system chokepoints; (4) a traffic limiter that regulates incoming packets by limiting the rate at which a transmitter can be synchronized with a receiver; and (5) a signaling synchronizer that allows a large number of nodes to communicate with a central node by partitioning the communication function between two separate entities

#### **BRIEF DESCRIPTION OF THE DRAWINGS**

FIG. 1 is an illustration of secure communications over the Internet according to a prior art embodiment.

FIG. 2 is an illustration of secure communications over the Internet according to a an embodiment of the invention.

FIG. 3a is an illustration of a process of forming a tunneled IP packet according to an embodiment of the invention.

FIG. 3b is an illustration of a process of forming a tunneled IP packet according to another embodiment of the invention.

5 FIG. 4 is an illustration of an OSI layer location of processes that may be used to implement the invention.

FIG. 5 is a flow chart illustrating a process for routing a tunneled packet according to an embodiment of the invention.

10 FIG. 6 is a flow chart illustrating a process for forming a tunneled packet according to an embodiment of the invention.

FIG. 7 is a flow chart illustrating a process for receiving a tunneled packet according to an embodiment of the invention.

FIG. 8 shows how a secure session is established and synchronized between a client and a TARP router.

15 FIG. 9 shows an IP address hopping scheme between a client computer and TARP router using transmit and receive tables in each computer.

FIG. 10 shows physical link redundancy among three Internet Service Providers (ISPs) and a client computer.

20 FIG. 11 shows how multiple IP packets can be embedded into a single "frame" such as an Ethernet frame, and further shows the use of a discriminator field to camouflage true packet recipients.

FIG. 12A shows a system that employs hopped hardware addresses, hopped IP addresses, and hopped discriminator fields.

25 FIG. 12B shows several different approaches for hopping hardware addresses, IP addresses, and discriminator fields in combination.

FIG. 13 shows a technique for automatically re-establishing synchronization between sender and receiver through the use of a partially public sync value.

FIG. 14 shows a “checkpoint” scheme for regaining synchronization between a sender and recipient.

FIG. 15 shows further details of the checkpoint scheme of FIG. 14.

5 FIG. 16 shows how two addresses can be decomposed into a plurality of segments for comparison with presence vectors.

FIG. 17 shows a storage array for a receiver’s active addresses.

FIG. 18 shows the receiver’s storage array after receiving a sync request.

FIG. 19 shows the receiver’s storage array after new addresses have been generated.

FIG. 20 shows a system employing distributed transmission paths.

10 FIG. 21 shows a plurality of link transmission tables that can be used to route packets in the system of FIG. 20.

FIG. 22A shows a flowchart for adjusting weight value distributions associated with a plurality of transmission links.

FIG. 22B shows a flowchart for setting a weight value to zero if a transmitter turns off.

15 FIG. 23 shows a system employing distributed transmission paths with adjusted weight value distributions for each path.

FIG. 24 shows an example using the system of FIG. 23.

FIG. 25 shows a conventional domain-name look-up service.

FIG. 26 shows a system employing a DNS proxy server with transparent VPN creation.

20 FIG. 27 shows steps that can be carried out to implement transparent VPN creation based on a DNS look-up function.

FIG. 28 shows a system including a link guard function that prevents packet overloading on a low-bandwidth link LOW BW.

FIG. 29 shows one embodiment of a system employing the principles of FIG. 28.

25 FIG. 30 shows a system that regulates packet transmission rates by throttling the rate at which synchronizations are performed.

FIG. 31 shows a signaling server 3101 and a transport server 3102 used to establish a VPN with a client computer.



FIG. 32 shows message flows relating to synchronization protocols of FIG. 31.

### **DETAILED DESCRIPTION OF THE INVENTION**

Referring to FIG. 2, a secure mechanism for communicating over the internet employs a number of special routers or servers, called TARP routers 122-127 that are similar to regular IP routers 128-132 in that each has one or more IP addresses and uses normal IP protocol to send normal-looking IP packet messages, called TARP packets 140. TARP packets 140 are identical to normal IP packet messages that are routed by regular IP routers 128-132 because each TARP packet 140 contains a destination address as in a normal IP packet. However, instead of indicating a final destination in the destination field of the IP header, the TARP packet's 140 IP header always points to a next-hop in a series of TARP router hops, or the final destination, TARP terminal 110. Because the header of the TARP packet contains only the next-hop destination, there is no overt indication from an intercepted TARP packet of the true destination of the TARP packet 140 since the destination could always be the next-hop TARP router as well as the final destination, TARP terminal 110.

Each TARP packet's true destination is concealed behind an outer layer of encryption generated using a link key 146. The link key 146 is the encryption key used for encrypted communication between the end points (TARP terminals or TARP routers) of a single link in the chain of hops connecting the originating TARP terminal 100 and the destination TARP terminal 110. Each TARP router 122-127, using the link key 146 it uses to communicate with the previous hop in a chain, can use the link key to reveal the true destination of a TARP packet. To identify the link key needed to decrypt the outer layer of encryption of a TARP packet, a receiving TARP or routing terminal may identify the transmitting terminal (which may indicate the link key used) by the sender field of the clear IP header. Alternatively, this identity may be hidden behind another layer of encryption in available bits in the clear IP header. Each TARP router, upon receiving a TARP message, determines if the message is a TARP message by using authentication data in the TARP packet. This could be recorded in available bytes in the TARP packet's IP header. Alternatively, TARP packets could be authenticated by attempting to decrypt

using the link key 146 and determining if the results are as expected. The former may have computational advantages because it does not involve a decryption process.

Once the outer layer of decryption is completed by a TARP router 122-127, the TARP router determines the final destination. The system is preferably designed to cause each TARP packet 140 to undergo a minimum number of hops to help foil traffic analysis. The time to live counter in the IP header of the TARP message may be used to indicate a number of TARP router hops yet to be completed. Each TARP router then would decrement the counter and determine from that whether it should forward the TARP packet 140 to another TARP router 122-127 or to the destination TARP terminal 110. If the time to live counter is zero or below zero after decrementing, for an example of usage, the TARP router receiving the TARP packet 140 may forward the TARP packet 140 to the destination TARP terminal 110. If the time to live counter is above zero after decrementing, for an example of usage, the TARP router receiving the TARP packet 140 may forward the TARP packet 140 to a TARP router 122-127 that the current TARP terminal chooses at random. As a result, each TARP packet 140 is routed through some minimum number of hops of TARP routers 122-127 which are chosen at random.

Thus, each TARP packet, irrespective of the traditional factors determining traffic in the Internet, makes random trips among a number of geographically disparate routers before reaching its destination and each trip is highly likely to be different for each packet composing a given message because each trip is independently randomly determined as described above. This feature is called *agile routing*. For reasons that will become clear shortly, the fact that different packets take different routes provides distinct advantages by making it difficult for an interloper to obtain all the packets forming an entire multi-packet message. Agile routing is combined with another feature that furthers this purpose, a feature that ensures that any message is broken into multiple packets.

A TARP router receives a TARP packet when an IP address used by the TARP router coincides with the IP address in the TARP packet's IP header  $IP_C$ . The IP address of a TARP router, however, may not remain constant. To avoid and manage attacks, each TARP router, independently or under direction from another TARP terminal or router, may change its IP

address. A separate, unchangeable identifier or address is also defined. This address, called the TARP address, is known only to TARP routers and terminals and may be correlated at any time by a TARP router or a TARP terminal using a Lookup Table (LUT). When a TARP router or terminal changes its IP address, it updates the other TARP routers and terminals which in turn update their respective LUTs. In reality, whenever a TARP router looks up the address of a destination in the encrypted header, it must convert a TARP address to a real IP address using its LUT.

While every TARP router receiving a TARP packet has the ability to determine the packet's final destination, the message payload is embedded behind an inner layer of encryption in the TARP packet that can only be unlocked using a session key. The session key is not available to any of the TARP routers 122-127 intervening between the originating 100 and destination 110 TARP terminals. The session key is used to decrypt the payloads of the TARP packets 140 permitting an entire message to be reconstructed.

In one embodiment, communication may be made private using link and session keys, which in turn may be shared and used according any desired method. For example, a public key or symmetric keys may be communicated between link or session endpoints using a public key method. Any of a variety of other mechanisms for securing data to ensure that only authorized computers can have access to the private information in the TARP packets 140 may be used as desired.

Referring to FIG. 3a, to construct a series of TARP packets, a data stream 300 of IP packets 207a, 207b, 207c, etc., such series of packets being formed by a network (IP) layer process, is broken into a series of small sized segments. In the present example, equal-sized segments 1-9 are defined and used to construct a set of interleaved data packets A, B, and C. Here it is assumed that the number of interleaved packets A, B, and C formed is three and that the number of IP packets 207a-207c used to form the three interleaved packets A, B, and C is exactly three. Of course, the number of IP packets spread over a group of interleaved packets may be any convenient number as may be the number of interleaved packets over which the

incoming data stream is spread. The latter, the number of interleaved packets over which the data stream is spread, is called the *interleave window*.

To create a packet, the transmitting software interleaves the normal IP packets 207a *et seq.* to form a new set of interleaved payload data 320. This payload data 320 is then encrypted using a session key to form a set of session-key-encrypted payload data 330, each of which, A, B, and C, will form the payload of a TARP packet. Using the IP header data, from the original packets 207a-207c, new TARP headers  $IP_T$  are formed. The TARP headers  $IP_T$  can be identical to normal IP headers or customized in some way. In a preferred embodiment, the TARP headers  $IP_T$  are IP headers with added data providing the following information required for routing and reconstruction of messages, some of which data is ordinarily, or capable of being, contained in normal IP headers:

1. A window sequence number – an identifier that indicates where the packet belongs in the original message sequence.
2. An interleave sequence number – an identifier that indicates the interleaving sequence used to form the packet so that the packet can be deinterleaved along with other packets in the interleave window.
3. A time-to-live (TTL) datum – indicates the number of TARP-router-hops to be executed before the packet reaches its destination. Note that the TTL parameter may provide a datum to be used in a probabilistic formula for determining whether to route the packet to the destination or to another hop.
4. Data type identifier – indicates whether the payload contains, for example, TCP or UDP data.
5. Sender's address – indicates the sender's address in the TARP network.
6. Destination address – indicates the destination terminal's address in the TARP network.
7. Decoy/Real – an indicator of whether the packet contains real message data or dummy decoy data or a combination.

Obviously, the packets going into a single interleave window must include only packets with a common destination. Thus, it is assumed in the depicted example that the IP headers of IP packets 207a-207c all contain the same destination address or at least will be received by the same terminal so that they can be deinterleaved. Note that dummy or decoy data or packets can be added to form a larger interleave window than would otherwise be required by the size of a given message. Decoy or dummy data can be added to a stream to help foil traffic analysis by leveling the load on the network. Thus, it may be desirable to provide the TARP process with an ability to respond to the time of day or other criteria to generate more decoy data during low traffic periods so that communication bursts at one point in the Internet cannot be tied to communication bursts at another point to reveal the communicating endpoints.

Dummy data also helps to break the data into a larger number of inconspicuously-sized packets permitting the interleave window size to be increased while maintaining a reasonable size for each packet. (The packet size can be a single standard size or selected from a fixed range of sizes.) One primary reason for desiring for each message to be broken into multiple packets is apparent if a chain block encryption scheme is used to form the first encryption layer prior to interleaving. A single block encryption may be applied to a portion, or the entirety, of a message, and that portion or entirety then interleaved into a number of separate packets.

Referring to FIG. 3b, in an alternative mode of TARP packet construction, a series of IP packets are accumulated to make up a predefined interleave window. The payloads of the packets are used to construct a single block for chain block encryption using the session key. The payloads used to form the block are presumed to be destined for the same terminal. The block size may coincide with the interleave window as depicted in the example embodiment of FIG. 3b. After encryption, the encrypted block is broken into separate payloads and segments which are interleaved as in the embodiment of Fig 3a. The resulting interleaved packets A, B, and C, are then packaged as TARP packets with TARP headers as in the Example of FIG. 3a. The remaining process is as shown in, and discussed with reference to, FIG. 3a.

Once the TARP packets 340 are formed, each entire TARP packet 340, including the TARP header  $IP_T$ , is encrypted using the link key for communication with the first-hop-TARP

router. The first hop TARP router is randomly chosen. A final unencrypted IP header  $IP_C$  is added to each encrypted TARP packet 340 to form a normal IP packet 360 that can be transmitted to a TARP router. Note that the process of constructing the TARP packet 360 does not have to be done in stages as described. The above description is just a useful heuristic for  
5 describing the final product, namely, the TARP packet.

Note that, TARP header  $IP_T$  could be a completely custom header configuration with no similarity to a normal IP header except that it contain the information identified above. This is so since this header is interpreted by only TARP routers.

The above scheme may be implemented entirely by processes operating between the data  
10 link layer and the network layer of each server or terminal participating in the TARP system. Referring to FIG. 4, a TARP transceiver 405 can be an originating terminal 100, a destination terminal 110, or a TARP router 122-127. In each TARP Transceiver 405, a transmitting process is generated to receive normal packets from the Network (IP) layer and generate TARP packets for communication over the network. A receiving process is generated to receive normal IP  
15 packets containing TARP packets and generate from these normal IP packets which are "passed up" to the Network (IP) layer. Note that where the TARP Transceiver 405 is a router, the received TARP packets 140 are not processed into a stream of IP packets 415 because they need only be authenticated as proper TARP packets and then passed to another TARP router or a TARP destination terminal 110. The intervening process, a "TARP Layer" 420, could be  
20 combined with either the data link layer 430 or the Network layer 410. In either case, it would intervene between the data link layer 430 so that the process would receive regular IP packets containing embedded TARP packets and "hand up" a series of reassembled IP packets to the Network layer 410. As an example of combining the TARP layer 420 with the data link layer  
25 430, a program may augment the normal processes running a communications card, for example, an Ethernet card. Alternatively, the TARP layer processes may form part of a dynamically loadable module that is loaded and executed to support communications between the network and data link layers.

Because the encryption system described above can be inserted between the data link and network layers, the processes involved in supporting the encrypted communication may be completely transparent to processes at the IP (network) layer and above. The TARP processes may also be completely transparent to the data link layer processes as well. Thus, no operations at or above the network layer, or at or below the data link layer, are affected by the insertion of the TARP stack. This provides additional security to all processes at or above the network layer, since the difficulty of unauthorized penetration of the network layer (by, for example, a hacker) is increased substantially. Even newly developed servers running at the session layer leave all processes below the session layer vulnerable to attack. Note that in this architecture, security is distributed. That is, notebook computers used by executives on the road, for example, can communicate over the Internet without any compromise in security.

Note that IP address changes made by TARP terminals and routers can be done at regular intervals, at random intervals, or upon detection of "attacks." The variation of IP addresses hinders traffic analysis that might reveal which computers are communicating, and also provides a degree of immunity from attack. The level of immunity from attack is roughly proportional to the rate at which the IP address of the host is changing.

As mentioned, IP addresses may be changed in response to attacks. An attack may be revealed, for example, by a regular series of messages indicates that a router is being probed in some way. Upon detection of an attack, the TARP layer process may respond to this event by changing its IP address. To accomplish this, the TARP process will construct a TARP-formatted message, in the style of Internet Control Message Protocol (ICMP) datagrams as an example; this message will contain the machine's TARP address, its previous IP address, and its new IP address. The TARP layer will transmit this packet to at least one known TARP router; then upon receipt and validation of the message, the TARP router will update its LUT with the new IP address for the stated TARP address. The TARP router will then format a similar message, and broadcast it to the other TARP routers so that they may update their LUTs. Since the total number of TARP routers on any given subnet is expected to be relatively small, this process of updating the LUTs should be relatively fast. It may not, however, work as well when there is a

relatively large number of TARP routers and/or a relatively large number of clients; this has motivated a refinement of this architecture to provide scalability; this refinement has led to a second embodiment, which is discussed below.

5 Upon detection of an attack, the TARP process may also create a subprocess that maintains the original IP address and continues interacting with the attacker. The latter may provide an opportunity to trace the attacker or study the attacker's methods (called "fishbowling" drawing upon the analogy of a small fish in a fish bowl that "thinks" it is in the ocean but is actually under captive observation). A history of the communication between the attacker and the abandoned (fishbowed) IP address can be recorded or transmitted for human analysis or further  
10 synthesized for purposes of responding in some way.

As mentioned above, decoy or dummy data or packets can be added to outgoing data streams by TARP terminals or routers. In addition to making it convenient to spread data over a larger number of separate packets, such decoy packets can also help to level the load on inactive portions of the Internet to help foil traffic analysis efforts.

15 Decoy packets may be generated by each TARP terminal 100, 110 or each router 122-127 on some basis determined by an algorithm. For example, the algorithm may be a random one which calls for the generation of a packet on a random basis when the terminal is idle. Alternatively, the algorithm may be responsive to time of day or detection of low traffic to generate more decoy packets during low traffic times. Note that packets are preferably generated  
20 in groups, rather than one by one, the groups being sized to simulate real messages. In addition, so that decoy packets may be inserted in normal TARP message streams, the background loop may have a latch that makes it more likely to insert decoy packets when a message stream is being received. That is, when a series of messages are received, the decoy packet generation rate may be increased. Alternatively, if a large number of decoy packets is received along with  
25 regular TARP packets, the algorithm may increase the rate of dropping of decoy packets rather than forwarding them. The result of dropping and generating decoy packets in this way is to make the apparent incoming message size different from the apparent outgoing message size to help foil traffic analysis. The rate of reception of packets, decoy or otherwise, may be indicated



to the decoy packet dropping and generating processes through perishable decoy and regular packet counters. (A perishable counter is one that resets or decrements its value in response to time so that it contains a high value when it is incremented in rapid succession and a small value when incremented either slowly or a small number of times in rapid succession.) Note that destination TARP terminal 110 may generate decoy packets equal in number and size to those TARP packets received to make it appear it is merely routing packets and is therefore not the destination terminal.

Referring to FIG. 5, the following particular steps may be employed in the above-described method for routing TARP packets.

- S0. A background loop operation is performed which applies an algorithm which determines the generation of decoy IP packets. The loop is interrupted when an encrypted TARP packet is received.
- S2. The TARP packet may be probed in some way to authenticate the packet before attempting to decrypt it using the link key. That is, the router may determine that the packet is an authentic TARP packet by performing a selected operation on some data included with the clear IP header attached to the encrypted TARP packet contained in the payload. This makes it possible to avoid performing decryption on packets that are not authentic TARP packets.
- S3. The TARP packet is decrypted to expose the destination TARP address and an indication of whether the packet is a decoy packet or part of a real message.
- S4. If the packet is a decoy packet, the perishable decoy counter is incremented.
- S5. Based on the decoy generation/dropping algorithm and the perishable decoy counter value, if the packet is a decoy packet, the router may choose to throw it away. If the received packet is a decoy packet and it is determined that it should be thrown away (S6), control returns to step S0.

- S7. The TTL parameter of the TARP header is decremented and it is determined if the TTL parameter is greater than zero.
- S8. If the TTL parameter is greater than zero, a TARP address is randomly chosen from a list of TARP addresses maintained by the router and the link key and IP address corresponding to that TARP address memorized for use in creating a new IP packet containing the TARP packet.
- S9. If the TTL parameter is zero or less, the link key and IP address corresponding to the TARP address of the destination are memorized for use in creating the new IP packet containing the TARP packet.
- S10. The TARP packet is encrypted using the memorized link key.
- S11. An IP header is added to the packet that contains the stored IP address, the encrypted TARP packet wrapped with an IP header, and the completed packet transmitted to the next hop or destination.

Referring to FIG. 6, the following particular steps may be employed in the above-described method for generating TARP packets.

- S20. A background loop operation applies an algorithm that determines the generation of decoy IP packets. The loop is interrupted when a data stream containing IP packets is received for transmission.
- S21. The received IP packets are grouped into a set consisting of messages with a constant IP destination address. The set is further broken down to coincide with a maximum size of an interleave window. The set is encrypted, and interleaved into a set of payloads destined to become TARP packets.
- S22. The TARP address corresponding to the IP address is determined from a lookup table and stored to generate the TARP header. An initial TTL count is generated and stored in the

header. The TTL count may be random with minimum and maximum values or it may be fixed or determined by some other parameter.

- S23. The window sequence numbers and interleave sequence numbers are recorded in the TARP headers of each packet.
- 5 • S24. One TARP router address is randomly chosen for each TARP packet and the IP address corresponding to it stored for use in the clear IP header. The link key corresponding to this router is identified and used to encrypt TARP packets containing interleaved and encrypted data and TARP headers.
- S25. A clear IP header with the first hop router's real IP address is generated and added to  
10 each of the encrypted TARP packets and the resulting packets.

Referring to FIG. 7, the following particular steps may be employed in the above-described method for receiving TARP packets.

- 15 • S40. A background loop operation is performed which applies an algorithm which determines the generation of decoy IP packets. The loop is interrupted when an encrypted TARP packet is received.
- S42. The TARP packet may be probed to authenticate the packet before attempting to decrypt it using the link key.
- 20 • S43. The TARP packet is decrypted with the appropriate link key to expose the destination TARP address and an indication of whether the packet is a decoy packet or part of a real message.
- S44. If the packet is a decoy packet, the perishable decoy counter is incremented.
- S45. Based on the decoy generation/dropping algorithm and the perishable decoy counter  
25 value, if the packet is a decoy packet, the receiver may choose to throw it away.
- S46. The TARP packets are cached until all packets forming an interleave window are received.

- S47. Once all packets of an interleave window are received, the packets are deinterleaved.
- S48. The packets block of combined packets defining the interleave window is then decrypted using the session key.
- S49. The decrypted block is then divided using the window sequence data and the  $IP_T$  headers are converted into normal  $IP_C$  headers. The window sequence numbers are integrated in the  $IP_C$  headers.
- S50. The packets are then handed up to the IP layer processes.

### 1. SCALABILITY ENHANCEMENTS

The IP agility feature described above relies on the ability to transmit IP address changes to all TARP routers. The embodiments including this feature will be referred to as “boutique” embodiments due to potential limitations in scaling these features up for a large network, such as the Internet. (The “boutique” embodiments would, however, be robust for use in smaller networks, such as small virtual private networks, for example). One problem with the boutique embodiments is that if IP address changes are to occur frequently, the message traffic required to update all routers sufficiently quickly creates a serious burden on the Internet when the TARP router and/or client population gets large. The bandwidth burden added to the networks, for example in ICMP packets, that would be used to update all the TARP routers could overwhelm the Internet for a large scale implementation that approached the scale of the Internet. In other words, the boutique system’s scalability is limited.

A system can be constructed which trades some of the features of the above embodiments to provide the benefits of IP agility without the additional messaging burden. This is accomplished by IP address-hopping according to shared algorithms that govern IP addresses used between links participating in communications sessions between nodes such as TARP nodes. (Note that the IP hopping technique is also applicable to the boutique embodiment.) The IP agility feature discussed with respect to the boutique system can be modified so that it becomes decentralized under this scalable regime and governed by the above-described shared

algorithm. Other features of the boutique system may be combined with this new type of IP-agility.

The new embodiment has the advantage of providing IP agility governed by a local algorithm and set of IP addresses exchanged by each communicating pair of nodes. This local governance is session-independent in that it may govern communications between a pair of nodes, irrespective of the session or end points being transferred between the directly communicating pair of nodes.

In the scalable embodiments, blocks of IP addresses are allocated to each node in the network. (This scalability will increase in the future, when Internet Protocol addresses are increased to 128-bit fields, vastly increasing the number of distinctly addressable nodes). Each node can thus use any of the IP addresses assigned to that node to communicate with other nodes in the network. Indeed, each pair of communicating nodes can use a plurality of source IP addresses and destination IP addresses for communicating with each other.

Each communicating pair of nodes in a chain participating in any session stores two blocks of IP addresses, called netblocks, and an algorithm and randomization seed for selecting, from each netblock, the next pair of source/destination IP addresses that will be used to transmit the next message. In other words, the algorithm governs the sequential selection of IP-address pairs, one sender and one receiver IP address, from each netblock. The combination of algorithm, seed, and netblock (IP address block) will be called a "hopblock." A router issues separate transmit and receive hopblocks to its clients. The send address and the receive address of the IP header of each outgoing packet sent by the client are filled with the send and receive IP addresses generated by the algorithm. The algorithm is "clocked" (indexed) by a counter so that each time a pair is used, the algorithm turns out a new transmit pair for the next packet to be sent.

The router's receive hopblock is identical to the client's transmit hopblock. The router uses the receive hopblock to predict what the send and receive IP address pair for the next expected packet from that client will be. Since packets can be received out of order, it is not possible for the router to predict with certainty what IP address pair will be on the next sequential packet. To account for this problem, the router generates a range of predictions

encompassing the number of possible transmitted packet send/receive addresses, of which the next packet received could leap ahead. Thus, if there is a vanishingly small probability that a given packet will arrive at the router ahead of 5 packets transmitted by the client before the given packet, then the router can generate a series of 6 send/receive IP address pairs (or “hop window”) to compare with the next received packet. When a packet is received, it is marked in the hop window as such, so that a second packet with the same IP address pair will be discarded. If an out-of-sequence packet does not arrive within a predetermined timeout period, it can be requested for retransmission or simply discarded from the receive table, depending upon the protocol in use for that communications session, or possibly by convention.

When the router receives the client’s packet, it compares the send and receive IP addresses of the packet with the next N predicted send and receive IP address pairs and rejects the packet if it is not a member of this set. Received packets that do not have the predicted source/destination IP addresses falling within the window are rejected, thus thwarting possible hackers. (With the number of possible combinations, even a fairly large window would be hard to fall into at random.) If it is a member of this set, the router accepts the packet and processes it further. This link-based IP-hopping strategy, referred to as “IHOP,” is a network element that stands on its own and is not necessarily accompanied by elements of the boutique system described above. If the routing agility feature described in connection with the boutique embodiment is combined with this link-based IP-hopping strategy, the router’s next step would be to decrypt the TARP header to determine the destination TARP router for the packet and determine what should be the next hop for the packet. The TARP router would then forward the packet to a random TARP router or the destination TARP router with which the source TARP router has a link-based IP hopping communication established.

Figure 8 shows how a client computer 801 and a TARP router 811 can establish a secure session. When client 801 seeks to establish an IHOP session with TARP router 811, the client 801 sends “secure synchronization” request (“SSYN”) packet 821 to the TARP router 811. This SYN packet 821 contains the client’s 801 authentication token, and may be sent to the router 811 in an encrypted format. The source and destination IP numbers on the packet 821 are the client’s

801 current fixed IP address, and a “known” fixed IP address for the router 811. (For security purposes, it may be desirable to reject any packets from outside of the local network that are destined for the router’s known fixed IP address.) Upon receipt and validation of the client’s 801 SSYN packet 821, the router 811 responds by sending an encrypted “secure synchronization acknowledgment” (“SSYN ACK”) 822 to the client 801. This SSYN ACK 822 will contain the transmit and receive hopblocks that the client 801 will use when communicating with the TARP router 811. The client 801 will acknowledge the TARP router’s 811 response packet 822 by generating an encrypted SSYN ACK ACK packet 823 which will be sent from the client’s 801 fixed IP address and to the TARP router’s 811 known fixed IP address. The client 801 will simultaneously generate a SSYN ACK ACK packet; this SSYN ACK packet, referred to as the Secure Session Initiation (SSI) packet 824, will be sent with the first {sender, receiver} IP pair in the client’s transmit table 921 (FIG. 9), as specified in the transmit hopblock provided by the TARP router 811 in the SSYN ACK packet 822. The TARP router 811 will respond to the SSI packet 824 with an SSI ACK packet 825, which will be sent with the first {sender, receiver} IP pair in the TARP router’s transmit table 923. Once these packets have been successfully exchanged, the secure communications session is established, and all further secure communications between the client 801 and the TARP router 811 will be conducted via this secure session, as long as synchronization is maintained. If synchronization is lost, then the client 801 and TARP router 802 may re-establish the secure session by the procedure outlined in Figure 8 and described above.

While the secure session is active, both the client 901 and TARP router 911 (FIG. 9) will maintain their respective transmit tables 921, 923 and receive tables 922, 924, as provided by the TARP router during session synchronization 822. It is important that the sequence of IP pairs in the client’s transmit table 921 be identical to those in the TARP router’s receive table 924; similarly, the sequence of IP pairs in the client’s receive table 922 must be identical to those in the router’s transmit table 923. This is required for the session synchronization to be maintained. The client 901 need maintain only one transmit table 921 and one receive table 922 during the course of the secure session. Each sequential packet sent by the client 901 will employ the next

{send, receive} IP address pair in the transmit table, regardless of TCP or UDP session. The TARP router 911 will expect each packet arriving from the client 901 to bear the next IP address pair shown in its receive table.

5 Since packets can arrive out of order, however, the router 911 can maintain a “look ahead” buffer in its receive table, and will mark previously-received IP pairs as invalid for future packets; any future packet containing an IP pair that is in the look-ahead buffer but is marked as previously received will be discarded. Communications from the TARP router 911 to the client 901 are maintained in an identical manner; in particular, the router 911 will select the next IP address pair from its transmit table 923 when constructing a packet to send to the client 901, and the client 901 will maintain a look-ahead buffer of expected IP pairs on packets that it is receiving. Each TARP router will maintain separate pairs of transmit and receive tables for each client that is currently engaged in a secure session with or through that TARP router.

10 While clients receive their hopblocks from the first server linking them to the Internet, routers exchange hopblocks. When a router establishes a link-based IP-hopping communication regime with another router, each router of the pair exchanges its transmit hopblock. The transmit hopblock of each router becomes the receive hopblock of the other router. The communication between routers is governed as described by the example of a client sending a packet to the first router.

15 While the above strategy works fine in the IP milieu, many local networks that are connected to the Internet are Ethernet systems. In Ethernet, the IP addresses of the destination devices must be translated into hardware addresses, and vice versa, using known processes (“address resolution protocol,” and “reverse address resolution protocol”). However, if the link-based IP-hopping strategy is employed, the correlation process would become explosive and burdensome. An alternative to the link-based IP hopping strategy may be employed within an Ethernet network. The solution is to provide that the node linking the Internet to the Ethernet (call it the border node) use the link-based IP-hopping communication regime to communicate with nodes outside the Ethernet LAN. Within the Ethernet LAN, each TARP node would have a single IP address which would be addressed in the conventional way. Instead of comparing the

20

25



{sender, receiver} IP address pairs to authenticate a packet, the intra-LAN TARP node would use one of the IP header extension fields to do so. Thus, the border node uses an algorithm shared by the intra-LAN TARP node to generate a symbol that is stored in the free field in the IP header, and the intra-LAN TARP node generates a range of symbols based on its prediction of the next expected packet to be received from that particular source IP address. The packet is rejected if it does not fall into the set of predicted symbols (for example, numerical values) or is accepted if it does. Communications from the intra-LAN TARP node to the border node are accomplished in the same manner, though the algorithm will necessarily be different for security reasons. Thus, each of the communicating nodes will generate transmit and receive tables in a similar manner to that of Figure 9; the intra-LAN TARP nodes transmit table will be identical to the border node's receive table, and the intra-LAN TARP node's receive table will be identical to the border node's transmit table.

The algorithm used for IP address-hopping can be any desired algorithm. For example, the algorithm can be a given pseudo-random number generator that generates numbers of the range covering the allowed IP addresses with a given seed. Alternatively, the session participants can assume a certain type of algorithm and specify simply a parameter for applying the algorithm. For example the assumed algorithm could be a particular pseudo-random number generator and the session participants could simply exchange seed values.

Note that there is no permanent physical distinction between the originating and destination terminal nodes. Either device at either end point can initiate a synchronization of the pair. Note also that the authentication/synchronization-request (and acknowledgment) and hopblock-exchange may all be served by a single message so that separate message exchanges may not be required.

As another extension to the stated architecture, multiple physical paths can be used by a client, in order to provide link redundancy and further thwart attempts at denial of service and traffic monitoring. As shown in Figure 10, for example, client 1001 can establish three simultaneous sessions with each of three TARP routers provided by different ISPs 1011, 1012, 1013. As an example, the client 1001 can use three different telephone lines 1021, 1022, 1023 to

connect to the ISPs, or two telephone lines and a cable modem, etc. In this scheme, transmitted packets will be sent in a random fashion among the different physical paths. This architecture provides a high degree of communications redundancy, with improved immunity from denial-of-service attacks and traffic monitoring.

5

## **2. FURTHER EXTENSIONS**

The following describes various extensions to the techniques, systems, and methods described above. As described above, the security of communications occurring between computers in a computer network (such as the Internet, an Ethernet, or others) can be enhanced by using seemingly random source and destination Internet Protocol (IP) addresses for data packets transmitted over the network. This feature prevents eavesdroppers from determining which computers in the network are communicating with each other while permitting the two communicating computers to easily recognize whether a given received data packet is legitimate or not. In one embodiment of the above-described systems, an IP header extension field is used to authenticate incoming packets on an Ethernet.

10

15

Various extensions to the previously described techniques described herein include: (1) use of hopped hardware or “MAC” addresses in broadcast type network; (2) a self-synchronization technique that permits a computer to automatically regain synchronization with a sender; (3) synchronization algorithms that allow transmitting and receiving computers to quickly re-establish synchronization in the event of lost packets or other events; and (4) a fast-packet rejection mechanism for rejecting invalid packets. Any or all of these extensions can be combined with the features described above in any of various ways.

20

### **A. Hardware Address Hopping**

Internet protocol-based communications techniques on a LAN—or across any dedicated physical medium—typically embed the IP packets within lower-level packets, often referred to as “frames.” As shown in FIG. 11, for example, a first Ethernet frame 1150 comprises a frame header 1101 and two embedded IP packets IP1 and IP2, while a second Ethernet frame 1160 comprises a different frame header 1104 and a single IP packet IP3. Each frame header generally includes a source hardware address 1101A and a destination hardware address 1101B;

25

other well-known fields in frame headers are omitted from FIG. 11 for clarity. Two hardware nodes communicating over a physical communication channel insert appropriate source and destination hardware addresses to indicate which nodes on the channel or network should receive the frame.

5           It may be possible for a nefarious listener to acquire information about the contents of a frame and/or its communicants by examining frames on a local network rather than (or in addition to) the IP packets themselves. This is especially true in broadcast media, such as Ethernet, where it is necessary to insert into the frame header the hardware address of the machine that generated the frame and the hardware address of the machine to which frame is  
10 being sent. All nodes on the network can potentially “see” all packets transmitted across the network. This can be a problem for secure communications, especially in cases where the communicants do not want for any third party to be able to identify who is engaging in the information exchange. One way to address this problem is to push the address-hopping scheme down to the hardware layer. In accordance with various embodiments of the invention, hardware  
15 addresses are “hopped” in a manner similar to that used to change IP addresses, such that a listener cannot determine which hardware node generated a particular message nor which node is the intended recipient.

FIG. 12A shows a system in which Media Access Control (“MAC”) hardware addresses are “hopped” in order to increase security over a network such as an Ethernet. While the  
20 description refers to the exemplary case of an Ethernet environment, the inventive principles are equally applicable to other types of communications media. In the Ethernet case, the MAC address of the sender and receiver are inserted into the Ethernet frame and can be observed by anyone on the LAN who is within the broadcast range for that frame. For secure communications, it becomes desirable to generate frames with MAC addresses that are not  
25 attributable to any specific sender or receiver.

As shown in FIG. 12A, two computer nodes 1201 and 1202 communicate over a communication channel such as an Ethernet. Each node executes one or more application programs 1203 and 1218 that communicate by transmitting packets through communication

software 1204 and 1217, respectively. Examples of application programs include video conferencing, e-mail, word processing programs, telephony, and the like. Communication software 1204 and 1217 can comprise, for example, an OSI layered architecture or “stack” that standardizes various services provided at different levels of functionality.

5 The lowest levels of communication software 1204 and 1217 communicate with hardware components 1206 and 1214 respectively, each of which can include one or more registers 1207 and 1215 that allow the hardware to be reconfigured or controlled in accordance with various communication protocols. The hardware components (an Ethernet network interface card, for example) communicate with each other over the communication medium.  
10 Each hardware component is typically pre-assigned a fixed hardware address or MAC number that identifies the hardware component to other nodes on the network. One or more interface drivers control the operation of each card and can, for example, be configured to accept or reject packets from certain hardware addresses. As will be described in more detail below, various embodiments of the inventive principles provide for “hopping” different addresses using one or  
15 more algorithms and one or more moving windows that track a range of valid addresses to validate received packets. Packets transmitted according to one or more of the inventive principles will be generally referred to as “secure” packets or “secure communications” to differentiate them from ordinary data packets that are transmitted in the clear using ordinary, machine-correlated addresses.

20 One straightforward method of generating non-attributable MAC addresses is an extension of the IP hopping scheme. In this scenario, two machines on the same LAN that desire to communicate in a secure fashion exchange random-number generators and seeds, and create sequences of quasi-random MAC addresses for synchronized hopping. The implementation and synchronization issues are then similar to that of IP hopping.

25 This approach, however, runs the risk of using MAC addresses that are currently active on the LAN—which, in turn, could interrupt communications for those machines. Since an Ethernet MAC address is at present 48 bits in length, the chance of randomly misusing an active MAC address is actually quite small. However, if that figure is multiplied by a large number of

nodes (as would be found on an extensive LAN), by a large number of frames (as might be the case with packet voice or streaming video), and by a large number of concurrent Virtual Private Networks (VPNs), then the chance that a non-secure machine's MAC address could be used in an address-hopped frame can become non-trivial. In short, any scheme that runs even a small risk of interrupting communications for other machines on the LAN is bound to receive resistance from prospective system administrators. Nevertheless, it is technically feasible, and can be implemented without risk on a LAN on which there is a small number of machines, or if all of the machines on the LAN are engaging in MAC-hopped communications.

Synchronized MAC address hopping may incur some overhead in the course of session establishment, especially if there are multiple sessions or multiple nodes involved in the communications. A simpler method of randomizing MAC addresses is to allow each node to receive and process *every* incident frame on the network. Typically, each network interface driver will check the destination MAC address in the header of every incident frame to see if it matches that machine's MAC address; if there is no match, then the frame is discarded. In one embodiment, however, these checks can be disabled, and every incident packet is passed to the TARP stack for processing. This will be referred to as "promiscuous" mode, since every incident frame is processed. Promiscuous mode allows the sender to use completely random, unsynchronized MAC addresses, since the destination machine is guaranteed to process the frame. The decision as to whether the packet was truly intended for that machine is handled by the TARP stack, which checks the source and destination IP addresses for a match in its IP synchronization tables. If no match is found, the packet is discarded; if there is a match, the packet is unwrapped, the inner header is evaluated, and if the inner header indicates that the packet is destined for that machine then the packet is forwarded to the IP stack—otherwise it is discarded.

One disadvantage of purely-random MAC address hopping is its impact on processing overhead; that is, since every incident frame must be processed, the machine's CPU is engaged considerably more often than if the network interface driver is discriminating and rejecting packets unilaterally. A compromise approach is to select either a single fixed MAC address or a

small number of MAC addresses (e.g., one for each virtual private network on an Ethernet) to use for MAC-hopped communications, regardless of the actual recipient for which the message is intended. In this mode, the network interface driver can check each incident frame against one (or a few) pre-established MAC addresses, thereby freeing the CPU from the task of physical-layer packet discrimination. This scheme does not betray any useful information to an interloper on the LAN; in particular, every secure packet can already be identified by a unique packet type in the outer header. However, since all machines engaged in secure communications would either be using the same MAC address, or be selecting from a small pool of predetermined MAC addresses, the association between a specific machine and a specific MAC address is effectively broken.

In this scheme, the CPU will be engaged more often than it would be in non-secure communications (or in synchronized MAC address hopping), since the network interface driver cannot always unilaterally discriminate between secure packets that are destined for that machine, and secure packets from other VPNs. However, the non-secure traffic is easily eliminated at the network interface, thereby reducing the amount of processing required of the CPU. There are boundary conditions where these statements would not hold, of course—e.g., if *all* of the traffic on the LAN is secure traffic, then the CPU would be engaged to the same degree as it is in the purely-random address hopping case; alternatively, if each VPN on the LAN uses a different MAC address, then the network interface can perfectly discriminate secure frames destined for the local machine from those constituting other VPNs. These are engineering tradeoffs that might be best handled by providing administrative options for the users when installing the software and/or establishing VPNs.

Even in this scenario, however, there still remains a slight risk of selecting MAC addresses that are being used by one or more nodes on the LAN. One solution to this problem is to formally assign one address or a range of addresses for use in MAC-hopped communications. This is typically done via an assigned numbers registration authority; e.g., in the case of Ethernet, MAC address ranges are assigned to vendors by the Institute of Electrical and Electronics Engineers (IEEE). A formally-assigned range of addresses would ensure that secure

frames do not conflict with any properly-configured and properly-functioning machines on the LAN.

Reference will now be made to FIGS. 12A and 12B in order to describe the many combinations and features that follow the inventive principles. As explained above, two computer nodes 1201 and 1202 are assumed to be communicating over a network or communication medium such as an Ethernet. A communication protocol in each node (1204 and 1217, respectively) contains a modified element 1205 and 1216 that performs certain functions that deviate from the standard communication protocols. In particular, computer node 1201 implements a first "hop" algorithm 1208X that selects seemingly random source and destination IP addresses (and, in one embodiment, seemingly random IP header discriminator fields) in order to transmit each packet to the other computer node. For example, node 1201 maintains a transmit table 1208 containing triplets of source (S), destination (D), and discriminator fields (DS) that are inserted into outgoing IP packet headers. The table is generated through the use of an appropriate algorithm (e.g., a random number generator that is seeded with an appropriate seed) that is known to the recipient node 1202. As each new IP packet is formed, the next sequential entry out of the sender's transmit table 1208 is used to populate the IP source, IP destination, and IP header extension field (e.g., discriminator field). It will be appreciated that the transmit table need not be created in advance but could instead be created on-the-fly by executing the algorithm when each packet is formed.

At the receiving node 1202, the same IP hop algorithm 1222X is maintained and used to generate a receive table 1222 that lists valid triplets of source IP address, destination IP address, and discriminator field. This is shown by virtue of the first five entries of transmit table 1208 matching the second five entries of receive table 1222. (The tables may be slightly offset at any particular time due to lost packets, misordered packets, or transmission delays). Additionally, node 1202 maintains a receive window W3 that represents a list of valid IP source, IP destination, and discriminator fields that will be accepted when received as part of an incoming IP packet. As packets are received, window W3 slides down the list of valid entries, such that the possible valid entries change over time. Two packets that arrive out of order but are

nevertheless matched to entries within window W3 will be accepted; those falling outside of window W3 will be rejected as invalid. The length of window W3 can be adjusted as necessary to reflect network delays or other factors.

Node 1202 maintains a similar transmit table 1221 for creating IP packets and frames destined for node 1201 using a potentially different hopping algorithm 1221X, and node 1201 maintains a matching receive table 1209 using the same algorithm 1209X. As node 1202 transmits packets to node 1201 using seemingly random IP source, IP destination, and/or discriminator fields, node 1201 matches the incoming packet values to those falling within window W1 maintained in its receive table. In effect, transmit table 1208 of node 1201 is synchronized (i.e., entries are selected in the same order) to receive table 1222 of receiving node 1202. Similarly, transmit table 1221 of node 1202 is synchronized to receive table 1209 of node 1201. It will be appreciated that although a common algorithm is shown for the source, destination and discriminator fields in FIG. 12A (using, e.g., a different seed for each of the three fields), an entirely different algorithm could in fact be used to establish values for each of these fields. It will also be appreciated that one or two of the fields can be “hopped” rather than all three as illustrated.

In accordance with another aspect of the invention, hardware or “MAC” addresses are hopped instead of or in addition to IP addresses and/or the discriminator field in order to improve security in a local area or broadcast-type network. To that end, node 1201 further maintains a transmit table 1210 using a transmit algorithm 1210X to generate source and destination hardware addresses that are inserted into frame headers (e.g., fields 1101A and 1101B in FIG. 11) that are synchronized to a corresponding receive table 1224 at node 1202. Similarly, node 1202 maintains a different transmit table 1223 containing source and destination hardware addresses that is synchronized with a corresponding receive table 1211 at node 1201. In this manner, outgoing hardware frames appear to be originating from and going to completely random nodes on the network, even though each recipient can determine whether a given packet is intended for it or not. It will be appreciated that the hardware hopping feature can be



implemented at a different level in the communications protocol than the IP hopping feature (e.g., in a card driver or in a hardware card itself to improve performance).

FIG. 12B shows three different embodiments or modes that can be employed using the aforementioned principles. In a first mode referred to as “promiscuous” mode, a common hardware address (e.g., a fixed address for source and another for destination) or else a completely random hardware address is used by all nodes on the network, such that a particular packet cannot be attributed to any one node. Each node must initially accept all packets containing the common (or random) hardware address and inspect the IP addresses or discriminator field to determine whether the packet is intended for that node. In this regard, either the IP addresses or the discriminator field or both can be varied in accordance with an algorithm as described above. As explained previously, this may increase each node’s overhead since additional processing is involved to determine whether a given packet has valid source and destination hardware addresses.

In a second mode referred to as “promiscuous per VPN” mode, a small set of fixed hardware addresses are used, with a fixed source/destination hardware address used for all nodes communicating over a virtual private network. For example, if there are six nodes on an Ethernet, and the network is to be split up into two private virtual networks such that nodes on one VPN can communicate with only the other two nodes on its own VPN, then two sets of hardware addresses could be used: one set for the first VPN and a second set for the second VPN. This would reduce the amount of overhead involved in checking for valid frames since only packets arriving from the designated VPN would need to be checked. IP addresses and one or more discriminator fields could still be hopped as before for secure communication within the VPN. Of course, this solution compromises the anonymity of the VPNs (i.e., an outsider can easily tell what traffic belongs in which VPN, though he cannot correlate it to a specific machine/person). It also requires the use of a discriminator field to mitigate the vulnerability to certain types of DoS attacks. (For example, without the discriminator field, an attacker on the LAN could stream frames containing the MAC addresses being used by the VPN; rejecting those

frames could lead to excessive processing overhead. The discriminator field would provide a low-overhead means of rejecting the false packets.)

In a third mode referred to as “hardware hopping” mode, hardware addresses are varied as illustrated in FIG. 12A, such that hardware source and destination addresses are changed constantly in order to provide non-attributable addressing. Variations on these embodiments are of course possible, and the invention is not intended to be limited in any respect by these illustrative examples.

### **B. Extending the Address Space**

Address hopping provides security and privacy. However, the level of protection is limited by the number of addresses in the blocks being hopped. A hopblock denotes a field or fields modulated on a packet-wise basis for the purpose of providing a VPN. For instance, if two nodes communicate with IP address hopping using hopblocks of 4 addresses (2 bits) each, there would be 16 possible address-pair combinations. A window of size 16 would result in most address pairs being accepted as valid most of the time. This limitation can be overcome by using a discriminator field in addition to or instead of the hopped address fields. The discriminator field would be hopped in exactly the same fashion as the address fields and it would be used to determine whether a packet should be processed by a receiver.

Suppose that two clients, each using four-bit hopblocks, would like the same level of protection afforded to clients communicating via IP hopping between two A blocks (24 address bits eligible for hopping). A discriminator field of 20 bits, used in conjunction with the 4 address bits eligible for hopping in the IP address field, provides this level of protection. A 24-bit discriminator field would provide a similar level of protection if the address fields were not hopped or ignored. Using a discriminator field offers the following advantages: (1) an arbitrarily high level of protection can be provided, and (2) address hopping is unnecessary to provide protection. This may be important in environments where address hopping would cause routing problems.

### C. Synchronization Techniques

It is generally assumed that once a sending node and receiving node have exchanged algorithms and seeds (or similar information sufficient to generate quasi-random source and destination tables), subsequent communication between the two nodes will proceed smoothly.

5 Realistically, however, two nodes may lose synchronization due to network delays or outages, or other problems. Consequently, it is desirable to provide means for re-establishing synchronization between nodes in a network that have lost synchronization.

One possible technique is to require that each node provide an acknowledgment upon successful receipt of each packet and, if no acknowledgment is received within a certain period  
10 of time, to re-send the unacknowledged packet. This approach, however, drives up overhead costs and may be prohibitive in high-throughput environments such as streaming video or audio, for example.

A different approach is to employ an automatic synchronizing technique that will be referred to herein as “self-synchronization.” In this approach, synchronization information is  
15 embedded into each packet, thereby enabling the receiver to re-synchronize itself upon receipt of a single packet if it determines that it has lost synchronization with the sender. (If communications are already in progress, and the receiver determines that it is still in sync with the sender, then there is no need to re-synchronize.) A receiver could detect that it was out of synchronization by, for example, employing a “dead-man” timer that expires after a certain  
20 period of time, wherein the timer is reset with each valid packet. A time stamp could be hashed into the public sync field (see below) to preclude packet-retry attacks.

In one embodiment, a “sync field” is added to the header of each packet sent out by the sender. This sync field could appear in the clear or as part of an encrypted portion of the packet. Assuming that a sender and receiver have selected a random-number generator (RNG) and seed  
25 value, this combination of RNG and seed can be used to generate a random-number sequence (RNS). The RNS is then used to generate a sequence of source/destination IP pairs (and, if desired, discriminator fields and hardware source and destination addresses), as described above. It is not necessary, however, to generate the entire sequence (or the first N-1 values) in order to

generate the Nth random number in the sequence; if the sequence index N is known, the random value corresponding to that index can be directly generated (see below). Different RNGs (and seeds) with different fundamental periods could be used to generate the source and destination IP sequences, but the basic concepts would still apply. For the sake of simplicity, the following discussion will assume that IP source and destination address pairs (only) are hopped using a single RNG sequencing mechanism.

In accordance with a “self-synchronization” feature, a sync field in each packet header provides an index (i.e., a sequence number) into the RNS that is being used to generate IP pairs. Plugging this index into the RNG that is being used to generate the RNS yields a specific random number value, which in turn yields a specific IP pair. That is, an IP pair can be generated directly from knowledge of the RNG, seed, and index number; it is not necessary, in this scheme, to generate the entire sequence of random numbers that precede the sequence value associated with the index number provided.

Since the communicants have presumably previously exchanged RNGs and seeds, the only new information that must be provided in order to generate an IP pair is the sequence number. If this number is provided by the sender in the packet header, then the receiver need only plug this number into the RNG in order to generate an IP pair – and thus verify that the IP pair appearing in the header of the packet is valid. In this scheme, if the sender and receiver lose synchronization, the receiver can immediately re-synchronize upon receipt of a single packet by simply comparing the IP pair in the packet header to the IP pair generated from the index number. Thus, synchronized communications can be resumed upon receipt of a single packet, making this scheme ideal for multicast communications. Taken to the extreme, it could obviate the need for synchronization tables entirely; that is, the sender and receiver could simply rely on the index number in the sync field to validate the IP pair on each packet, and thereby eliminate the tables entirely.

The aforementioned scheme may have some inherent security issues associated with it — namely, the placement of the sync field. If the field is placed in the outer header, then an interloper could observe the values of the field and their relationship to the IP stream. This could

potentially compromise the algorithm that is being used to generate the IP-address sequence, which would compromise the security of the communications. If, however, the value is placed in the inner header, then the sender must decrypt the inner header before it can extract the sync value and validate the IP pair; this opens up the receiver to certain types of denial-of-service (DoS) attacks, such as packet replay. That is, if the receiver must decrypt a packet before it can validate the IP pair, then it could potentially be forced to expend a significant amount of processing on decryption if an attacker simply retransmits previously valid packets. Other attack methodologies are possible in this scenario.

10 A possible compromise between algorithm security and processing speed is to split up the sync value between an inner (encrypted) and outer (unencrypted) header. That is, if the sync value is sufficiently long, it could potentially be split into a rapidly-changing part that can be viewed in the clear, and a fixed (or very slowly changing) part that must be protected. The part that can be viewed in the clear will be called the “public sync” portion and the part that must be protected will be called the “private sync” portion.

15 Both the public sync and private sync portions are needed to generate the complete sync value. The private portion, however, can be selected such that it is fixed or will change only occasionally. Thus, the private sync value can be stored by the recipient, thereby obviating the need to decrypt the header in order to retrieve it. If the sender and receiver have previously agreed upon the frequency with which the private part of the sync will change, then the receiver can selectively decrypt a single header in order to extract the new private sync if the communications gap that has led to lost synchronization has exceeded the lifetime of the previous private sync. This should not represent a burdensome amount of decryption, and thus should not open up the receiver to denial-of-service attack simply based on the need to occasionally decrypt a single header.

25 One implementation of this is to use a hashing function with a one-to-one mapping to generate the private and public sync portions from the sync value. This implementation is shown in FIG. 13, where (for example) a first ISP 1302 is the sender and a second ISP 1303 is the receiver. (Other alternatives are possible from FIG. 13.) A transmitted packet comprises a public

or “outer” header 1305 that is not encrypted, and a private or “inner” header 1306 that is encrypted using for example a link key. Outer header 1305 includes a public sync portion while inner header 1306 contains the private sync portion. A receiving node decrypts the inner header using a decryption function 1307 in order to extract the private sync portion. This step is necessary only if the lifetime of the currently buffered private sync has expired. (If the currently-buffered private sync is still valid, then it is simply extracted from memory and “added” (which could be an inverse hash) to the public sync, as shown in step 1308.) The public and decrypted private sync portions are combined in function 1308 in order to generate the combined sync 1309. The combined sync (1309) is then fed into the RNG (1310) and compared to the IP address pair (1311) to validate or reject the packet.

An important consideration in this architecture is the concept of “future” and “past” where the public sync values are concerned. Though the sync values, themselves, should be random to prevent spoofing attacks, it may be important that the receiver be able to quickly identify a sync value that has already been sent — even if the packet containing that sync value was never actually received by the receiver. One solution is to hash a time stamp or sequence number into the public sync portion, which could be quickly extracted, checked, and discarded, thereby validating the public sync portion itself.

In one embodiment, packets can be checked by comparing the source/destination IP pair generated by the sync field with the pair appearing in the packet header. If (1) they match, (2) the time stamp is valid, and (3) the dead-man timer has expired, then re-synchronization occurs; otherwise, the packet is rejected. If enough processing power is available, the dead-man timer and synchronization tables can be avoided altogether, and the receiver would simply resynchronize (e.g., validate) on every packet.

The foregoing scheme may require large-integer (e.g., 160-bit) math, which may affect its implementation. Without such large-integer registers, processing throughput would be affected, thus potentially affecting security from a denial-of-service standpoint. Nevertheless, as large-integer math processing features become more prevalent, the costs of implementing such a feature will be reduced.

#### D. Other Synchronization Schemes

As explained above, if  $W$  or more consecutive packets are lost between a transmitter and receiver in a VPN (where  $W$  is the window size), the receiver's window will not have been updated and the transmitter will be transmitting packets not in the receiver's window. The sender and receiver will not recover synchronization until perhaps the random pairs in the window are repeated by chance. Therefore, there is a need to keep a transmitter and receiver in synchronization whenever possible and to re-establish synchronization whenever it is lost.

A "checkpoint" scheme can be used to regain synchronization between a sender and a receiver that have fallen out of synchronization. In this scheme, a checkpoint message comprising a random IP address pair is used for communicating synchronization information. In one embodiment, two messages are used to communicate synchronization information between a sender and a recipient:

1. SYNC\_REQ is a message used by the sender to indicate that it wants to synchronize; and
2. SYNC\_ACK is a message used by the receiver to inform the transmitter that it has been synchronized.

According to one variation of this approach, both the transmitter and receiver maintain three checkpoints (see FIG. 14):

1. In the transmitter, ckpt\_o ("checkpoint old") is the IP pair that was used to re-send the last SYNC\_REQ packet to the receiver. In the receiver, ckpt\_o ("checkpoint old") is the IP pair that receives repeated SYNC\_REQ packets from the transmitter.
2. In the transmitter, ckpt\_n ("checkpoint new") is the IP pair that will be used to send the next SYNC\_REQ packet to the receiver. In the receiver, ckpt\_n ("checkpoint new") is the IP pair that receives a new SYNC\_REQ packet from the transmitter and which causes the receiver's window to be re-aligned, ckpt\_o set to ckpt\_n, a new ckpt\_n to be generated and a new ckpt\_r to be generated.
3. In the transmitter, ckpt\_r is the IP pair that will be used to send the next SYNC\_ACK packet to the receiver. In the receiver, ckpt\_r is the IP pair that receives a new

SYNC\_ACK packet from the transmitter and which causes a new ckpt\_n to be generated. Since SYNC\_ACK is transmitted from the receiver ISP to the sender ISP, the transmitter ckpt\_r refers to the ckpt\_r of the receiver and the receiver ckpt\_r refers to the ckpt\_r of the transmitter (see FIG. 14).

- 5 When a transmitter initiates synchronization, the IP pair it will use to transmit the next data packet is set to a predetermined value and when a receiver first receives a SYNC\_REQ, the receiver window is updated to be centered on the transmitter's next IP pair. This is the primary mechanism for checkpoint synchronization.

10 Synchronization can be initiated by a packet counter (e.g., after every N packets transmitted, initiate a synchronization) or by a timer (every S seconds, initiate a synchronization) or a combination of both. See FIG. 15. From the transmitter's perspective, this technique operates as follows: (1) Each transmitter periodically transmits a "sync request" message to the receiver to make sure that it is in sync. (2) If the receiver is still in sync, it sends back a "sync ack" message. (If this works, no further action is necessary). (3) If no "sync ack" has been  
 15 received within a period of time, the transmitter retransmits the sync request again. If the transmitter reaches the next checkpoint without receiving a "sync ack" response, then synchronization is broken, and the transmitter should stop transmitting. The transmitter will continue to send sync\_reqs until it receives a sync\_ack, at which point transmission is reestablished.

20 From the receiver's perspective, the scheme operates as follows: (1) when it receives a "sync request" request from the transmitter, it advances its window to the next checkpoint position (even skipping pairs if necessary), and sends a "sync ack" message to the transmitter. If sync was never lost, then the "jump ahead" really just advances to the next available pair of addresses in the table (i.e., normal advancement).

25 If an interloper intercepts the "sync request" messages and tries to interfere with communication by sending new ones, it will be ignored if the synchronization has been established or it it will actually help to re-establish synchronization.



A window is realigned whenever a re-synchronization occurs. This realignment entails updating the receiver's window to straddle the address pairs used by the packet transmitted immediately after the transmission of the SYNC\_REQ packet. Normally, the transmitter and receiver are in synchronization with one another. However, when network events occur, the receiver's window may have to be advanced by many steps during resynchronization. In this case, it is desirable to move the window ahead without having to step through the intervening random numbers sequentially. (This feature is also desirable for the auto-sync approach discussed above).

**E. Random Number Generator with a Jump-Ahead capability**

An attractive method for generating randomly hopped addresses is to use identical random number generators in the transmitter and receiver and advance them as packets are transmitted and received. There are many random number generation algorithms that could be used. Each one has strengths and weaknesses for address hopping applications.

Linear congruential random number generators (LCRs) are fast, simple and well characterized random number generators that can be made to jump ahead *n* steps efficiently. An LCR generates random numbers  $X_1, X_2, X_3 \dots X_k$  starting with seed  $X_0$  using a recurrence

$$X_i = (a X_{i-1} + b) \text{ mod } c, \quad (1)$$

where *a*, *b* and *c* define a particular LCR. Another expression for  $X_i$ ,

$$X_i = ((a^i(X_0 + b) - b) / (a - 1)) \text{ mod } c \quad (2)$$

enables the jump-ahead capability. The factor  $a^i$  can grow very large even for modest *i* if left unfettered. Therefore some special properties of the modulo operation can be used to control the size and processing time required to compute (2). (2) can be rewritten as:

$$X_i = (a^i (X_0(a-1) + b) - b) / (a-1) \text{ mod } c. \quad (3)$$

It can be shown that:

$$(a^i(X_0(a-1) + b) - b) / (a-1) \text{ mod } c = ((a^i \text{ mod } ((a-1)c)(X_0(a-1) + b) - b) / (a-1)) \text{ mod } c \quad (4).$$

$(X_0(a-1) + b)$  can be stored as  $(X_0(a-1) + b) \text{ mod } c$ , *b* as  $b \text{ mod } c$  and compute  $a^i \text{ mod } ((a-1)c)$  (this requires  $O(\log(i))$  steps).

A practical implementation of this algorithm would jump a fixed distance,  $n$ , between synchronizations; this is tantamount to synchronizing every  $n$  packets. The window would commence  $n$  IP pairs from the start of the previous window. Using  $X_j^w$ , the random number at the  $j^{\text{th}}$  checkpoint, as  $X_0$  and  $n$  as  $i$ , a node can store  $a^n \bmod((a-1)c)$  once per LCR and set

$$5 \quad X_{j+1}^w = X_{n(j+1)} = ((a^n \bmod((a-1)c) (X_j^w (a-1) + b) - b) / (a-1)) \bmod c, \quad (5)$$

to generate the random number for the  $j+1^{\text{th}}$  synchronization. Using this construction, a node could jump ahead an arbitrary (but fixed) distance between synchronizations in a constant amount of time (independent of  $n$ ).

Pseudo-random number generators, in general, and LCRs, in particular, will eventually repeat their cycles. This repetition may present vulnerability in the IP hopping scheme. An adversary would simply have to wait for a repeat to predict future sequences. One way of coping with this vulnerability is to create a random number generator with a known long cycle. A random sequence can be replaced by a new random number generator before it repeats. LCRs can be constructed with known long cycles. This is not currently true of many random number generators.

Random number generators can be cryptographically insecure. An adversary can derive the RNG parameters by examining the output or part of the output. This is true of LCGs. This vulnerability can be mitigated by incorporating an encryptor, designed to scramble the output as part of the random number generator. The random number generator prevents an adversary from mounting an attack—e.g., a known plaintext attack—against the encryptor.

#### **F. Random Number Generator Example**

Consider a RNG where  $a=31, b=4$  and  $c=15$ . For this case equation (1) becomes:

$$X_i = (31 X_{i-1} + 4) \bmod 15. \quad (6)$$

If one sets  $X_0=1$ , equation (6) will produce the sequence 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 0, 4, 8, 12. This sequence will repeat indefinitely. For a jump ahead of 3 numbers in this sequence  $a^n = 31^3 = 29791$ ,  $c*(a-1) = 15*30 = 450$  and  $a^n \bmod((a-1)c) = 31^3 \bmod(15*30) = 29791 \bmod(450) = 91$ . Equation (5) becomes:

$$((91 (X_i 30 + 4) - 4) / 30) \bmod 15 \quad (7).$$

Table 1 shows the jump ahead calculations from (7) . The calculations start at 5 and jump ahead 3.

TABLE 1

I	$X_i$	$(X_i30+4)$	$91 (X_i30+4)-4$	$((91 (X_i30+4)-4)/30)$	$X_{i+3}$
1	5	154	14010	467	2
4	2	64	5820	194	14
7	14	424	38580	1286	11
10	11	334	30390	1013	8
13	8	244	22200	740	5

5

### G. Fast Packet Filter

Address hopping VPNs must rapidly determine whether a packet has a valid header and thus requires further processing, or has an invalid header (a hostile packet) and should be immediately rejected. Such rapid determinations will be referred to as “fast packet filtering.” This capability protects the VPN from attacks by an adversary who streams hostile packets at the receiver at a high rate of speed in the hope of saturating the receiver’s processor (a so-called “denial of service” attack). Fast packet filtering is an important feature for implementing VPNs on shared media such as Ethernet.

10

Assuming that all participants in a VPN share an unassigned “A” block of addresses, one possibility is to use an experimental “A” block that will never be assigned to any machine that is not address hopping on the shared medium. “A” blocks have a 24 bits of address that can be hopped as opposed to the 8 bits in “C” blocks. In this case a hopblock will be the “A” block. The use of the experimental “A” block is a likely option on an Ethernet because:

15

1. The addresses have no validity outside of the Ethernet and will not be routed out to a valid outside destination by a gateway.
2. There are  $2^{24}$  (~16 million) addresses that can be hopped within each “A” block. This yields >280 trillion possible address pairs making it very unlikely that an adversary would guess a

20

valid address. It also provides acceptably low probability of collision between separate VPNs (all VPNs on a shared medium independently generate random address pairs from the same “A” block).

3. The packets will not be received by someone on the Ethernet who is not on a VPN (unless the machine is in promiscuous mode) minimizing impact on non-VPN computers.

The Ethernet example will be used to describe one implementation of fast packet filtering. The ideal algorithm would quickly examine a packet header, determine whether the packet is hostile, and reject any hostile packets or determine which active IP pair the packet header matches. The problem is a classical associative memory problem. A variety of techniques have been developed to solve this problem (hashing, B-trees etc). Each of these approaches has its strengths and weaknesses. For instance, hash tables can be made to operate quite fast in a statistical sense, but can occasionally degenerate into a much slower algorithm. This slowness can persist for a period of time. Since there is a need to discard hostile packets quickly at all times, hashing would be unacceptable.

#### H. Presence Vector Algorithm

A presence vector is a bit vector of length  $2^n$  that can be indexed by  $n$ -bit numbers (each ranging from 0 to  $2^n-1$ ). One can indicate the presence of  $k$   $n$ -bit numbers (not necessarily unique), by setting the bits in the presence vector indexed by each number to 1. Otherwise, the bits in the presence vector are 0. An  $n$ -bit number,  $x$ , is one of the  $k$  numbers if and only if the  $x^{\text{th}}$  bit of the presence vector is 1. A fast packet filter can be implemented by indexing the presence vector and looking for a 1, which will be referred to as the “test.”

For example, suppose one wanted to represent the number 135 using a presence vector. The 135<sup>th</sup> bit of the vector would be set. Consequently, one could very quickly determine whether an address of 135 was valid by checking only one bit: the 135<sup>th</sup> bit. The presence vectors could be created in advance corresponding to the table entries for the IP addresses. In effect, the incoming addresses can be used as indices into a long vector, making comparisons very fast. As each RNG generates a new address, the presence vector is updated to reflect the

information. As the window moves, the presence vector is updated to zero out addresses that are no longer valid.

There is a trade-off between efficiency of the test and the amount of memory required for storing the presence vector(s). For instance, if one were to use the 48 bits of hopping addresses as an index, the presence vector would have to be 35 terabytes. Clearly, this is too large for practical purposes. Instead, the 48 bits can be divided into several smaller fields. For instance, one could subdivide the 48 bits into four 12-bit fields (see FIG. 16). This reduces the storage requirement to 2048 bytes at the expense of occasionally having to process a hostile packet. In effect, instead of one long presence vector, the decomposed address portions must match all four shorter presence vectors before further processing is allowed. (If the first part of the address portion doesn't match the first presence vector, there is no need to check the remaining three presence vectors).

A presence vector will have a 1 in the  $y^{\text{th}}$  bit if and only if one or more addresses with a corresponding field of  $y$  are active. An address is active only if each presence vector indexed by the appropriate sub-field of the address is 1.

Consider a window of 32 active addresses and 3 checkpoints. A hostile packet will be rejected by the indexing of one presence vector more than 99% of the time. A hostile packet will be rejected by the indexing of all 4 presence vectors more than 99.9999995% of the time. On average, hostile packets will be rejected in less than 1.02 presence vector index operations.

The small percentage of hostile packets that pass the fast packet filter will be rejected when matching pairs are not found in the active window or are active checkpoints. Hostile packets that serendipitously match a header will be rejected when the VPN software attempts to decrypt the header. However, these cases will be extremely rare. There are many other ways this method can be configured to arbitrate the space/speed tradeoffs.

### **I. Further Synchronization Enhancements**

A slightly modified form of the synchronization techniques described above can be employed. The basic principles of the previously described checkpoint synchronization scheme remain unchanged. The actions resulting from the reception of the checkpoints are, however,

slightly different. In this variation, the receiver will maintain between OoO (“Out of Order”) and  $2 \times \text{WINDOW\_SIZE} + \text{OoO}$  active addresses ( $1 \leq \text{OoO} \leq \text{WINDOW\_SIZE}$  and  $\text{WINDOW\_SIZE} \geq 1$ ). OoO and WINDOW\_SIZE are engineerable parameters, where OoO is the minimum number of addresses needed to accommodate lost packets due to events in the network or out of order arrivals and WINDOW\_SIZE is the number of packets transmitted before a SYNC\_REQ is issued. FIG. 17 depicts a storage array for a receiver’s active addresses.

The receiver starts with the first  $2 \times \text{WINDOW\_SIZE}$  addresses loaded and active (ready to receive data). As packets are received, the corresponding entries are marked as “used” and are no longer eligible to receive packets. The transmitter maintains a packet counter, initially set to 0, containing the number of data packets transmitted since the last *initial* transmission of a SYNC\_REQ for which SYNC\_ACK has been received. When the transmitter packet counter equals WINDOW\_SIZE, the transmitter generates a SYNC\_REQ and does its initial transmission. When the receiver receives a SYNC\_REQ corresponding to its current CKPT\_N, it generates the next WINDOW\_SIZE addresses and starts loading them in order starting at the first location after the last active address wrapping around to the beginning of the array after the end of the array has been reached. The receiver’s array might look like FIG. 18 when a SYNC\_REQ has been received. In this case a couple of packets have been either lost or will be received out of order when the SYNC\_REQ is received.

FIG. 19 shows the receiver’s array after the new addresses have been generated. If the transmitter does not receive a SYNC\_ACK, it will re-issue the SYNC\_REQ at regular intervals. When the transmitter receives a SYNC\_ACK, the packet counter is decremented by WINDOW\_SIZE. If the packet counter reaches  $2 \times \text{WINDOW\_SIZE} - \text{OoO}$  then the transmitter ceases sending data packets until the appropriate SYNC\_ACK is finally received. The transmitter then resumes sending data packets. Future behavior is essentially a repetition of this initial cycle. The advantages of this approach are:

1. There is no need for an efficient jump ahead in the random number generator,
2. No packet is ever transmitted that does not have a corresponding entry in the receiver side

3. No timer based re-synchronization is necessary. This is a consequence of 2.
4. The receiver will always have the ability to accept data messages transmitted within OoO messages of the most recently transmitted message.

#### **J. Distributed Transmission Path Variant**

5           Another embodiment incorporating various inventive principles is shown in FIG. 20. In this embodiment, a message transmission system includes a first computer 2001 in communication with a second computer 2002 through a network 2011 of intermediary computers. In one variant of this embodiment, the network includes two edge routers 2003 and 2004 each of which is linked to a plurality of Internet Service Providers (ISPs) 2005 through 10 2010. Each ISP is coupled to a plurality of other ISPs in an arrangement as shown in FIG. 20, which is a representative configuration only and is not intended to be limiting. Each connection between ISPs is labeled in FIG. 20 to indicate a specific physical transmission path (e.g., AD is a physical path that links ISP A (element 2005) to ISP D (element 2008)). Packets arriving at each edge router are selectively transmitted to one of the ISPs to which the router is attached on the 15 basis of a randomly or quasi-randomly selected basis.

          As shown in FIG. 21, computer 2001 or edge router 2003 incorporates a plurality of link transmission tables 2100 that identify, for each potential transmission path through the network, valid sets of IP addresses that can be used to transmit the packet. For example, AD table 2101 contains a plurality of IP source/destination pairs that are randomly or quasi-randomly generated. 20 When a packet is to be transmitted from first computer 2001 to second computer 2002, one of the link tables is randomly (or quasi-randomly) selected, and the next valid source/destination address pair from that table is used to transmit the packet through the network. If path AD is randomly selected, for example, the next source/destination IP address pair (which is pre-determined to transmit between ISP A (element 2005) and ISP B (element 2008)) is used to 25 transmit the packet. If one of the transmission paths becomes degraded or inoperative, that link table can be set to a “down” condition as shown in table 2105, thus preventing addresses from being selected from that table. Other transmission paths would be unaffected by this broken link.

### **3. CONTINUATION-IN-PART IMPROVEMENTS**

The following describes various improvements and features that can be applied to the embodiments described above. The improvements include: (1) a load balancer that distributes packets across different transmission paths according to transmission path quality; (2) a DNS proxy server that transparently creates a virtual private network in response to a domain name inquiry; (3) a large-to-small link bandwidth management feature that prevents denial-of-service attacks at system chokepoints; (4) a traffic limiter that regulates incoming packets by limiting the rate at which a transmitter can be synchronized with a receiver; and (5) a signaling synchronizer that allows a large number of nodes to communicate with a central node by partitioning the communication function between two separate entities. Each is discussed separately below.

#### **A. Load Balancer**

Various embodiments described above include a system in which a transmitting node and a receiving node are coupled through a plurality of transmission paths, and wherein successive packets are distributed quasi-randomly over the plurality of paths. See, for example, FIGS. 20 and 21 and accompanying description. The improvement extends this basic concept to encompass distributing packets across different paths in such a manner that the loads on the paths are generally balanced according to transmission link quality.

In one embodiment, a system includes a transmitting node and a receiving node that are linked via a plurality of transmission paths having potentially varying transmission quality. Successive packets are transmitted over the paths based on a weight value distribution function for each path. The rate that packets will be transmitted over a given path can be different for each path. The relative "health" of each transmission path is monitored in order to identify paths that have become degraded. In one embodiment, the health of each path is monitored in the transmitter by comparing the number of packets transmitted to the number of packet acknowledgements received. Each transmission path may comprise a physically separate path (e.g., via dial-up phone line, computer network, router, bridge, or the like), or may comprise logically separate paths contained within a broadband communication medium (e.g., separate



channels in an FDM, TDM, CDMA, or other type of modulated or unmodulated transmission link).

When the transmission quality of a path falls below a predetermined threshold and there are other paths that can transmit packets, the transmitter changes the weight value used for that path, making it less likely that a given packet will be transmitted over that path. The weight will preferably be set no lower than a minimum value that keeps nominal traffic on the path. The weights of the other available paths are altered to compensate for the change in the affected path. When the quality of a path degrades to where the transmitter is turned off by the synchronization function (i.e., no packets are arriving at the destination), the weight is set to zero. If all transmitters are turned off, no packets are sent.

Conventional TCP/IP protocols include a "throttling" feature that reduces the transmission rate of packets when it is determined that delays or errors are occurring in transmission. In this respect, timers are sometimes used to determine whether packets have been received. These conventional techniques for limiting transmission of packets, however, do not involve multiple transmission paths between two nodes wherein transmission across a particular path relative to the others is changed based on link quality.

According to certain embodiments, in order to damp oscillations that might otherwise occur if weight distributions are changed drastically (e.g., according to a step function), a linear or an exponential decay formula can be applied to gradually decrease the weight value over time that a degrading path will be used. Similarly, if the health of a degraded path improves, the weight value for that path is gradually increased.

Transmission link health can be evaluated by comparing the number of packets that are acknowledged within the transmission window (see embodiments discussed above) to the number of packets transmitted within that window and by the state of the transmitter (i.e., on or off). In other words, rather than accumulating general transmission statistics over time for a path, one specific implementation uses the "windowing" concepts described above to evaluate transmission path health.

The same scheme can be used to shift virtual circuit paths from an “unhealthy” path to a “healthy” one, and to select a path for a new virtual circuit.

FIG. 22A shows a flowchart for adjusting weight values associated with a plurality of transmission links. It is assumed that software executing in one or more computer nodes executes the steps shown in FIG. 22A. It is also assumed that the software can be stored on a computer-readable medium such as a magnetic or optical disk for execution by a computer.

Beginning in step 2201, the transmission quality of a given transmission path is measured. As described above, this measurement can be based on a comparison between the number of packets transmitted over a particular link to the number of packet acknowledgements received over the link (e.g., per unit time, or in absolute terms). Alternatively, the quality can be evaluated by comparing the number of packets that are acknowledged within the transmission window to the number of packets that were transmitted within that window. In yet another variation, the number of missed synchronization messages can be used to indicate link quality. Many other variations are of course possible.

In step 2202, a check is made to determine whether more than one transmitter (e.g., transmission path) is turned on. If not, the process is terminated and resumes at step 2201.

In step 2203, the link quality is compared to a given threshold (e.g., 50%, or any arbitrary number). If the quality falls below the threshold, then in step 2207 a check is made to determine whether the weight is above a minimum level (e.g., 1%). If not, then in step 2209 the weight is set to the minimum level and processing resumes at step 2201. If the weight is above the minimum level, then in step 2208 the weight is gradually decreased for the path, then in step 2206 the weights for the remaining paths are adjusted accordingly to compensate (e.g., they are increased).

If in step 2203 the quality of the path was greater than or equal to the threshold, then in step 2204 a check is made to determine whether the weight is less than a steady-state value for that path. If so, then in step 2205 the weight is increased toward the steady-state value, and in step 2206 the weights for the remaining paths are adjusted accordingly to compensate (e.g., they

are decreased). If in step 2204 the weight is not less than the steady-state value, then processing resumes at step 2201 without adjusting the weights.

The weights can be adjusted incrementally according to various functions, preferably by changing the value gradually. In one embodiment, a linearly decreasing function is used to adjust the weights; according to another embodiment, an exponential decay function is used. Gradually changing the weights helps to damp oscillators that might otherwise occur if the probabilities were abruptly.

Although not explicitly shown in FIG. 22A the process can be performed only periodically (e.g., according to a time schedule), or it can be continuously run, such as in a background mode of operation. In one embodiment, the combined weights of all potential paths should add up to unity (e.g., when the weighting for one path is decreased, the corresponding weights that the other paths will be selected will increase).

Adjustments to weight values for other paths can be prorated. For example, a decrease of 10% in weight value for one path could result in an evenly distributed increase in the weights for the remaining paths. Alternatively, weightings could be adjusted according to a weighted formula as desired (e.g., favoring healthy paths over less healthy paths). In yet another variation, the difference in weight value can be amortized over the remaining links in a manner that is proportional to their traffic weighting.

FIG. 22B shows steps that can be executed to shut down transmission links where a transmitter turns off. In step 2210, a transmitter shut-down event occurs. In step 2211, a test is made to determine whether at least one transmitter is still turned on. If not, then in step 2215 all packets are dropped until a transmitter turns on. If in step 2211 at least one transmitter is turned on, then in step 2212 the weight for the path is set to zero, and the weights for the remaining paths are adjusted accordingly.

FIG. 23 shows a computer node 2301 employing various principles of the above-described embodiments. It is assumed that two computer nodes of the type shown in FIG. 23 communicate over a plurality of separate physical transmission paths. As shown in FIG. 23, four transmission paths X1 through X4 are defined for communicating between the two nodes. Each

node includes a packet transmitter 2302 that operates in accordance with a transmit table 2308 as described above. (The packet transmitter could also operate without using the IP-hopping features described above, but the following description assumes that some form of hopping is employed in conjunction with the path selection mechanism.). The computer node also includes  
5 a packet receiver 2303 that operates in accordance with a receive table 2309, including a moving window *W* that moves as valid packets are received. Invalid packets having source and destination addresses that do not fall within window *W* are rejected.

As each packet is readied for transmission, source and destination IP addresses (or other discriminator values) are selected from transmit table 2308 according to any of the various  
10 algorithms described above, and packets containing these source/destination address pairs, which correspond to the node to which the four transmission paths are linked, are generated to a transmission path switch 2307. Switch 2307, which can comprise a software function, selects from one of the available transmission paths according to a weight distribution table 2306. For example, if the weight for path X1 is 0.2, then every fifth packet will be transmitted on path X1.  
15 A similar regime holds true for the other paths as shown. Initially, each link's weight value can be set such that it is proportional to its bandwidth, which will be referred to as its "steady-state" value.

Packet receiver 2303 generates an output to a link quality measurement function 2304 that operates as described above to determine the quality of each transmission path. (The input  
20 to packet receiver 2303 for receiving incoming packets is omitted for clarity). Link quality measurement function 2304 compares the link quality to a threshold for each transmission link and, if necessary, generates an output to weight adjustment function 2305. If a weight adjustment is required, then the weights in table 2306 are adjusted accordingly, preferably according to a gradual (e.g., linearly or exponentially declining) function. In one embodiment,  
25 the weight values for all available paths are initially set to the same value, and only when paths degrade in quality are the weights changed to reflect differences.

Link quality measurement function 2304 can be made to operate as part of a synchronizer function as described above. That is, if resynchronization occurs and the receiver detects that

synchronization has been lost (e.g., resulting in the synchronization window W being advanced out of sequence), that fact can be used to drive link quality measurement function 2304. According to one embodiment, load balancing is performed using information garnered during the normal synchronization, augmented slightly to communicate link health from the receiver to the transmitter. The receiver maintains a count, MESS\_R(W), of the messages received in synchronization window W. When it receives a synchronization request (SYNC\_REQ) corresponding to the end of window W, the receiver includes counter MESS\_R in the resulting synchronization acknowledgement (SYNC\_ACK) sent back to the transmitter. This allows the transmitter to compare messages sent to messages received in order to assess the health of the link.

If synchronization is completely lost, weight adjustment function 2305 decreases the weight value on the affected path to zero. When synchronization is regained, the weight value for the affected path is gradually increased to its original value. Alternatively, link quality can be measured by evaluating the length of time required for the receiver to acknowledge a synchronization request. In one embodiment, separate transmit and receive tables are used for each transmission path.

When the transmitter receives a SYNC\_ACK, the MESS\_R is compared with the number of messages transmitted in a window (MESS\_T). When the transmitter receives a SYNC\_ACK, the traffic probabilities will be examined and adjusted if necessary. MESS\_R is compared with the number of messages transmitted in a window (MESS\_T). There are two possibilities:

1. If MESS\_R is less than a threshold value, THRESH, then the link will be deemed to be unhealthy. If the transmitter was turned off, the transmitter is turned on and the weight P for that link will be set to a minimum value MIN. This will keep a trickle of traffic on the link for monitoring purposes until it recovers. If the transmitter was turned on, the weight P for that link will be set to:

$$P' = \alpha \times \text{MIN} + (1 - \alpha) \times P \quad (1)$$

Equation 1 will exponentially damp the traffic weight value to MIN during sustained periods of degraded service.

2. If MESS\_R for a link is greater than or equal to THRESH, the link will be deemed healthy. If the weight P for that link is greater than or equal to the steady state value S for that link, then P is left unaltered. If the weight P for that link is less than THRESH then P will be set to:

$$5 \quad P' = \beta \times S + (1 - \beta) \times P \quad (2)$$

where  $\beta$  is a parameter such that  $0 \leq \beta \leq 1$  that determines the damping rate of P.

Equation 2 will increase the traffic weight to S during sustained periods of acceptable service in a damped exponential fashion.

10 A detailed example will now be provided with reference to FIG. 24. As shown in FIG. 24, a first computer 2401 communicates with a second computer 2402 through two routers 2403 and 2404. Each router is coupled to the other router through three transmission links. As described above, these may be physically diverse links or logical links (including virtual private networks).

15 Suppose that a first link L1 can sustain a transmission bandwidth of 100 Mb/s and has a window size of 32; link L2 can sustain 75 Mb/s and has a window size of 24; and link L3 can sustain 25 Mb/s and has a window size of 8. The combined links can thus sustain 200Mb/s. The steady state traffic weights are 0.5 for link L1; 0.375 for link L2, and 0.125 for link L3. MIN=1Mb/s, THRESH =0.8 MESS\_T for each link,  $\alpha=.75$  and  $\beta=.5$ . These traffic weights will remain stable until a link stops for synchronization or reports a number of packets received less than its THRESH. Consider the following sequence of events:

20

1. Link L1 receives a SYNC\_ACK containing a MESS\_R of 24, indicating that only 75% of the MESS\_T (32) messages transmitted in the last window were successfully received. Link 1 would be below THRESH (0.8). Consequently, link L1's traffic weight value would be reduced to 0.12825, while link L2's traffic weight value would be increased to 0.65812 and link L3's traffic weight value would be increased to 0.217938.

25

2. Link L2 and L3 remained healthy and link L1 stopped to synchronize. Then link L1's traffic weight value would be set to 0, link L2's traffic weight value would be set to 0.75, and link L3's traffic weight value would be set to 0.25.

3. Link L1 finally received a SYNC\_ACK containing a MESS\_R of 0 indicating that none of the MESS\_T (32) messages transmitted in the last window were successfully received. Link L1 would be below THRESH. Link L1's traffic weight value would be increased to .005, link L2's traffic weight value would be decreased to 0.74625, and link L3's traffic weight value would be decreased to 0.24875.

4. Link L1 received a SYNC\_ACK containing a MESS\_R of 32 indicating that 100% of the MESS\_T (32) messages transmitted in the last window were successfully received. Link L1 would be above THRESH. Link L1's traffic weight value would be increased to 0.2525, while link L2's traffic weight value would be decreased to 0.560625 and link L3's traffic weight value would be decreased to .186875.

5. Link L1 received a SYNC\_ACK containing a MESS\_R of 32 indicating that 100% of the MESS\_T (32) messages transmitted in the last window were successfully received. Link L1 would be above THRESH. Link L1's traffic weight value would be increased to 0.37625; link L2's traffic weight value would be decreased to 0.4678125, and link L3's traffic weight value would be decreased to 0.1559375.

6. Link L1 remains healthy and the traffic probabilities approach their steady state traffic probabilities.

#### **B. Use of a DNS Proxy to Transparently Create Virtual Private Networks**

A second improvement concerns the automatic creation of a virtual private network (VPN) in response to a domain-name server look-up function.

Conventional Domain Name Servers (DNSs) provide a look-up function that returns the IP address of a requested computer or host. For example, when a computer user types in the web name "Yahoo.com," the user's web browser transmits a request to a DNS, which converts the name into a four-part IP address that is returned to the user's browser and then used by the browser to contact the destination web site.

This conventional scheme is shown in FIG. 25. A user's computer 2501 includes a client application 2504 (for example, a web browser) and an IP protocol stack 2505. When the user enters the name of a destination host, a request DNS REQ is made (through IP protocol stack 2505) to a DNS 2502 to look up the IP address associated with the name. The DNS returns the IP address DNS RESP to client application 2504, which is then able to use the IP address to communicate with the host 2503 through separate transactions such as PAGE REQ and PAGE RESP.

In the conventional architecture shown in FIG. 25, nefarious listeners on the Internet could intercept the DNS REQ and DNS RESP packets and thus learn what IP addresses the user was contacting. For example, if a user wanted to set up a secure communication path with a web site having the name "Target.com," when the user's browser contacted a DNS to find the IP address for that web site, the true IP address of that web site would be revealed over the Internet as part of the DNS inquiry. This would hamper anonymous communications on the Internet.

One conventional scheme that provides secure virtual private networks over the Internet provides the DNS server with the public keys of the machines that the DNS server has the addresses for. This allows hosts to retrieve automatically the public keys of a host that the host is to communicate with so that the host can set up a VPN without having the user enter the public key of the destination host. One implementation of this standard is presently being developed as part of the FreeS/WAN project(RFC 2535).

The conventional scheme suffers from certain drawbacks. For example, any user can perform a DNS request. Moreover, DNS requests resolve to the same value for all users.

According to certain aspects of the invention, a specialized DNS server traps DNS requests and, if the request is from a special type of user (e.g., one for which secure communication services are defined), the server does not return the true IP address of the target node, but instead automatically sets up a virtual private network between the target node and the user. The VPN is preferably implemented using the IP address "hopping" features of the basic invention described above, such that the true identity of the two nodes cannot be determined even if packets during the communication are intercepted. For DNS requests that are determined



to not require secure services (e.g., an unregistered user), the DNS server transparently “passes through” the request to provide a normal look-up function and return the IP address of the target web server, provided that the requesting host has permissions to resolve unsecured sites. Different users who make an identical DNS request could be provided with different results.

5           FIG. 26 shows a system employing various principles summarized above. A user’s computer 2601 includes a conventional client (e.g., a web browser) 2605 and an IP protocol stack 2606 that preferably operates in accordance with an IP hopping function 2607 as outlined above. A modified DNS server 2602 includes a conventional DNS server function 2609 and a DNS proxy 2610. A gatekeeper server 2603 is interposed between the modified DNS server and  
10 a secure target site 2704. An “unsecure” target site 2611 is also accessible via conventional IP protocols.

          According to one embodiment, DNS proxy 2610 intercepts all DNS lookup functions from client 2605 and determines whether access to a secure site has been requested. If access to a secure site has been requested (as determined, for example, by a domain name extension, or by  
15 reference to an internal table of such sites), DNS proxy 2610 determines whether the user has sufficient security privileges to access the site. If so, DNS proxy 2610 transmits a message to gatekeeper 2603 requesting that a virtual private network be created between user computer 2601 and secure target site 2604. In one embodiment, gatekeeper 2603 creates “hopblocks” to be used by computer 2601 and secure target site 2604 for secure communication. Then, gatekeeper 2603  
20 communicates these to user computer 2601. Thereafter, DNS proxy 2610 returns to user computer 2601 the resolved address passed to it by the gatekeeper (this address could be different from the actual target computer) 2604, preferably using a secure administrative VPN. The address that is returned need not be the actual address of the destination computer.

          Had the user requested lookup of a non-secure web site such as site 2611, DNS proxy  
25 would merely pass through to conventional DNS server 2609 the look-up request, which would be handled in a conventional manner, returning the IP address of non-secure web site 2611. If the user had requested lookup of a secure web site but lacked credentials to create such a connection, DNS proxy 2610 would return a “host unknown” error to the user. In this manner,

different users requesting access to the same DNS name could be provided with different look-up results.

Gatekeeper 2603 can be implemented on a separate computer (as shown in FIG. 26) or as a function within modified DNS server 2602. In general, it is anticipated that gatekeeper 2703 facilitates the allocation and exchange of information needed to communicate securely, such as using “hopped” IP addresses. Secure hosts such as site 2604 are assumed to be equipped with a secure communication function such as an IP hopping function 2608.

It will be appreciated that the functions of DNS proxy 2610 and DNS server 2609 can be combined into a single server for convenience. Moreover, although element 2602 is shown as combining the functions of two servers, the two servers can be made to operate independently.

FIG. 27 shows steps that can be executed by DNS proxy server 2610 to handle requests for DNS look-up for secure hosts. In step 2701, a DNS look-up request is received for a target host. In step 2702, a check is made to determine whether access to a secure host was requested. If not, then in step 2703 the DNS request is passed to conventional DNS server 2609, which looks up the IP address of the target site and returns it to the user’s application for further processing.

In step 2702, if access to a secure host was requested, then in step 2704 a further check is made to determine whether the user is authorized to connect to the secure host. Such a check can be made with reference to an internally stored list of authorized IP addresses, or can be made by communicating with gatekeeper 2603 (e.g., over an “administrative” VPN that is secure). It will be appreciated that different levels of security can also be provided for different categories of hosts. For example, some sites may be designated as having a certain security level, and the security level of the user requesting access must match that security level. The user’s security level can also be determined by transmitting a request message back to the user’s computer requiring that it prove that it has sufficient privileges.

If the user is not authorized to access the secure site, then a “host unknown” message is returned (step 2705). If the user has sufficient security privileges, then in step 2706 a secure VPN is established between the user’s computer and the secure target site. As described above,

this is preferably done by allocating a hopping regime that will be carried out between the user's computer and the secure target site, and is preferably performed transparently to the user (i.e., the user need not be involved in creating the secure link). As described in various embodiments of this application, any of various fields can be "hopped" (e.g., IP source/destination addresses; a field in the header; etc.) in order to communicate securely.

Some or all of the security functions can be embedded in gatekeeper 2603, such that it handles all requests to connect to secure sites. In this embodiment, DNS proxy 2610 communicates with gatekeeper 2603 to determine (preferably over a secure administrative VPN) whether the user has access to a particular web site. Various scenarios for implementing these features are described by way of example below:

Scenario #1: Client has permission to access target computer, and gatekeeper has a rule to make a VPN for the client. In this scenario, the client's DNS request would be received by the DNS proxy server 2610, which would forward the request to gatekeeper 2603. The gatekeeper would establish a VPN between the client and the requested target. The gatekeeper would provide the address of the destination to the DNS proxy, which would then return the resolved name as a result. The resolved address can be transmitted back to the client in a secure administrative VPN.

Scenario #2: Client does not have permission to access target computer. In this scenario, the client's DNS request would be received by the DNS proxy server 2610, which would forward the request to gatekeeper 2603. The gatekeeper would reject the request, informing DNS proxy server 2610 that it was unable to find the target computer. The DNS proxy 2610 would then return a "host unknown" error message to the client.

Scenario #3: Client has permission to connect using a normal non-VPN link, and the gatekeeper does not have a rule to set up a VPN for the client to the target site. In this scenario, the client's DNS request is received by DNS proxy server 2610, which would check its rules and determine that no VPN is needed. Gatekeeper 2603 would then inform the DNS proxy server to forward the request to conventional DNS server 2609, which would resolve the request and return the result to the DNS proxy server and then back to the client.

Scenario #4: Client does not have permission to establish a normal/non-VPN link, and the gatekeeper does not have a rule to make a VPN for the client to the target site. In this scenario, the DNS proxy server would receive the client’s DNS request and forward it to gatekeeper 2603. Gatekeeper 2603 would determine that no special VPN was needed, but that the client is not authorized to communicate with non-VPN members. The gatekeeper would reject the request, causing DNS proxy server 2610 to return an error message to the client.

**C. Large Link to Small Link Bandwidth Management**

One feature of the basic architecture is the ability to prevent so-called “denial of service” attacks that can occur if a computer hacker floods a known Internet node with packets, thus preventing the node from communicating with other nodes. Because IP addresses or other fields are “hopped” and packets arriving with invalid addresses are quickly discarded, Internet nodes are protected against flooding targeted at a single IP address.

In a system in which a computer is coupled through a link having a limited bandwidth (e.g., an edge router) to a node that can support a much higher-bandwidth link (e.g., an Internet Service Provider), a potential weakness could be exploited by a determined hacker. Referring to FIG. 28, suppose that a first host computer 2801 is communicating with a second host computer 2804 using the IP address hopping principles described above. The first host computer is coupled through an edge router 2802 to an Internet Service Provider (ISP) 2803 through a low bandwidth link (LOW BW), and is in turn coupled to second host computer 2804 through parts of the Internet through a high bandwidth link (HIGH BW). In this architecture, the ISP is able to support a high bandwidth to the internet, but a much lower bandwidth to the edge router 2802.

Suppose that a computer hacker is able to transmit a large quantity of dummy packets addressed to first host computer 2801 across high bandwidth link HIGH BW. Normally, host computer 2801 would be able to quickly reject the packets since they would not fall within the acceptance window permitted by the IP address hopping scheme. However, because the packets must travel across low bandwidth link LOW BW, the packets overwhelm the lower bandwidth link before they are received by host computer 2801. Consequently, the link to host computer 2801 is effectively flooded before the packets can be discarded.

According to one inventive improvement, a “link guard” function 2805 is inserted into the high-bandwidth node (e.g., ISP 2803) that quickly discards packets destined for a low-bandwidth target node if they are not valid packets. Each packet destined for a low-bandwidth node is cryptographically authenticated to determine whether it belongs to a VPN. If it is not a valid VPN packet, the packet is discarded at the high-bandwidth node. If the packet is authenticated as belonging to a VPN, the packet is passed with high preference. If the packet is a valid non-VPN packet, it is passed with a lower quality of service (e.g., lower priority).

In one embodiment, the ISP distinguishes between VPN and non-VPN packets using the protocol of the packet. In the case of IPSEC [rfc 2401], the packets have IP protocols 420 and 421. In the case of the TARP VPN, the packets will have an IP protocol that is not yet defined. The ISP’s link guard, 2805, maintains a table of valid VPNs which it uses to validate whether VPN packets are cryptographically valid. According to one embodiment, packets that do not fall within any hop windows used by nodes on the low-bandwidth link are rejected, or are sent with a lower quality of service. One approach for doing this is to provide a copy of the IP hopping tables used by the low-bandwidth nodes to the high-bandwidth node, such that both the high-bandwidth and low-bandwidth nodes track hopped packets (e.g., the high-bandwidth node moves its hopping window as valid packets are received). In such a scenario, the high-bandwidth node discards packets that do not fall within the hopping window before they are transmitted over the low-bandwidth link. Thus, for example, ISP 2903 maintains a copy 2910 of the receive table used by host computer 2901. Incoming packets that do not fall within this receive table are discarded. According to a different embodiment, link guard 2805 validates each VPN packet using a keyed hashed message authentication code (HMAC) [rfc 2104].

According to another embodiment, separate VPNs (using, for example, hopblocks) can be established for communicating between the low-bandwidth node and the high-bandwidth node (i.e., packets arriving at the high-bandwidth node are converted into different packets before being transmitted to the low-bandwidth node).

As shown in FIG. 29, for example, suppose that a first host computer 2900 is communicating with a second host computer 2902 over the Internet, and the path includes a high

bandwidth link HIGH BW to an ISP 2901 and a low bandwidth link LOW BW through an edge router 2904. In accordance with the basic architecture described above, first host computer 2900 and second host computer 2902 would exchange hopblocks (or a hopblock algorithm) and would be able to create matching transmit and receive tables 2905, 2906, 2912 and 2913. Then in accordance with the basic architecture, the two computers would transmit packets having seemingly random IP source and destination addresses, and each would move a corresponding hopping window in its receive table as valid packets were received.

Suppose that a nefarious computer hacker 2903 was able to deduce that packets having a certain range of IP addresses (e.g., addresses 100 to 200 for the sake of simplicity) are being transmitted to ISP 2901, and that these packets are being forwarded over a low-bandwidth link. Hacker computer 2903 could thus “flood” packets having addresses falling into the range 100 to 200, expecting that they would be forwarded along low bandwidth link LOW BW, thus causing the low bandwidth link to become overwhelmed. The fast packet reject mechanism in first host computer 3000 would be of little use in rejecting these packets, since the low bandwidth link was effectively jammed before the packets could be rejected. In accordance with one aspect of the improvement, however, VPN link guard 2911 would prevent the attack from impacting the performance of VPN traffic because the packets would either be rejected as invalid VPN packets or given a lower quality of service than VPN traffic over the lower bandwidth link. A denial-of-service flood attack could, however, still disrupt non-VPN traffic.

According to one embodiment of the improvement, ISP 2901 maintains a separate VPN with first host computer 2900, and thus translates packets arriving at the ISP into packets having a different IP header before they are transmitted to host computer 2900. The cryptographic keys used to authenticate VPN packets at the link guard 2911 and the cryptographic keys used to encrypt and decrypt the VPN packets at host 2902 and host 2901 can be different, so that link guard 2911 does not have access to the private host data; it only has the capability to authenticate those packets.

According to yet a third embodiment, the low-bandwidth node can transmit a special message to the high-bandwidth node instructing it to shut down all transmissions on a particular

IP address, such that only hopped packets will pass through to the low-bandwidth node. This embodiment would prevent a hacker from flooding packets using a single IP address. According to yet a fourth embodiment, the high-bandwidth node can be configured to discard packets transmitted to the low-bandwidth node if the transmission rate exceeds a certain predetermined threshold for any given IP address; this would allow hopped packets to go through. In this respect, link guard 2911 can be used to detect that the rate of packets on a given IP address are exceeding a threshold rate; further packets addressed to that same IP address would be dropped or transmitted at a lower priority (e.g., delayed).

#### **D. Traffic Limiter**

In a system in which multiple nodes are communicating using “hopping” technology, a treasonous insider could internally flood the system with packets. In order to prevent this possibility, one inventive improvement involves setting up “contracts” between nodes in the system, such that a receiver can impose a bandwidth limitation on each packet sender. One technique for doing this is to delay acceptance of a checkpoint synchronization request from a sender until a certain time period (e.g., one minute) has elapsed. Each receiver can effectively control the rate at which its hopping window moves by delaying “SYNC ACK” responses to “SYNC\_REQ” messages.

A simple modification to the checkpoint synchronizer will serve to protect a receiver from accidental or deliberate overload from an internally treasonous client. This modification is based on the observation that a receiver will not update its tables until a SYNC\_REQ is received on hopped address CKPT\_N. It is a simple matter of deferring the generation of a new CKPT\_N until an appropriate interval after previous checkpoints.

Suppose a receiver wished to restrict reception from a transmitter to 100 packets a second, and that checkpoint synchronization messages were triggered every 50 packets. A compliant transmitter would not issue new SYNC\_REQ messages more often than every 0.5 seconds. The receiver could delay a non-compliant transmitter from synchronizing by delaying the issuance of CKPT\_N for 0.5 second after the last SYNC\_REQ was accepted.

In general, if M receivers need to restrict N transmitters issuing new SYNC\_REQ messages after every W messages to sending R messages a second in aggregate, each receiver could defer issuing a new CKPT\_N until  $M \times N \times W / R$  seconds have elapsed since the last SYNC\_REQ has been received and accepted. If the transmitter exceeds this rate between a pair of checkpoints, it will issue the new checkpoint before the receiver is ready to receive it, and the SYNC\_REQ will be discarded by the receiver. After this, the transmitter will re-issue the SYNC\_REQ every T1 seconds until it receives a SYNC\_ACK. The receiver will eventually update CKPT\_N and the SYNC\_REQ will be acknowledged. If the transmission rate greatly exceeds the allowed rate, the transmitter will stop until it is compliant. If the transmitter exceeds the allowed rate by a little, it will eventually stop after several rounds of delayed synchronization until it is in compliance. Hacking the transmitter's code to not shut off only permits the transmitter to lose the acceptance window. In this case it can recover the window and proceed only after it is compliant again.

Two practical issues should be considered when implementing the above scheme:

1. The receiver rate should be slightly higher than the permitted rate in order to allow for statistical fluctuations in traffic arrival times and non-uniform load balancing.

2. Since a transmitter will rightfully continue to transmit for a period after a SYNC\_REQ is transmitted, the algorithm above can artificially reduce the transmitter's bandwidth. If events prevent a compliant transmitter from synchronizing for a period (e.g. the network dropping a SYNC\_REQ or a SYNC\_ACK) a SYNC\_REQ will be accepted later than expected. After this, the transmitter will transmit fewer than expected messages before encountering the next checkpoint. The new checkpoint will not have been activated and the transmitter will have to retransmit the SYNC\_REQ. This will appear to the receiver as if the transmitter is not compliant. Therefore, the next checkpoint will be accepted late from the transmitter's perspective. This has the effect of reducing the transmitter's allowed packet rate until the transmitter transmits at a packet rate below the agreed upon rate for a period of time.

To guard against this, the receiver should keep track of the times that the last C SYNC\_REQs were received and accepted and use the minimum of  $M \times N \times W / R$  seconds after the



last SYNC\_REQ has been received and accepted,  $2xMxNxW/R$  seconds after next to the last SYNC\_REQ has been received and accepted,  $CxMxNxW/R$  seconds after  $(C-1)^{th}$  to the last SYNC\_REQ has been received, as the time to activate CKPT\_N. This prevents the receiver from inappropriately limiting the transmitter's packet rate if at least one out of the last C SYNC\_REQs was processed on the first attempt.

FIG. 30 shows a system employing the above-described principles. In FIG. 30, two computers 3000 and 3001 are assumed to be communicating over a network N in accordance with the "hopping" principles described above (e.g., hopped IP addresses, discriminator values, etc.). For the sake of simplicity, computer 3000 will be referred to as the receiving computer and computer 3001 will be referred to as the transmitting computer, although full duplex operation is of course contemplated. Moreover, although only a single transmitter is shown, multiple transmitters can transmit to receiver 3000.

As described above, receiving computer 3000 maintains a receive table 3002 including a window W that defines valid IP address pairs that will be accepted when appearing in incoming data packets. Transmitting computer 3001 maintains a transmit table 3003 from which the next IP address pairs will be selected when transmitting a packet to receiving computer 3000. (For the sake of illustration, window W is also illustrated with reference to transmit table 3003). As transmitting computer moves through its table, it will eventually generate a SYNC\_REQ message as illustrated in function 3010. This is a request to receiver 3000 to synchronize the receive table 3002, from which transmitter 3001 expects a response in the form of a CKPT\_N (included as part of a SYNC\_ACK message). If transmitting computer 3001 transmits more messages than its allotment, it will prematurely generate the SYNC\_REQ message. (If it has been altered to remove the SYNC\_REQ message generation altogether, it will fall out of synchronization since receiver 3000 will quickly reject packets that fall outside of window W, and the extra packets generated by transmitter 3001 will be discarded).

In accordance with the improvements described above, receiving computer 3000 performs certain steps when a SYNC\_REQ message is received, as illustrated in FIG. 30. In step 3004, receiving computer 3000 receives the SYNC\_REQ message. In step 3005, a check is

made to determine whether the request is a duplicate. If so, it is discarded in step 3006. In step 3007, a check is made to determine whether the SYNC\_REQ received from transmitter 3001 was received at a rate that exceeds the allowable rate R (i.e., the period between the time of the last SYNC\_REQ message). The value R can be a constant, or it can be made to fluctuate as desired.

5 If the rate exceeds R, then in step 3008 the next activation of the next CKPT\_N hopping table entry is delayed by W/R seconds after the last SYNC\_REQ has been accepted.

Otherwise, if the rate has not been exceeded, then in step 3109 the next CKPT\_N value is calculated and inserted into the receiver's hopping table prior to the next SYNC\_REQ from the transmitter 3101. Transmitter 3101 then processes the SYNC\_REQ in the normal manner.

#### 10 **E. Signaling Synchronizer**

In a system in which a large number of users communicate with a central node using secure hopping technology, a large amount of memory must be set aside for hopping tables and their supporting data structures. For example, if one million subscribers to a web site occasionally communicate with the web site, the site must maintain one million hopping tables, thus using up valuable computer resources, even though only a small percentage of the users may actually be using the system at any one time. A desirable solution would be a system that permits a certain maximum number of simultaneous links to be maintained, but which would "recognize" millions of registered users at any one time. In other words, out of a population of a million registered users, a few thousand at a time could simultaneously communicate with a central server, without requiring that the server maintain one million hopping tables of appreciable size.

20 One solution is to partition the central node into two nodes: a signaling server that performs session initiation for user log-on and log-off (and requires only minimally sized tables), and a transport server that contains larger hopping tables for the users. The signaling server listens for the millions of known users and performs a fast-packet reject of other (bogus) packets. When a packet is received from a known user, the signaling server activates a virtual private link (VPL) between the user and the transport server, where hopping tables are allocated and maintained. When the user logs onto the signaling server, the user's computer is provided with

hop tables for communicating with the transport server, thus activating the VPL. The VPLs can be torn down when they become inactive for a time period, or they can be torn down upon user log-out. Communication with the signaling server to allow user log-on and log-off can be accomplished using a specialized version of the checkpoint scheme described above.

5           FIG. 31 shows a system employing certain of the above-described principles. In FIG. 31, a signaling server 3101 and a transport server 3102 communicate over a link. Signaling server 3101 contains a large number of small tables 3106 and 3107 that contain enough information to authenticate a communication request with one or more clients 3103 and 3104. As described in more detail below, these small tables may advantageously be constructed as a special case of the  
10           synchronizing checkpoint tables described previously. Transport server 3102, which is preferably a separate computer in communication with signaling server 3101, contains a smaller number of larger hopping tables 3108, 3109, and 3110 that can be allocated to create a VPN with one of the client computers.

          According to one embodiment, a client that has previously registered with the system  
15           (e.g., via a system administration function, a user registration procedure, or some other method) transmits a request for information from a computer (e.g., a web site). In one variation, the request is made using a “hopped” packet, such that signaling server 3101 will quickly reject invalid packets from unauthorized computers such as hacker computer 3105. An “administrative” VPN can be established between all of the clients and the signaling server in  
20           order to ensure that a hacker cannot flood signaling server 3101 with bogus packets. Details of this scheme are provided below.

          Signaling server 3101 receives the request 3111 and uses it to determine that client 3103 is a validly registered user. Next, signaling server 3101 issues a request to transport server 3102 to allocate a hopping table (or hopping algorithm or other regime) for the purpose of creating a  
25           VPN with client 3103. The allocated hopping parameters are returned to signaling server 3101 (path 3113), which then supplies the hopping parameters to client 3103 via path 3114, preferably in encrypted form.

Thereafter, client 3103 communicates with transport server 3102 using the normal hopping techniques described above. It will be appreciated that although signaling server 3101 and transport server 3102 are illustrated as being two separate computers, they could of course be combined into a single computer and their functions performed on the single computer. Alternatively, it is possible to partition the functions shown in FIG. 31 differently from as shown without departing from the inventive principles.

One advantage of the above-described architecture is that signaling server 3101 need only maintain a small amount of information on a large number of potential users, yet it retains the capability of quickly rejecting packets from unauthorized users such as hacker computer 3105. Larger data tables needed to perform the hopping and synchronization functions are instead maintained in a transport server 3102, and a smaller number of these tables are needed since they are only allocated for "active" links. After a VPN has become inactive for a certain time period (e.g., one hour), the VPN can be automatically torn down by transport server 3102 or signaling server 3101.

A more detailed description will now be provided regarding how a special case of the checkpoint synchronization feature can be used to implement the signaling scheme described above.

The signaling synchronizer may be required to support many (millions) of standing, low bandwidth connections. It therefore should minimize per-VPL memory usage while providing the security offered by hopping technology. In order to reduce memory usage in the signaling server, the data hopping tables can be completely eliminated and data can be carried as part of the SYNC\_REQ message. The table used by the server side (receiver) and client side (transmitter) is shown schematically as element 3106 in FIG. 31.

The meaning and behaviors of CKPT\_N, CKPT\_O and CKPT\_R remain the same from the previous description, except that CKPT\_N can receive a combined data and SYNC\_REQ message or a SYNC\_REQ message without the data.

The protocol is a straightforward extension of the earlier synchronizer. Assume that a client transmitter is on and the tables are synchronized. The initial tables can be generated "out

of band.” For example, a client can log into a web server to establish an account over the Internet. The client will receive keys etc encrypted over the Internet. Meanwhile, the server will set up the signaling VPN on the signaling server.

5 Assuming that a client application wishes to send a packet to the server on the client’s standing signaling VPL:

1. The client sends the message marked as a data message on the inner header using the transmitter’s CKPT\_N address. It turns the transmitter off and starts a timer T1 noting CKPT\_O. Messages can be one of three types: DATA, SYNC\_REQ and SYNC\_ACK. In the normal algorithm, some potential problems can be prevented by identifying each message type as part of the encrypted inner header field. In this algorithm, it is important to distinguish a data packet and a SYNC\_REQ in the signaling synchronizer since the data and the SYNC\_REQ come in on the same address.

10  
15  
2. When the server receives a data message on its CKPT\_N, it verifies the message and passes it up the stack. The message can be verified by checking message type and other information (i.e user credentials) contained in the inner header. It replaces its CKPT\_O with CKPT\_N and generates the next CKPT\_N. It updates its transmitter side CKPT\_R to correspond to the client’s receiver side CKPT\_R and transmits a SYNC\_ACK containing CKPT\_O in its payload.

20  
3. When the client side receiver receives a SYNC\_ACK on its CKPT\_R with a payload matching its transmitter side CKPT\_O and the transmitter is off, the transmitter is turned on and the receiver side CKPT\_R is updated. If the SYNC\_ACK’s payload does not match the transmitter side CKPT\_O or the transmitter is on, the SYNC\_ACK is simply discarded.

25  
4. T1 expires: If the transmitter is off and the client’s transmitter side CKPT\_O matches the CKPT\_O associated with the timer, it starts timer T1 noting CKPT\_O again, and a SYNC\_REQ is sent using the transmitter’s CKPT\_O address. Otherwise, no action is taken.

5. When the server receives a SYNC\_REQ on its CKPT\_N, it replaces its CKPT\_O with CKPT\_N and generates the next CKPT\_N. It updates its transmitter side CKPT\_R to correspond

to the client's receiver side CKPT\_R and transmits a SYNC\_ACK containing CKPT\_O in its payload.

6. When the server receives a SYNC\_REQ on its CKPT\_O, it updates its transmitter side CKPT\_R to correspond to the client's receiver side CKPT\_R and transmits a SYNC\_ACK containing CKPT\_O in its payload.

FIG. 32 shows message flows to highlight the protocol. Reading from top to bottom, the client sends data to the server using its transmitter side CKPT\_N. The client side transmitter is turned off and a retry timer is turned off. The transmitter will not transmit messages as long as the transmitter is turned off. The client side transmitter then loads CKPT\_N into CKPT\_O and updates CKPT\_N. This message is successfully received and a passed up the stack. It also synchronizes the receiver i.e, the server loads CKPT\_N into CKPT\_O and generates a new CKPT\_N, it generates a new CKPT\_R in the server side transmitter and transmits a SYNC\_ACK containing the server side receiver's CKPT\_O the server. The SYNC\_ACK is successfully received at the client. The client side receiver's CKPT\_R is updated, the transmitter is turned on and the retry timer is killed. The client side transmitter is ready to transmit a new data message.

Next, the client sends data to the server using its transmitter side CKPT\_N. The client side transmitter is turned off and a retry timer is turned off. The transmitter will not transmit messages as long as the transmitter is turned off. The client side transmitter then loads CKPT\_N into CKPT\_O and updates CKPT\_N. This message is lost. The client side timer expires and as a result a SYNC\_REQ is transmitted on the client side transmitter's CKPT\_O (this will keep happening until the SYNC\_ACK has been received at the client). The SYNC\_REQ is successfully received at the server. It synchronizes the receiver i.e, the server loads CKPT\_N into CKPT\_O and generates a new CKPT\_N, it generates an new CKPT\_R in the server side transmitter and transmits a SYNC\_ACK containing the server side receiver's CKPT\_O the server. The SYNC\_ACK is successfully received at the client. The client side receiver's CKPT\_R is updated, the transmitter is turned off and the retry timer is killed. The client side transmitter is ready to transmit a new data message.

There are numerous other scenarios that follow this flow. For example, the SYNC\_ACK could be lost. The transmitter would continue to re-send the SYNC\_REQ until the receiver synchronizes and responds.

5 The above-described procedures allow a client to be authenticated at signaling server 3201 while maintaining the ability of signaling server 3201 to quickly reject invalid packets, such as might be generated by hacker computer 3205. In various embodiments, the signaling synchronizer is really a derivative of the synchronizer. It provides the same protection as the hopping protocol, and it does so for a large number of low bandwidth connections.

**CLAIMS**

1. A method of transmitting data packets between a first computer and a second computer, wherein the first computer and the second computer are linked via a plurality of separate transmission paths, the method comprising the steps of:

- 5 (1) assigning a weight value to each of the plurality of transmission paths, wherein each respective weight value represents the relative number of packets that a respective transmission path will transmit;
- (2) for each data packet that is to be transmitted from the first computer to the second computer, selecting one of the plurality of transmission paths on the basis of each respective  
10 transmission path's assigned weight value;
- (3) measuring the transmission quality for each of the plurality of transmission paths; and
- (4) adjusting downwardly to a non-zero value the assigned weight value for a transmission path for which the transmission quality has declined.
2. The method of claim 1, wherein step (4) comprises the step of gradually decreasing  
15 over time the assigned weight value in relation to weight values assigned to the remaining transmission paths.
3. The method of claim 2, wherein step (4) comprises the step of gradually decreasing the assigned weight value according to an incrementally decreasing function.
4. The method of claim 2, wherein step (4) comprises the step of gradually decreasing  
20 the assigned weight value according to an exponentially decaying function.
5. The method of claim 1, wherein step (3) comprises the step of determining that one or more packets transmitted to the second computer was not acknowledged by the second computer.
6. The method of claim 1, wherein step (3) comprises the step of evaluating the contents of a synchronization packet that maintains synchronization with a moving window of valid  
25 values.
7. The method of claim 1, further comprising the step of inserting into each data packet a source and destination IP address pair that is selected according to a pseudo-random sequence.



8. The method of claim 1, wherein step (4) comprises the step of adjusting downwardly the assigned weight value for a transmission path only if the transmission quality has declined below a predetermined threshold.

5 9. The method of claim 1, further comprising the step of adjusting upwardly the assigned weight value that was adjusted in step (4) if it is later determined that the transmission quality has improved.

10. The method of claim 1, further comprising the step of adjusting upwardly the weight values of the remaining transmission links in an amount that compensates for the downwardly adjusted weight value.

10 11. The method of claim 10, wherein the step of adjusting upwardly comprises the step of equally distributing the amount that was downwardly adjusted across the remaining transmission links.

15 12. The method of claim 1, further comprising the step of adjusting downwardly to zero the assigned weight value for any transmission link whose quality has degraded below a preset threshold.

13. The method of claim 1, wherein steps (2) through (4) are repeated periodically.

14. A first computer that transmits data packets to a second computer over a plurality of separate transmission paths, wherein the first computer performs the steps of:

20 (1) assigning a weight value to each of the plurality of transmission paths, wherein each respective weight value represents the relative number of packets that a respective transmission path will transmit;

(2) for each data packet that is to be transmitted to the second computer, selecting one of the plurality of transmission paths on the basis of each respective transmission path's assigned weight value;

25 (3) measuring the transmission quality for each of the plurality of transmission paths; and

(4) adjusting downwardly to a non-zero value the assigned weight value for a transmission path for which the transmission quality has declined.

15. The first computer of claim 14, wherein the first computer gradually decreases over time the assigned weight value in relation to weight values assigned to the remaining transmission paths.

16. The first computer of claim 15, wherein the first computer gradually decreases the assigned weight value according to an incrementally decreasing function.

17. The first computer of claim 15, wherein the first computer gradually decreases the assigned weight value according to an exponentially decaying function.

18. The first computer of claim 14, wherein the first computer measures the transmission quality by determining that one or more packets transmitted to the second computer was not acknowledged by the second computer.

19. The first computer of claim 14, wherein the first computer measures the transmission quality by evaluating the contents of a synchronization packet that maintains synchronization with a moving window of valid values.

20. The first computer of claim 14, wherein the first computer inserts into each data packet a source and destination IP address pair that is selected according to a pseudo-random sequence.

21. The first computer of claim 14, wherein the first computer adjusts downwardly the assigned weight value for any transmission path only if the transmission quality has declined below a predetermined threshold.

22. The first computer of claim 14, wherein the first computer adjusts upwardly the assigned weight value that was adjusted in step (4) if it is later determined that the transmission quality has improved.

23. The first computer of claim 14, wherein the first computer adjusts upwardly the weight values of the remaining transmission links in an amount that compensates for the downwardly adjusted weight value.

24. The first computer of claim 23, wherein the first computer upwardly adjusts probabilities across the remaining transmission links in an amount equal to the downwardly adjusted weight value.

25. The first computer of claim 14, wherein the first computer adjusts downwardly to zero the assigned weight value for any transmission link whose quality has degraded below a preset threshold.

26. The first computer of claim 14, wherein the first computer repeats steps (2) through  
5 (4) periodically.

27. A system comprising the first computer of claim 14 and a second computer constructed in accordance with the first computer of claim 14.

28. A method of transparently creating a virtual private network (VPN) between a client computer and a target computer, comprising the steps of:

10 (1) generating from the client computer a Domain Name Service (DNS) request that requests an IP address corresponding to a domain name associated with the target computer;

(2) determining whether the DNS request transmitted in step (1) is requesting access to a secure web site; and

15 (3) in response to determining that the DNS request in step (2) is requesting access to a secure target web site, automatically initiating the VPN between the client computer and the target computer.

29. The method of claim 28, wherein steps (2) and (3) are performed at a DNS server separate from the client computer.

30. The method of claim 28, further comprising the step of:

20 (4) in response to determining that the DNS request in step (2) is not requesting access to a secure target web site, resolving the IP address for the domain name and returning the IP address to the client computer.

31. The method of claim 28, wherein step (3) comprises the step of, prior to automatically initiating the VPN between the client computer and the target computer,  
25 determining whether the client computer is authorized to establish a VPN with the target computer and, if not so authorized, returning an error from the DNS request.

32. The method of claim 28, wherein step (3) comprises the step of, prior to automatically initiating the VPN between the client computer and the target computer,

determining whether the client computer is authorized to resolve addresses of non secure target computers and, if not so authorized, returning an error from the DNS request.

33. The method of claim 28, wherein step (3) comprises the step of establishing the VPN by creating an IP address hopping scheme between the client computer and the target computer.

5 34. The method of claim 28, wherein step (3) comprises the step of using a gatekeeper computer that allocates VPN resources for communicating between the client computer and the target computer.

10 35. The method of claim 28, wherein step (2) is performed in a DNS proxy server that passes through the request to a DNS server if it is determined in step (3) that access is not being requested to a secure target web site.

36. The method of claim 32, wherein step (3) comprises the step of transmitting a message to the client computer to determine whether the client computer is authorized to establish the VPN target computer.

15 37. A system that transparently creates a virtual private network (VPN) between a client computer and a secure target computer, comprising:

20 a DNS proxy server that receives a request from the client computer to look up an IP address for a domain name, wherein the DNS proxy server returns the IP address for the requested domain name if it is determined that access to a non-secure web site has been requested, and wherein the DNS proxy server generates a request to create the VPN between the client computer and the secure target computer if it is determined that access to a secure web site has been requested; and

a gatekeeper computer that allocates resources for the VPN between the client computer and the secure web computer in response to the request by the DNS proxy server.

25 38. The system of claim 37, wherein the gatekeeper computer creates the VPN by establishing an IP address hopping regime that is used to pseudorandomly change IP addresses in packets transmitted between the client computer and the secure target computer.

39. The system of claim 37, wherein the gatekeeper computer determines whether the client computer has sufficient security privileges to create the VPN and, if the client computer lacks sufficient security privileges, rejecting the request to create the VPN.

40. A method of preventing data packets received from a high bandwidth link from flooding a low bandwidth link, comprising the steps of:

(1) receiving data packets from the high bandwidth link that are ostensibly addressed to a computer residing on the low-bandwidth link;

(2) for each data packet, determining whether the data packet is validly addressed to the computer on the low-bandwidth link;

10 (3) in response to determining that the data packet is not validly addressed to the computer on the low-bandwidth link, rejecting the data packet; and

(4) in response to determining that the data packet is validly addressed to the computer on the low-bandwidth link, forwarding the data packet to the computer over the low-bandwidth link.

41. The method of claim 40, wherein step (3) comprises the step of comparing a value in a header of each data packet to a set of valid values maintained for the computer on the low-bandwidth link.

42. The method of claim 41, wherein step (3) comprises the step of comparing a value in a header of each data packet to a moving window of valid values.

43. The method of claim 42, wherein step (3) comprises the step of comparing the IP address in the header of each data packet to a moving window of valid IP addresses, wherein the moving window is also maintained by the computer on the low-bandwidth link.

44. The method of claim 40, wherein step (3) comprises the step of reducing a priority level of the packet in relation to other data packets, wherein the priority level determines whether a particular data packet will be transmitted before another data packet having a different priority level.

45. The method of claim 40, wherein step (3) comprises the step of performing a cryptographic check on each data packet to determine whether each data packet is validly addressed.

46. The method of claim 40, wherein step (3) comprises the step of receiving a message from the computer on the low-bandwidth link to stop accepting messages having a particular characteristic.

47. The method of claim 46, wherein step (3) comprises the step of receiving a message  
5 from the computer on the low-bandwidth link to stop accepting messages addressed to a particular IP address.

48. The method of claim 40, wherein step (3) comprises the step of determining that a packet transmission rate has been exceeded for a given packet parameter.

49. The method of claim 48, wherein step (3) comprises the step of determining that a  
10 packet transmission rate has been exceeded for a given IP destination address.

50. In a system having a low bandwidth data link, a first computer coupled to the low bandwidth data link, and a high bandwidth data link, an improvement comprising:

a second computer coupled between the low bandwidth data link and the high bandwidth data link, wherein the second computer receives data packets from the high bandwidth data link and, if they are addressed to the first computer, routes them to the first computer over the low  
15 bandwidth data link,

wherein the second computer prevents invalid data packets ostensibly addressed to the first computer from being transmitted over the low bandwidth data link.

51. The system of claim 50, wherein the second computer prevents invalid data packets  
20 from being transmitted over the low bandwidth data link by comparing a discriminator field in a header of each data packet to a table of valid discriminator fields maintained for the first computer.

52. The system of claim 50, wherein the second computer compares an Internet Protocol (IP) address in a header of each data packet to a table of valid IP addresses.

53. The system of claim 52, wherein the second computer compares the IP address in the  
25 header of each data packet to a moving window of valid IP addresses, wherein the moving window is also maintained by the first computer.

54. The system of claim 50, wherein the second computer reduces a priority level of a data packet in relation to other data packets, wherein the priority level determines whether a particular data packet will be transmitted before another data packet having a different priority level.

5 55. The system of claim 50, wherein the second computer performs a cryptographic check on each data packet to determine whether each data packet is validly addressed.

56. The system of claim 50, wherein the second computer receives a message from the first computer that causes the second computer to stop accepting messages having a particular characteristic.

10 57. The system of claim 56, wherein the second computer receiving a message from the first computer to stop accepting messages addressed to a particular IP address.

58. The system of claim 50, wherein the second computer rejects invalid packets by determining that a packet transmission rate has been exceeded for a given packet parameter.

15 59. The system of claim 58, wherein the second computer determines that a packet transmission rate has been exceeded for a given IP destination address.

60. In a system comprising a first computer that transmits data packets to a second computer over a network according to a scheme by which at least one field in a series of data packets is periodically changed according to a sequence known by the first and second computers, and wherein the second computer periodically receives a synchronization request  
20 from the first computer to maintain synchronization of the sequence between the first and second computers, a method comprising the steps of:

(1) receiving at the first computer the synchronization request from the second computer;

(2) determining whether the synchronization request was received in less than a predetermined interval;

25 (3) in response to determining that the synchronization request was received in less than the predetermined interval, ignoring the synchronization request; and

(4) in response to determining that the synchronization request was not received in less than the predetermined interval, providing the synchronization response to the first computer.

61. The method of claim 60, wherein step (3) comprises the step of delaying the acceptance of a SYNC\_REQ for  $W/R$  seconds, where  $W$  is the number of data packets between synchronization requests according to an agreed schedule, and  $R$  is the agreed rate at which synchronization requests should be received according to the agreed schedule.

5           62. The method of claim 60, further comprising the step of determining whether the synchronization request is a duplicate of a previously received synchronization request and, if it is a duplicate, discarding it.

63. The method of claim 60, wherein step (4) comprises the step of providing a response that includes a new checkpoint for synchronizing a window in a hopping table.

10           64. A computer that receives data packets from a second computer over a network according to a scheme by which at least one field in a series of data packets is periodically changed according to a known sequence, wherein the second computer periodically transmits a synchronization request to maintain synchronization of the sequence, wherein the computer performs the steps of:

15           (1) receiving the synchronization request from the second computer;

          (2) determining whether the synchronization request was received in less than a predetermined interval;

          (3) in response to determining that the synchronization request was received in less than a predetermined interval ignoring the synchronization request; and

20           (4) in response to determining that the synchronization request was not received in less than a predetermined interval, providing the response to the first computer.

65. The computer of claim 64, wherein the computer delays the acceptance of a SYNC\_REQ in step (3) for  $W/R$  seconds, where  $W$  is the number of data packets between synchronization requests according to an agreed schedule, and  $R$  is the agreed rate at which synchronization requests should be received according to the agreed schedule.

66. The computer of claim 64, wherein the computer further performs the step of determining whether the synchronization request is a duplicate of a previously received synchronization request and, if it is a duplicate, discarding it.



67. A method of establishing communication between one of a plurality of client computers and a central computer that maintains a plurality of authentication tables each corresponding to one of the client computers, the method comprising the steps of:

5 (1) in the central computer, receiving from one of the plurality of client computers a request to establish a connection;

(2) authenticating, with reference to one of the plurality of authentication tables, that the request received in step (1) is from an authorized client;

(3) responsive to a determination that the request is from an authorized client, allocating resources to establish a virtual private link between the client and a second computer; and

10 (4) communicating between the authorized client and the second computer using the virtual private link.

68. The method of claim 67, wherein step (4) comprises the step of communicating according to a scheme by which at least one field in a series of data packets is periodically changed according to a known sequence.

15 69. The method of claim 68, wherein step (4) comprises the step of comparing an Internet Protocol (IP) address in a header of each data packet to a table of valid IP addresses maintained in a table in the second computer.

20 70. The method of claim 69, wherein step (4) comprises the step of comparing the IP address in the header of each data packet to a moving window of valid IP addresses, and rejecting data packets having IP addresses that do not fall within the moving window.

71. The method of claim 67, wherein step (2) comprises the step of using a checkpoint data structure that maintains synchronization of a periodically changing parameter known by the central computer and the client computer to authenticate the client.

**ABSTRACT**

A plurality of computer nodes communicate using seemingly random Internet Protocol source and destination addresses. Data packets matching criteria defined by a moving window of valid addresses are accepted for further processing, while those that do not meet the criteria are quickly rejected. Improvements to the basic design include (1) a load balancer that distributes packets across different transmission paths according to transmission path quality; (2) a DNS proxy server that transparently creates a virtual private network in response to a domain name inquiry; (3) a large-to-small link bandwidth management feature that prevents denial-of-service attacks at system chokepoints; (4) a traffic limiter that regulates incoming packets by limiting the rate at which a transmitter can be synchronized with a receiver; and (5) a signaling synchronizer that allows a large number of nodes to communicate with a central node by partitioning the communication function between two separate entities.

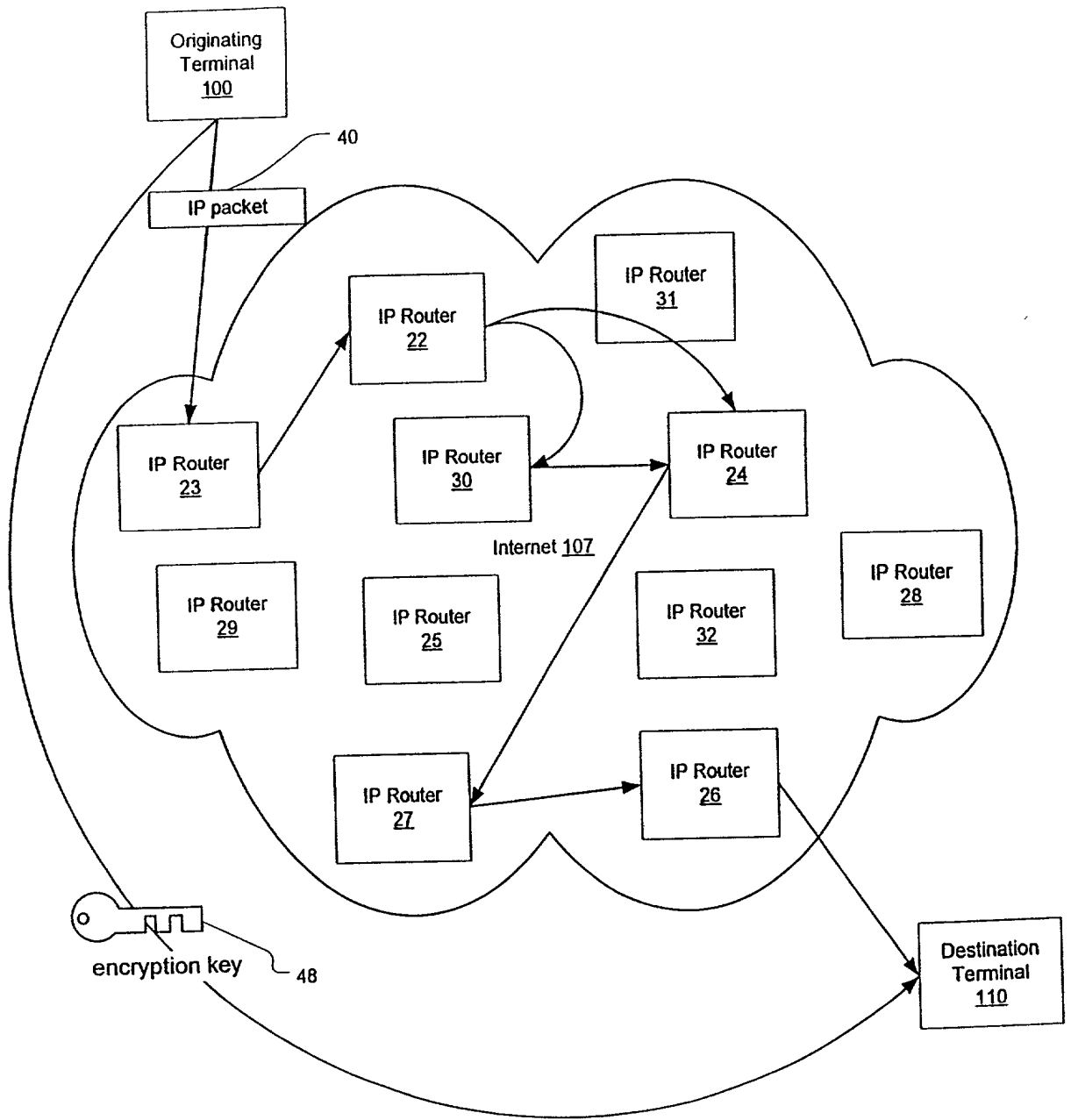


Fig. 1

FIG. 2 is a schematic diagram of a network architecture for TARP (Traffic Analysis and Reporting Protocol) routers and terminals. The network is represented by a cloud shape labeled 'Internet 107'. Inside the cloud, there are several TARP Routers (122, 123, 124, 125, 126, 127, 128) and IP Routers (129, 130, 131, 132). TARP Router 122 is connected to TARP Router 123, TARP Router 124, and TARP Router 125. TARP Router 123 is connected to TARP Router 124. TARP Router 124 is connected to TARP Router 125 and TARP Router 126. TARP Router 125 is connected to TARP Router 126. TARP Router 126 is connected to TARP Router 127. TARP Router 127 is connected to TARP Router 128. IP Router 129 is connected to TARP Router 123. IP Router 130 is connected to TARP Router 122 and TARP Router 124. IP Router 131 is connected to TARP Router 122. IP Router 132 is connected to TARP Router 124. Outside the cloud, TARP Terminal 100 is connected to TARP Router 123 via a 'link key' (146) and a 'TARP packet' (140). TARP Terminal 110 is connected to TARP Router 126 via a 'link key' and a 'TARP packet' (140). A 'session key' (148) is shown at the bottom left, connected to TARP Router 127. The diagram illustrates the flow of TARP packets and the distribution of link keys between routers and terminals.

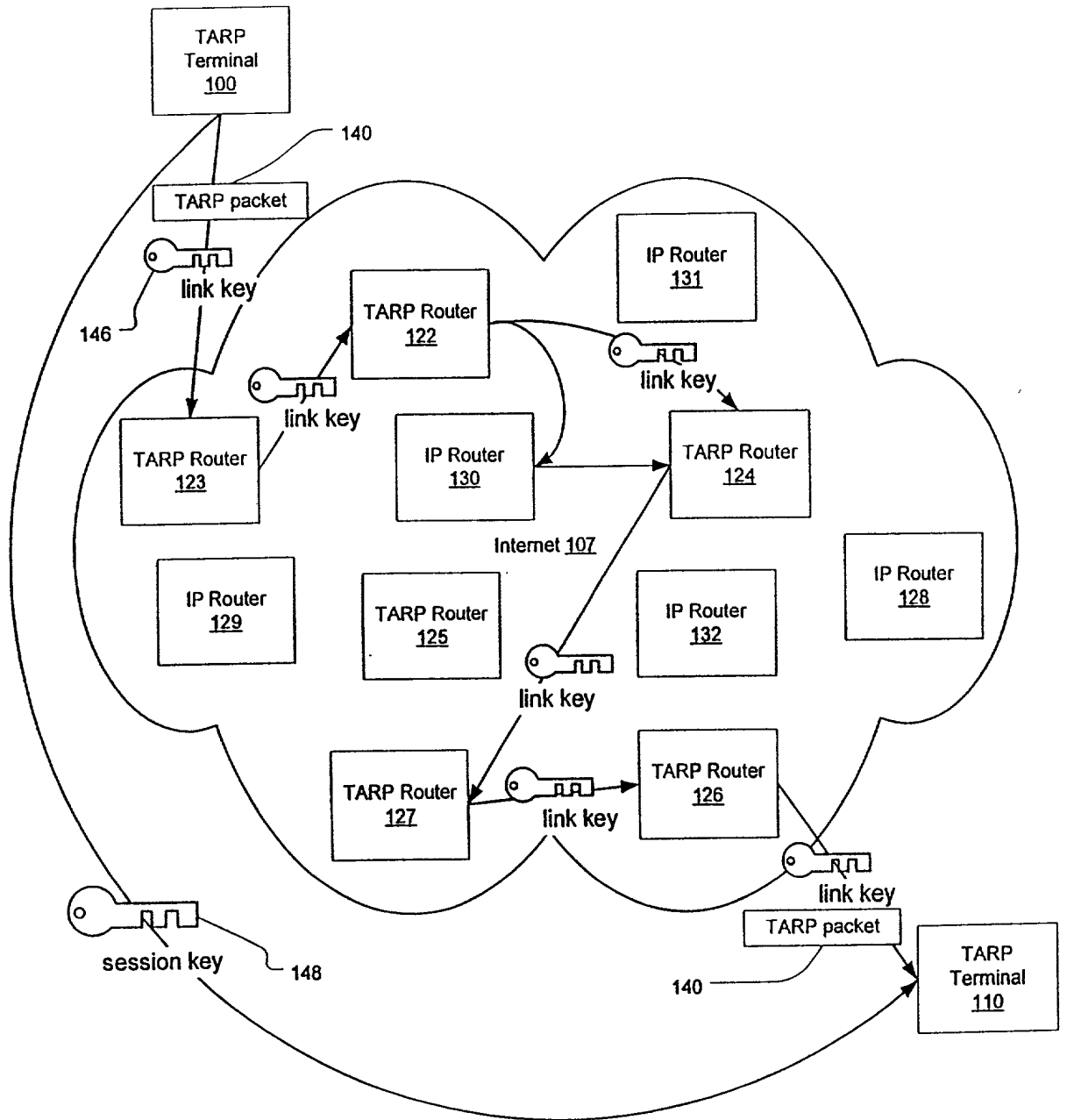


Fig. 2

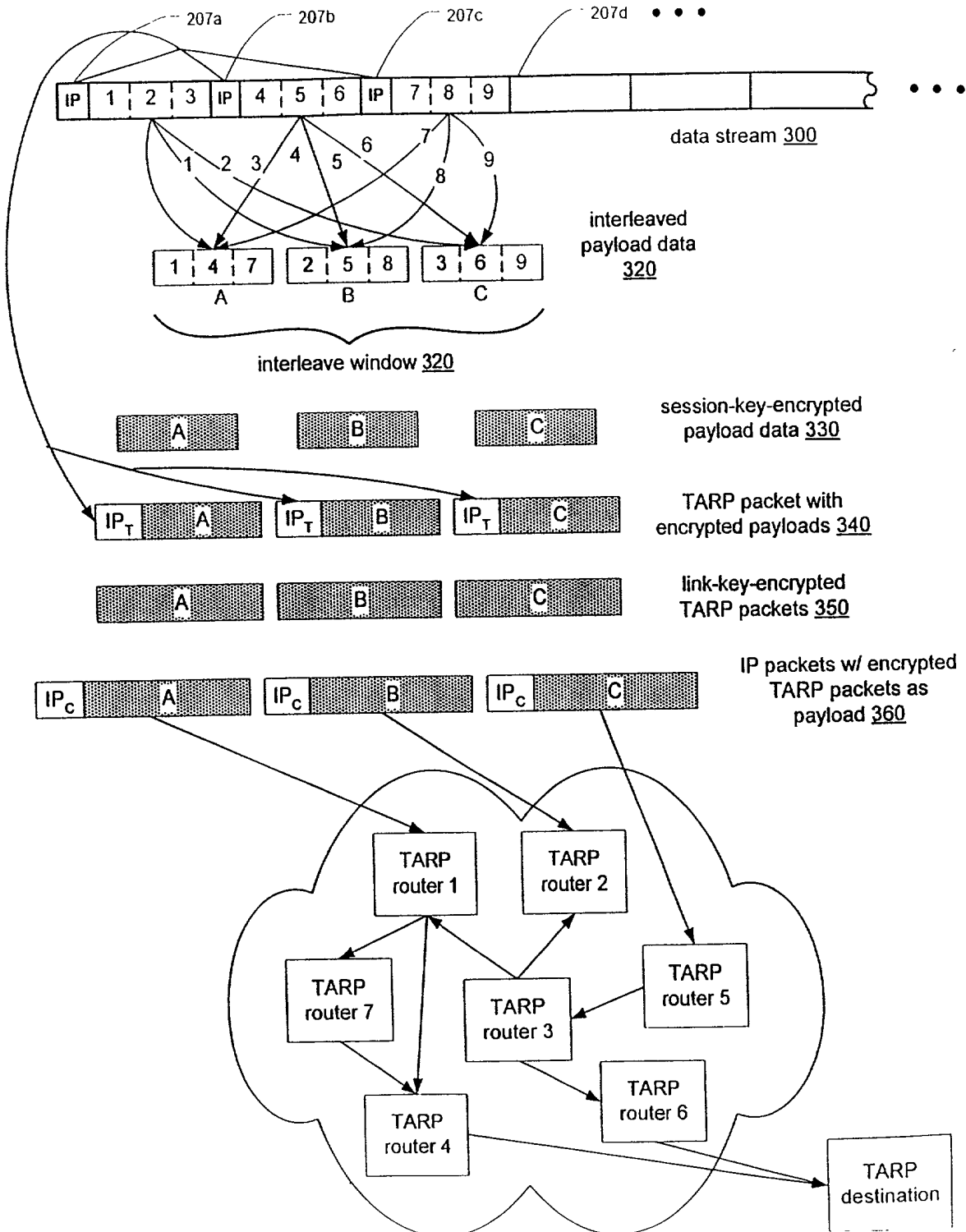


Fig. 3a

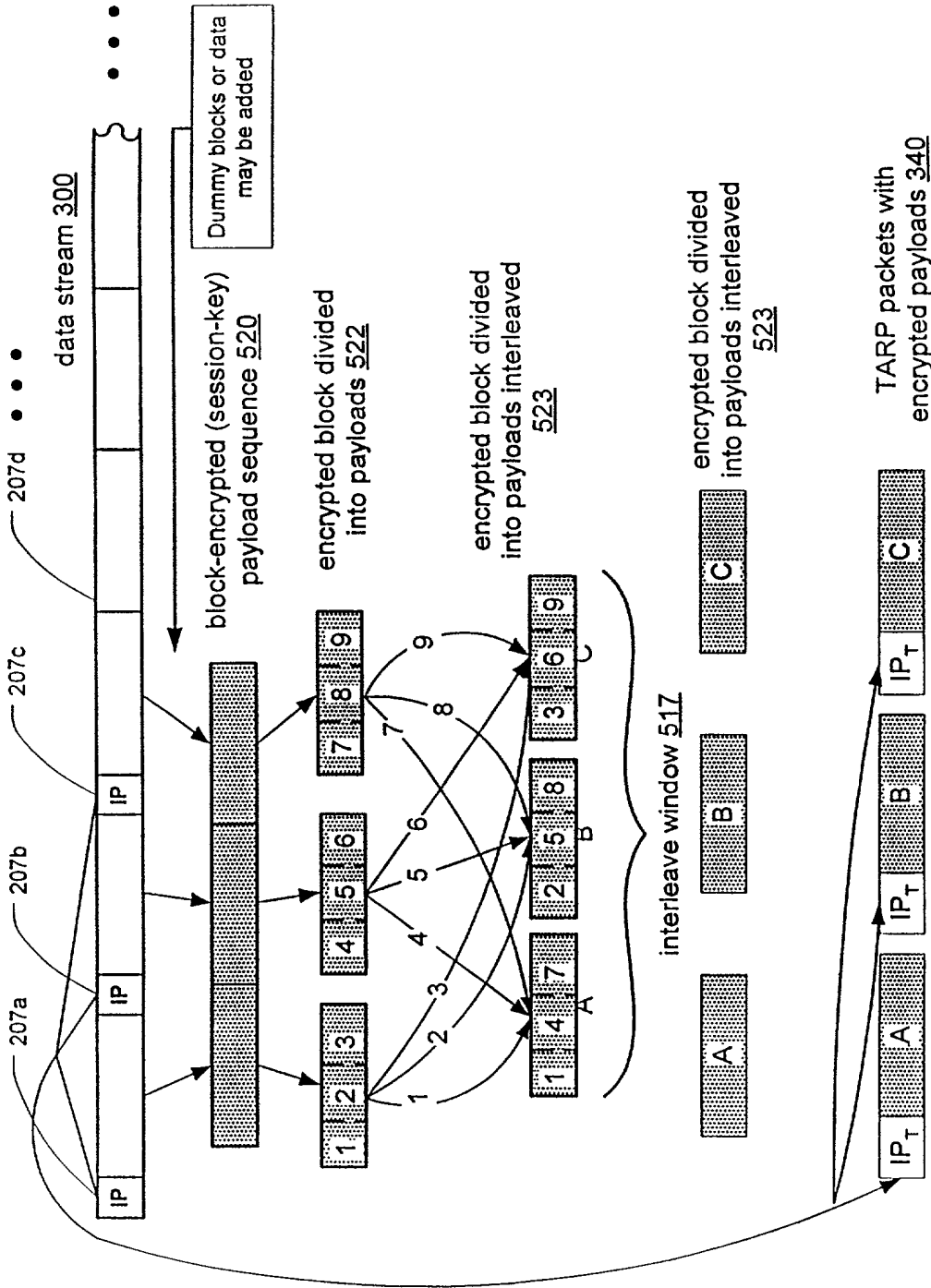


Fig. 3b

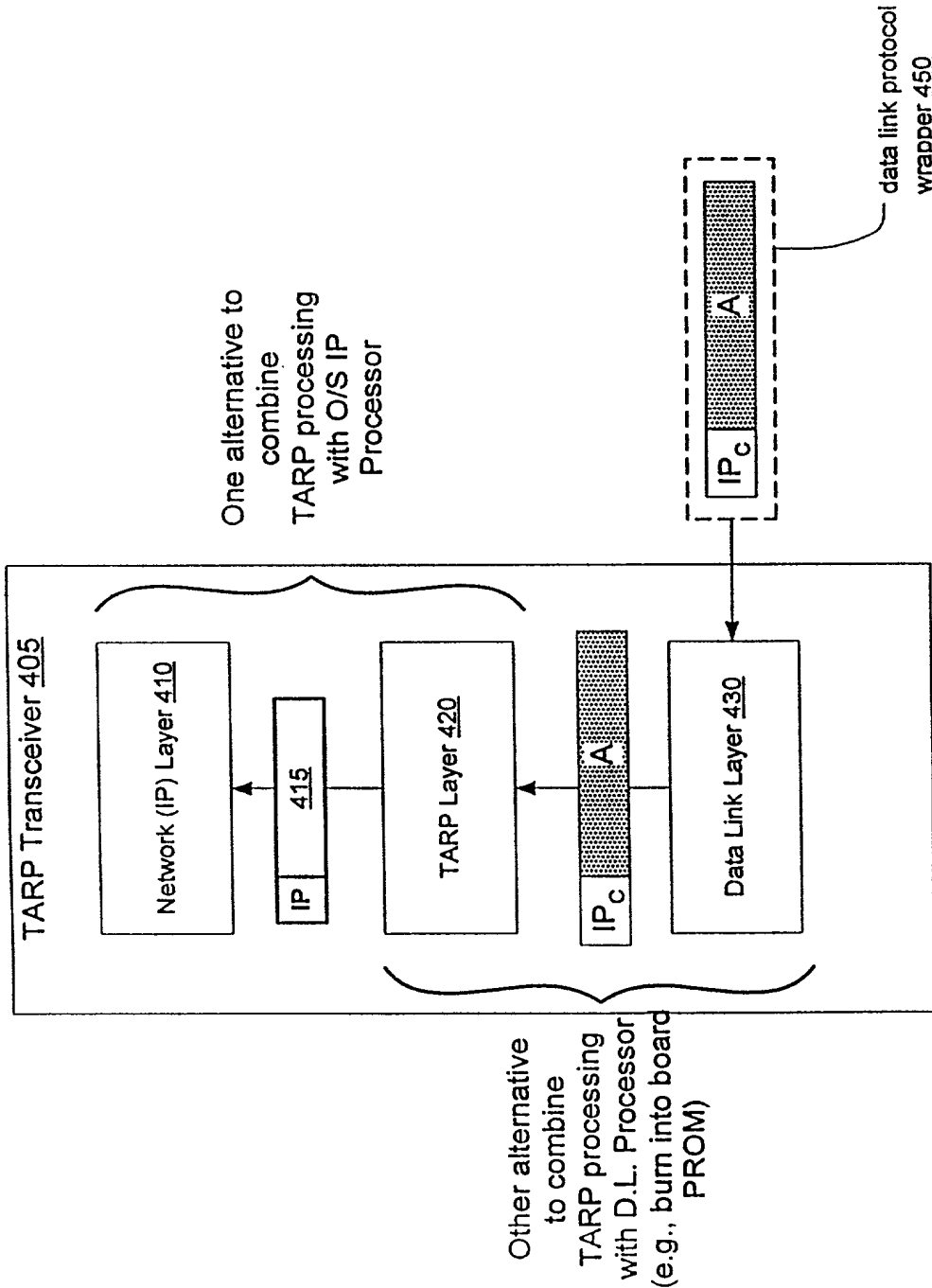


Fig. 4

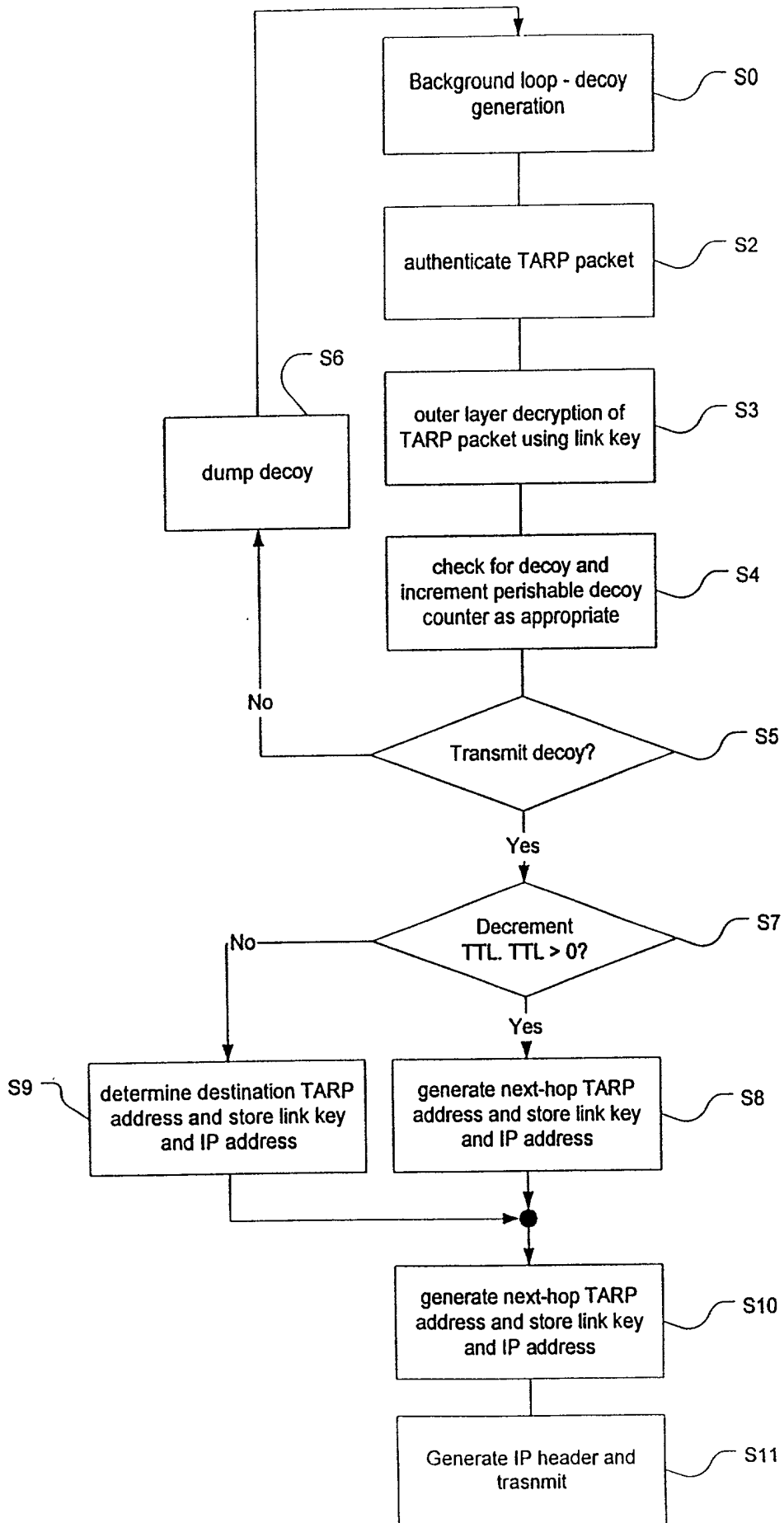
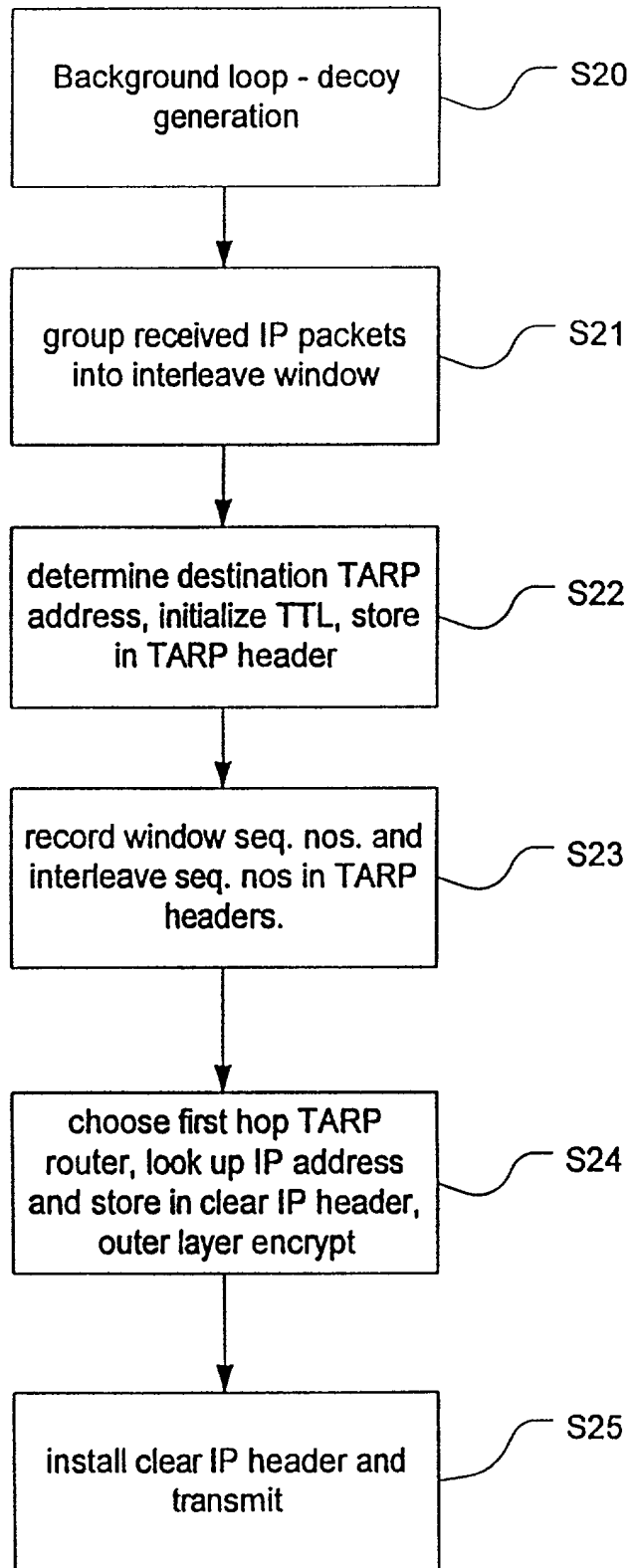


Fig. 5





**Fig. 6**

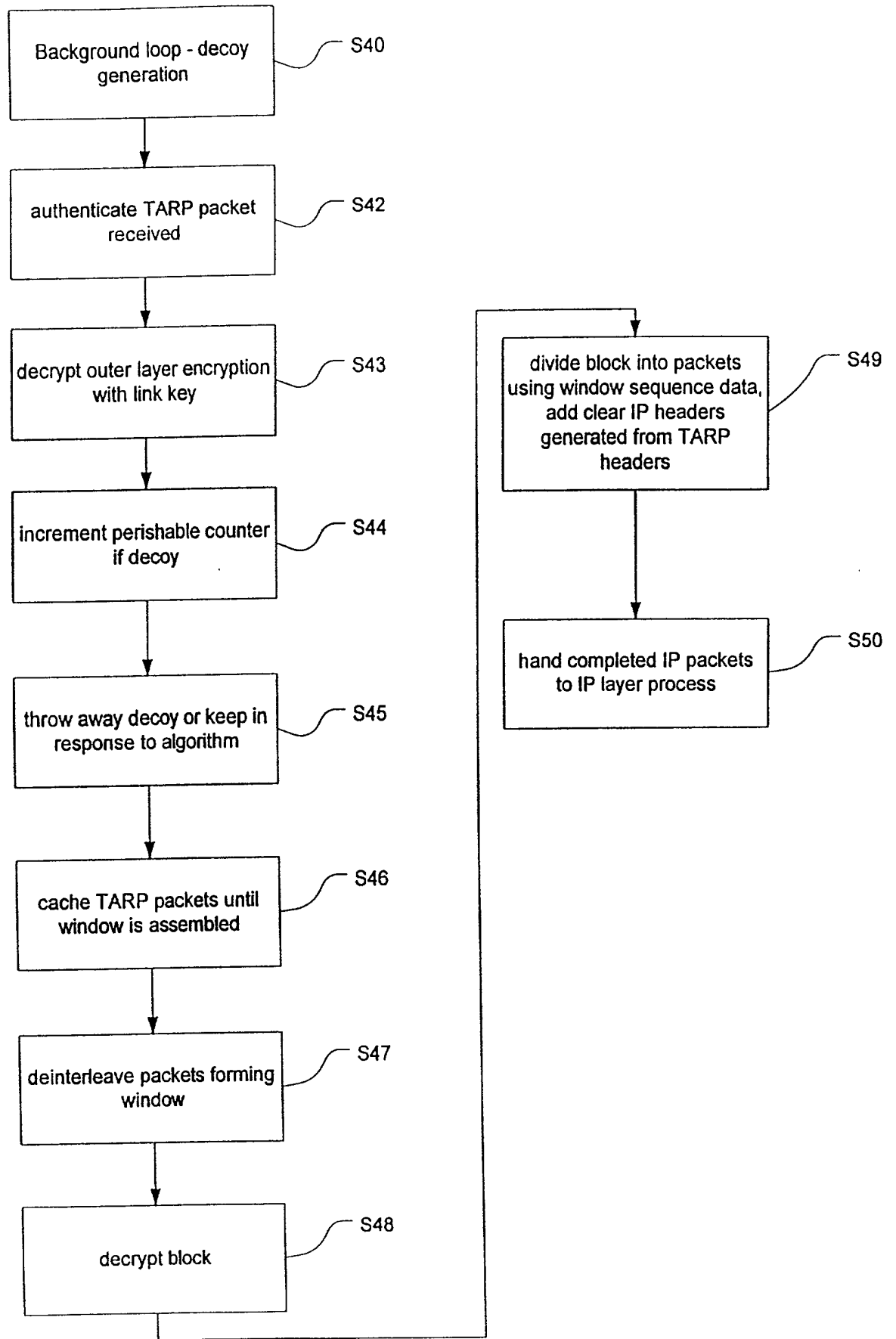
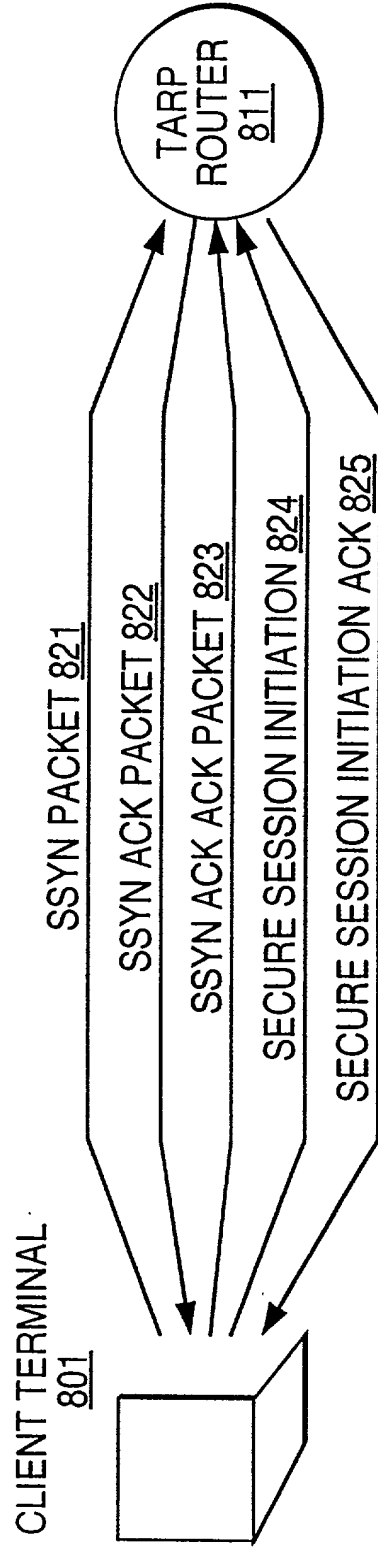


Fig. 7

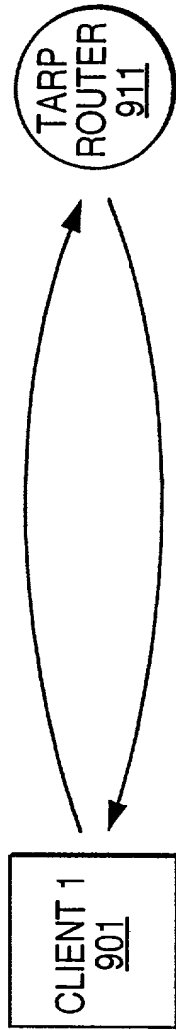
**FIG. 8**

**SECURE SESSION ESTABLISHMENT  
AND SYNCHRONIZATION**



**FIG. 9**

IHOP TRANSMIT AND RECEIVE TABLES



TRANSMIT TABLE 921

131.218.204.98 , 131.218.204.65  
 131.218.204.221 , 131.218.204.97  
 131.218.204.139 , 131.218.204.186  
 131.218.204.12 , 131.218.204.55

.  
 .  
 .

RECEIVE TABLE 922

131.218.204.161 , 131.218.204.89  
 131.218.204.66 , 131.218.204.212  
 131.218.204.201 , 131.218.204.127  
 131.218.204.119 , 131.218.204.49

.  
 .  
 .

RECEIVE TABLE 924

131.218.204.98 , 131.218.204.65  
 131.218.204.221 , 131.218.204.97  
 131.218.204.139 , 131.218.204.186  
 131.218.204.12 , 131.218.204.55

.  
 .  
 .

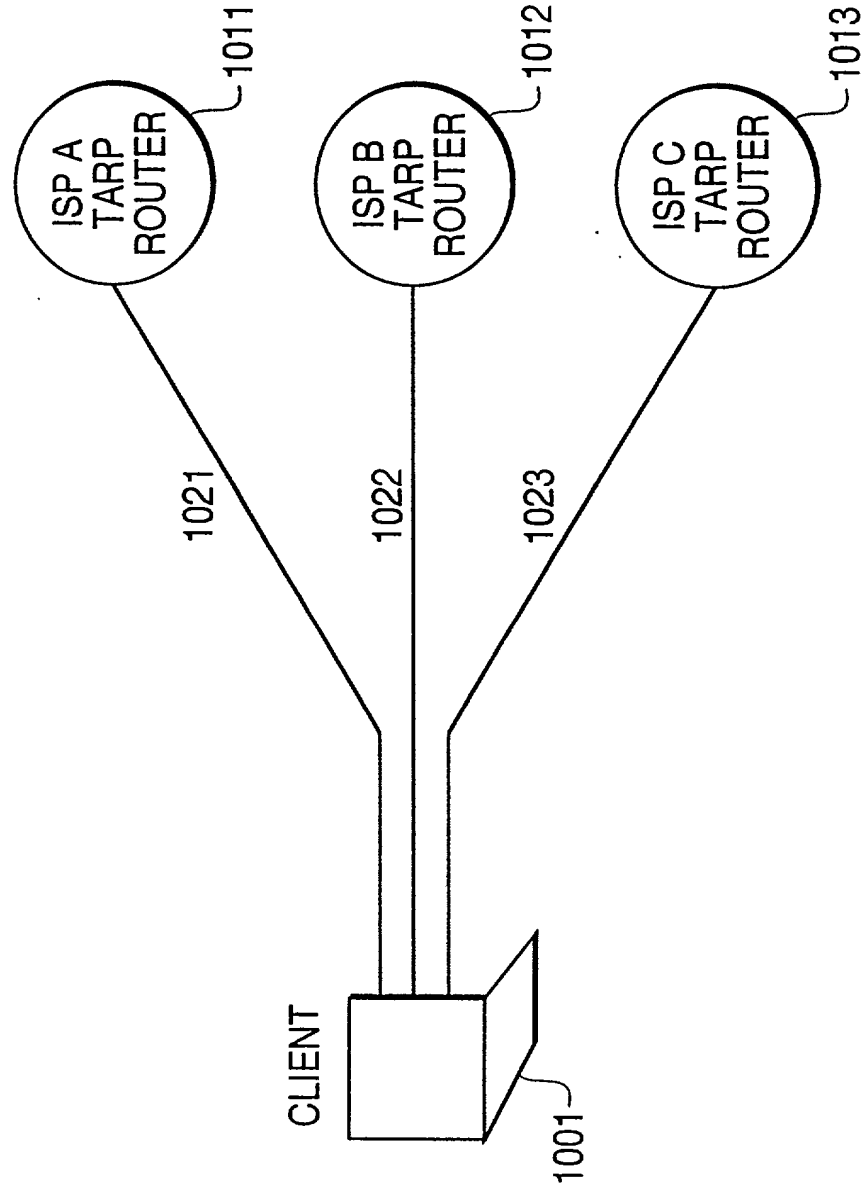
TRANSMIT TABLE 923

131.218.204.161 , 131.218.204.89  
 131.218.204.66 , 131.218.204.212  
 131.218.204.201 , 131.218.204.127  
 131.218.204.119 , 131.218.204.49

.  
 .  
 .

**FIG. 10**

PHYSICAL LINK REDUNDANCY



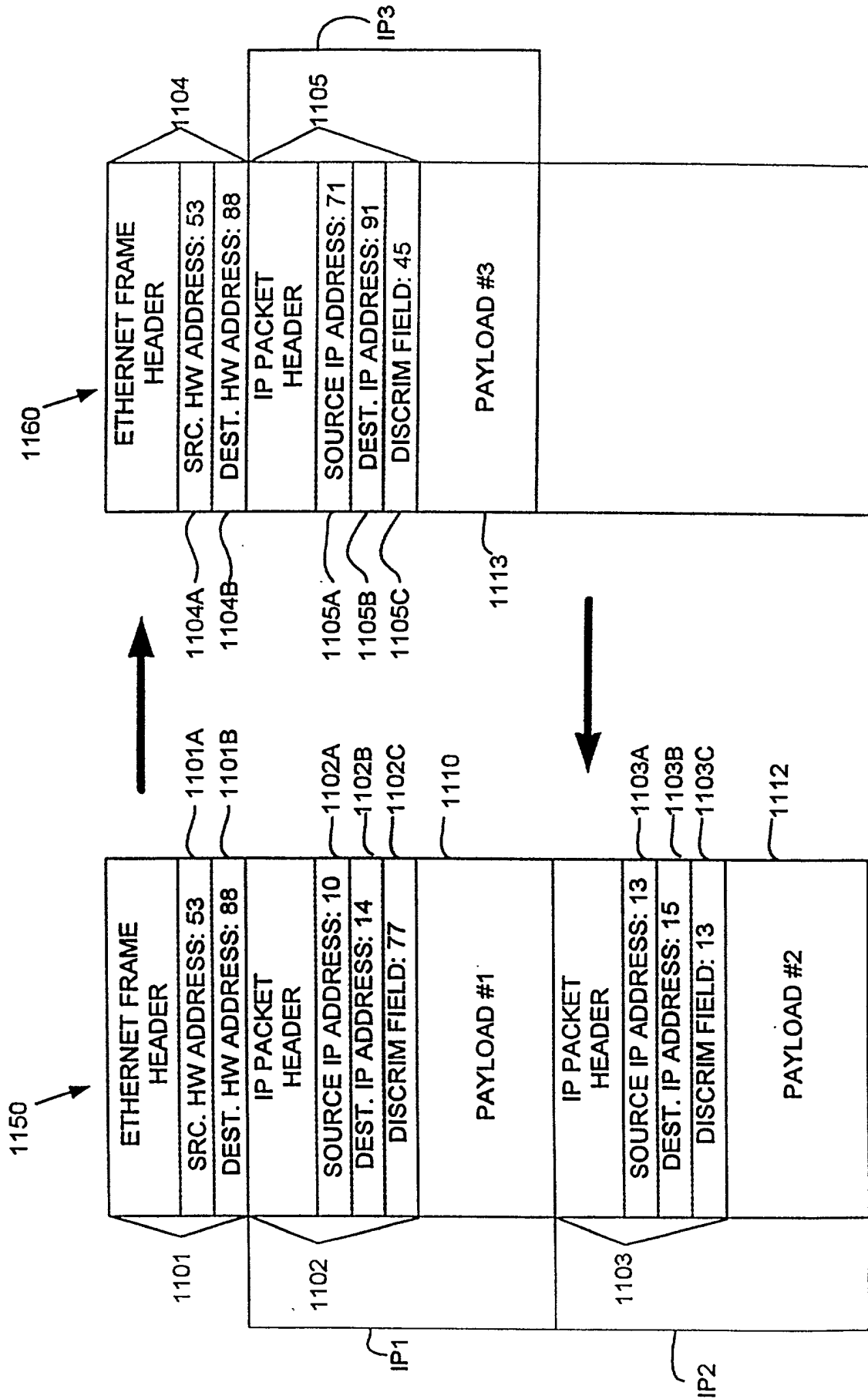


FIG. 11

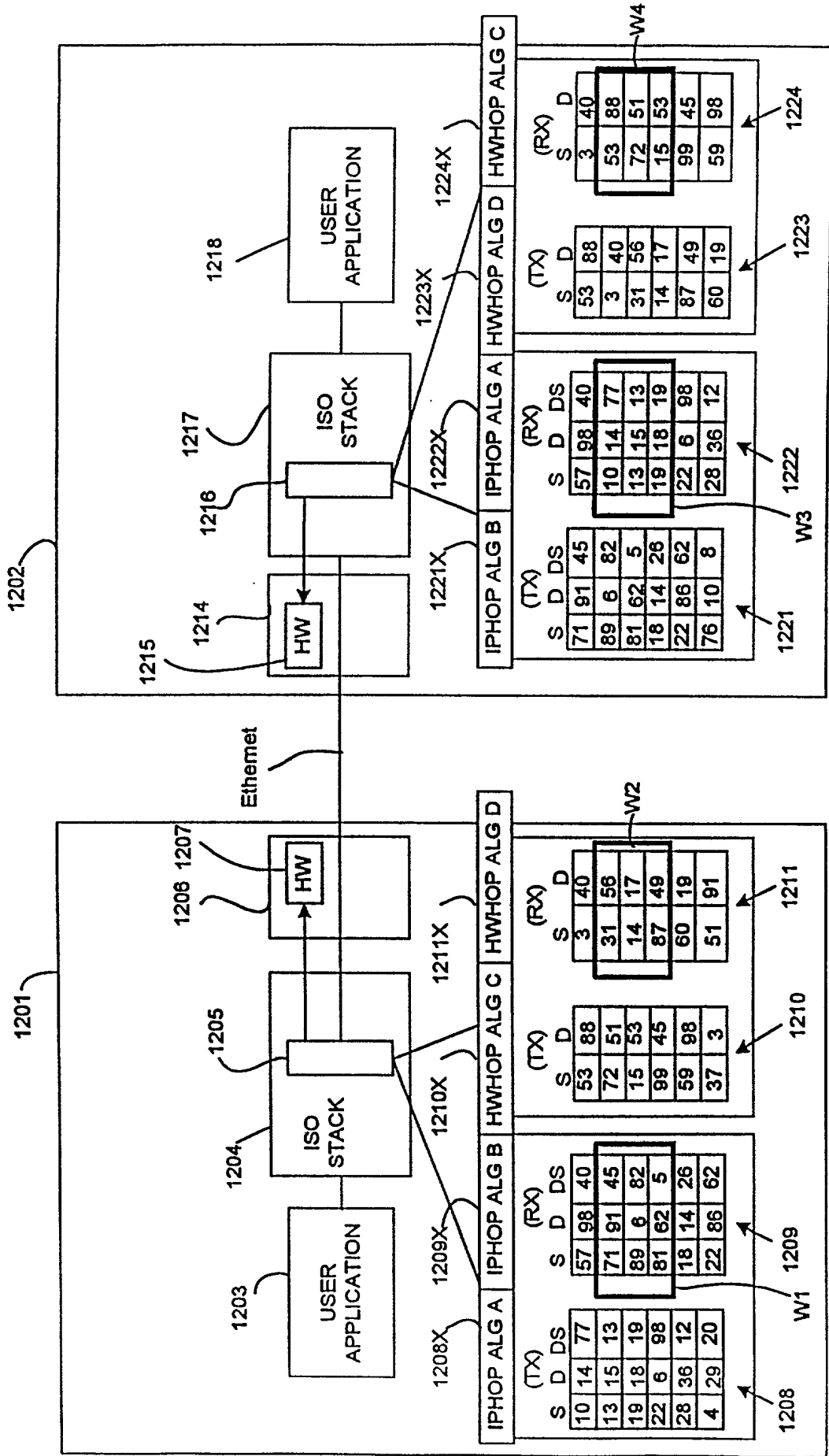


FIG. 12A

MODE OR EMBODIMENT	HARDWARE ADDRESSES	IP ADDRESSES	DISCRIMINATOR FIELD VALUES
1. PROMISCUOUS	SAME FOR ALL NODES OR COMPLETELY RANDOM	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC
2. PROMISCUOUS PER VPN	FIXED FOR EACH VPN	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC
3. HARDWARE HOPPING	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC

FIG. 12B



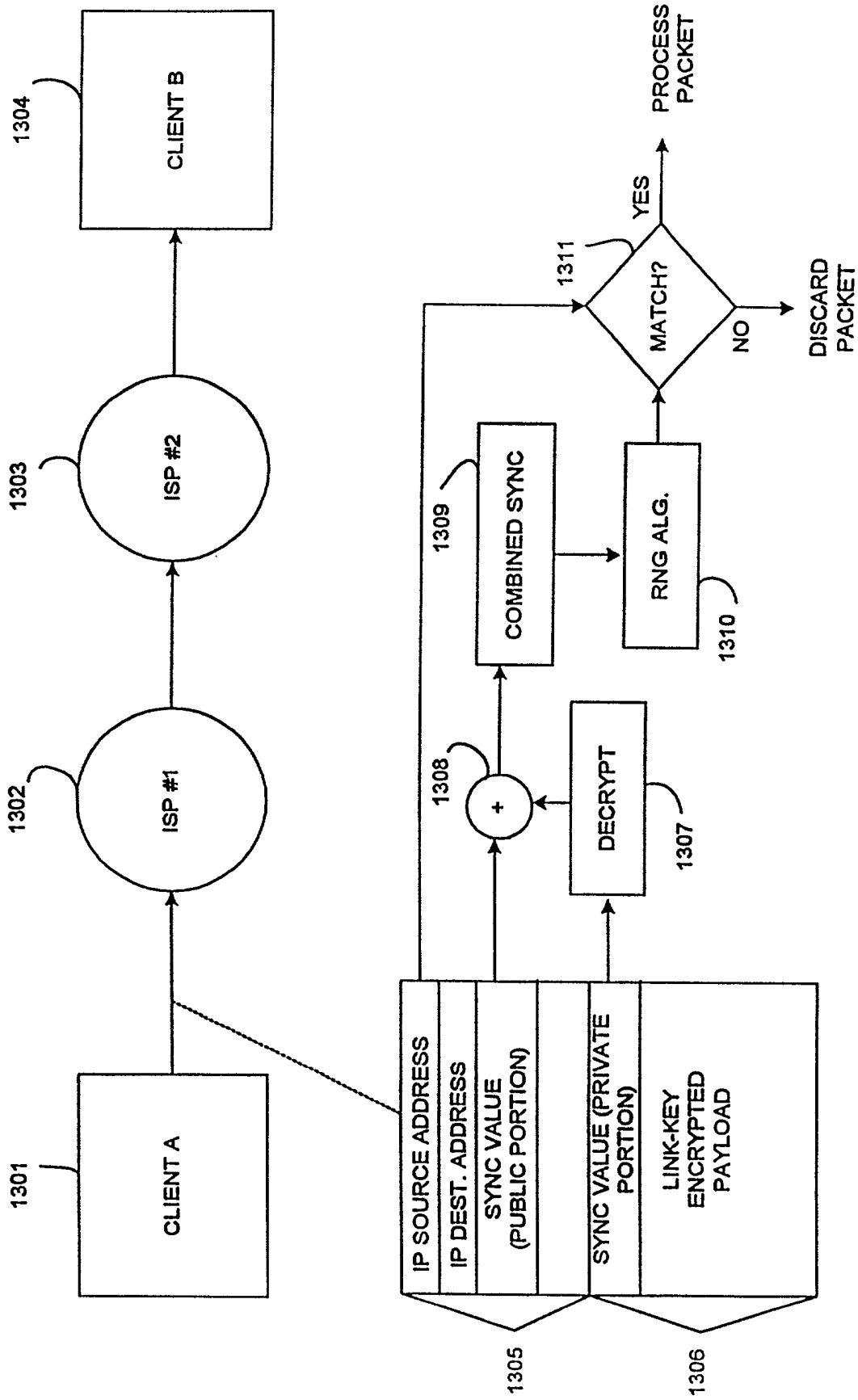


FIG. 13



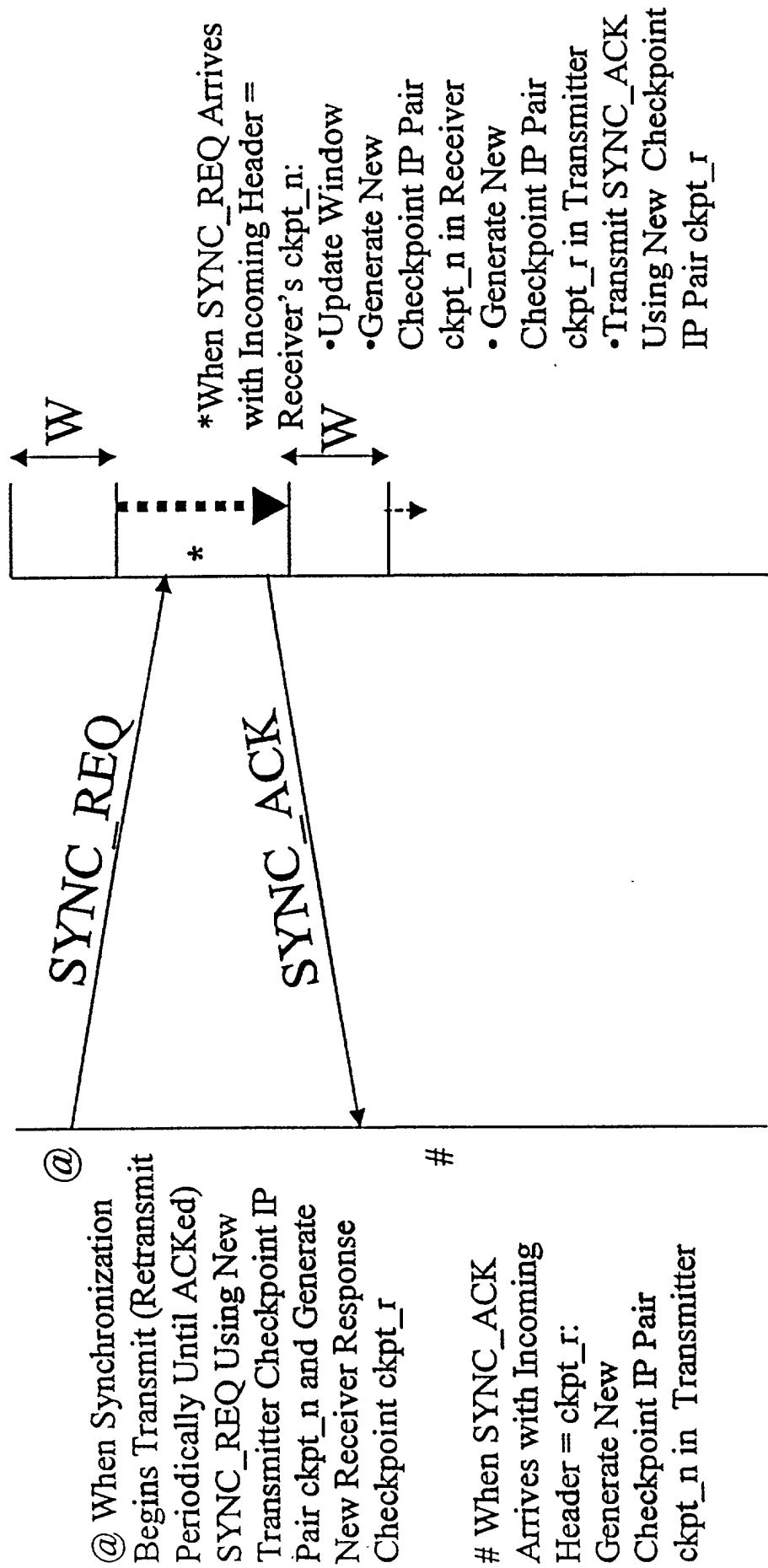


FIG. 15

### (Ethernet Lan - Two A Address Blocks)

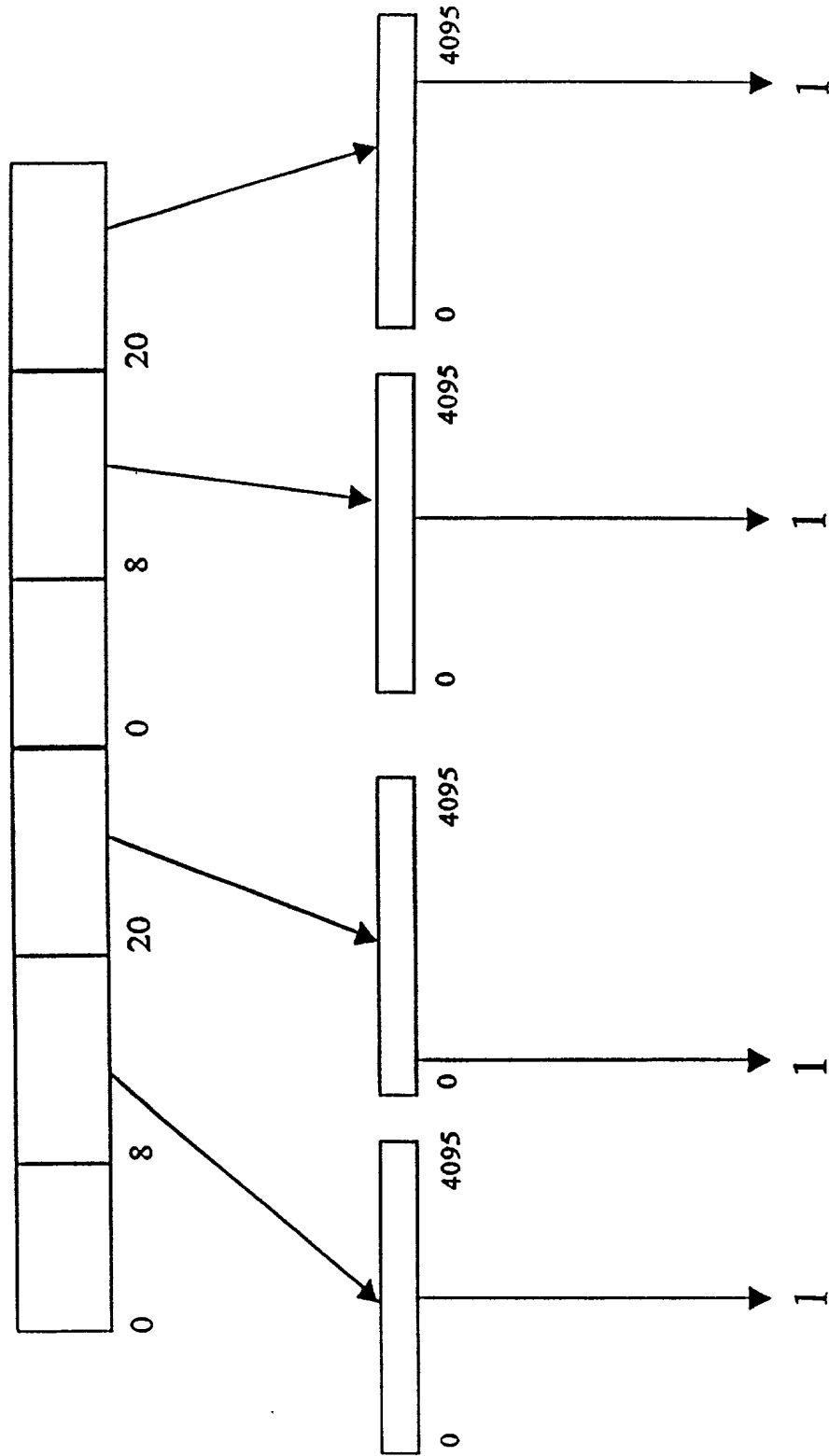


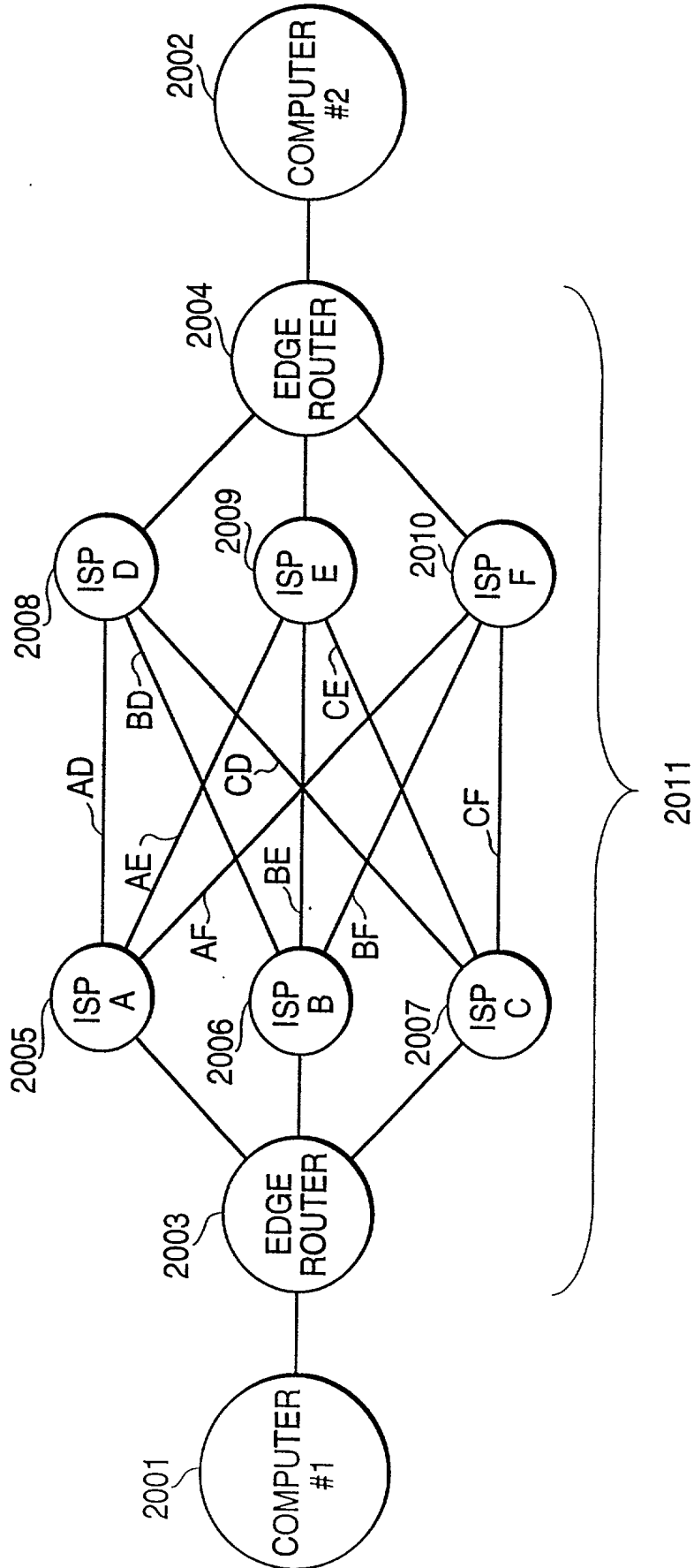
FIG. 16





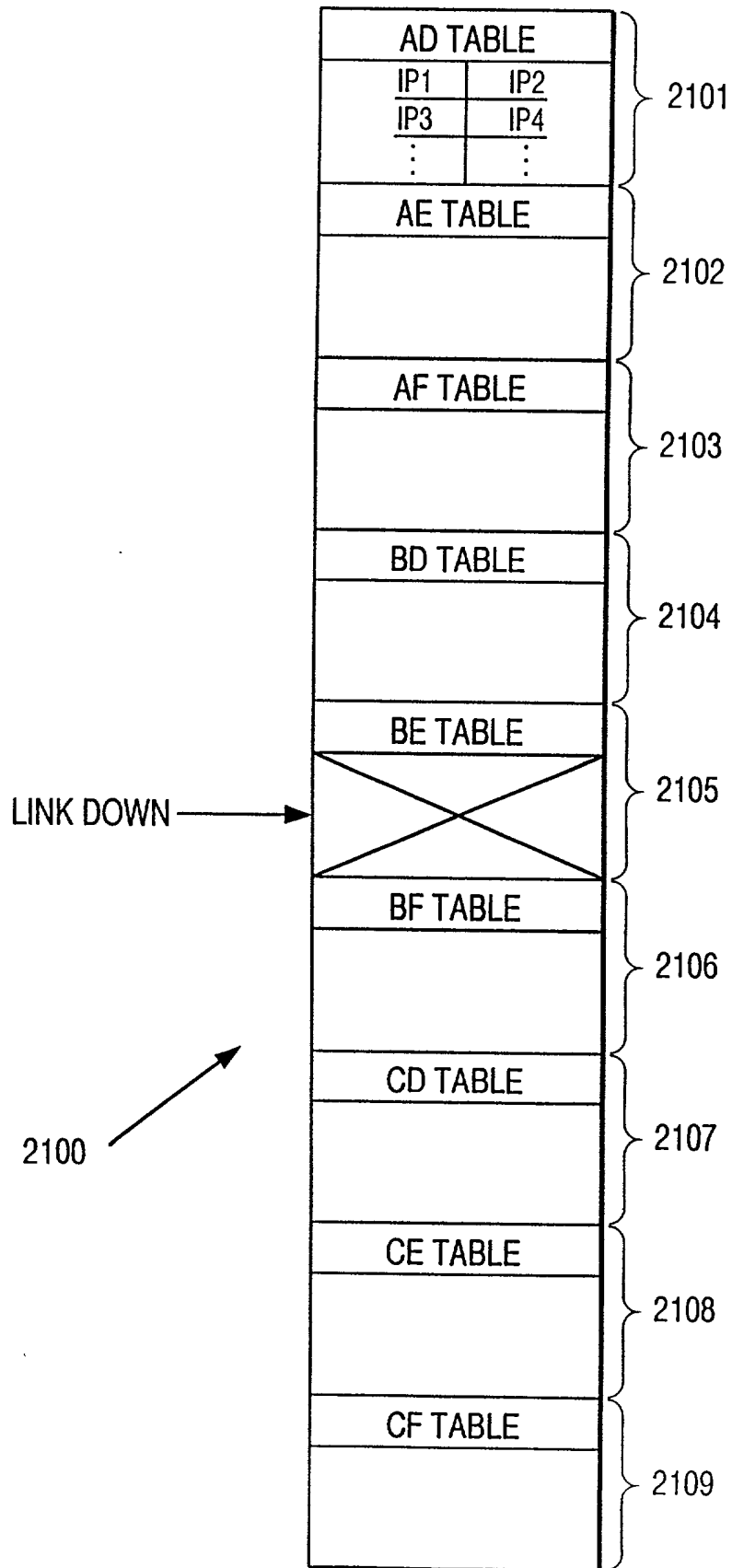


FIG. 20





**FIG. 21**



Patent Application No. 10/200,000

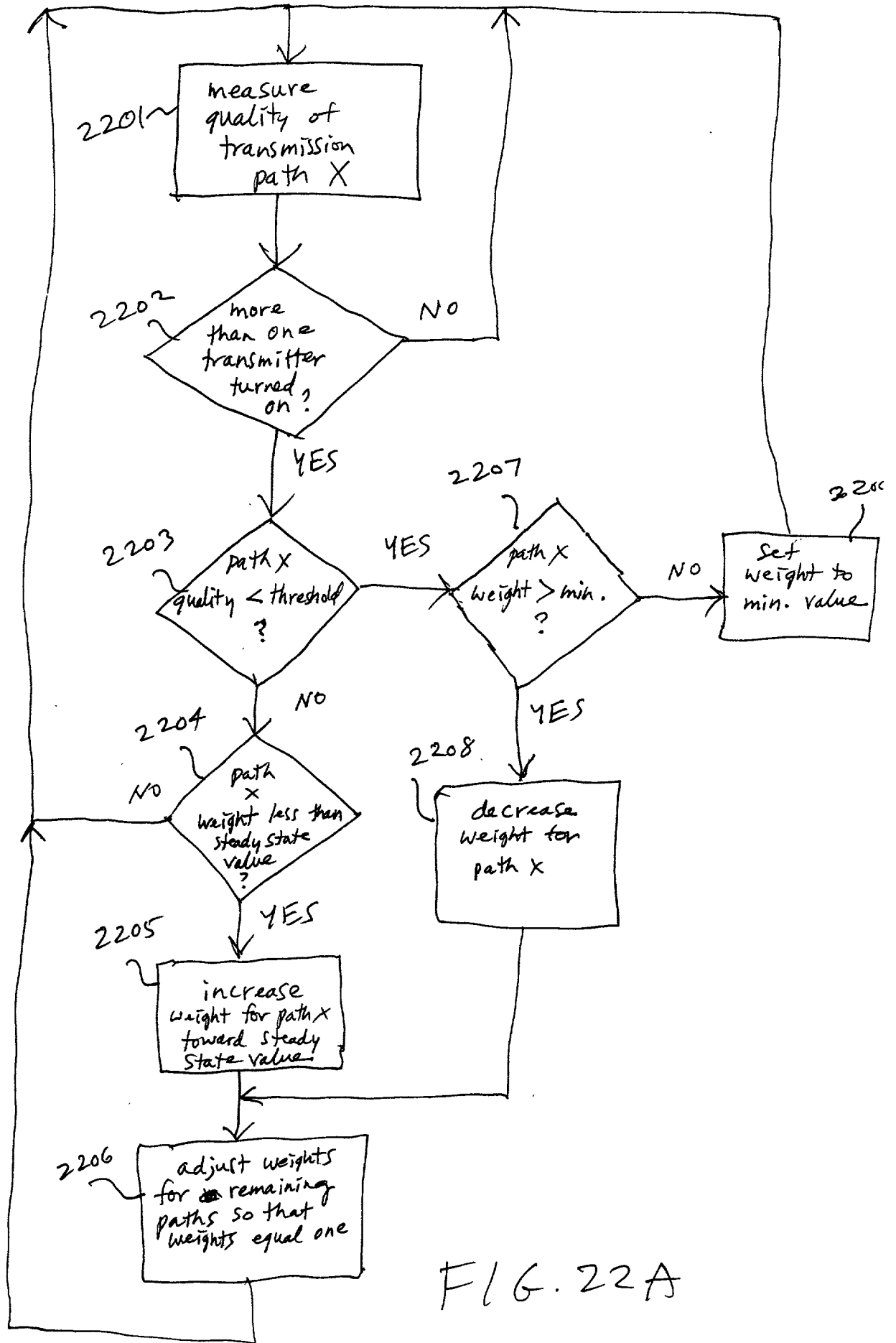


FIG. 22A

01/02/2012 10:00 AM

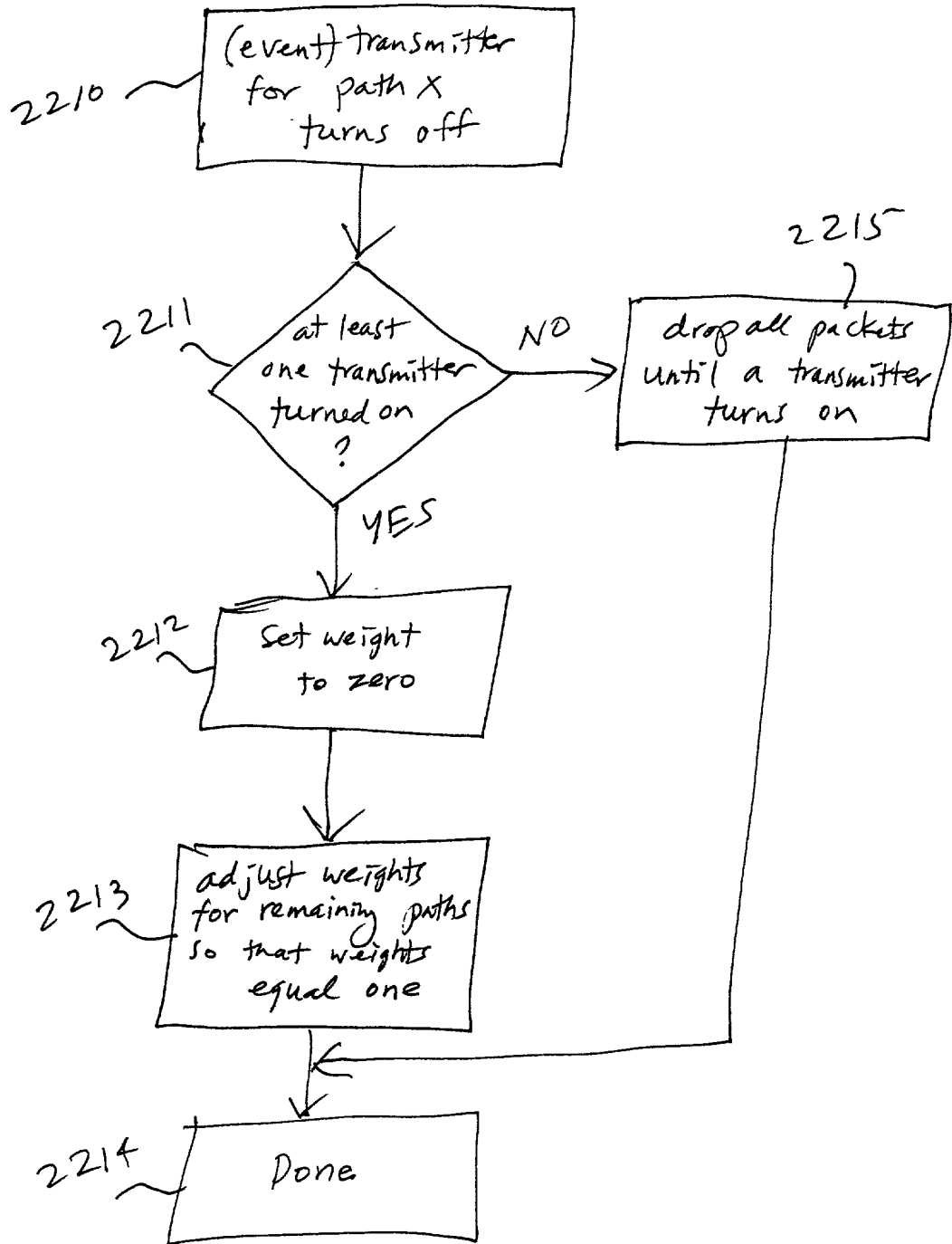
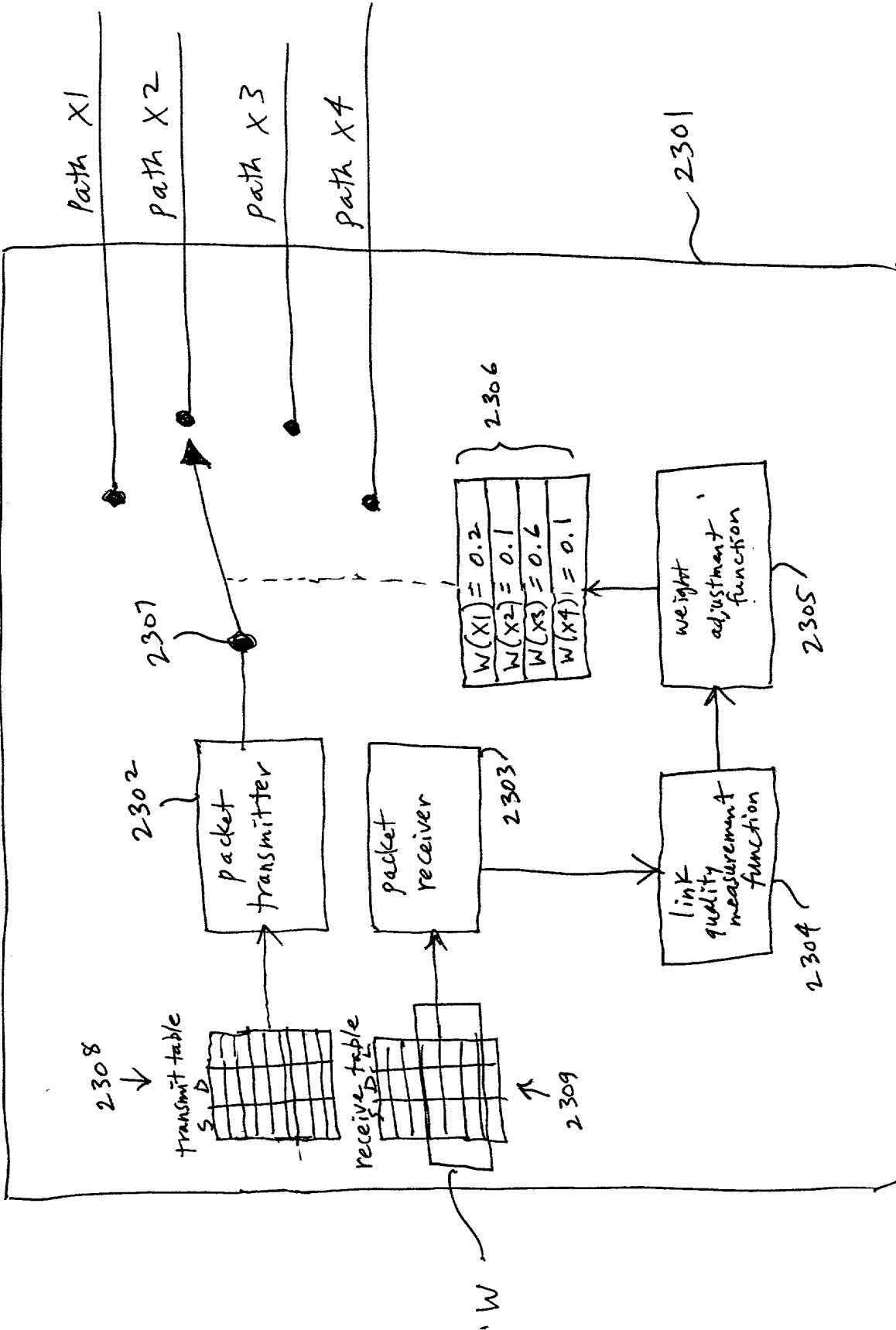


FIG. 22B



F/G.23

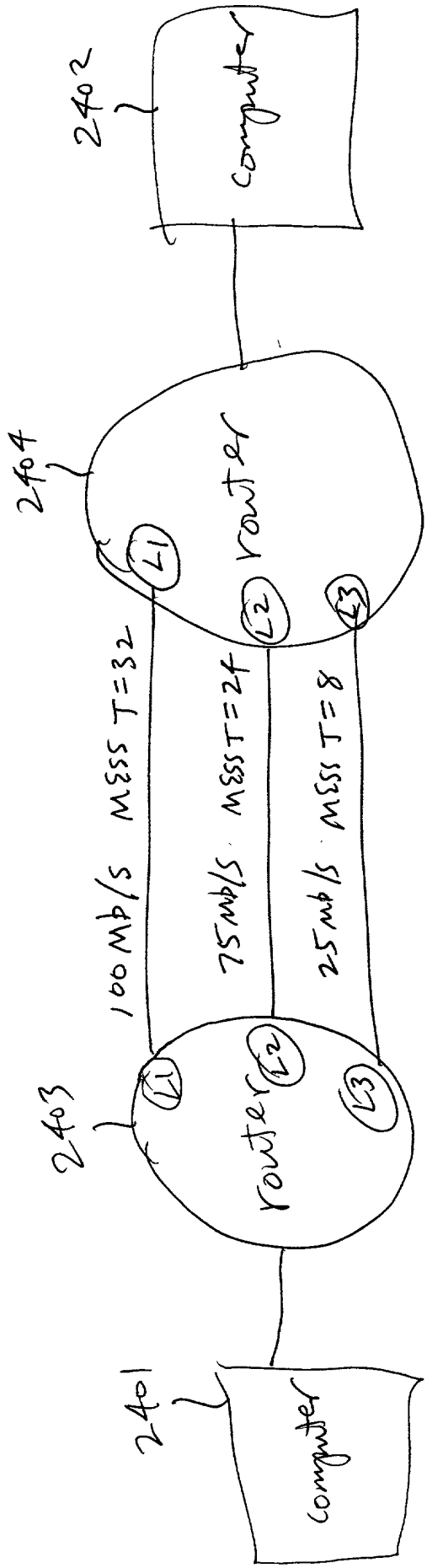


FIG. 24

2502

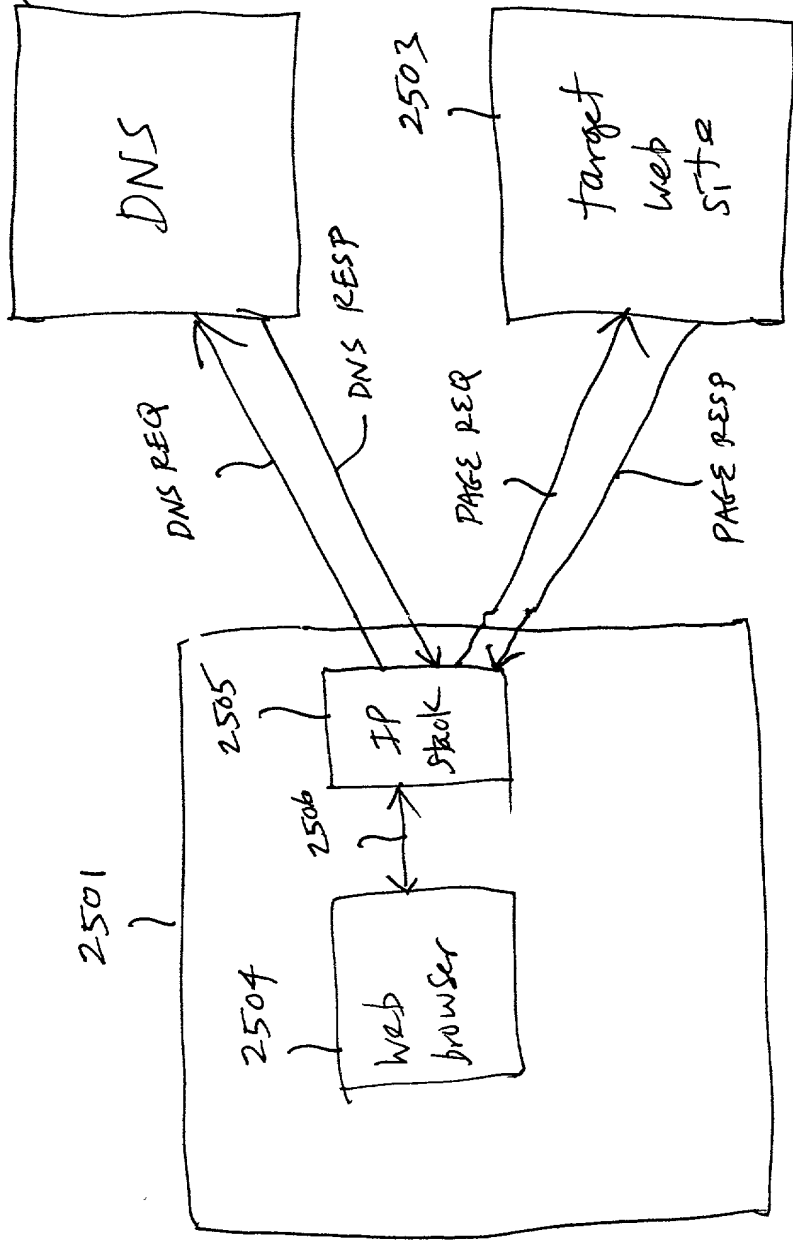


FIG- 25  
(prior art)

Copyright © 2012 by RPX Corporation. All rights reserved. No part of this document may be reproduced without the prior written permission of RPX Corporation.

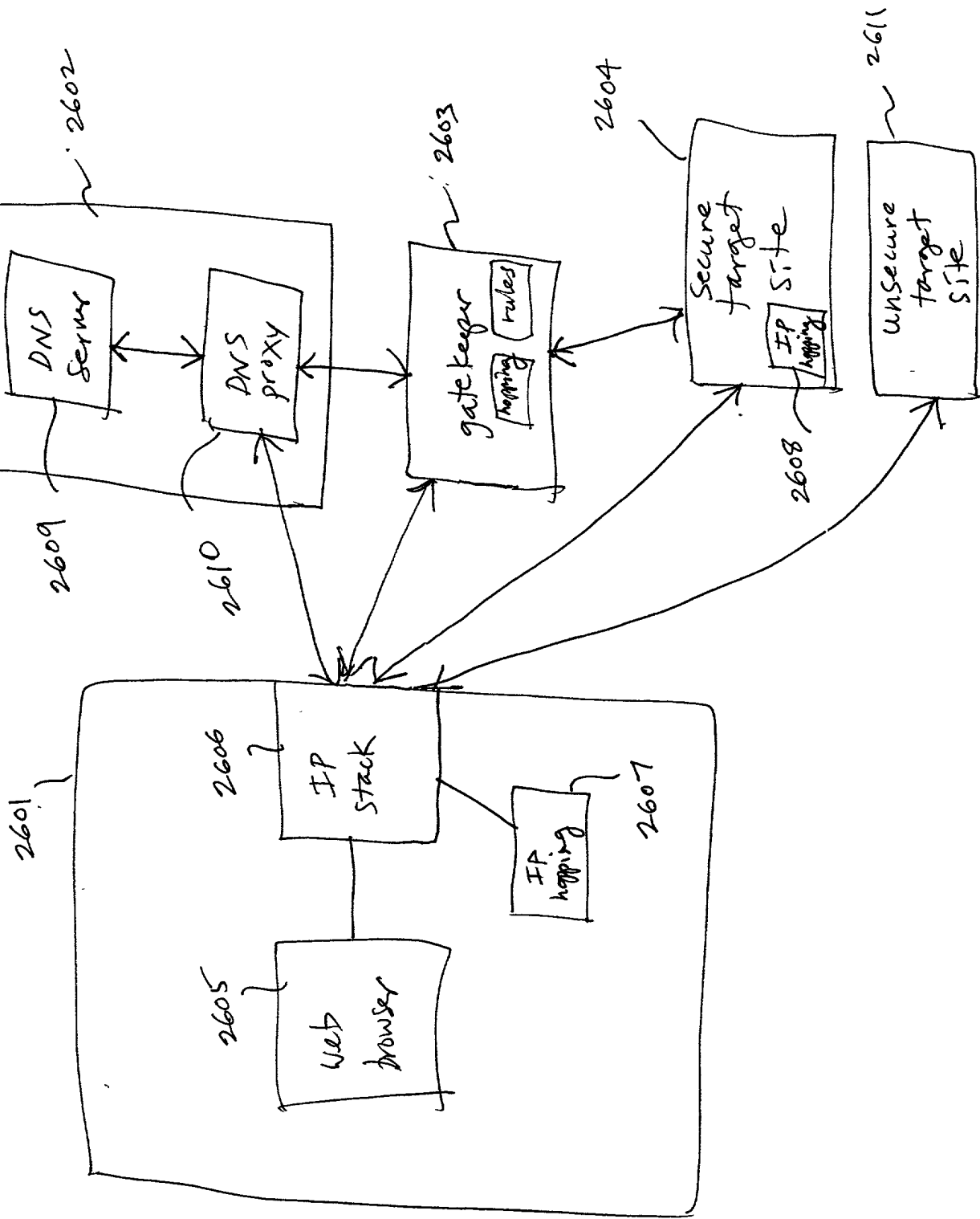


FIG. 26

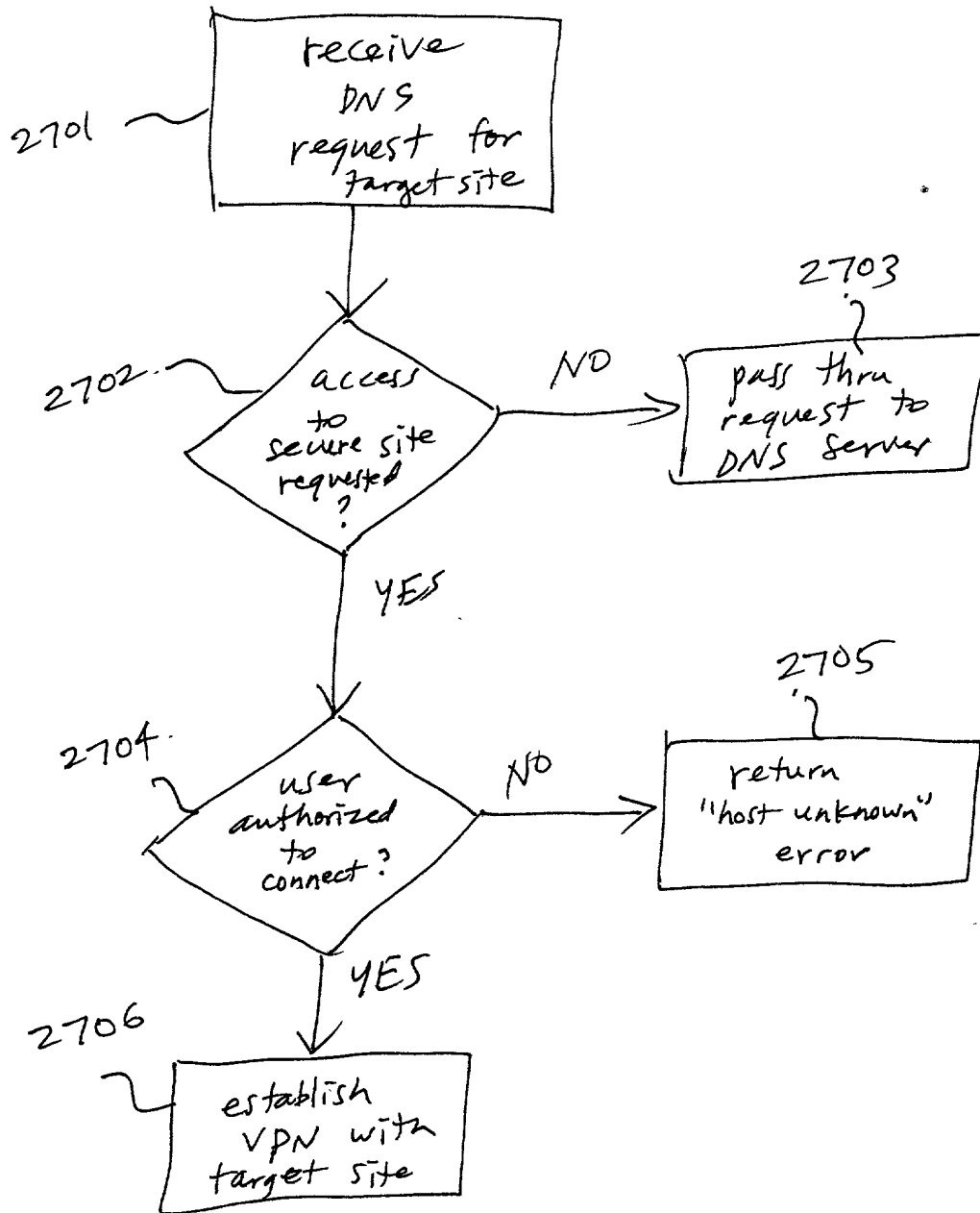


FIG. 27



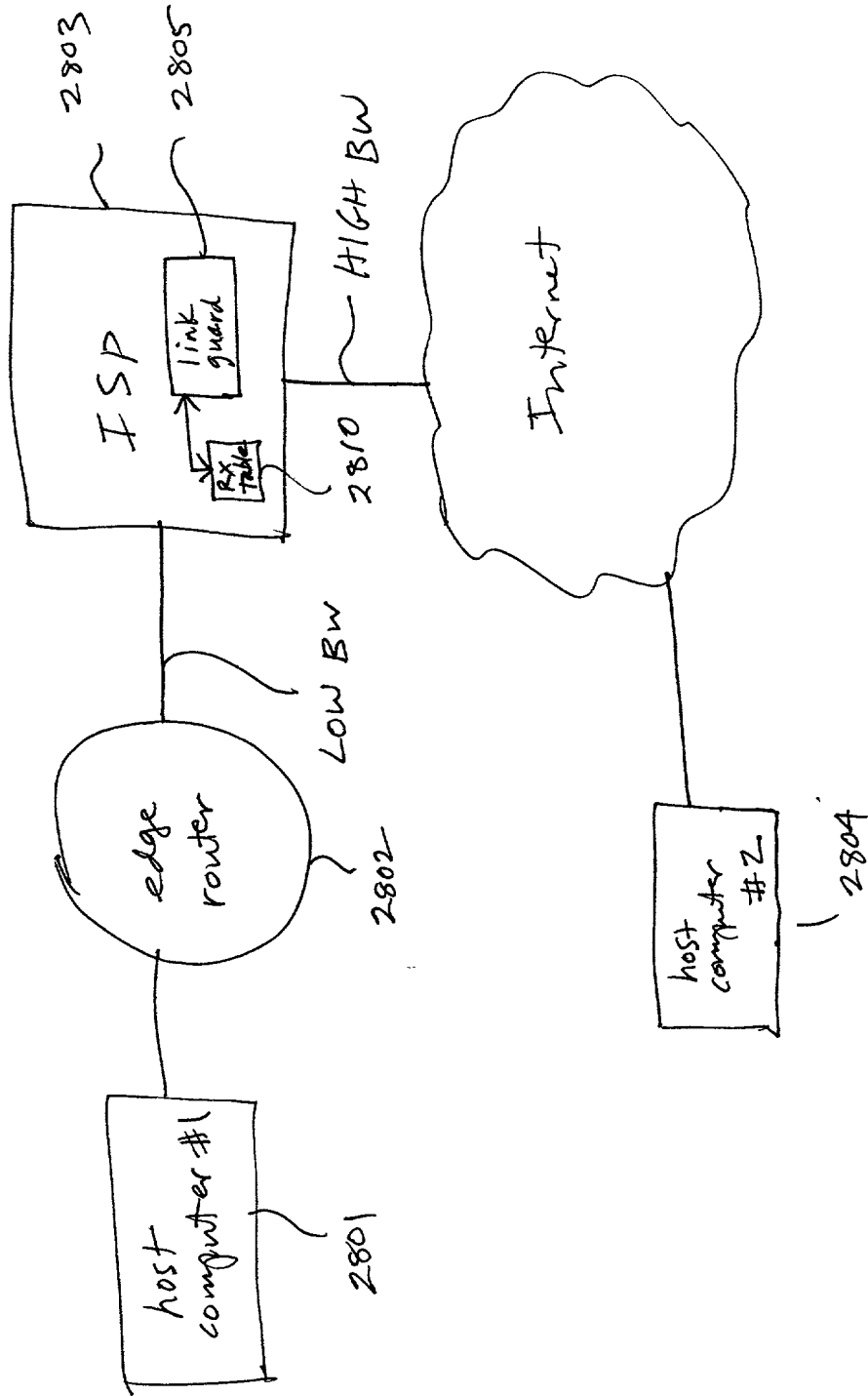


FIG. 28

FIG. 29

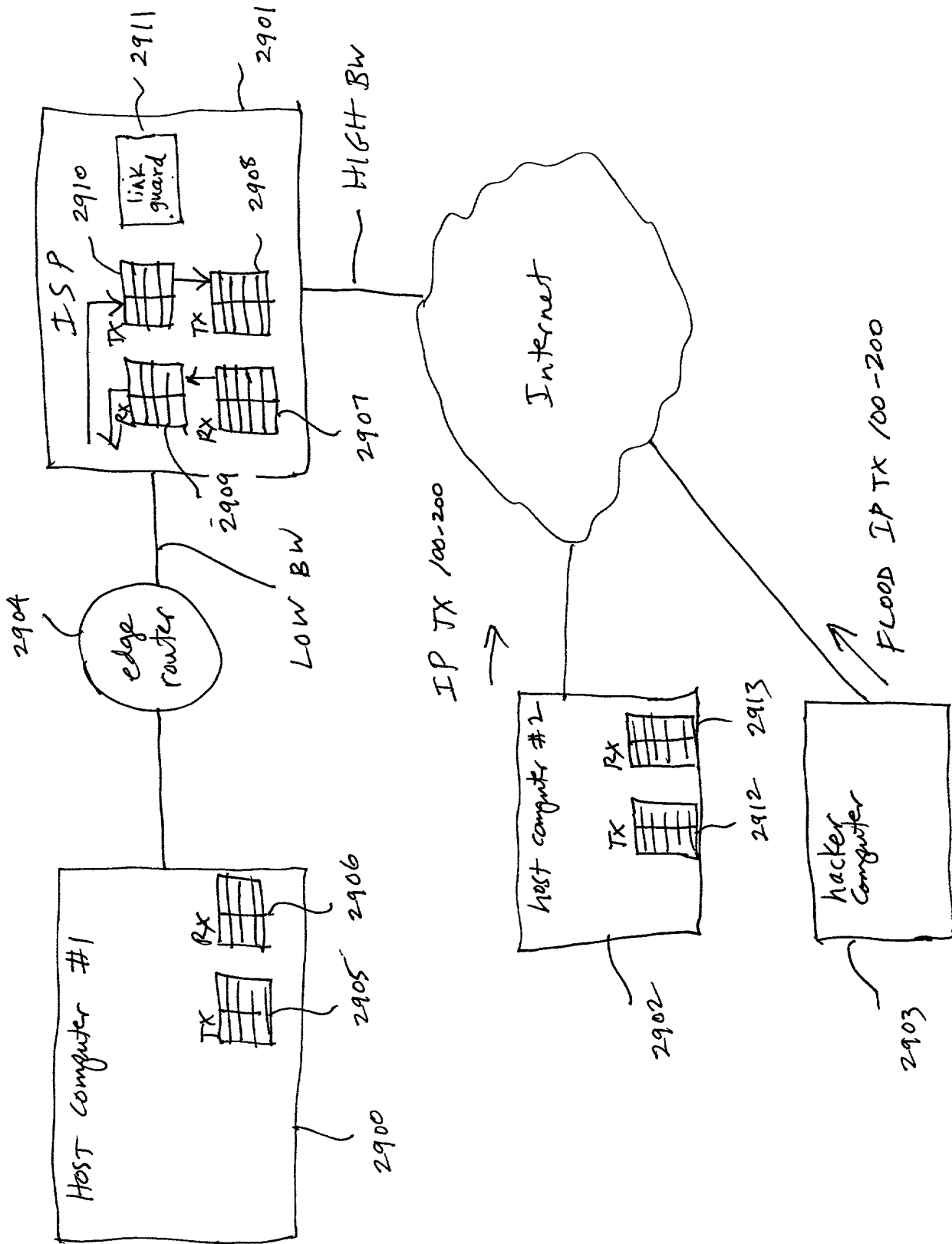


FIG. 29

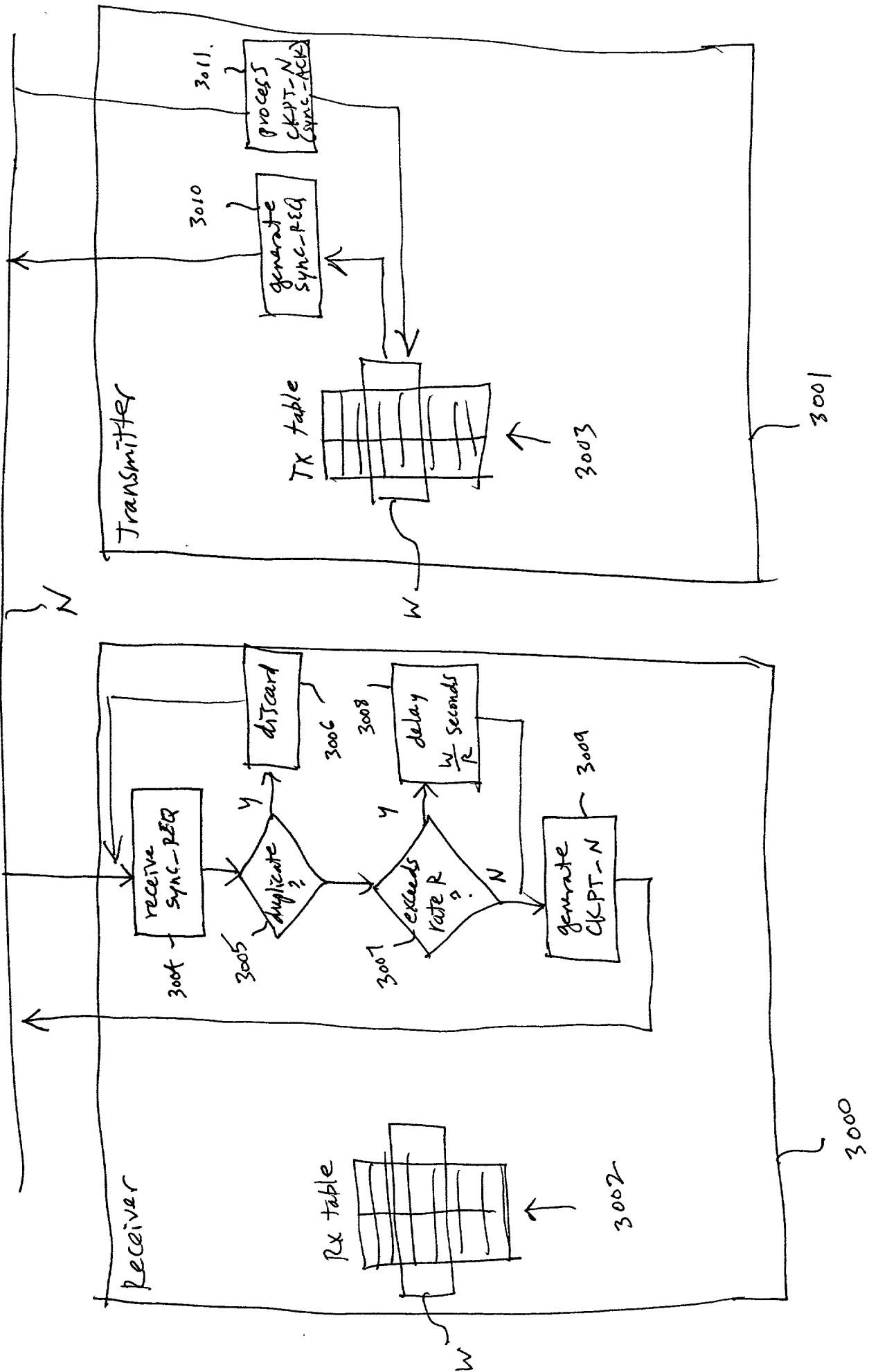


FIG. 30

FIG. 31 is a block diagram of a system architecture. The diagram shows a central server (3101) connected to three clients (3103, 3104, 3105) and a hacker (3105). The server (3101) contains two sets of components (3106, 3107) and a TX/RX interface (3102). The TX/RX interface (3102) is connected to three TX/RX units (3208, 3209, 3210). The server (3101) is connected to Client #1 (3103) via a bidirectional link (3111). Client #1 (3103) contains a set of components (3106). Client #1 (3103) is connected to the server (3101) via a bidirectional link (3114). Client #2 (3104) is connected to the server (3101) via a bidirectional link (3115). The hacker (3105) is connected to the server (3101) via a bidirectional link (3113).

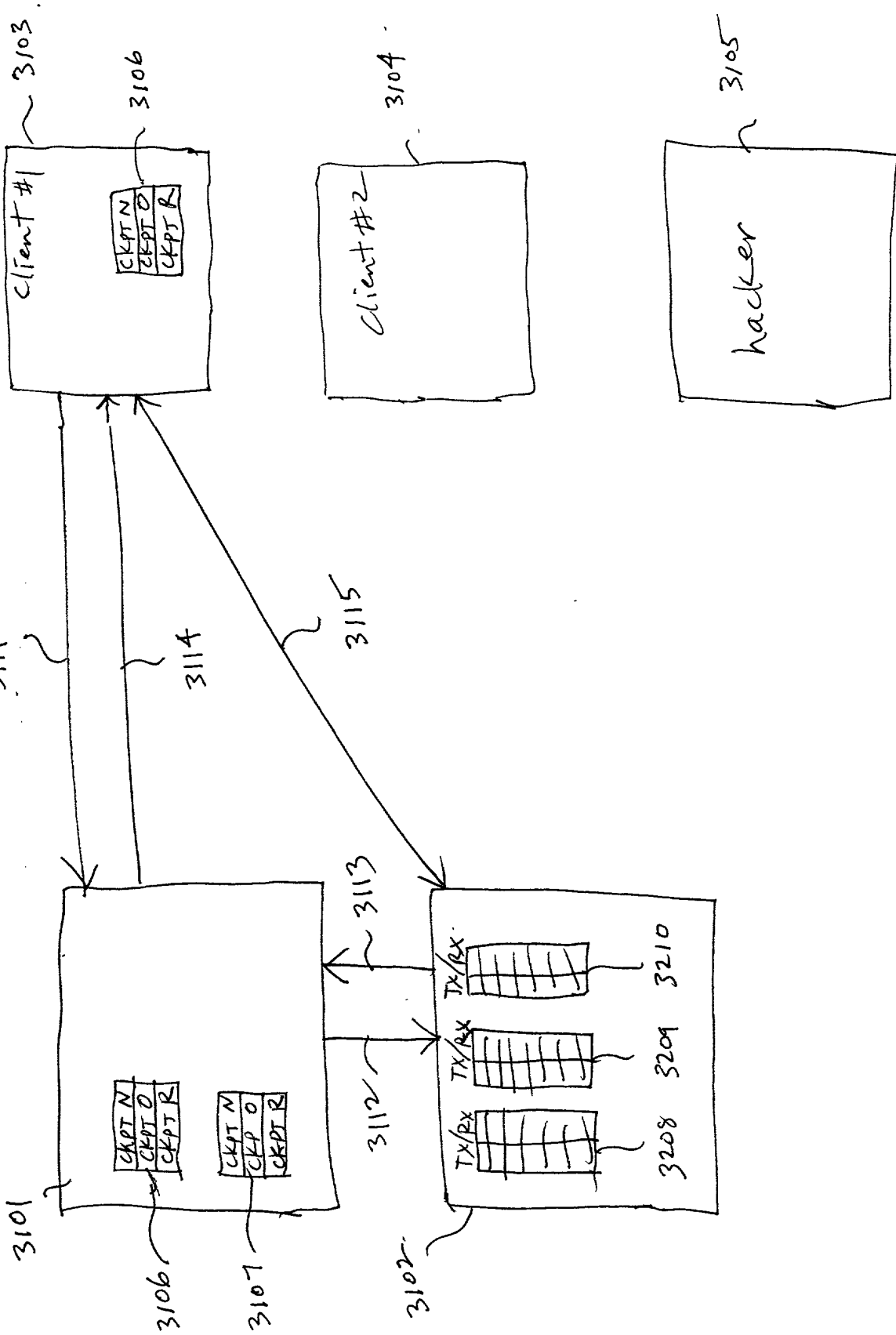


FIG. 31

Client

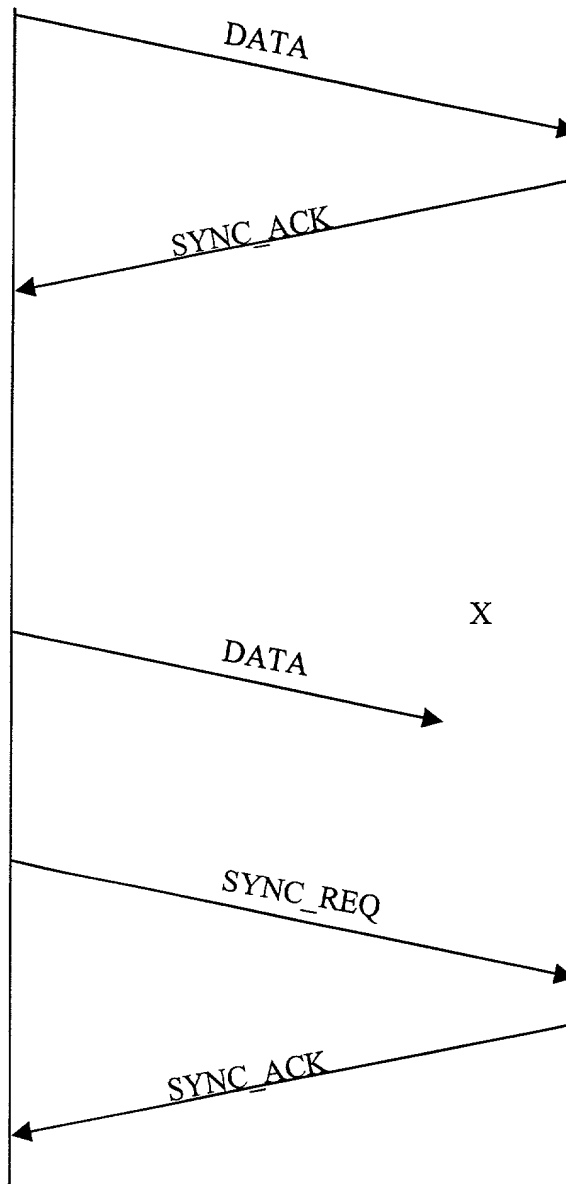
Server

Send Data Packet  
Using CKPT\_N  
CKPT\_O=CKPT\_N  
Generate New CKPT\_N  
Start timer, Shut transmitter  
Off  
If CKPT\_O in SYNC\_ACK  
matches Transmitter's  
CKPT\_O  
Update Receiver's  
CKPT\_R  
Kill Timer, Turn  
Transmitter On

Send Data Packet  
Using CKPT\_N  
CKPT\_O=CKPT\_N  
Generate New CKPT\_N  
Start timer, Shut transmitter  
Off

When timer expires  
Transmit SYNC\_REQ  
using Transmitters  
CKPT\_O, Start Timer

If CKPT\_O in SYNC\_ACK  
matches Transmitter's  
CKPT\_O  
Update Receiver's  
CKPT\_R  
Kill Timer, Turn  
Transmitter On



Pass Data Up Stack  
CKPT\_O=CKPT\_N  
Generate new CKPT\_N  
Generate New CKPT\_R for  
Transmitter Side  
Transmit SYNC\_ACK  
containing CKPT\_O

CKPT\_O=CKPT\_N  
Generate new CKPT\_N  
Generate New CKPT\_R for  
Transmitter Side  
Transmit SYNC\_ACK  
containing CKPT\_O

FIG. 32

## JOINT DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As the below named inventors, we hereby declare that:

Our residences, post office addresses and citizenships are as stated below next to our names:

We believe we are the original, first and joint inventors of the subject matter which is claimed and for which a patent is sought on the invention entitled:

### IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

the specification of which

- is attached hereto.  
 was filed on \_\_\_\_\_ as Application Serial Number \_\_\_\_\_ and was amended on \_\_\_\_\_ (if applicable).

We hereby state that we have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

We acknowledge the duty to disclose information which is material to patentability in accordance with Title 37, Code of Federal Regulations, §1.56.

### Prior Foreign Application(s)

We hereby claim foreign priority benefits under Title 35, United States Code, §119(a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below any foreign application(s) for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Country	Application Number	Date of Filing (day, month, year)	Date of Issue (day, month, year)	Priority Claimed Under 35 U.S.C. 119	Certified Copy Attached
				Yes / No	Yes / No

### Prior United States Application(s)

We hereby claim the benefit under 35 U.S.C. 119(e) of any United States provisional application(s) listed below:

Application Number(s)	Filing Date (MM/DD/YYYY)	<input type="checkbox"/> Additional provisional application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto.
60/106,261	10/30/98	
60/137,704	6/7/99	

We hereby claim the benefit under Title 35, United States Code, §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, §112, We acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Application Serial Number	Date of Filing (Day, Month, Year)	Status Patented, Pending, Abandoned
09/429,643	10/29/99	Pending

**Power of Attorney**

And we hereby appoint, both jointly and severally, as our attorneys with full power of substitution and revocation, to prosecute this application and transact all business in the U.S. Patent and Trademark Office connected herewith as well as before any office or agency of a foreign country or any international organization in connection with any foreign counterpart application claiming priority to this application, including the power to appoint agents and local representatives in connection with such foreign applications, the following attorneys of Banner & Witcoff, their registration numbers being listed after their names:

Robert Altherr, Reg. No. 31,810, Donald W. Banner, Reg. No. 17,037; Edward F. McKie, Jr., Reg. No. 17,335; William W. Beckett, Reg. No. 18,262; Dale H. Hoscheit, Reg. No. 19,090; Joseph M. Potenza, Reg. No. 28,175; James A. Niegowski, Reg. No. 28,331; Joseph M. Skerpon, Reg. No. 29,864; Thomas L. Peterson, Reg. No. 30,969; Nina L. Medlock, Reg. No. 29,673; William J. Fisher, Reg. No. 32,133; Thomas H. Jackson, Reg. No. 29,808; Franklin D. Wolffe, Reg. No. 19,724; Susan A. Wolffe, Reg. No. 33,568; Daniel E. Fisher, Reg. No. 34,162; Kevin A. Wolff, Reg. No. 42,233 and Bradley C. Wright, Reg. No. 38,061.

All correspondence and telephone communications should be addressed to:

Banner & Witcoff, Ltd.  
 Eleventh Floor  
 1001 G Street, N.W.  
 Washington, D.C. 20001-4597  
 Tel. No. (202) 508-9100

We hereby declare that all statements made herein of our 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 18 U.S.C. 1001 and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature \_\_\_\_\_ Date \_\_\_\_\_

Full Name of \_\_\_\_\_

Joint Inventor MUNGER Edmund Colby  
 Family Name First Given Name Second Given Name

Residence 1101 Opaca Court, Crownsville, Maryland 21032

Citizenship U.S.

Post Office Address 1101 Opaca Court, Crownsville, Maryland 21032

Signature \_\_\_\_\_ Date \_\_\_\_\_

Full Name of Joint Inventor SCHMIDT Douglas Charles  
Family Name First Given Name Second Given Name

Residence 230 Oak Court, Severna Park, Maryland 21146

Citizenship U.S.

Post Office Address 230 Oak Court, Severna Park, Maryland 21146

Signature *Robert D. Short* Date 2/14/00

Full Name of Joint Inventor SHORT Robert Dunham, III  
Family Name First Given Name Second Given Name

Residence 38710 Goose Creek Lane, Leesburg, Virginia 20175

Citizenship U.S.

Post Office Address 38710 Goose Creek Lane, Leesburg, Virginia 20175

Signature *Victor Larson* Date 2/14/2000

Full Name of Joint Inventor LARSON Victor  
Family Name First Given Name Second Given Name

Residence 12026 Lisa Marie Court, Fairfax, Virginia 22033

Citizenship U.S.

Post Office Address 12026 Lisa Marie Court, Fairfax, Virginia 22033





## JOINT DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As the below named inventors, we hereby declare that:

Our residences, post office addresses and citizenships are as stated below next to our names:

We believe we are the original, first and joint inventors of the subject matter which is claimed and for which a patent is sought on the invention entitled:

### IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

the specification of which

is attached hereto.

was filed on \_\_\_\_\_ as Application Serial Number \_\_\_\_\_ and was amended on \_\_\_\_\_ (if applicable).

We hereby state that we have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

We acknowledge the duty to disclose information which is material to patentability in accordance with Title 37, Code of Federal Regulations, §1.56.

### Prior Foreign Application(s)

We hereby claim foreign priority benefits under Title 35, United States Code, §119(a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below any foreign application(s) for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Country	Application Number	Date of Filing (day, month, year)	Date of Issue (day, month, year)	Priority Claimed Under 35 U.S.C. 119	Certified Copy Attached
				Yes / No	Yes / No

### Prior United States Application(s)

We hereby claim the benefit under 35 U.S.C. 119(e) of any United States provisional application(s) listed below:

Application Number(s)	Filing Date (MM/DD/YYYY)	
60/106,261	10/30/98	<input type="checkbox"/> Additional provisional application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto.
60/137,704	6/7/99	

We hereby claim the benefit under Title 35, United States Code, §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, §112, We acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Application Serial Number	Date of Filing (Day, Month, Year)	Status Patented, Pending, Abandoned
09/429,643	10/29/99	Pending

**Power of Attorney**

And we hereby appoint, both jointly and severally, as our attorneys with full power of substitution and revocation, to prosecute this application and transact all business in the U.S. Patent and Trademark Office connected herewith as well as before any office or agency of a foreign country or any international organization in connection with any foreign counterpart application claiming priority to this application, including the power to appoint agents and local representatives in connection with such foreign applications, the following attorneys of Banner & Witcoff, their registration numbers being listed after their names:

Robert Altherr, Reg. No. 31,810, Donald W. Banner, Reg. No. 17,037; Edward F. McKie, Jr., Reg. No. 17,335; William W. Beckett, Reg. No. 18,262; Dale H. Hoscheit, Reg. No. 19,090; Joseph M. Potenza, Reg. No. 28,175; James A. Niegowski, Reg. No. 28,331; Joseph M. Skerpon, Reg. No. 29,864; Thomas L. Peterson, Reg. No. 30,969; Nina L. Medlock, Reg. No. 29,673; William J. Fisher, Reg. No. 32,133; Thomas H. Jackson, Reg. No. 29,808; Franklin D. Wolffe, Reg. No. 19,724; Susan A. Wolffe, Reg. No. 33,568; Daniel E. Fisher, Reg. No. 34,162; Kevin A. Wolff, Reg. No. 42,233 and Bradley C. Wright, Reg. No. 38,061.

All correspondence and telephone communications should be addressed to:

Banner & Witcoff, Ltd.  
 Eleventh Floor  
 1001 G Street, N.W.  
 Washington, D.C. 20001-4597  
 Tel. No. (202) 508-9100

We hereby declare that all statements made herein of our 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 18 U.S.C. 1001 and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature \_\_\_\_\_ Date \_\_\_\_\_

Full Name of Joint Inventor MUNGER Edmund Colby  
 Family Name First Given Name Second Given Name

Residence 1101 Opaca Court, Crownsville, Maryland 21032

Citizenship U.S.

Post Office Address 1101 Opaca Court, Crownsville, Maryland 21032

Signature Dy C Sch Date 2/14/00

Full Name of Joint Inventor SCHMIDT Douglas Charles  
Family Name First Given Name Second Given Name

Residence 230 Oak Court, Severna Park, Maryland 21146

Citizenship U.S.

Post Office Address 230 Oak Court, Severna Park, Maryland 21146

Signature \_\_\_\_\_ Date \_\_\_\_\_

Full Name of Joint Inventor SHORT Robert Dunham, III  
Family Name First Given Name Second Given Name

Residence 38710 Goose Creek Lane, Leesburg, Virginia 20175

Citizenship U.S.

Post Office Address 38710 Goose Creek Lane, Leesburg, Virginia 20175

Signature \_\_\_\_\_ Date \_\_\_\_\_

Full Name of Joint Inventor LARSON Victor  
Family Name First Given Name Second Given Name

Residence 12026 Lisa Marie Court, Fairfax, Virginia 22033

Citizenship U.S.

Post Office Address 12026 Lisa Marie Court, Fairfax, Virginia 22033



Attorney Doctort No. 00479.85672

### JOINT DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As the below named inventors, we hereby declare that:

Our residences, post office addresses and citizenships are as stated below next to our names:

We believe we are the original, first and joint inventors of the subject matter which is claimed and for which a patent is sought on the invention entitled:

#### IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

the specification of which

is attached hereto.

was filed on \_\_\_\_\_ as Application Serial Number \_\_\_\_\_ and was amended on \_\_\_\_\_ (if applicable).

We hereby state that we have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

We acknowledge the duty to disclose information which is material to patentability in accordance with Title 37, Code of Federal Regulations, §1.56.

#### Prior Foreign Application(s)

We hereby claim foreign priority benefits under Title 35, United States Code, §119(a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below any foreign application(s) for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Country	Application Number	Date of Filing (Month/Day/Year)	Date of Issue (Month/Day/Year)	Priority Claimed (Under 35 U.S.C. 119)	Certified Copy Submitted
				Yes / No	Yes / No

#### Prior United States Application(s)

We hereby claim the benefit under 35 U.S.C. 119(e) of any United States provisional application(s) listed below:

Application Number(s)	Filing Date (Month/Day/Year)	<input type="checkbox"/> Additional provisional application numbers are listed on a supplemental priority data sheet PTO/SB/028 attached hereto.
60/106,261	10/30/98	
60/137,704	6/7/99	

02/15/00 14:22 FAX

003

Attorney Docket No. 00479.85672

We hereby claim the benefit under Title 35, United States Code, §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, §112. We acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Application No.	Date of Filing (App. No., Date)	Status (Foreign Patent Application)
09/429,643	10/29/99	Pending

Power of Attorney

And we hereby appoint, both jointly and severally, as our attorneys with full power of substitution and revocation, to prosecute this application and transact all business in the U.S. Patent and Trademark Office connected herewith as well as before any office or agency of a foreign country or any international organization in connection with any foreign counterpart application claiming priority to this application, including the power to appoint agents and local representatives in connection with such foreign applications, the following attorneys of Banner & Witcoff, their registration numbers being listed after their names:

Robert Alther, Reg. No. 31,810; Donald W. Barner, Reg. No. 17,037; Edward F. McKie, Jr., Reg. No. 17,335; William W. Beckett, Reg. No. 18,262; Dale H. Hoscheit, Reg. No. 19,090; Joseph M. Potenza, Reg. No. 28,175; James A. Niegowski, Reg. No. 28,331; Joseph M. Skerpton, Reg. No. 29,864; Thomas L. Peterson, Reg. No. 30,989; Nina L. Medlock, Reg. No. 29,673; William J. Fisher, Reg. No. 32,133; Thomas H. Jackson, Reg. No. 29,808; Franklin D. Wolffe, Reg. No. 19,724; Susan A. Wolffe, Reg. No. 33,568; Daniel E. Fisher, Reg. No. 34,162; Kevin A. Wolff, Reg. No. 42,233 and Bradley C. Wright, Reg. No. 38,081.

All correspondence and telephone communications should be addressed to:

Banner & Witcoff, Ltd.  
Eleventh Floor  
1001 G Street, N.W.  
Washington, D.C. 20001-4597  
Tel. No. (202) 508-9100

We hereby declare that all statements made herein of our 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 18 U.S.C. 1001 and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature Edmund Colby Munger Date 15 FEB 2000  
Full Name of Joint Inventor MUNGER Edmund Colby  
Family Name First Given Name Second Given Name

Residence 1101 Opaca Court, Crownsville, Maryland 21032

Citizenship U.S.

Post Office Address 1101 Opaca Court, Crownsville, Maryland 21032

02/15/00 14:23 FAX

004

Attorney Docket No. 00479.85672

Signature \_\_\_\_\_ Date \_\_\_\_\_

Full Name of Joint Inventor SCHMIDT Douglas Charles  
Family Name First Given Name Second Given Name

Residence 230 Oak Court, Severna Park, Maryland 21146

Citizenship U.S.

Post Office Address 230 Oak Court, Severna Park, Maryland 21146

Signature \_\_\_\_\_ Date \_\_\_\_\_

Full Name of Joint Inventor SHORT Robert Dunham, III  
Family Name First Given Name Second Given Name

Residence 38710 Goose Creek Lane, Leesburg, Virginia 20175

Citizenship U.S.

Post Office Address 38710 Goose Creek Lane, Leesburg, Virginia 20175

Signature \_\_\_\_\_ Date \_\_\_\_\_

Full Name of Joint Inventor LARSON Victor \_\_\_\_\_  
Family Name First Given Name Second Given Name

Residence 12026 Lisa Marie Court, Fairfax, Virginia 22033

Citizenship U.S.

Post Office Address 12026 Lisa Marie Court, Fairfax, Virginia 22033



02/15/00 14:23 FAX

005

Attorney Docket No. 00479.85672

Signature \_\_\_\_\_ Date \_\_\_\_\_

Full Name of	_____	
Joint Inventor	WILLIAMSON	Michael
	Family Name	First Given Name
		Second Given Name

Residence 26203 Ocala Circle, South Riding, Virginia 20152

Citizenship U.S.

Post Office Address 26203 Ocala Circle, South Riding, Virginia 20152

Vertical text on the left margin, possibly a scanning artifact or page number.

LAW OFFICES  
 BANNER & WITCOFF, LTD.  
 1001 G STREET, N.W.  
 WASHINGTON, D.C. 20001-4597  
 (202) 308-9100

Jc490 U.S. PTO  
02/15/00

NEW UNITED STATES UTILITY PATENT APPLICATION  
under 37 C.F.R. 1.53(b)

A

Page 1

Atty. Docket No. 00479.85672

Commissioner of Patents  
Box Patent Applications  
Washington, D.C. 20231

Jc135 U.S. PTO  
09/504783  
02/15/00

Enclosed herewith is a new patent application and the following papers:

First Named Inventor (or application identifier): Edmund Colby Munger, et al.

Title of Invention: Improvements To An Agile Network Protocol For Secure Communications With Assured System Availability

1.  Specification 84 pages (including specification, claims, abstract) / 71 claims (9 independent)
2.  Declaration/Power of Attorney is:
  - attached in the regular manner.
  - NOT included, but deferred under 37 C.F.R. § 1.53(f).
3.  35 Distinct sheets of  Formal  Informal Drawings
4.  Preliminary Amendment.
5.  Information Disclosure Statement
  - Form 1449
  - A copy of each cited prior art reference
6.  Assignment with Cover Sheet.
7.  Priority is hereby claimed under 35 U.S.C. § 119(e) and §120 based upon the following application(s):

Country	Application Number	Date of Filing (day, month, year)
US	60/106,261	10/30/98
US	60/137,704	6/7/99
US	09/429,643	10/29/99

8.  Priority document(s).
9.  Statement Claiming Small Entity Status.
10.  Microfiche Computer Program (Appendix).

**NEW UNITED STATES UTILITY PATENT APPLICATION**  
under 37 C.F.R. 1.53(b)

Page 2

Atty. Docket No. 00479.85672

11. Calculation of Fees:

FEES FOR	EXCESS CLAIMS	FEE	AMOUNT DUE
Basic Filing Fee (37 C.F.R. § 1.16(a))			\$690.00
Total Claims in Excess of 20 (37 C.F.R. § 1.16(c))	51	18.00	\$918.00
Independent Claims in Excess of 3 (37 C.F.R. § 1.16(b))	6	78.00	\$468.00
Multiple Dependent Claims (37 C.F.R. § 1.16(d))	0	260.00	\$0.00
Subtotal - Filing Fee Due			\$2,076.00
			REDUCE BY (%)   (\$)
Reduction by 50%, if Small Entity (37 C.F.R. §§ 1.9, 1.27, 1.28)	0		\$0.00
<b>TOTAL FILING FEE DUE</b>			\$2,076.00
Assignment Recordation Fee (if applicable) (37 C.F.R. § 1.21(h))	1	40.00	\$40.00
<b>GRAND TOTAL DUE</b>			\$2,116.00

12. PAYMENT is:

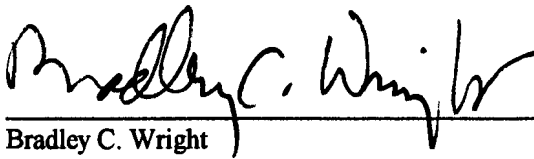
- included in the amount of the GRAND TOTAL by our enclosed check. A general authorization under 37 C.F.R. § 1.25(b), second sentence, is hereby given to credit or debit our Deposit Account No. 19-0733 for the instant filing and for any other fees during the pendency of this application under 37 C.F.R. §§ 1.16, 1.17 and 1.18.
- not included, but deferred under 37 C.F.R. § 1.53(f).

13. All correspondence for the attached application should be directed to:

Banner & Witcoff, Ltd.  
1001 G Street, N.W.  
Washington, D. C. 20001-4597  
Telephone: (202) 508-9100  
Facsimile: (202) 508-9299

14. Other: \_\_\_\_\_

Date: February 15, 2000

By:   
Bradley C. Wright  
Reg. No. 38,061

BCW/pp

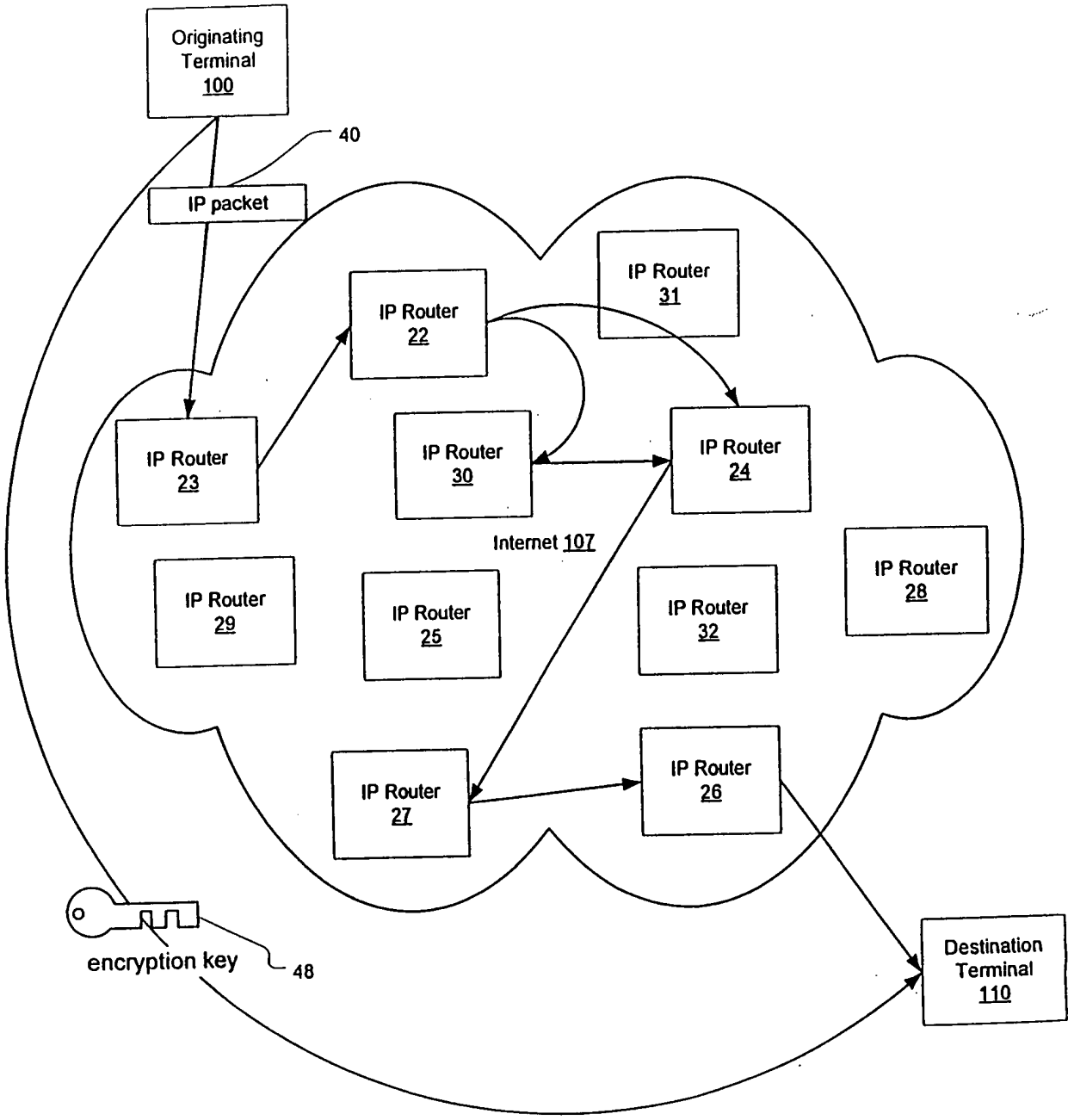


Fig. 1



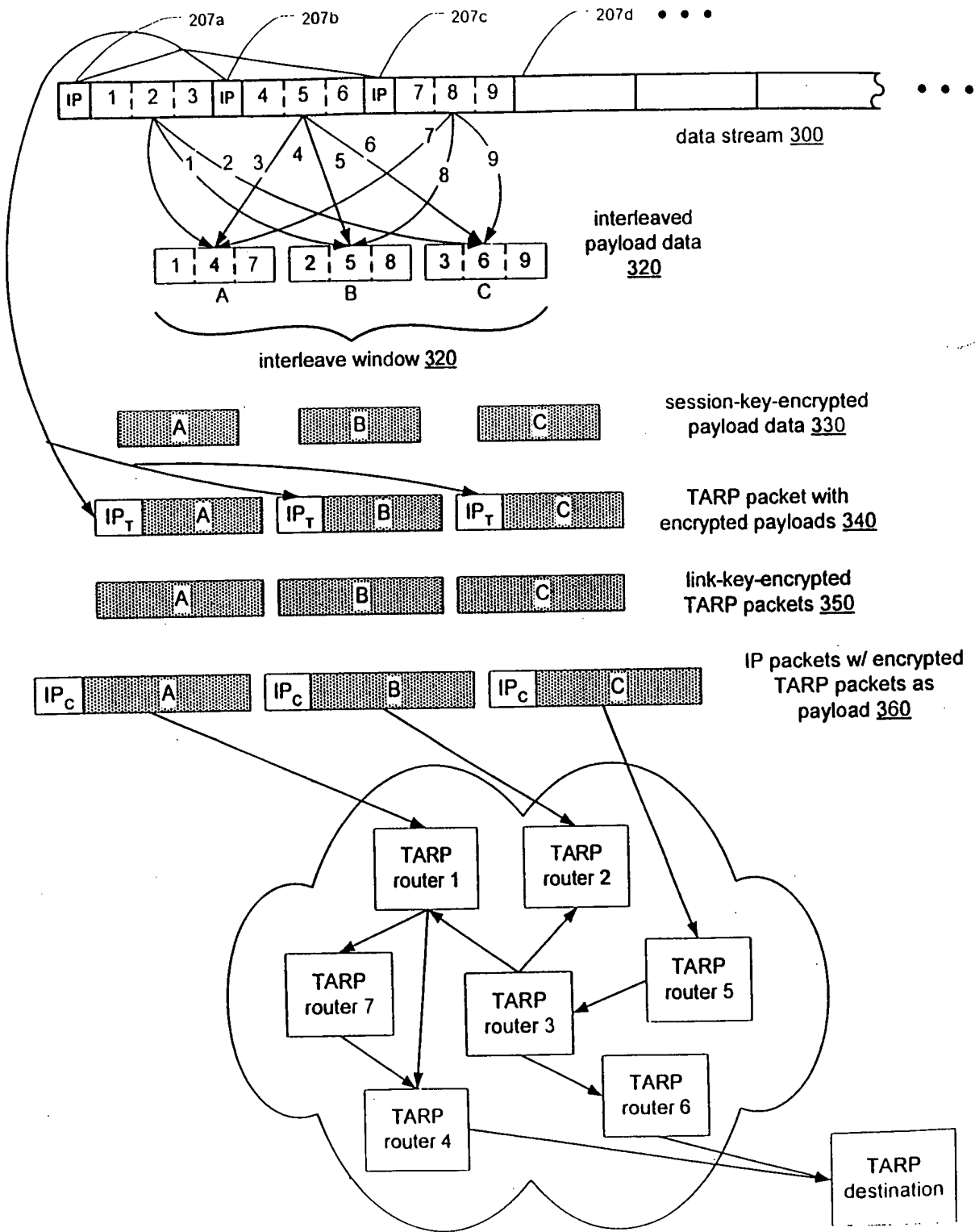


Fig. 3a

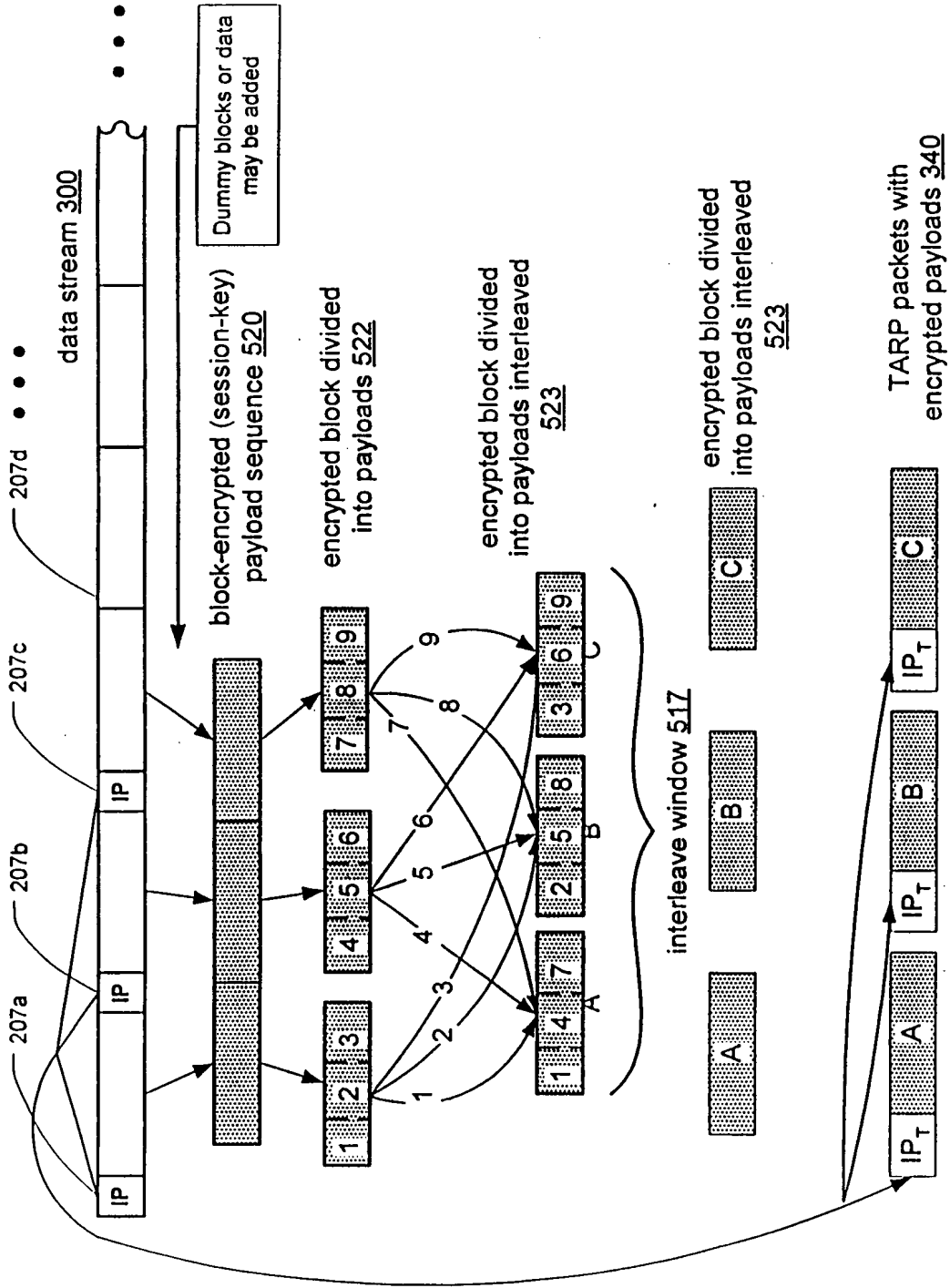


Fig. 3b

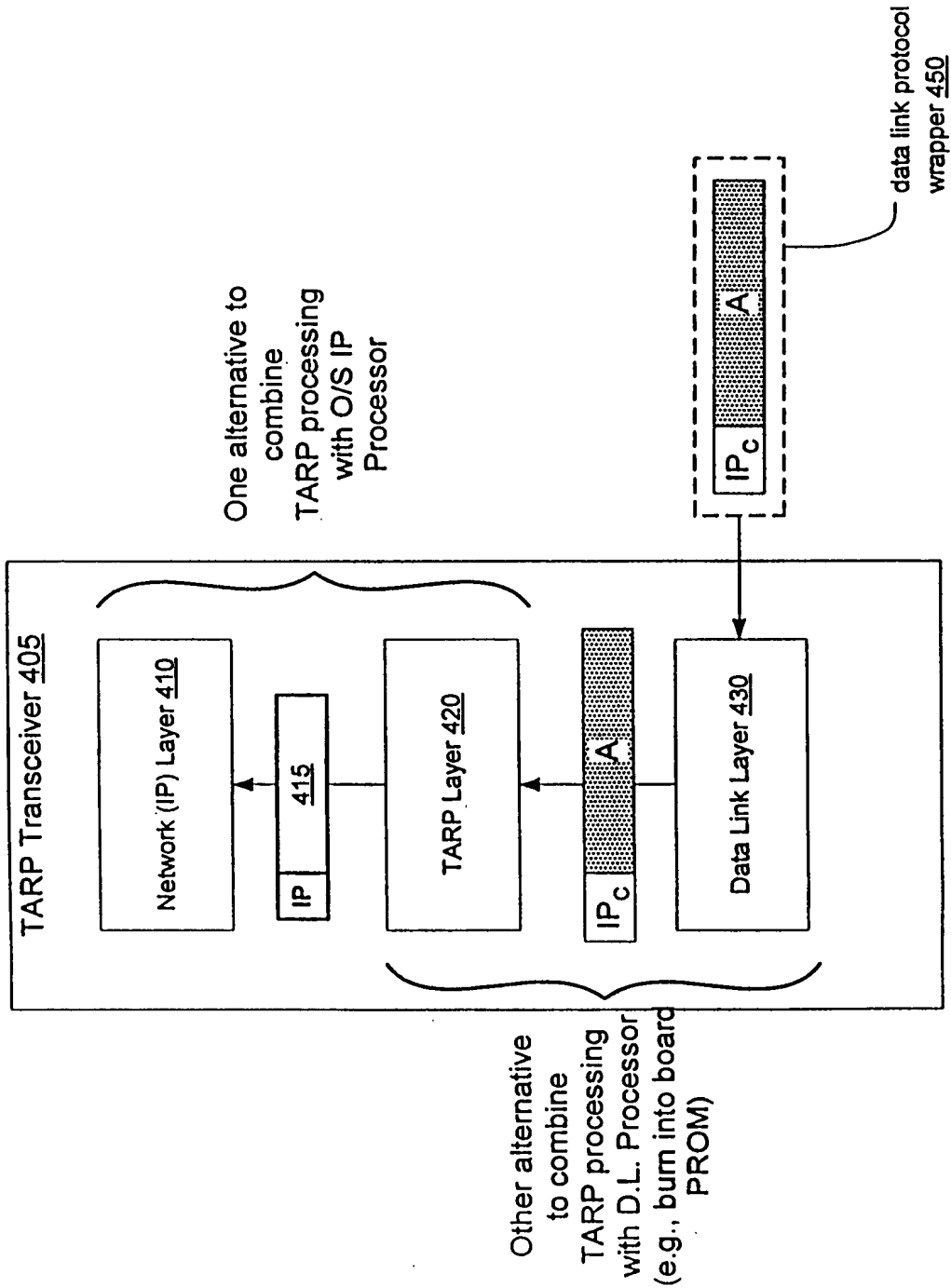


Fig. 4



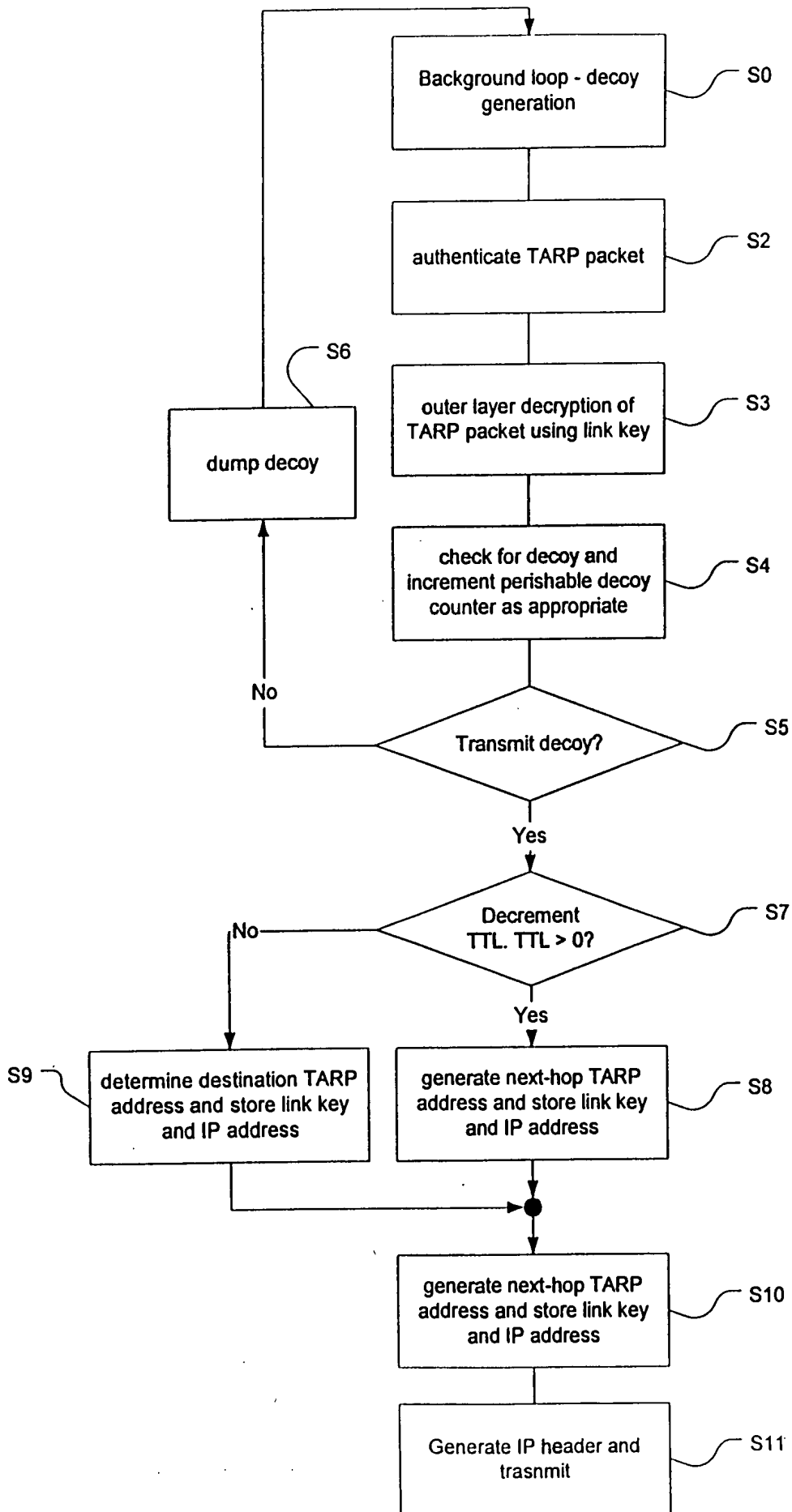
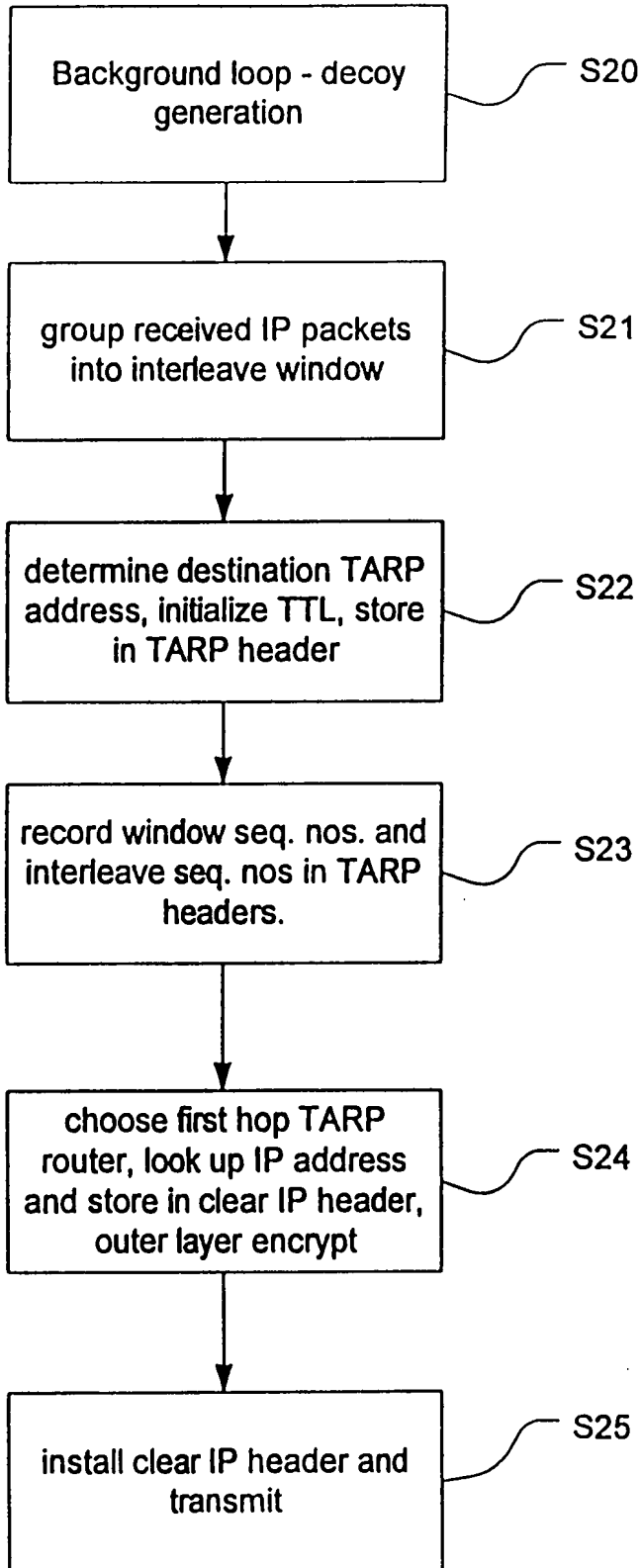


Fig. 5



**Fig. 6**

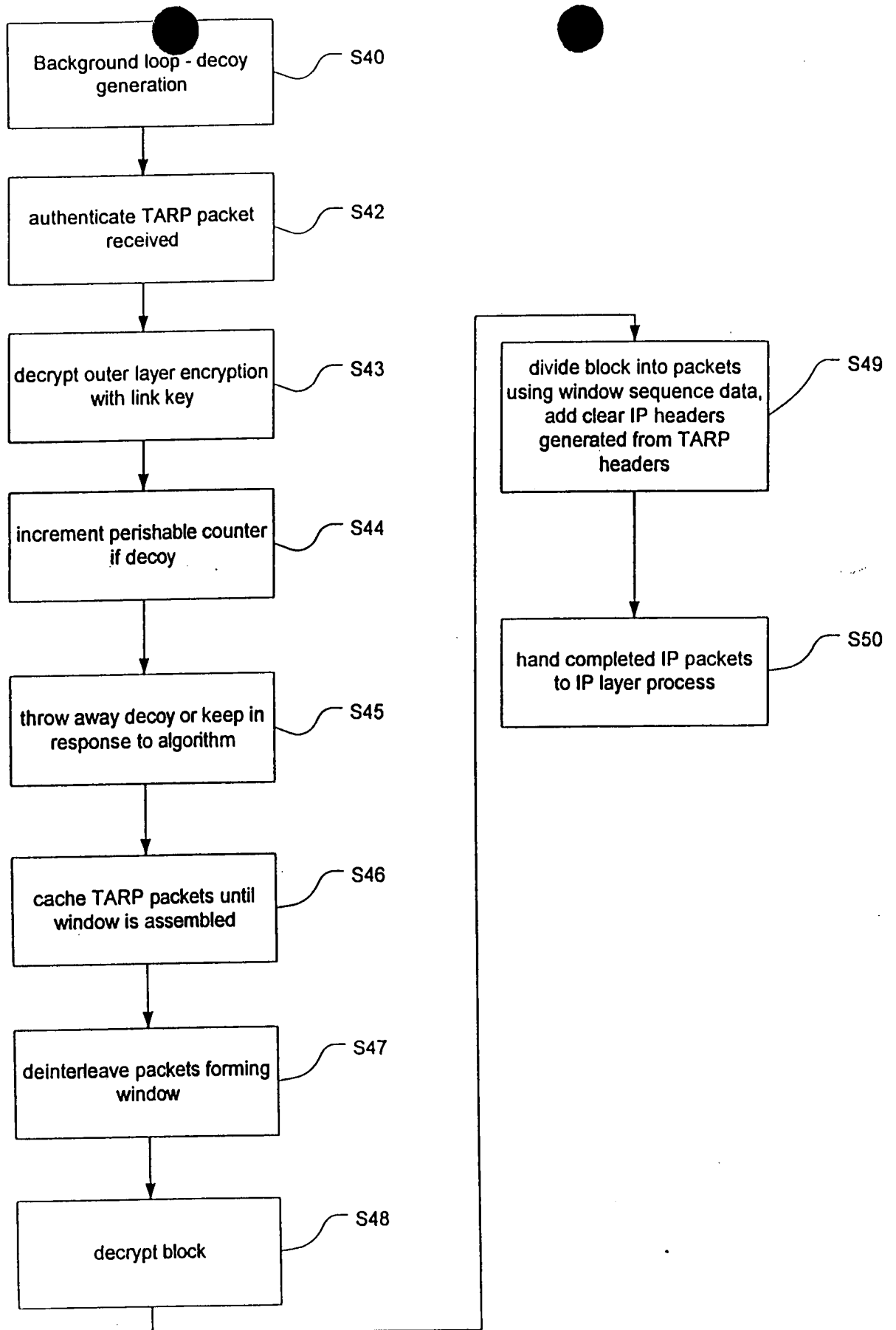
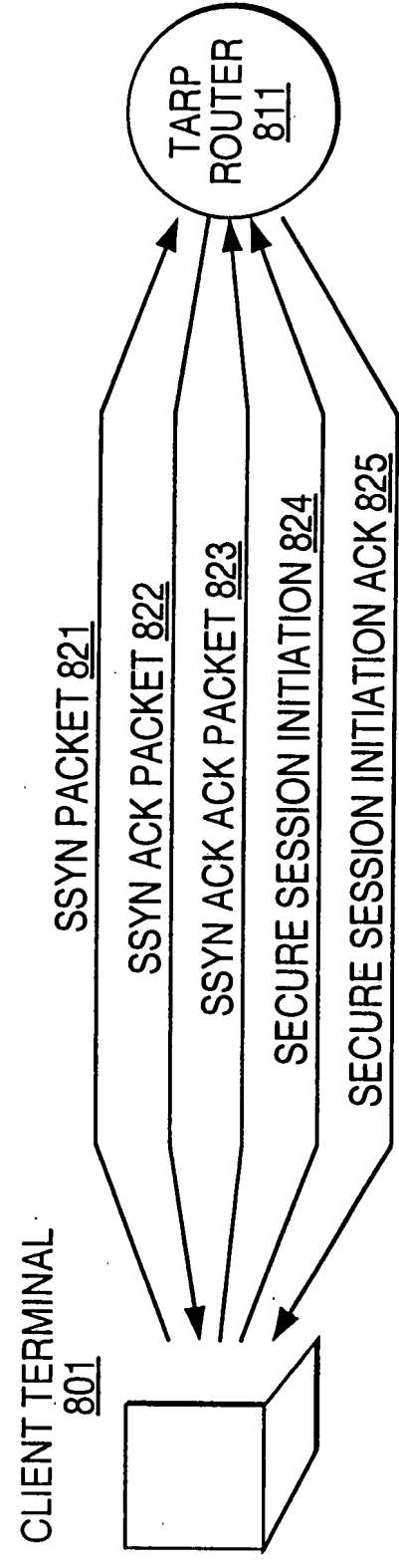


Fig. 7

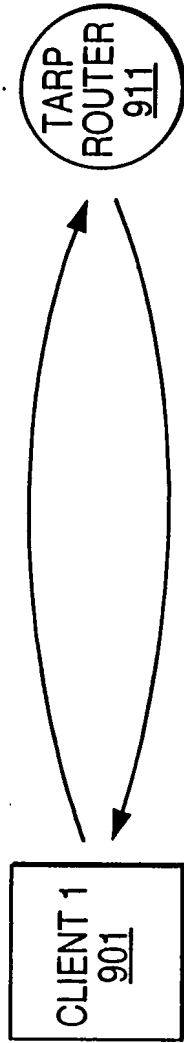
**FIG. 8**

**SECURE SESSION ESTABLISHMENT  
AND SYNCHRONIZATION**



**FIG. 9**

IHOP TRANSMIT AND RECEIVE TABLES



TRANSMIT TABLE 921

131.218.204.98 , 131.218.204.65  
 131.218.204.221 , 131.218.204.97  
 131.218.204.139 , 131.218.204.186  
 131.218.204.12 , 131.218.204.55

RECEIVE TABLE 922

131.218.204.161 , 131.218.204.89  
 131.218.204.66 , 131.218.204.212  
 131.218.204.201 , 131.218.204.127  
 131.218.204.119 , 131.218.204.49

RECEIVE TABLE 924

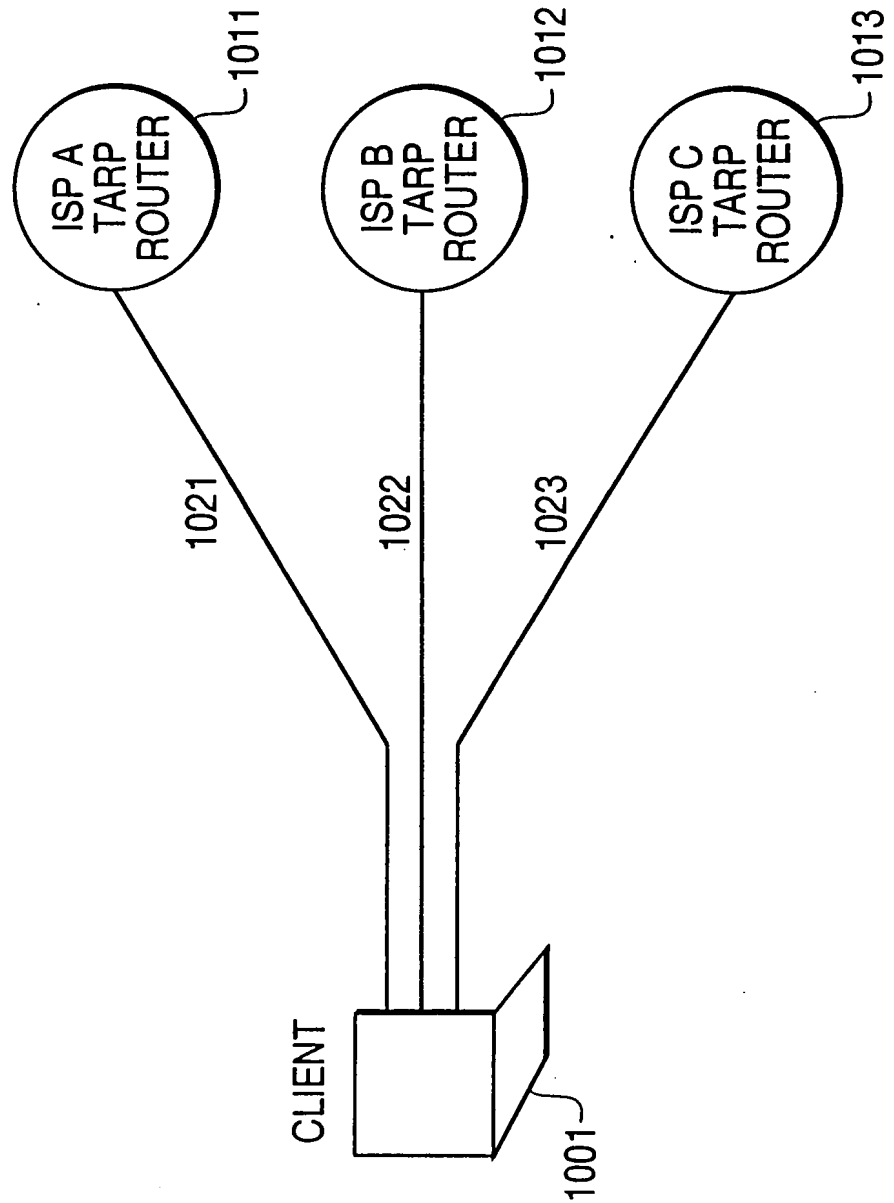
131.218.204.98 , 131.218.204.65  
 131.218.204.221 , 131.218.204.97  
 131.218.204.139 , 131.218.204.186  
 131.218.204.12 , 131.218.204.55

TRANSMIT TABLE 923

131.218.204.161 , 131.218.204.89  
 131.218.204.66 , 131.218.204.212  
 131.218.204.201 , 131.218.204.127  
 131.218.204.119 , 131.218.204.49

**FIG. 10**

PHYSICAL LINK REDUNDANCY



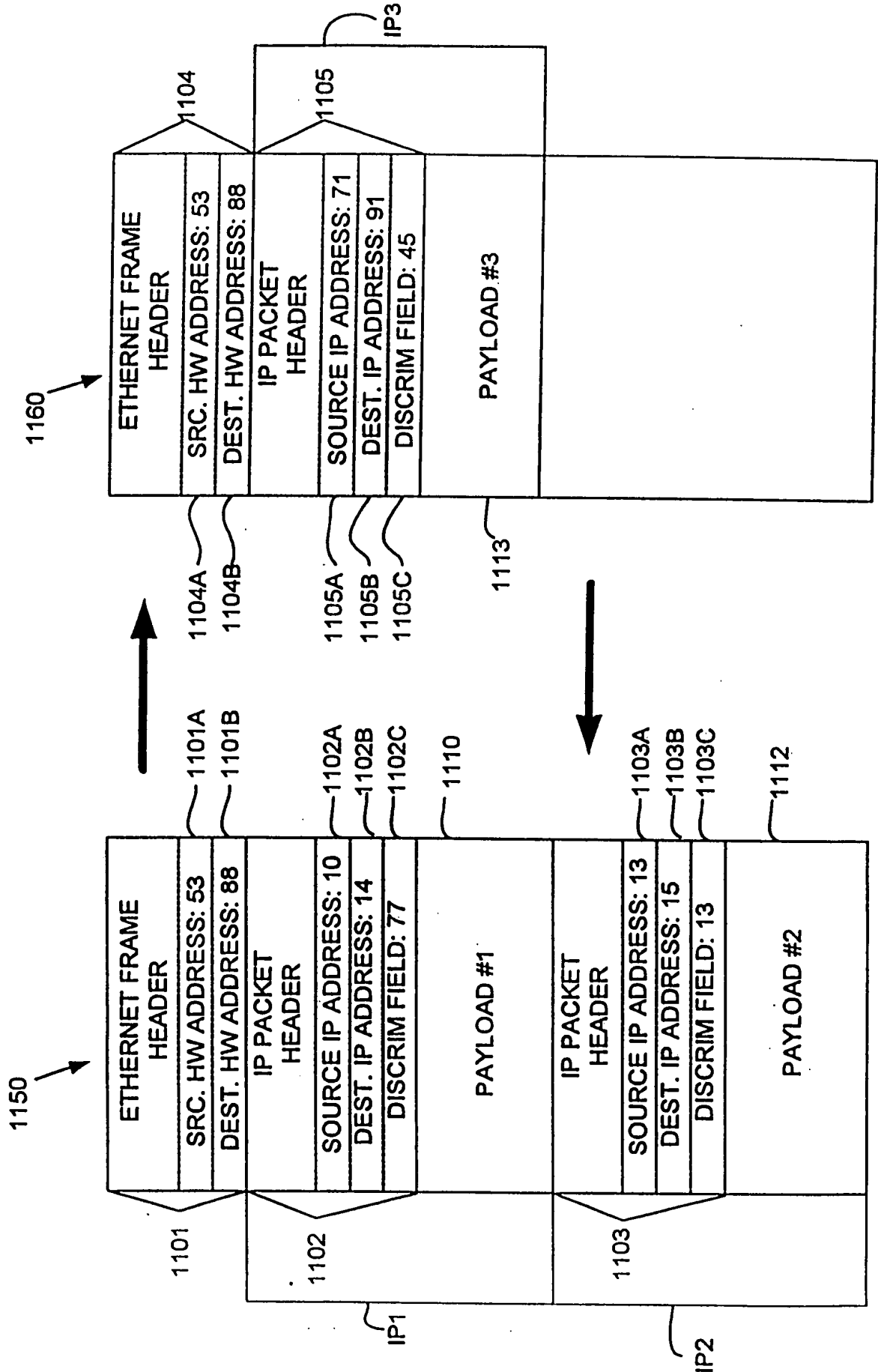


FIG. 11

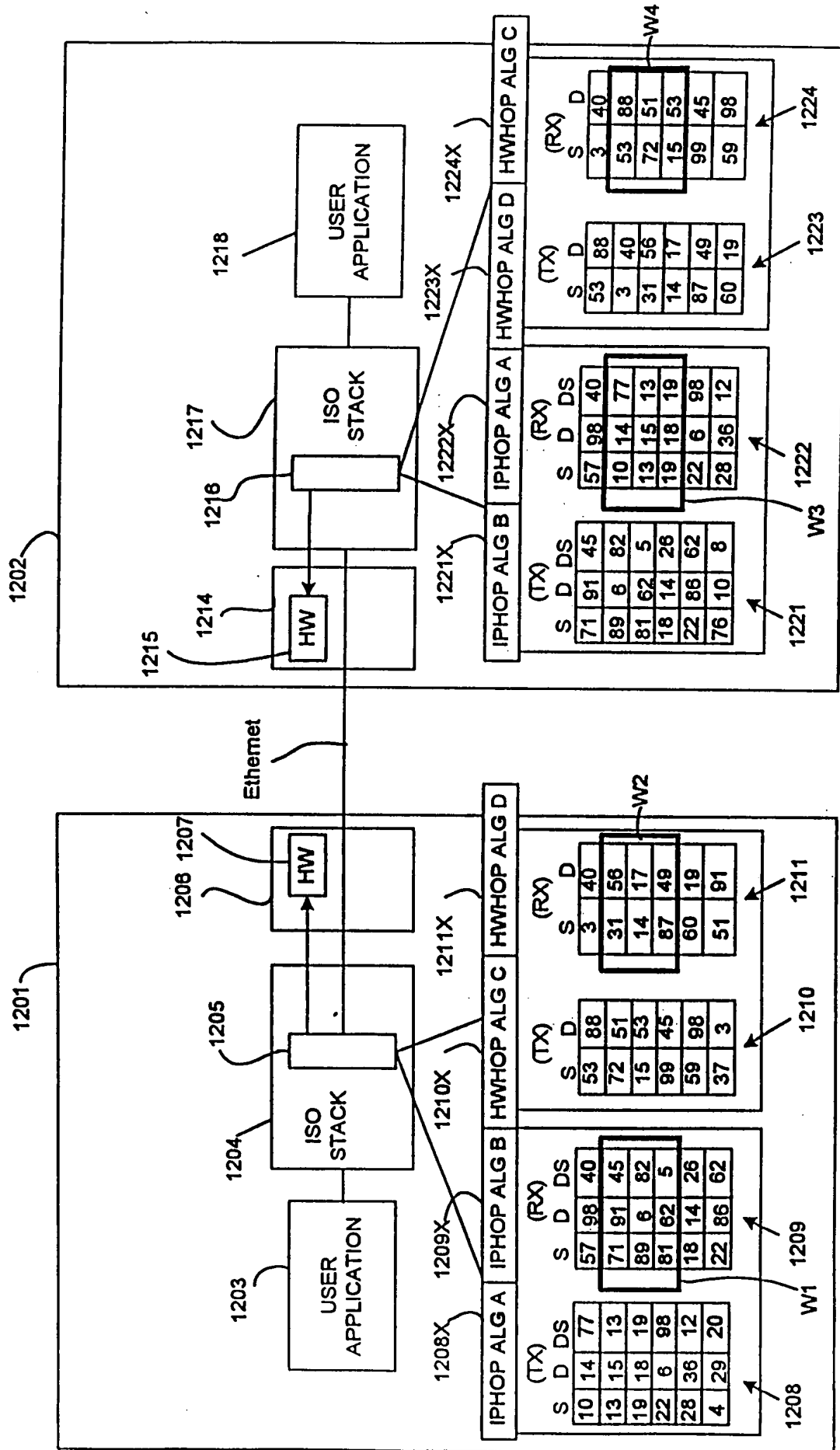


FIG. 12A



MODE OR EMBODIMENT	HARDWARE ADDRESSES	IP ADDRESSES	DISCRIMINATOR FIELD VALUES
1. PROMISCUOUS	SAME FOR ALL NODES OR COMPLETELY RANDOM	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC
2. PROMISCUOUS PER VPN	FIXED FOR EACH VPN	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC
3. HARDWARE HOPPING	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC

FIG. 12B

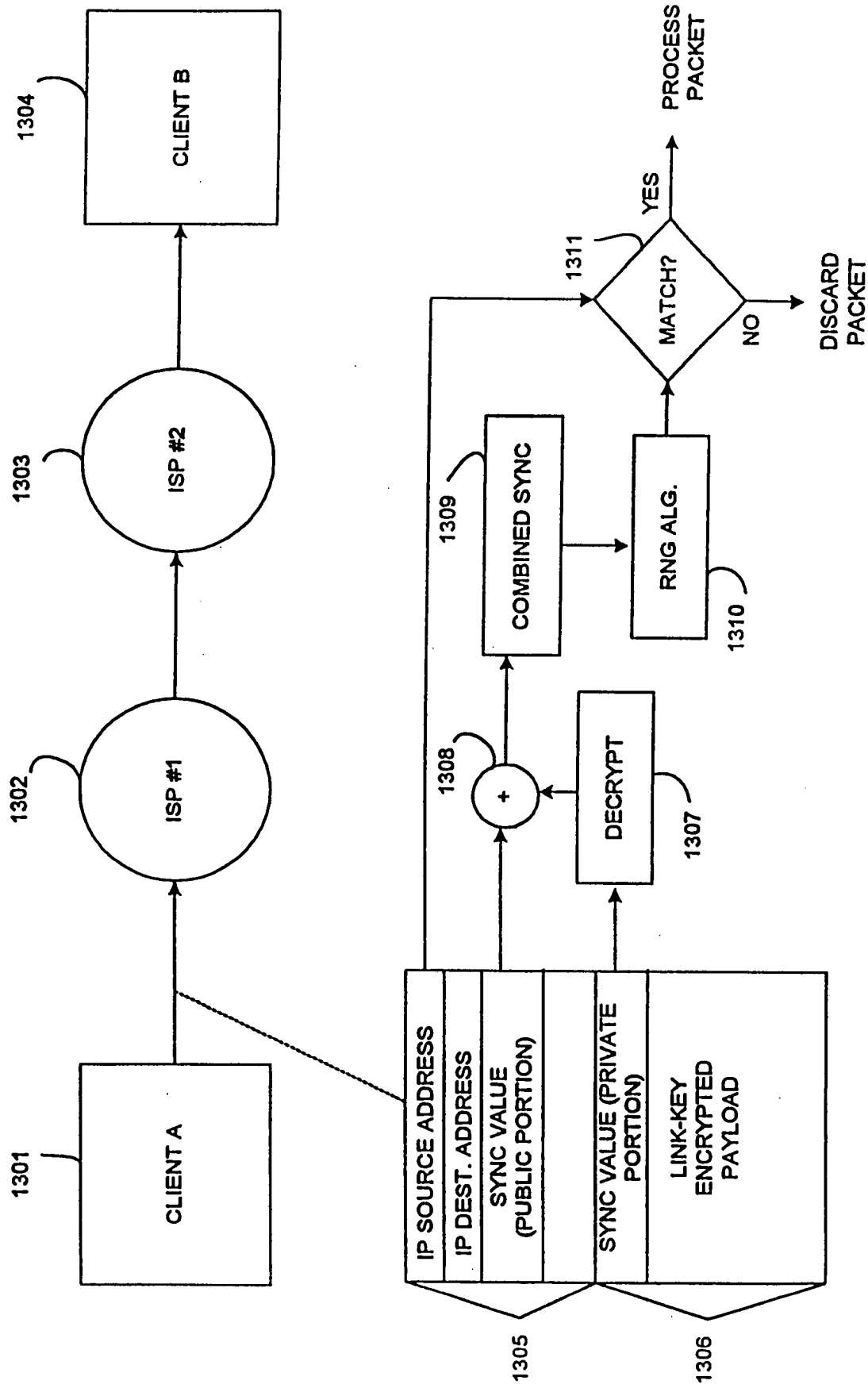


FIG. 13



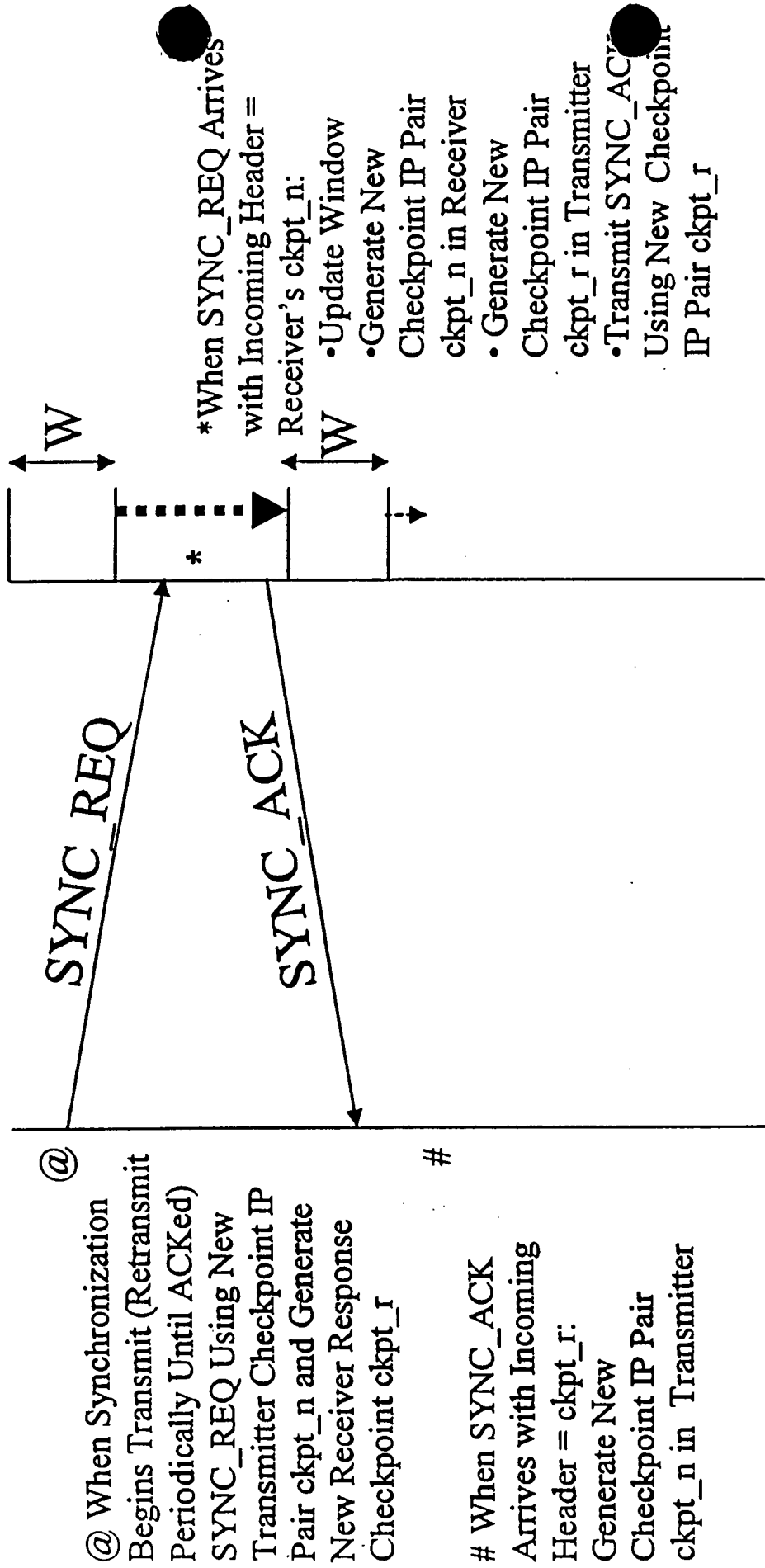


FIG. 15

### (Ethernet Lan - Two A Address Blocks)

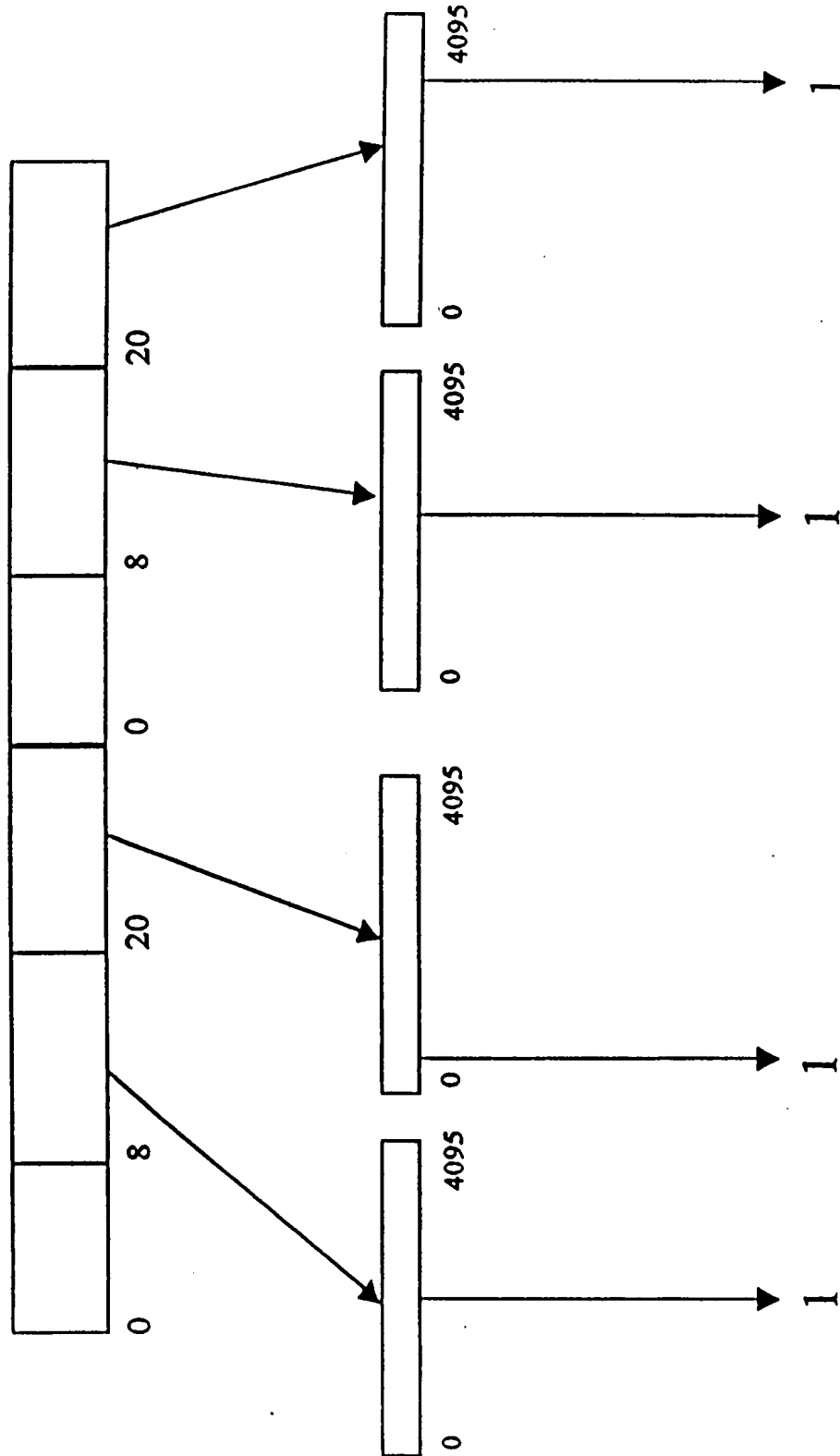


FIG. 16

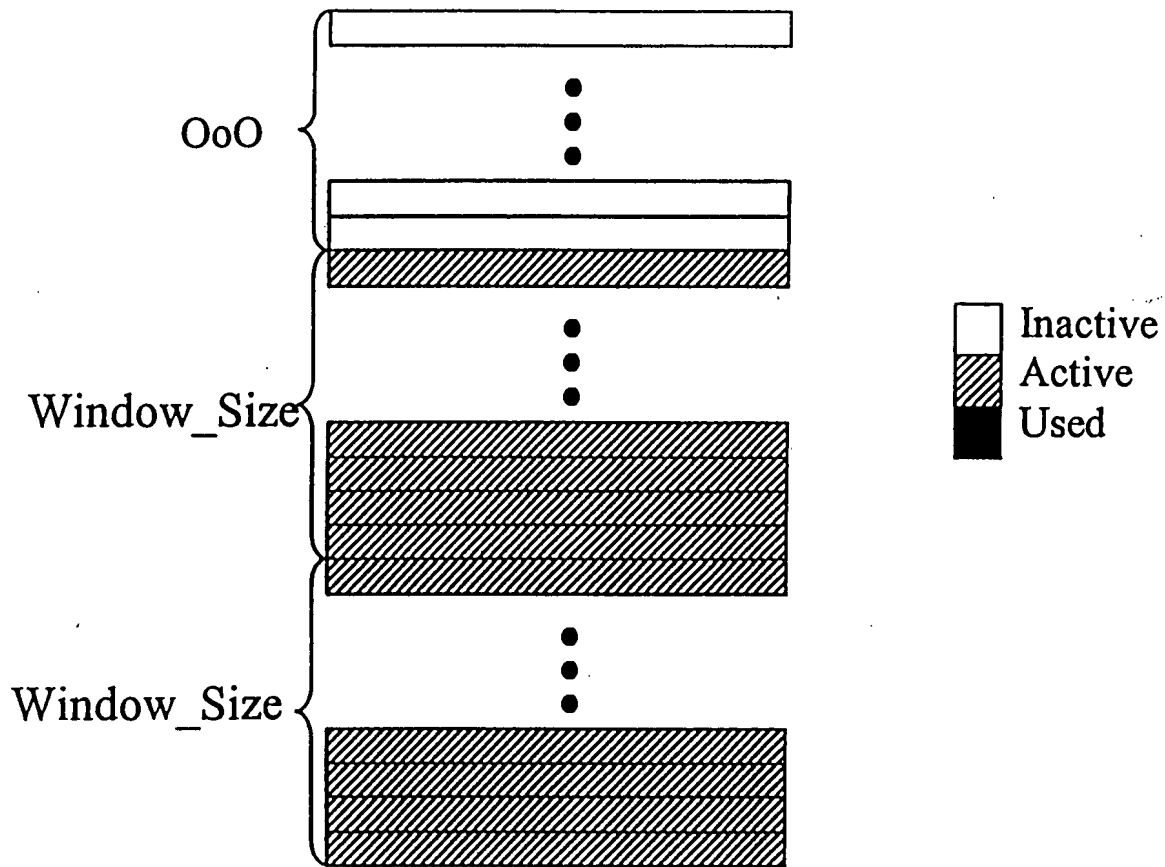


FIG. 17

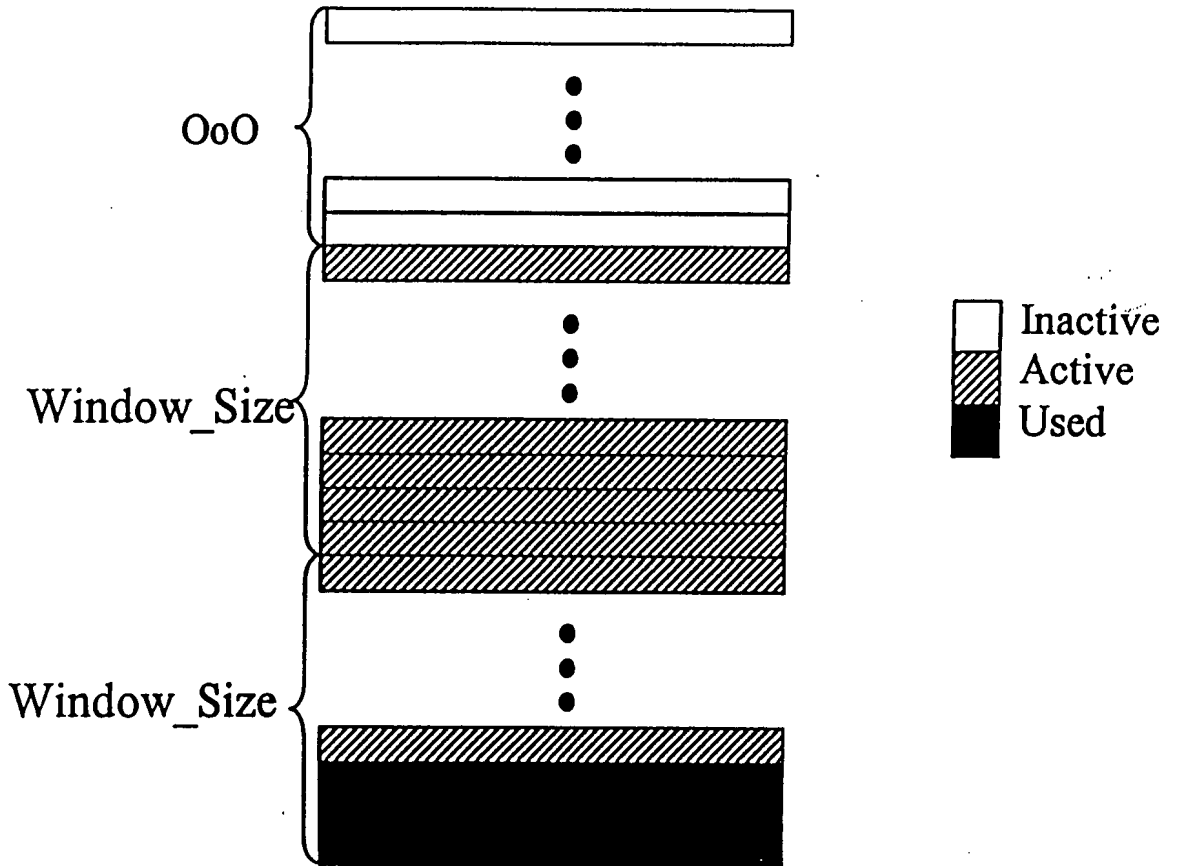


FIG. 18

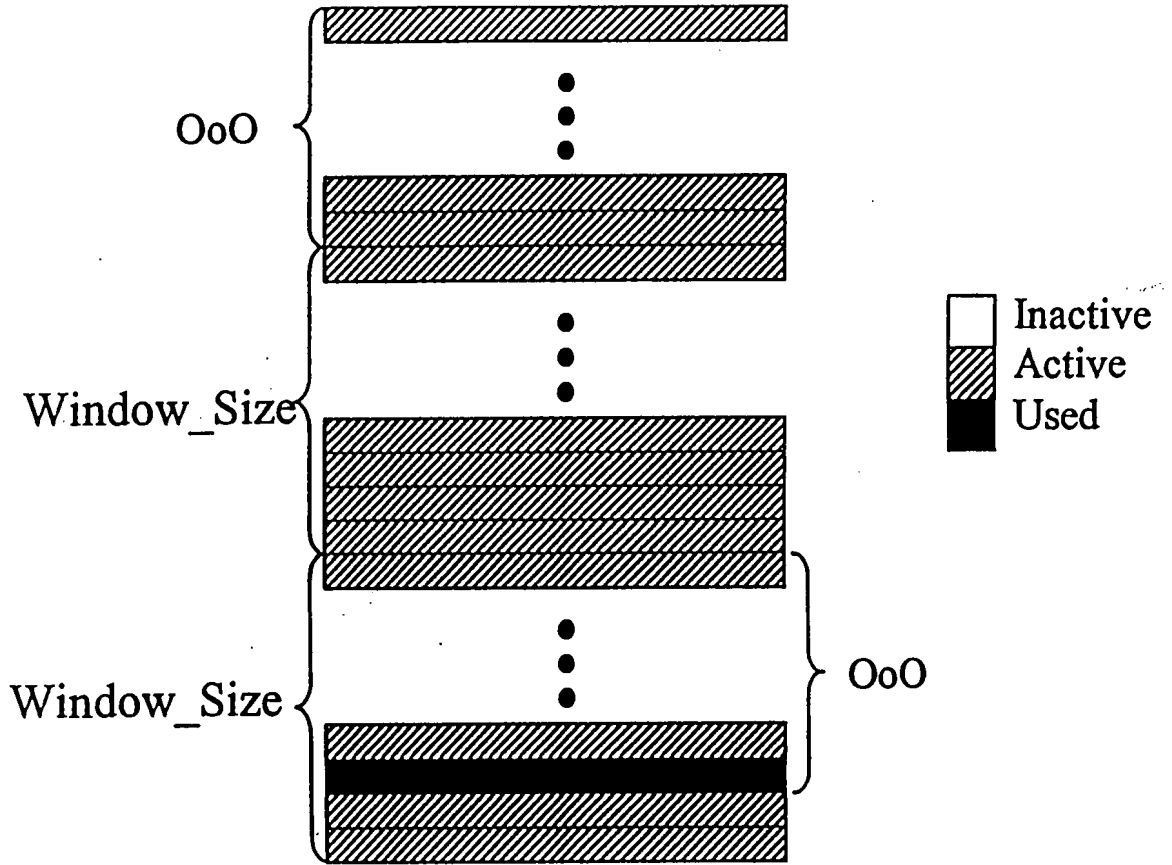


FIG. 19



FIG. 20

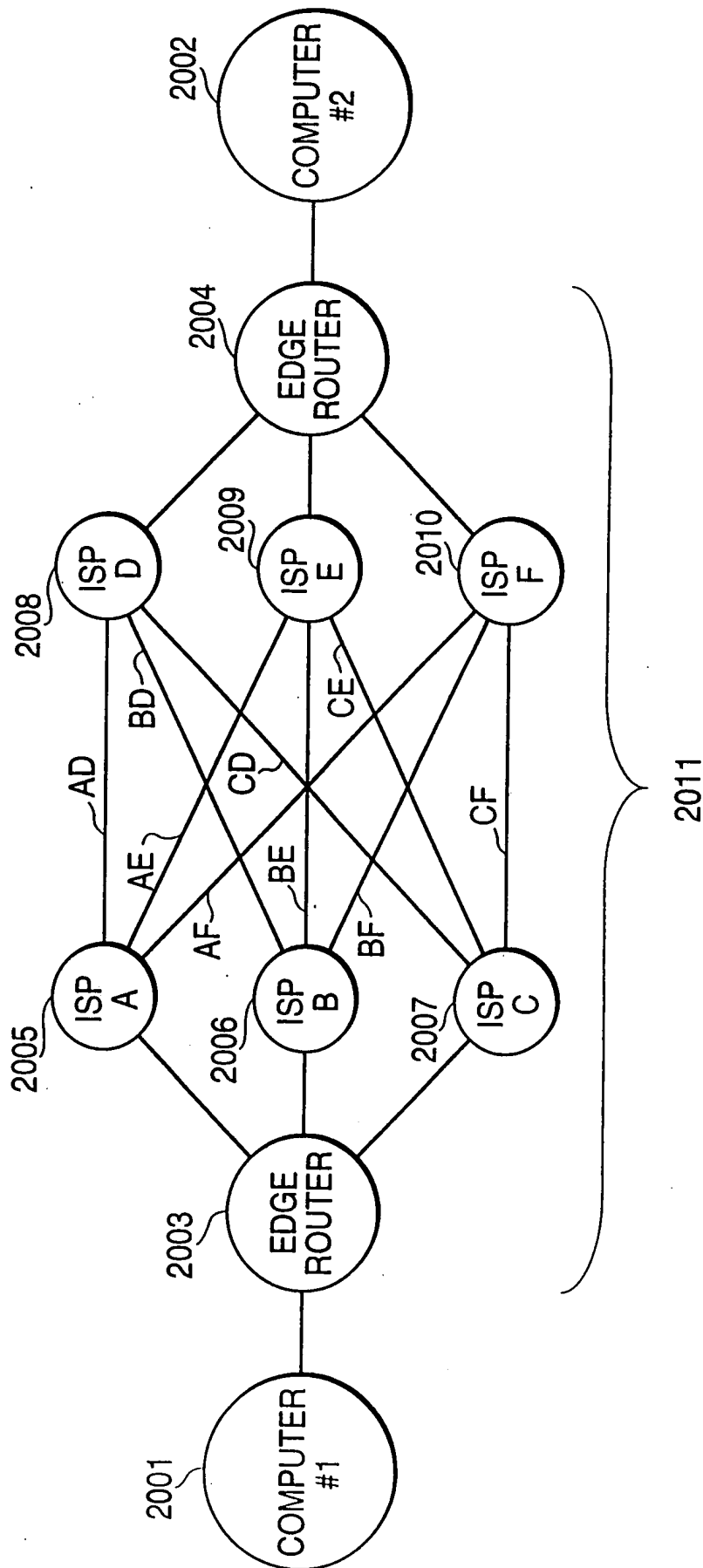
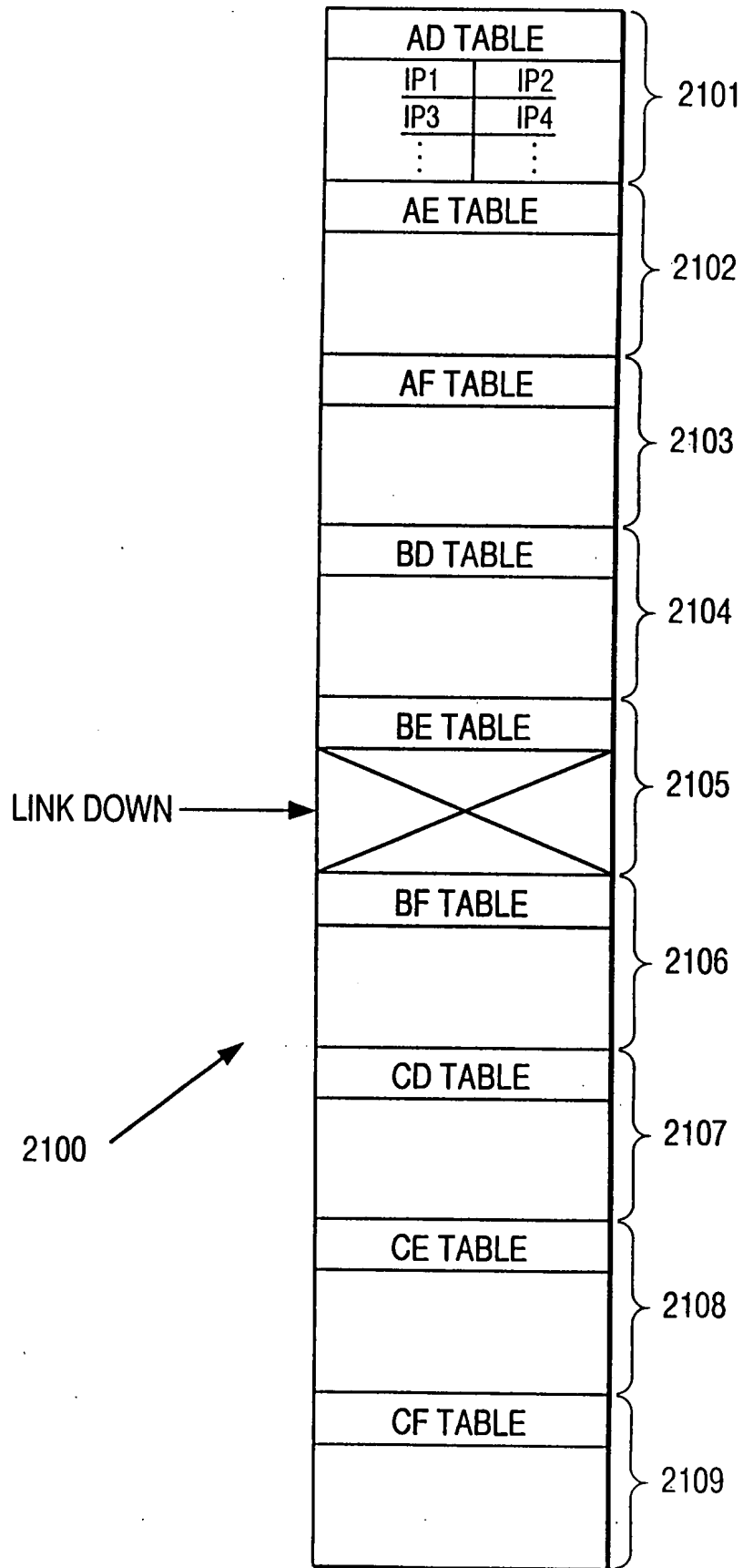


FIG. 21



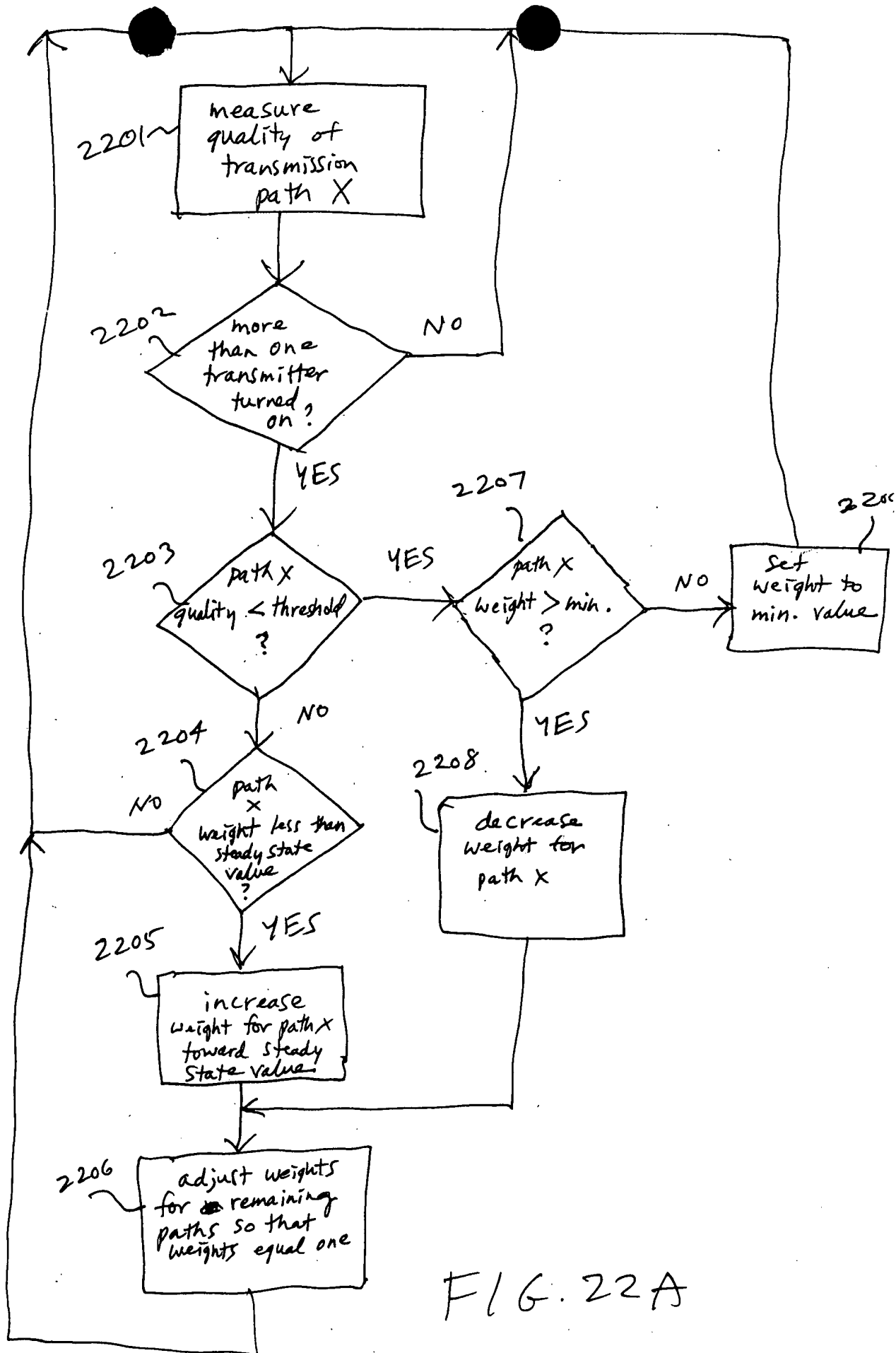


FIG. 22A

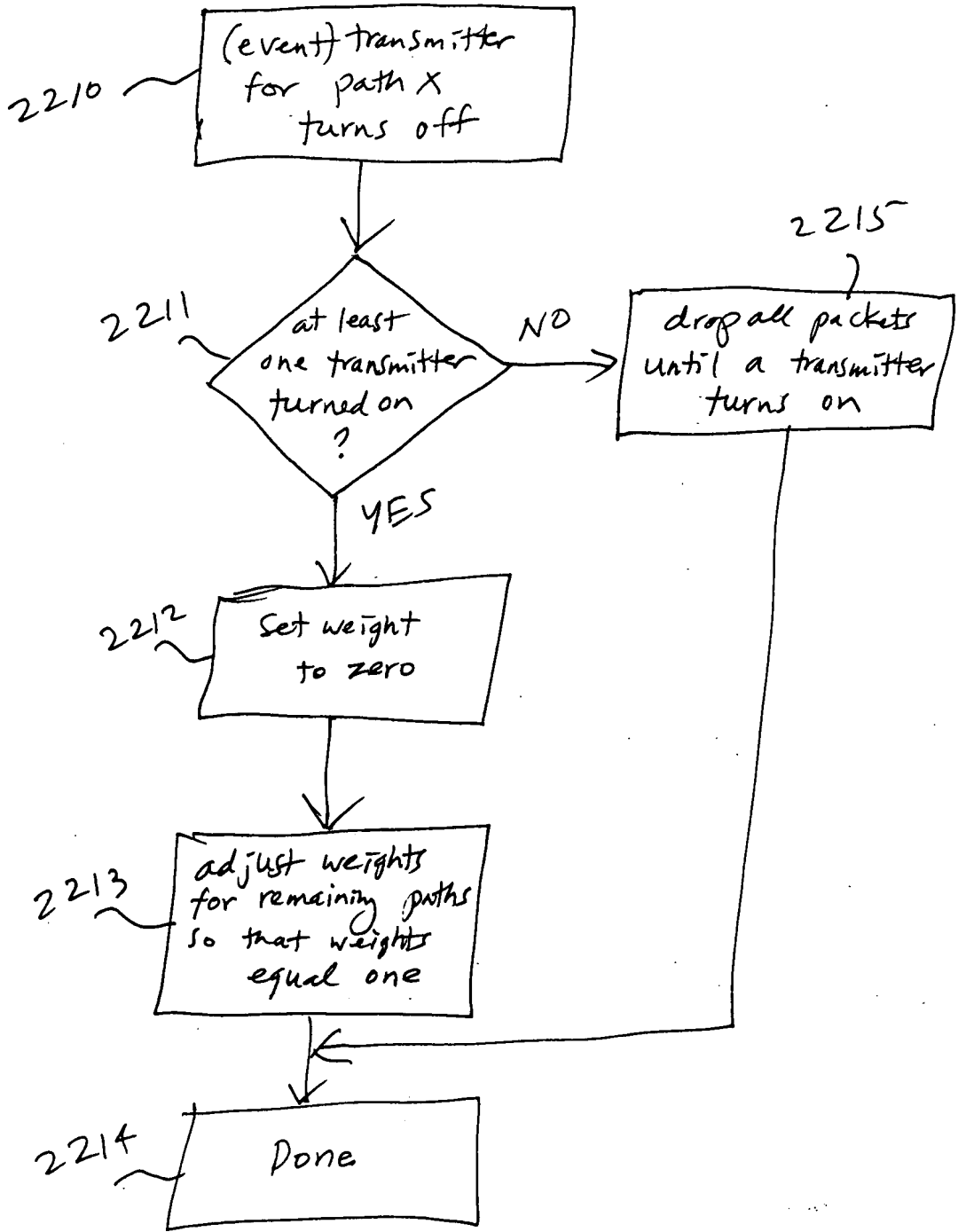
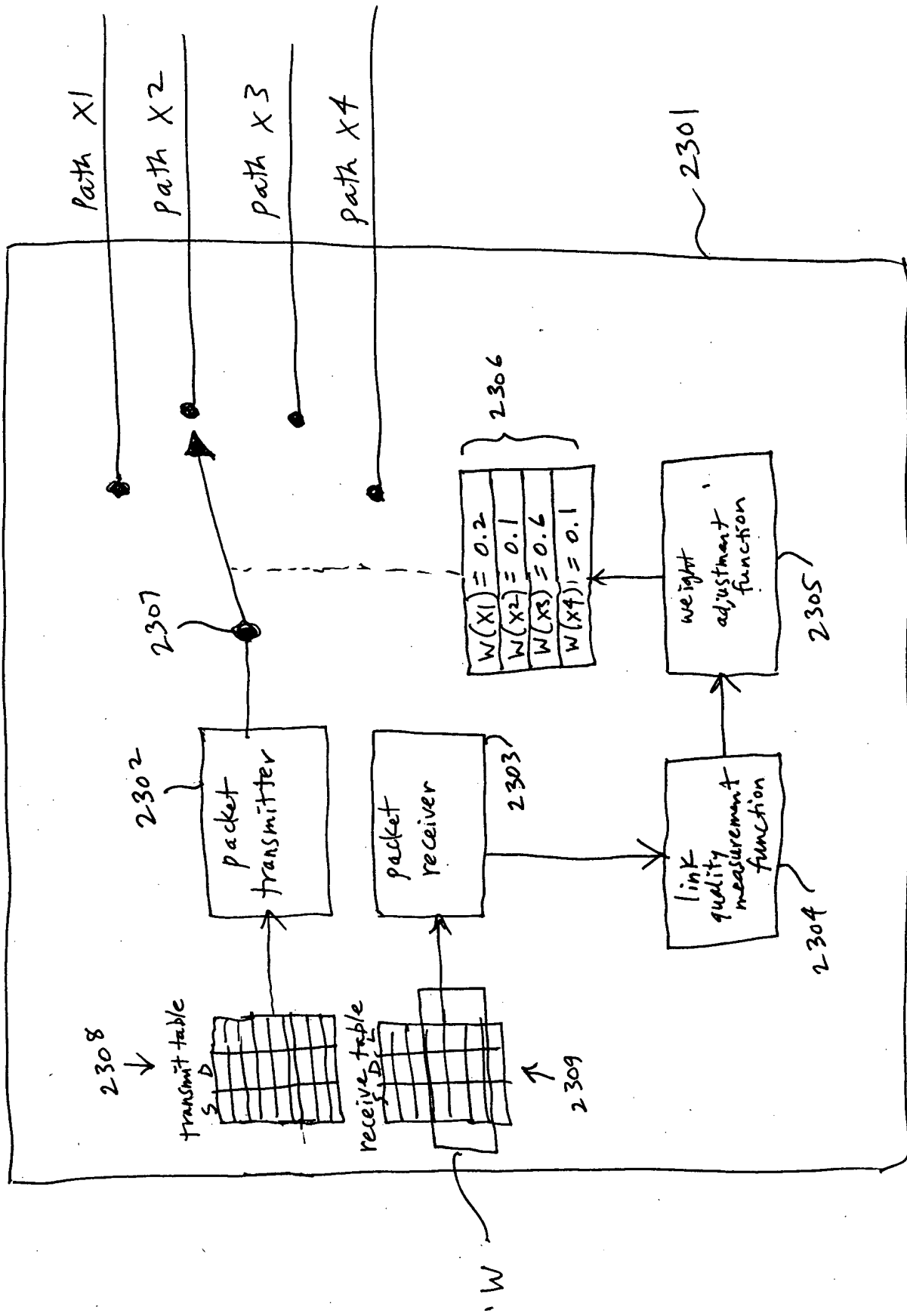


FIG. 22B

Small vertical text on the left margin, possibly a reference or page number.



F/6.23



FIG. 25 is a block diagram of a system for intercepting and redirecting network traffic. The system includes a DNS server (2502) and a target web site (2503). A client device (2501) contains a web browser (2504) and an IP stack (2505). The web browser (2504) sends a DNS request (DNS REQ) to the DNS server (2502). The DNS server (2502) returns a DNS response (DNS RESP) to the IP stack (2505). The IP stack (2505) sends a page request (PAGE REQ) to the target web site (2503). The target web site (2503) returns a page response (PAGE RESP) to the IP stack (2505).

2502

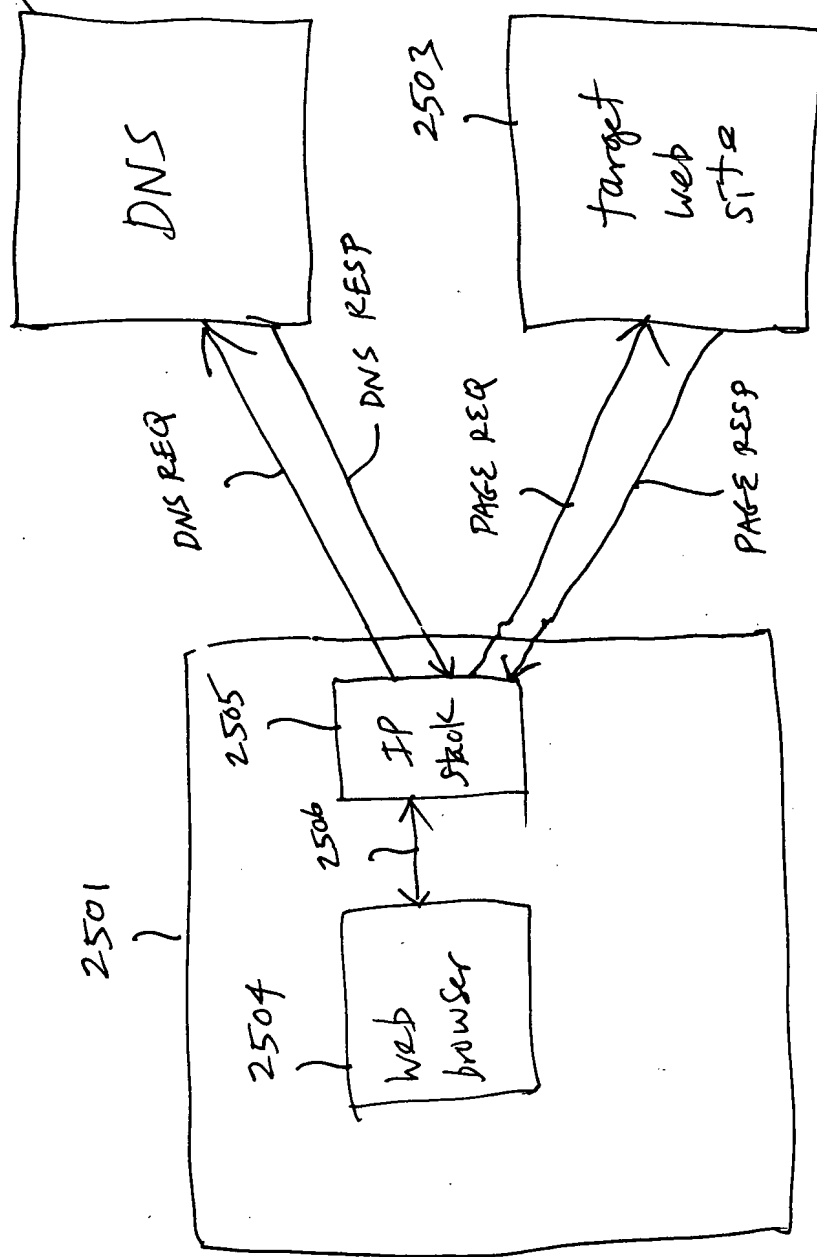


FIG. 25  
(prior art)

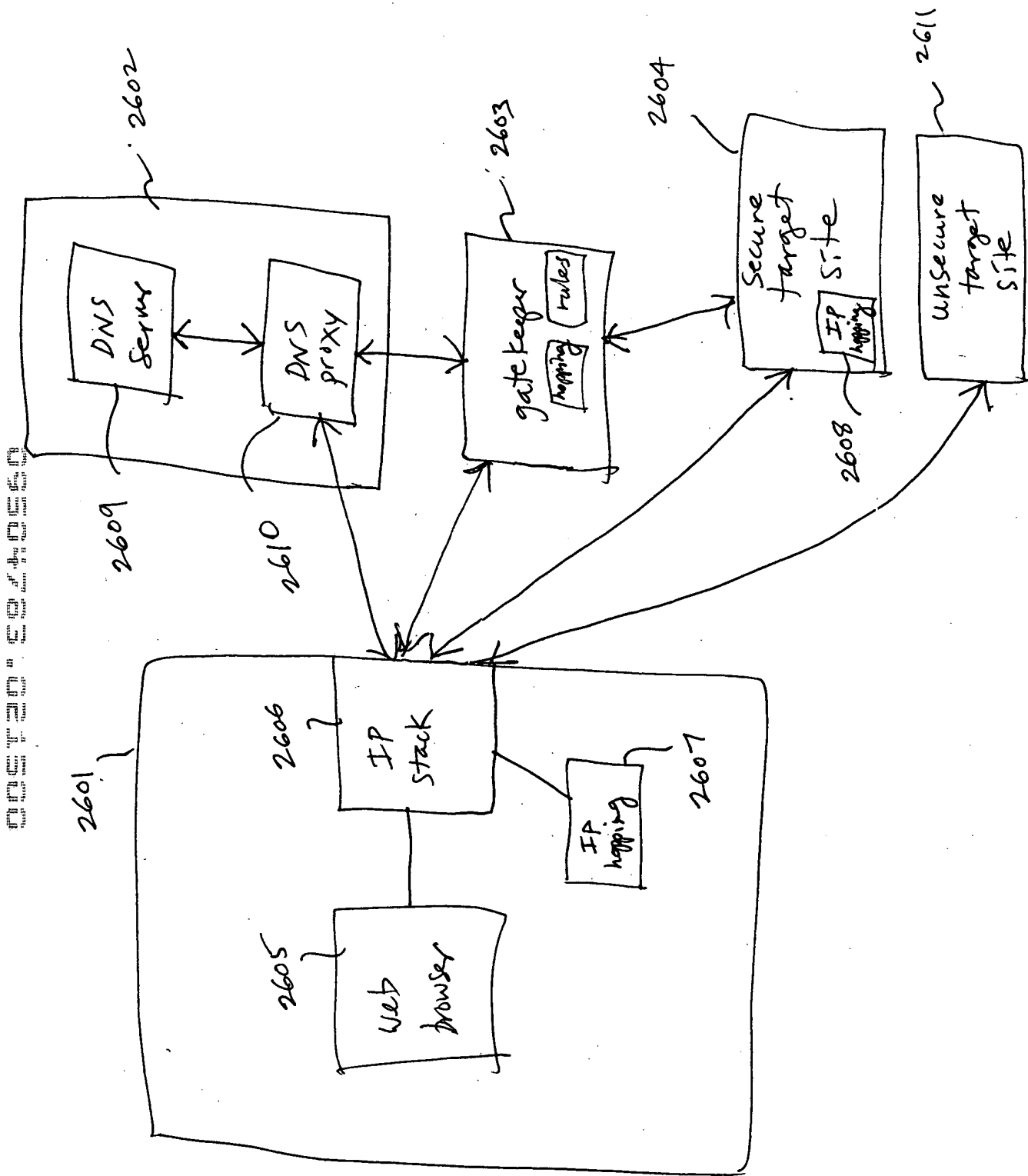


FIG. 26



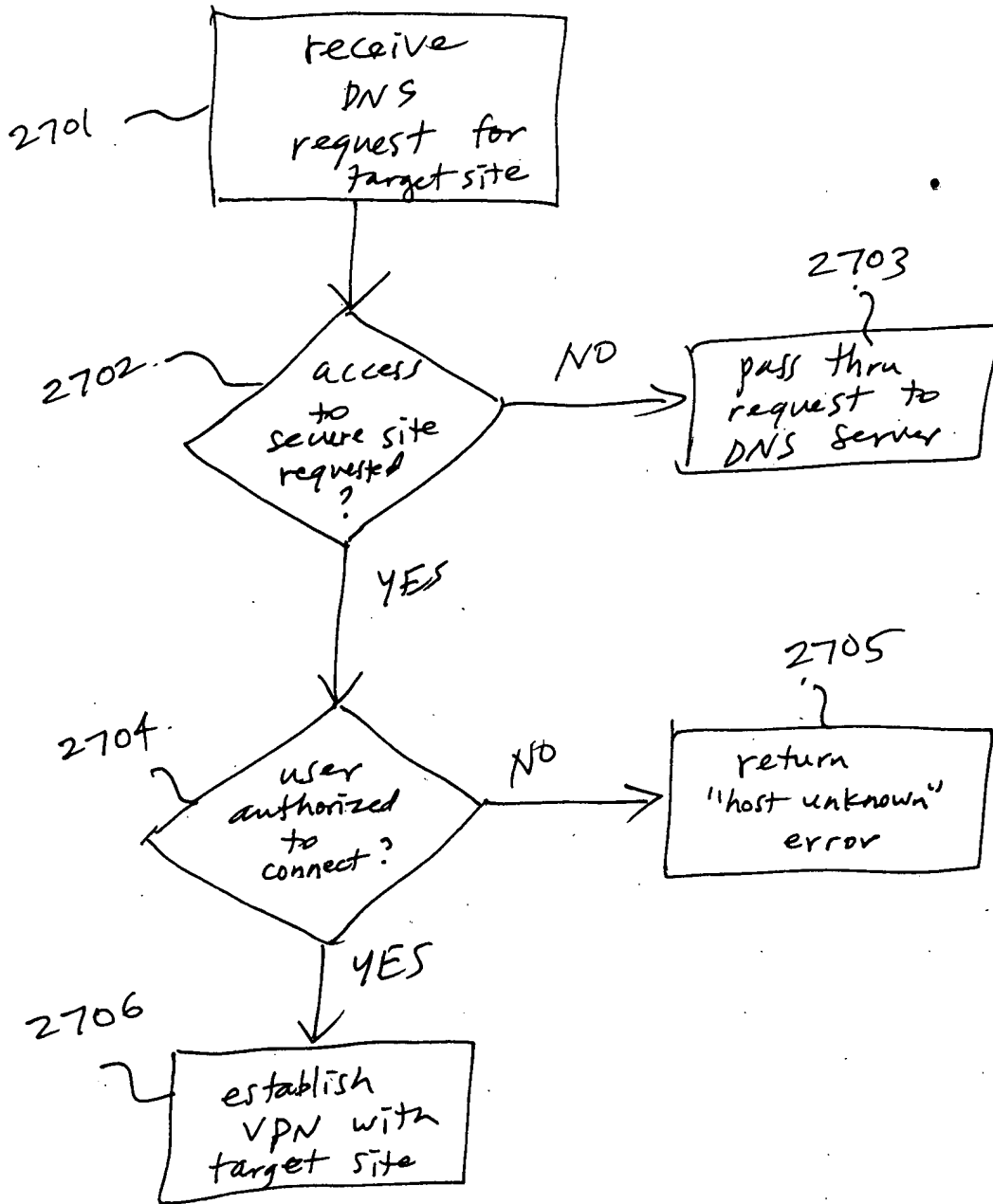


FIG. 27

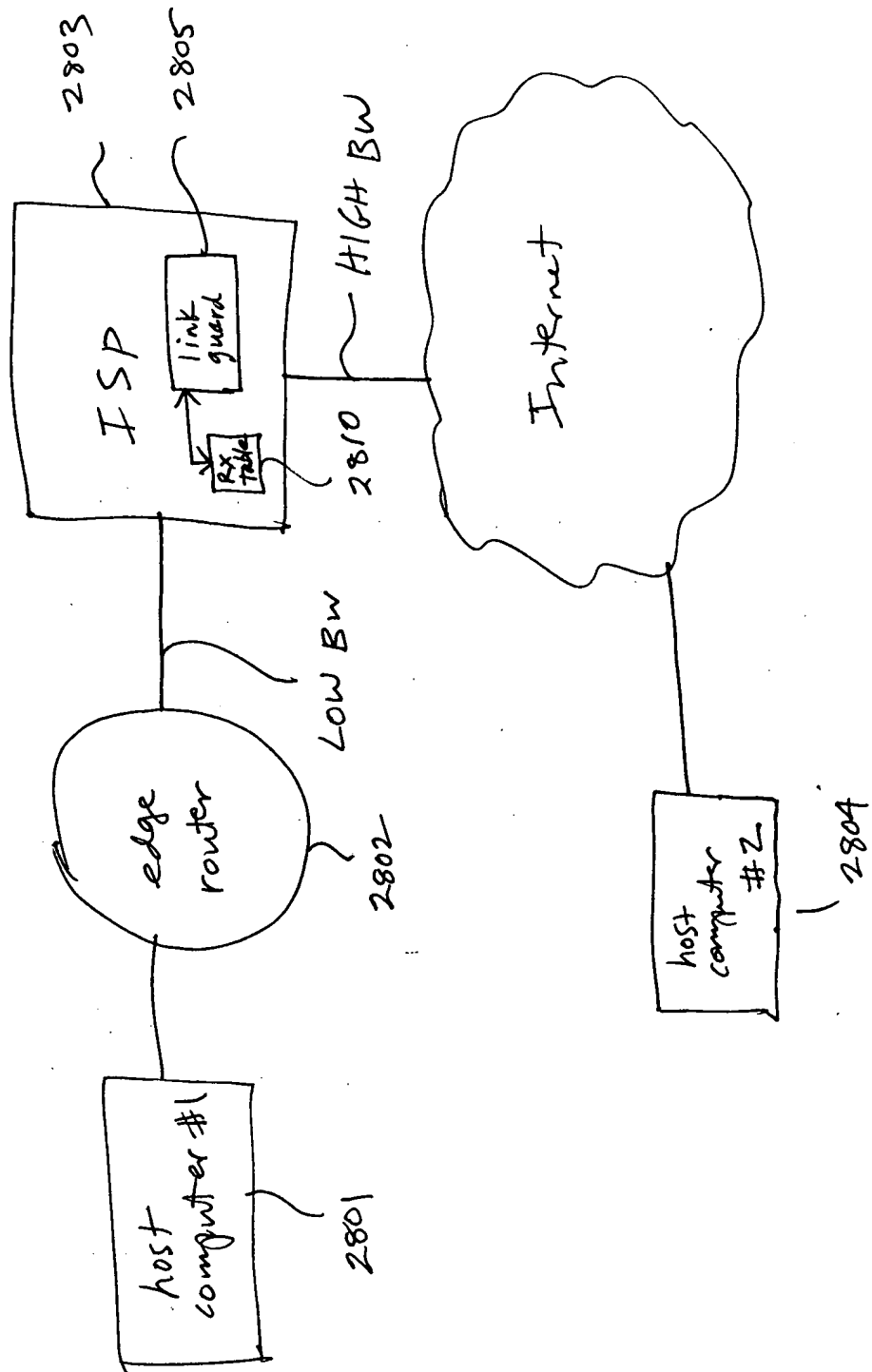
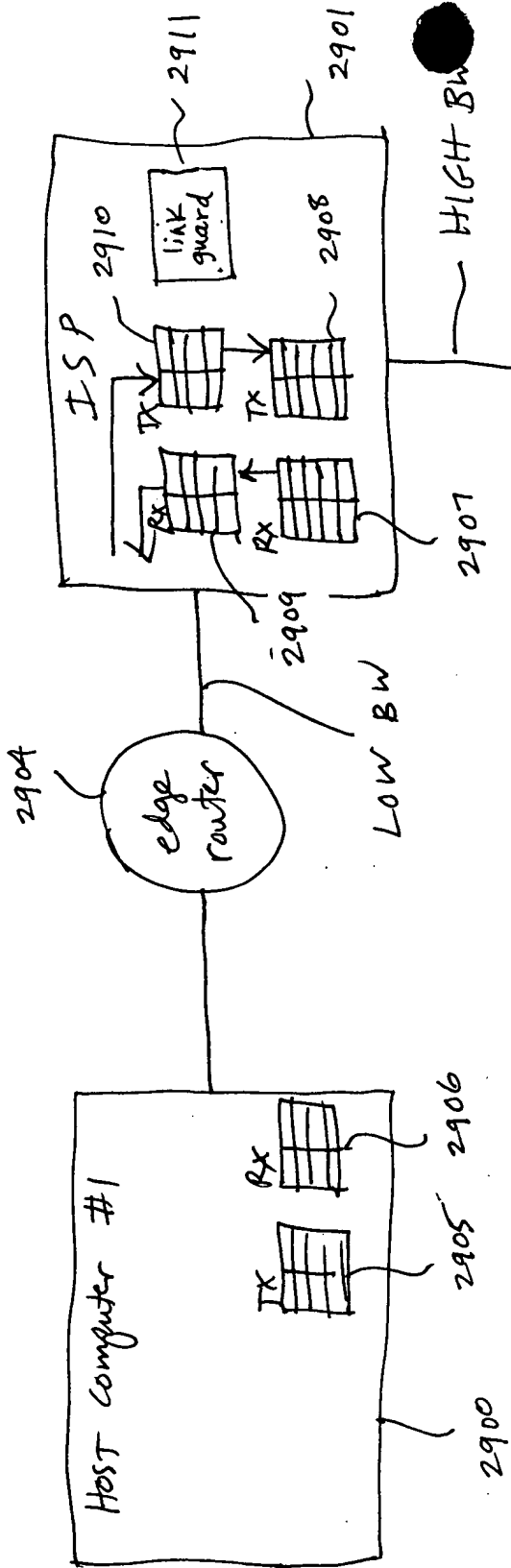
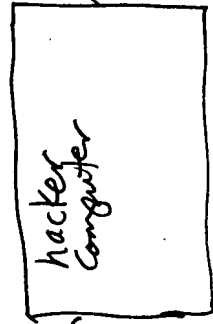
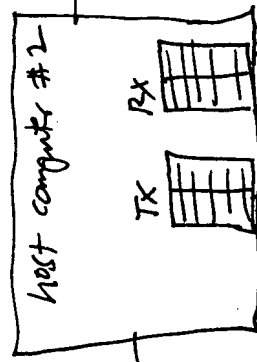


FIG. 28

2904



IP TX / 100-200



FLOOD IP TX / 100-200

FIG. 29

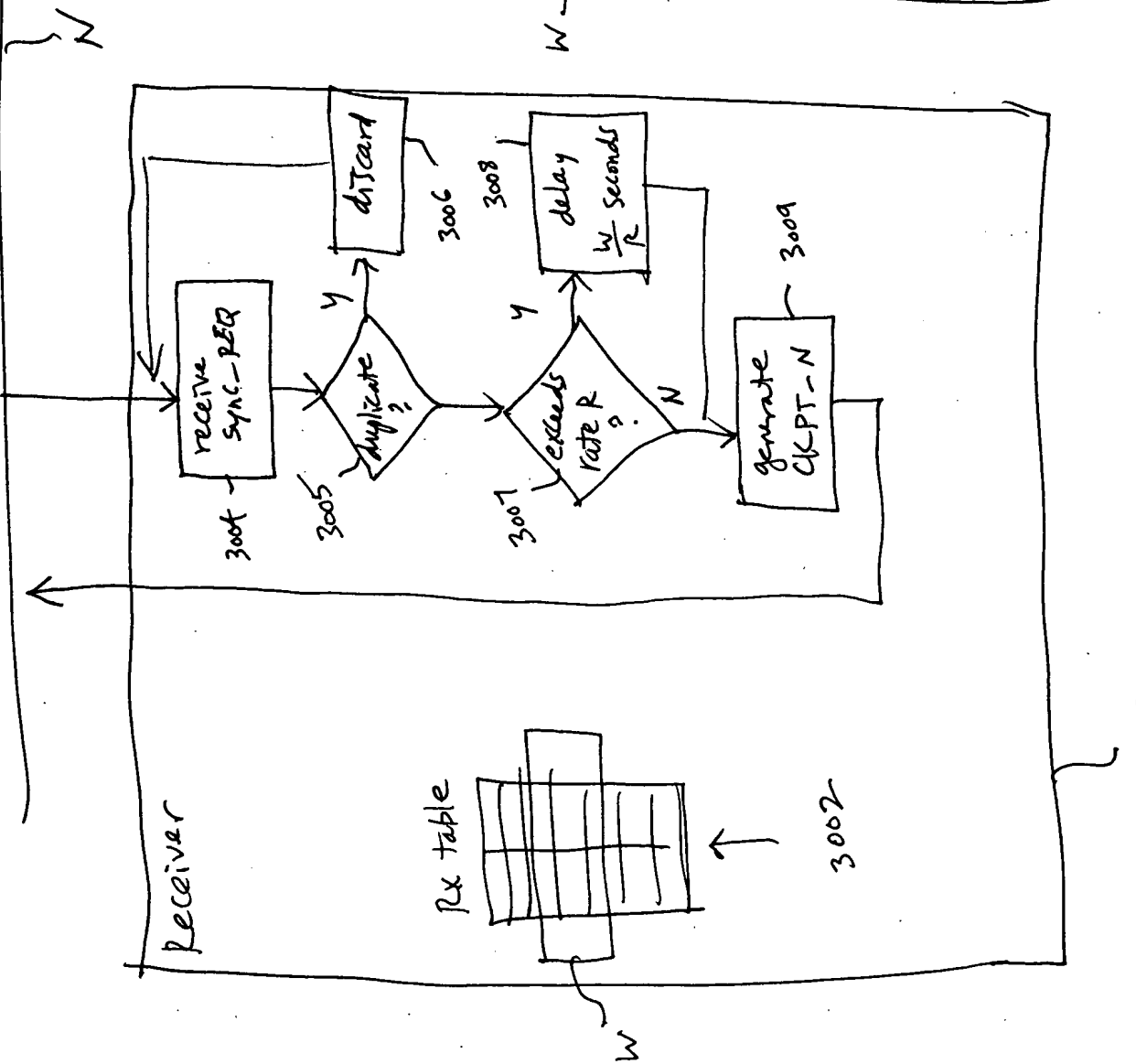
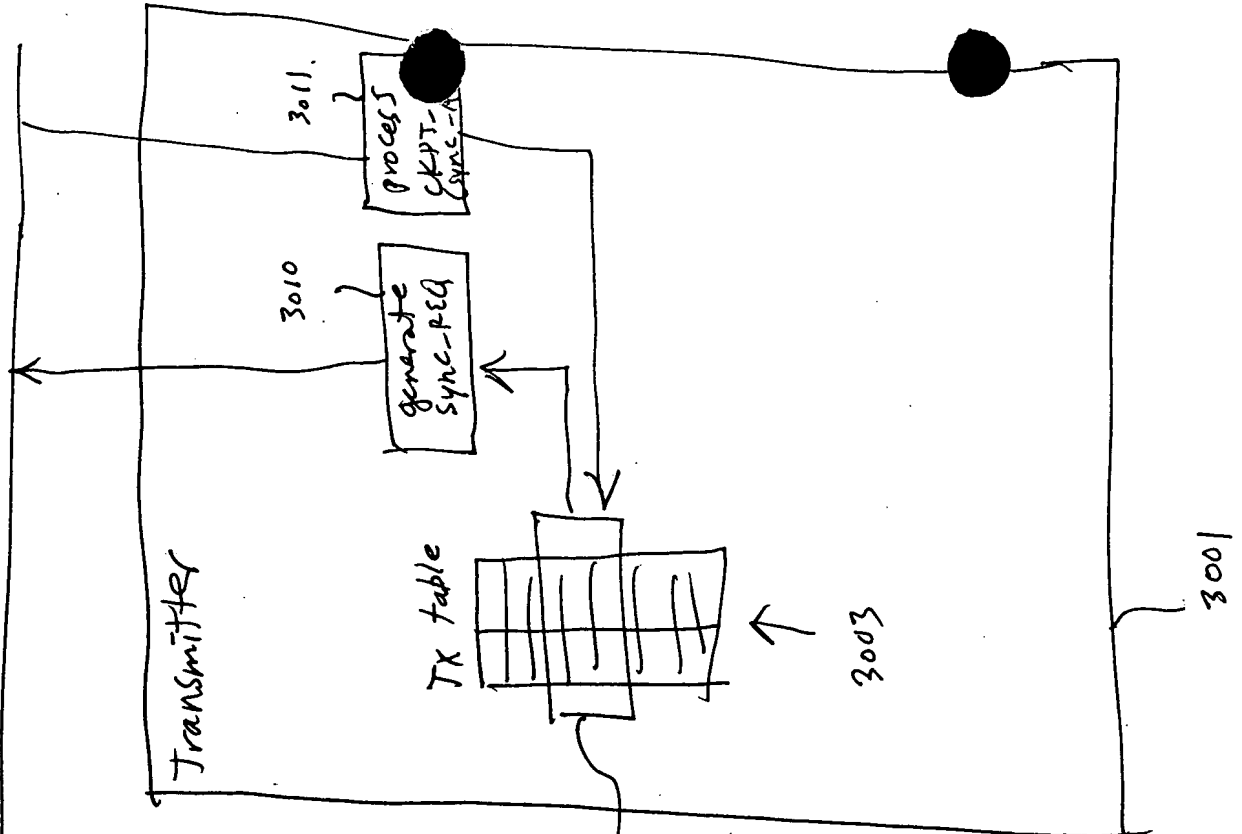


FIG. 30

FIG. 31 is a schematic diagram of a network architecture.

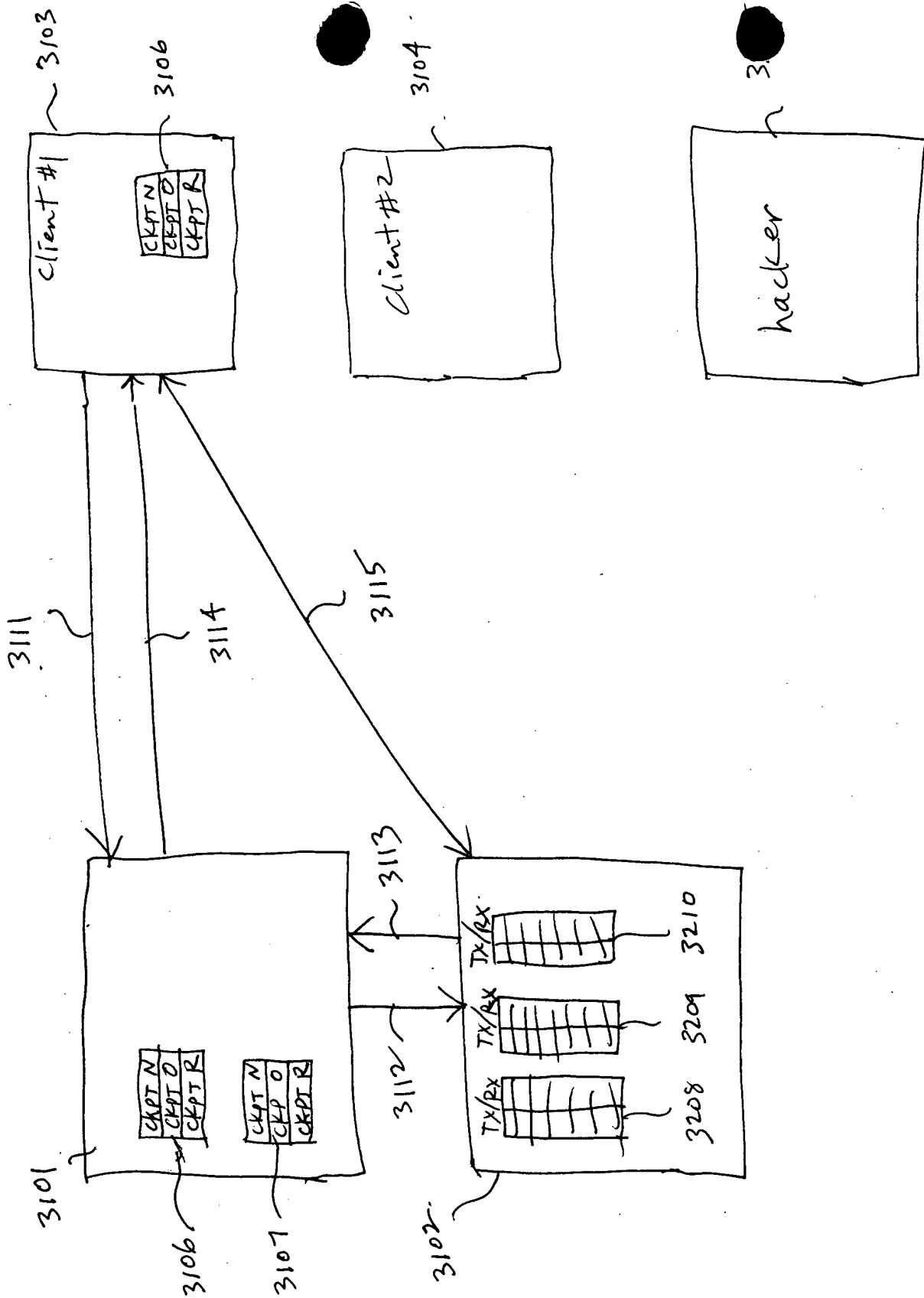


FIG- 31

Client

Server

Send Data Packet  
Using CKPT\_N  
CKPT\_O=CKPT\_N  
Generate New CKPT\_N  
Start timer, Shut transmitter  
Off

If CKPT\_O in SYNC\_ACK  
matches Transmitter's  
CKPT\_O  
Update Receiver's  
CKPT\_R  
Kill Timer, Turn  
Transmitter On

Send Data Packet  
Using CKPT\_N  
CKPT\_O=CKPT\_N  
Generate New CKPT\_N  
Start timer, Shut transmitter  
Off

When timer expires  
Transmit SYNC\_REQ  
using Transmitters  
CKPT\_O, Start Timer

If CKPT\_O in SYNC\_ACK  
matches Transmitter's  
CKPT\_O  
Update Receiver's  
CKPT\_R  
Kill Timer, Turn  
Transmitter On

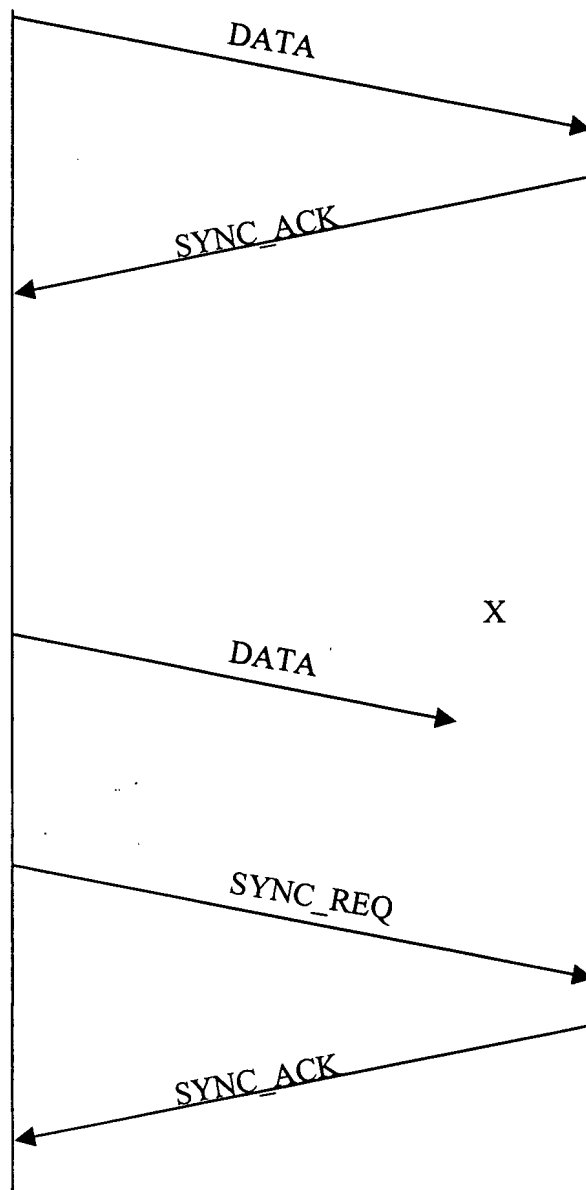


FIG. 32

**IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL  
FOR SECURE COMMUNICATIONS  
WITH ASSURED SYSTEM AVAILABILITY**

5

**CROSS-REFERENCE TO RELATED APPLICATION**

This application claims priority from and is a continuation-in-part of previously filed U.S. application serial number 09/429,643, filed on October 29, 1999. The subject matter of that application, which is bodily incorporated herein, derives from provisional U.S. application numbers 60/106,261 (filed October 30, 1998) and 60/137,704 (filed June 7, 1999).

10

**BACKGROUND OF THE INVENTION**

15

20

25

A tremendous variety of methods have been proposed and implemented to provide security and anonymity for communications over the Internet. The variety stems, in part, from the different needs of different Internet users. A basic heuristic framework to aid in discussing these different security techniques is illustrated in FIG. 1. Two terminals, an originating terminal 100 and a destination terminal 110 are in communication over the Internet. It is desired for the communications to be secure, that is, immune to eavesdropping. For example, terminal 100 may transmit secret information to terminal 110 over the Internet 107. Also, it may be desired to prevent an eavesdropper from discovering that terminal 100 is in communication with terminal 110. For example, if terminal 100 is a user and terminal 110 hosts a web site, terminal 100's user may not want anyone in the intervening networks to know what web sites he is "visiting." Anonymity would thus be an issue, for example, for companies that want to keep their market research interests private and thus would prefer to prevent outsiders from knowing which web-sites or other Internet resources they are "visiting." These two security issues may be called data security and anonymity, respectively.

Data security is usually tackled using some form of data encryption. An encryption key 48 is known at both the originating and terminating terminals 100 and 110. The keys may be private and public at the originating and destination terminals 100 and 110, respectively or they

may be symmetrical keys (the same key is used by both parties to encrypt and decrypt). Many encryption methods are known and usable in this context.

To hide traffic from a local administrator or ISP, a user can employ a local proxy server in communicating over an encrypted channel with an outside proxy such that the local administrator or ISP only sees the encrypted traffic. Proxy servers prevent destination servers  
5 from determining the identities of the originating clients. This system employs an intermediate server interposed between client and destination server. The destination server sees only the Internet Protocol (IP) address of the proxy server and not the originating client. The target server only sees the address of the outside proxy. This scheme relies on a trusted outside proxy server.

Also, proxy schemes are vulnerable to traffic analysis methods of determining identities of transmitters and receivers. Another important limitation of proxy servers is that the server knows the identities of both calling and called parties. In many instances, an originating terminal, such as terminal A, would prefer to keep its identity concealed from the proxy, for example, if the proxy server is provided by an Internet service provider (ISP).  
10

To defeat traffic analysis, a scheme called Chaum's mixes employs a proxy server that transmits and receives fixed length messages, including dummy messages. Multiple originating terminals are connected through a mix (a server) to multiple target servers. It is difficult to tell which of the originating terminals are communicating to which of the connected target servers, and the dummy messages confuse eavesdroppers' efforts to detect communicating pairs by  
15 analyzing traffic. A drawback is that there is a risk that the mix server could be compromised. One way to deal with this risk is to spread the trust among multiple mixes. If one mix is compromised, the identities of the originating and target terminals may remain concealed. This strategy requires a number of alternative mixes so that the intermediate servers interposed between the originating and target terminals are not determinable except by compromising more  
20 than one mix. The strategy wraps the message with multiple layers of encrypted addresses. The first mix in a sequence can decrypt only the outer layer of the message to reveal the next destination mix in sequence. The second mix can decrypt the message to reveal the next mix and so on. The target server receives the message and, optionally, a multi-layer encrypted payload  
25





## SUMMARY OF THE INVENTION

A secure mechanism for communicating over the internet, including a protocol referred to as the Tunneled Agile Routing Protocol (TARP), uses a unique two-layer encryption format and special TARP routers. TARP routers are similar in function to regular IP routers. Each TARP router has one or more IP addresses and uses normal IP protocol to send IP packet messages ("packets" or "datagrams"). The IP packets exchanged between TARP terminals via TARP routers are actually encrypted packets whose true destination address is concealed except to TARP routers and servers. The normal or "clear" or "outside" IP header attached to TARP IP packets contains only the address of a next hop router or destination server. That is, instead of indicating a final destination in the destination field of the IP header, the TARP packet's IP header always points to a next-hop in a series of TARP router hops, or to the final destination. This means there is no overt indication from an intercepted TARP packet of the true destination of the TARP packet since the destination could always be next-hop TARP router as well as the final destination.

Each TARP packet's true destination is concealed behind a layer of encryption generated using a link key. The link key is the encryption key used for encrypted communication between the hops intervening between an originating TARP terminal and a destination TARP terminal. Each TARP router can remove the outer layer of encryption to reveal the destination router for each TARP packet. To identify the link key needed to decrypt the outer layer of encryption of a TARP packet, a receiving TARP or routing terminal may identify the transmitting terminal by the sender/receiver IP numbers in the cleartext IP header.

Once the outer layer of encryption is removed, the TARP router determines the final destination. Each TARP packet undergoes a minimum number of hops to help foil traffic analysis. The hops may be chosen at random or by a fixed value. As a result, each TARP packet may make random trips among a number of geographically disparate routers before reaching its destination. Each trip is highly likely to be different for each packet composing a given message because each trip is independently randomly determined. This feature is called *agile routing*. The fact that different packets take different routes provides distinct advantages by making it difficult

for an interloper to obtain all the packets forming an entire multi-packet message. The associated advantages have to do with the inner layer of encryption discussed below. Agile routing is combined with another feature that furthers this purpose; a feature that ensures that any message is broken into multiple packets.

5           The IP address of a TARP router can be changed, a feature called *IP agility*. Each TARP router, independently or under direction from another TARP terminal or router, can change its IP address. A separate, unchangeable identifier or address is also defined. This address, called the TARP address, is known only to TARP routers and terminals and may be correlated at any time by a TARP router or a TARP terminal using a Lookup Table (LUT). When a TARP router or  
10 terminal changes its IP address, it updates the other TARP routers and terminals which in turn update their respective LUTs.

          The message payload is hidden behind an inner layer of encryption in the TARP packet that can only be unlocked using a session key. The session key is not available to any of the intervening TARP routers. The session key is used to decrypt the payloads of the TARP packets  
15 permitting the data stream to be reconstructed.

          Communication may be made private using link and session keys, which in turn may be shared and used according to any desired method. For example, public/private keys or symmetric keys may be used.

          To transmit a data stream, a TARP originating terminal constructs a series of TARP  
20 packets from a series of IP packets generated by a network (IP) layer process. (Note that the terms "network layer," "data link layer," "application layer," etc. used in this specification correspond to the Open Systems Interconnection (OSI) network terminology.) The payloads of these packets are assembled into a block and chain-block encrypted using the session key. This assumes, of course, that all the IP packets are destined for the same TARP terminal. The block is  
25 then interleaved and the interleaved encrypted block is broken into a series of payloads, one for each TARP packet to be generated. Special TARP headers  $IP_T$  are then added to each payload using the IP headers from the data stream packets. The TARP headers can be identical to normal IP headers or customized in some way. They should contain a formula or data for deinterleaving

the data at the destination TARP terminal, a time-to-live (TTL) parameter to indicate the number of hops still to be executed, a data type identifier which indicates whether the payload contains, for example, TCP or UDP data, the sender's TARP address, the destination TARP address, and an indicator as to whether the packet contains real or decoy data or a formula for filtering out  
5 decoy data if decoy data is spread in some way through the TARP payload data.

Note that although chain-block encryption is discussed here with reference to the session key, any encryption method may be used. Preferably, as in chain block encryption, a method should be used that makes unauthorized decryption difficult without an entire result of the encryption process. Thus, by separating the encrypted block among multiple packets and making  
10 it difficult for an interloper to obtain access to all of such packets, the contents of the communications are provided an extra layer of security.

Decoy or dummy data can be added to a stream to help foil traffic analysis by reducing the peak-to-average network load. It may be desirable to provide the TARP process with an ability to respond to the time of day or other criteria to generate more decoy data during low  
15 traffic periods so that communication bursts at one point in the Internet cannot be tied to communication bursts at another point to reveal the communicating endpoints.

Dummy data also helps to break the data into a larger number of inconspicuously-sized packets permitting the interleave window size to be increased while maintaining a reasonable size for each packet. (The packet size can be a single standard size or selected from a fixed range  
20 of sizes.) One primary reason for desiring for each message to be broken into multiple packets is apparent if a chain block encryption scheme is used to form the first encryption layer prior to interleaving. A single block encryption may be applied to portion, or entirety, of a message, and that portion or entirety then interleaved into a number of separate packets. Considering the agile IP routing of the packets, and the attendant difficulty of reconstructing an entire sequence of  
25 packets to form a single block-encrypted message element, decoy packets can significantly increase the difficulty of reconstructing an entire data stream.

The above scheme may be implemented entirely by processes operating between the data link layer and the network layer of each server or terminal participating in the TARP system.

0479.85672



likely to insert decoy packets when a message stream is being received. Alternatively, if a large number of decoy packets is received along with regular TARP packets, the algorithm may increase the rate of dropping of decoy packets rather than forwarding them. The result of dropping and generating decoy packets in this way is to make the apparent incoming message size different from the apparent outgoing message size to help foil traffic analysis.

In various other embodiments of the invention, a scalable version of the system may be constructed in which a plurality of IP addresses are preassigned to each pair of communicating nodes in the network. Each pair of nodes agrees upon an algorithm for "hopping" between IP addresses (both sending and receiving), such that an eavesdropper sees apparently continuously random IP address pairs (source and destination) for packets transmitted between the pair. Overlapping or "reusable" IP addresses may be allocated to different users on the same subnet, since each node merely verifies that a particular packet includes a valid source/destination pair from the agreed-upon algorithm. Source/destination pairs are preferably not reused between any two nodes during any given end-to-end session, though limited IP block sizes or lengthy sessions might require it.

Further improvements described in this continuation-in-part application include: (1) a load balancer that distributes packets across different transmission paths according to transmission path quality; (2) a DNS proxy server that transparently creates a virtual private network in response to a domain name inquiry; (3) a large-to-small link bandwidth management feature that prevents denial-of-service attacks at system chokepoints; (4) a traffic limiter that regulates incoming packets by limiting the rate at which a transmitter can be synchronized with a receiver; and (5) a signaling synchronizer that allows a large number of nodes to communicate with a central node by partitioning the communication function between two separate entities

**BRIEF DESCRIPTION OF THE DRAWINGS**

FIG. 1 is an illustration of secure communications over the Internet according to a prior art embodiment.

FIG. 2 is an illustration of secure communications over the Internet according to a an embodiment of the invention.

FIG. 3a is an illustration of a process of forming a tunneled IP packet according to an embodiment of the invention.

FIG. 3b is an illustration of a process of forming a tunneled IP packet according to another embodiment of the invention.

5 FIG. 4 is an illustration of an OSI layer location of processes that may be used to implement the invention.

FIG. 5 is a flow chart illustrating a process for routing a tunneled packet according to an embodiment of the invention.

10 FIG. 6 is a flow chart illustrating a process for forming a tunneled packet according to an embodiment of the invention.

FIG. 7 is a flow chart illustrating a process for receiving a tunneled packet according to an embodiment of the invention.

FIG. 8 shows how a secure session is established and synchronized between a client and a TARP router.

15 FIG. 9 shows an IP address hopping scheme between a client computer and TARP router using transmit and receive tables in each computer.

FIG. 10 shows physical link redundancy among three Internet Service Providers (ISPs) and a client computer.

20 FIG. 11 shows how multiple IP packets can be embedded into a single "frame" such as an Ethernet frame, and further shows the use of a discriminator field to camouflage true packet recipients.

FIG. 12A shows a system that employs hopped hardware addresses, hopped IP addresses, and hopped discriminator fields.

25 FIG. 12B shows several different approaches for hopping hardware addresses, IP addresses, and discriminator fields in combination.

FIG. 13 shows a technique for automatically re-establishing synchronization between sender and receiver through the use of a partially public sync value.

0479.85672

9 10

FIG. 14 shows a "checkpoint" scheme for regaining synchronization between a sender and recipient.

FIG. 15 shows further details of the checkpoint scheme of FIG. 14.

5 FIG. 16 shows how two addresses can be decomposed into a plurality of segments for comparison with presence vectors.

FIG. 17 shows a storage array for a receiver's active addresses.

FIG. 18 shows the receiver's storage array after receiving a sync request.

FIG. 19 shows the receiver's storage array after new addresses have been generated.

FIG. 20 shows a system employing distributed transmission paths.

10 FIG. 21 shows a plurality of link transmission tables that can be used to route packets in the system of FIG. 20.

FIG. 22A shows a flowchart for adjusting weight value distributions associated with a plurality of transmission links.

FIG. 22B shows a flowchart for setting a weight value to zero if a transmitter turns off.

15 FIG. 23 shows a system employing distributed transmission paths with adjusted weight value distributions for each path.

FIG. 24 shows an example using the system of FIG. 23.

FIG. 25 shows a conventional domain-name look-up service.

FIG. 26 shows a system employing a DNS proxy server with transparent VPN creation.

20 FIG. 27 shows steps that can be carried out to implement transparent VPN creation based on a DNS look-up function.

FIG. 28 shows a system including a link guard function that prevents packet overloading on a low-bandwidth link LOW BW.

FIG. 29 shows one embodiment of a system employing the principles of FIG. 28.

25 FIG. 30 shows a system that regulates packet transmission rates by throttling the rate at which synchronizations are performed.

FIG. 31 shows a signaling server 3101 and a transport server 3102 used to establish a VPN with a client computer.

||



FIG. 32 shows message flows relating to synchronization protocols of FIG. 31.

**DETAILED DESCRIPTION OF THE INVENTION**

Referring to FIG. 2, a secure mechanism for communicating over the internet employs a number of special routers or servers, called TARP routers 122-127 that are similar to regular IP routers 128-132 in that each has one or more IP addresses and uses normal IP protocol to send normal-looking IP packet messages, called TARP packets 140. TARP packets 140 are identical to normal IP packet messages that are routed by regular IP routers 128-132 because each TARP packet 140 contains a destination address as in a normal IP packet. However, instead of indicating a final destination in the destination field of the IP header, the TARP packet's 140 IP header always points to a next-hop in a series of TARP router hops, or the final destination, TARP terminal 110. Because the header of the TARP packet contains only the next-hop destination, there is no overt indication from an intercepted TARP packet of the true destination of the TARP packet 140 since the destination could always be the next-hop TARP router as well as the final destination, TARP terminal 110.

Each TARP packet's true destination is concealed behind an outer layer of encryption generated using a link key 146. The link key 146 is the encryption key used for encrypted communication between the end points (TARP terminals or TARP routers) of a single link in the chain of hops connecting the originating TARP terminal 100 and the destination TARP terminal 110. Each TARP router 122-127, using the link key 146 it uses to communicate with the previous hop in a chain, can use the link key to reveal the true destination of a TARP packet. To identify the link key needed to decrypt the outer layer of encryption of a TARP packet, a receiving TARP or routing terminal may identify the transmitting terminal (which may indicate the link key used) by the sender field of the clear IP header. Alternatively, this identity may be hidden behind another layer of encryption in available bits in the clear IP header. Each TARP router, upon receiving a TARP message, determines if the message is a TARP message by using authentication data in the TARP packet. This could be recorded in available bytes in the TARP packet's IP header. Alternatively, TARP packets could be authenticated by attempting to decrypt

using the link key 146 and determining if the results are as expected. The former may have computational advantages because it does not involve a decryption process.

Once the outer layer of decryption is completed by a TARP router 122-127, the TARP router determines the final destination. The system is preferably designed to cause each TARP packet 140 to undergo a minimum number of hops to help foil traffic analysis. The time to live counter in the IP header of the TARP message may be used to indicate a number of TARP router hops yet to be completed. Each TARP router then would decrement the counter and determine from that whether it should forward the TARP packet 140 to another TARP router 122-127 or to the destination TARP terminal 110. If the time to live counter is zero or below zero after decrementing, for an example of usage, the TARP router receiving the TARP packet 140 may forward the TARP packet 140 to the destination TARP terminal 110. If the time to live counter is above zero after decrementing, for an example of usage, the TARP router receiving the TARP packet 140 may forward the TARP packet 140 to a TARP router 122-127 that the current TARP terminal chooses at random. As a result, each TARP packet 140 is routed through some minimum number of hops of TARP routers 122-127 which are chosen at random.

Thus, each TARP packet, irrespective of the traditional factors determining traffic in the Internet, makes random trips among a number of geographically disparate routers before reaching its destination and each trip is highly likely to be different for each packet composing a given message because each trip is independently randomly determined as described above. This feature is called *agile routing*. For reasons that will become clear shortly, the fact that different packets take different routes provides distinct advantages by making it difficult for an interloper to obtain all the packets forming an entire multi-packet message. Agile routing is combined with another feature that furthers this purpose, a feature that ensures that any message is broken into multiple packets.

A TARP router receives a TARP packet when an IP address used by the TARP router coincides with the IP address in the TARP packet's IP header IP<sub>C</sub>. The IP address of a TARP router, however, may not remain constant. To avoid and manage attacks, each TARP router, independently or under direction from another TARP terminal or router, may change its IP

address. A separate, unchangeable identifier or address is also defined. This address, called the TARP address, is known only to TARP routers and terminals and may be correlated at any time by a TARP router or a TARP terminal using a Lookup Table (LUT). When a TARP router or terminal changes its IP address, it updates the other TARP routers and terminals which in turn update their respective LUTs. In reality, whenever a TARP router looks up the address of a destination in the encrypted header, it must convert a TARP address to a real IP address using its LUT.

While every TARP router receiving a TARP packet has the ability to determine the packet's final destination, the message payload is embedded behind an inner layer of encryption in the TARP packet that can only be unlocked using a session key. The session key is not available to any of the TARP routers 122-127 intervening between the originating 100 and destination 110 TARP terminals. The session key is used to decrypt the payloads of the TARP packets 140 permitting an entire message to be reconstructed.

In one embodiment, communication may be made private using link and session keys, which in turn may be shared and used according any desired method. For example, a public key or symmetric keys may be communicated between link or session endpoints using a public key method. Any of a variety of other mechanisms for securing data to ensure that only authorized computers can have access to the private information in the TARP packets 140 may be used as desired.

Referring to FIG. 3a, to construct a series of TARP packets, a data stream 300 of IP packets 207a, 207b, 207c, etc., such series of packets being formed by a network (IP) layer process, is broken into a series of small sized segments. In the present example, equal-sized segments 1-9 are defined and used to construct a set of interleaved data packets A, B, and C. Here it is assumed that the number of interleaved packets A, B, and C formed is three and that the number of IP packets 207a-207c used to form the three interleaved packets A, B, and C is exactly three. Of course, the number of IP packets spread over a group of interleaved packets may be any convenient number as may be the number of interleaved packets over which the

incoming data stream is spread. The latter, the number of interleaved packets over which the data stream is spread, is called the *interleave window*.

To create a packet, the transmitting software interleaves the normal IP packets 207a *et seq.* to form a new set of interleaved payload data 320. This payload data 320 is then encrypted using a session key to form a set of session-key-encrypted payload data 330, each of which, A, B, and C, will form the payload of a TARP packet. Using the IP header data, from the original packets 207a-207c, new TARP headers  $IP_T$  are formed. The TARP headers  $IP_T$  can be identical to normal IP headers or customized in some way. In a preferred embodiment, the TARP headers  $IP_T$  are IP headers with added data providing the following information required for routing and reconstruction of messages, some of which data is ordinarily, or capable of being, contained in normal IP headers:

1. A window sequence number – an identifier that indicates where the packet belongs in the original message sequence.
2. An interleave sequence number – an identifier that indicates the interleaving sequence used to form the packet so that the packet can be deinterleaved along with other packets in the interleave window.
3. A time-to-live (TTL) datum – indicates the number of TARP-router-hops to be executed before the packet reaches its destination. Note that the TTL parameter may provide a datum to be used in a probabilistic formula for determining whether to route the packet to the destination or to another hop.
4. Data type identifier – indicates whether the payload contains, for example, TCP or UDP data.
5. Sender's address – indicates the sender's address in the TARP network.
6. Destination address – indicates the destination terminal's address in the TARP network.
7. Decoy/Real – an indicator of whether the packet contains real message data or dummy decoy data or a combination.

Obviously, the packets going into a single interleave window must include only packets with a common destination. Thus, it is assumed in the depicted example that the IP headers of IP packets 207a-207c all contain the same destination address or at least will be received by the same terminal so that they can be deinterleaved. Note that dummy or decoy data or packets can be added to form a larger interleave window than would otherwise be required by the size of a given message. Decoy or dummy data can be added to a stream to help foil traffic analysis by leveling the load on the network. Thus, it may be desirable to provide the TARP process with an ability to respond to the time of day or other criteria to generate more decoy data during low traffic periods so that communication bursts at one point in the Internet cannot be tied to communication bursts at another point to reveal the communicating endpoints.

Dummy data also helps to break the data into a larger number of inconspicuously-sized packets permitting the interleave window size to be increased while maintaining a reasonable size for each packet. (The packet size can be a single standard size or selected from a fixed range of sizes.) One primary reason for desiring for each message to be broken into multiple packets is apparent if a chain block encryption scheme is used to form the first encryption layer prior to interleaving. A single block encryption may be applied to a portion, or the entirety, of a message, and that portion or entirety then interleaved into a number of separate packets.

Referring to FIG. 3b, in an alternative mode of TARP packet construction, a series of IP packets are accumulated to make up a predefined interleave window. The payloads of the packets are used to construct a single block 520 for chain block encryption using the session key. The payloads used to form the block are presumed to be destined for the same terminal. The block size may coincide with the interleave window as depicted in the example embodiment of FIG. 3b. After encryption, the encrypted block is broken into separate payloads and segments which are interleaved as in the embodiment of Fig 3a. The resulting interleaved packets A, B, and C, are then packaged as TARP packets with TARP headers as in the Example of FIG. 3a. The remaining process is as shown in, and discussed with reference to, FIG. 3a.

Once the TARP packets 340 are formed, each entire TARP packet 340, including the TARP header  $IP_T$ , is encrypted using the link key for communication with the first-hop-TARP

router. The first hop TARP router is randomly chosen. A final unencrypted IP header  $IP_C$  is added to each encrypted TARP packet 340 to form a normal IP packet 360 that can be transmitted to a TARP router. Note that the process of constructing the TARP packet 360 does not have to be done in stages as described. The above description is just a useful heuristic for describing the final product, namely, the TARP packet.

Note that, TARP header  $IP_T$  could be a completely custom header configuration with no similarity to a normal IP header except that it contain the information identified above. This is so since this header is interpreted by only TARP routers.

The above scheme may be implemented entirely by processes operating between the data link layer and the network layer of each server or terminal participating in the TARP system. Referring to FIG. 4, a TARP transceiver 405 can be an originating terminal 100, a destination terminal 110, or a TARP router 122-127. In each TARP Transceiver 405, a transmitting process is generated to receive normal packets from the Network (IP) layer and generate TARP packets for communication over the network. A receiving process is generated to receive normal IP packets containing TARP packets and generate from these normal IP packets which are "passed up" to the Network (IP) layer. Note that where the TARP Transceiver 405 is a router, the received TARP packets 140 are not processed into a stream of IP packets 415 because they need only be authenticated as proper TARP packets and then passed to another TARP router or a TARP destination terminal 110. The intervening process, a "TARP Layer" 420, could be combined with either the data link layer 430 or the Network layer 410. In either case, it would intervene between the data link layer 430 so that the process would receive regular IP packets containing embedded TARP packets and "hand up" a series of reassembled IP packets to the Network layer 410. As an example of combining the TARP layer 420 with the data link layer 430, a program may augment the normal processes running a communications card, for example, an Ethernet card. Alternatively, the TARP layer processes may form part of a dynamically loadable module that is loaded and executed to support communications between the network and data link layers.

Because the encryption system described above can be inserted between the data link and network layers, the processes involved in supporting the encrypted communication may be completely transparent to processes at the IP (network) layer and above. The TARP processes may also be completely transparent to the data link layer processes as well. Thus, no operations at or above the network layer, or at or below the data link layer, are affected by the insertion of the TARP stack. This provides additional security to all processes at or above the network layer, since the difficulty of unauthorized penetration of the network layer (by, for example, a hacker) is increased substantially. Even newly developed servers running at the session layer leave all processes below the session layer vulnerable to attack. Note that in this architecture, security is distributed. That is, notebook computers used by executives on the road, for example, can communicate over the Internet without any compromise in security.

Note that IP address changes made by TARP terminals and routers can be done at regular intervals, at random intervals, or upon detection of "attacks." The variation of IP addresses hinders traffic analysis that might reveal which computers are communicating, and also provides a degree of immunity from attack. The level of immunity from attack is roughly proportional to the rate at which the IP address of the host is changing.

As mentioned, IP addresses may be changed in response to attacks. An attack may be revealed, for example, by a regular series of messages indicates that a router is being probed in some way. Upon detection of an attack, the TARP layer process may respond to this event by changing its IP address. To accomplish this, the TARP process will construct a TARP-formatted message, in the style of Internet Control Message Protocol (ICMP) datagrams as an example; this message will contain the machine's TARP address, its previous IP address, and its new IP address. The TARP layer will transmit this packet to at least one known TARP router; then upon receipt and validation of the message, the TARP router will update its LUT with the new IP address for the stated TARP address. The TARP router will then format a similar message, and broadcast it to the other TARP routers so that they may update their LUTs. Since the total number of TARP routers on any given subnet is expected to be relatively small, this process of updating the LUTs should be relatively fast. It may not, however, work as well when there is a

0479.85672

relatively large number of TARP routers and/or a relatively large number of clients; this has motivated a refinement of this architecture to provide scalability; this refinement has led to a second embodiment, which is discussed below.

5 Upon detection of an attack, the TARP process may also create a subprocess that maintains the original IP address and continues interacting with the attacker. The latter may provide an opportunity to trace the attacker or study the attacker's methods (called "fishbowling" drawing upon the analogy of a small fish in a fish bowl that "thinks" it is in the ocean but is actually under captive observation). A history of the communication between the attacker and the abandoned (fishbowed) IP address can be recorded or transmitted for human analysis or further  
10 synthesized for purposes of responding in some way.

As mentioned above, decoy or dummy data or packets can be added to outgoing data streams by TARP terminals or routers. In addition to making it convenient to spread data over a larger number of separate packets, such decoy packets can also help to level the load on inactive portions of the Internet to help foil traffic analysis efforts.

15 Decoy packets may be generated by each TARP terminal 100, 110 or each router 122-127 on some basis determined by an algorithm. For example, the algorithm may be a random one which calls for the generation of a packet on a random basis when the terminal is idle. Alternatively, the algorithm may be responsive to time of day or detection of low traffic to generate more decoy packets during low traffic times. Note that packets are preferably generated  
20 in groups, rather than one by one, the groups being sized to simulate real messages. In addition, so that decoy packets may be inserted in normal TARP message streams, the background loop may have a latch that makes it more likely to insert decoy packets when a message stream is being received. That is, when a series of messages are received, the decoy packet generation rate may be increased. Alternatively, if a large number of decoy packets is received along with  
25 regular TARP packets, the algorithm may increase the rate of dropping of decoy packets rather than forwarding them. The result of dropping and generating decoy packets in this way is to make the apparent incoming message size different from the apparent outgoing message size to help foil traffic analysis. The rate of reception of packets, decoy or otherwise, may be indicated

19



to the decoy packet dropping and generating processes through perishable decoy and regular packet counters. (A perishable counter is one that resets or decrements its value in response to time so that it contains a high value when it is incremented in rapid succession and a small value when incremented either slowly or a small number of times in rapid succession.) Note that  
5 destination TARP terminal 110 may generate decoy packets equal in number and size to those TARP packets received to make it appear it is merely routing packets and is therefore not the destination terminal.

Referring to FIG. 5, the following particular steps may be employed in the above-described method for routing TARP packets.

10

• S0. A background loop operation is performed which applies an algorithm which determines the generation of decoy IP packets. The loop is interrupted when an encrypted TARP packet is received.

15

• S2. The TARP packet may be probed in some way to authenticate the packet before attempting to decrypt it using the link key. That is, the router may determine that the packet is an authentic TARP packet by performing a selected operation on some data included with the clear IP header attached to the encrypted TARP packet contained in the payload. This makes it possible to avoid performing decryption on packets that are not authentic TARP packets.

20

• S3. The TARP packet is decrypted to expose the destination TARP address and an indication of whether the packet is a decoy packet or part of a real message.

• S4. If the packet is a decoy packet, the perishable decoy counter is incremented.

25

• S5. Based on the decoy generation/dropping algorithm and the perishable decoy counter value, if the packet is a decoy packet, the router may choose to throw it away. If the received packet is a decoy packet and it is determined that it should be thrown away (S6), control returns to step S0.

- S7. The TTL parameter of the TARP header is decremented and it is determined if the TTL parameter is greater than zero.
- S8. If the TTL parameter is greater than zero, a TARP address is randomly chosen from a list of TARP addresses maintained by the router and the link key and IP address corresponding to that TARP address memorized for use in creating a new IP packet containing the TARP packet.
- S9. If the TTL parameter is zero or less, the link key and IP address corresponding to the TARP address of the destination are memorized for use in creating the new IP packet containing the TARP packet.
- S10. The TARP packet is encrypted using the memorized link key.
- S11. An IP header is added to the packet that contains the stored IP address, the encrypted TARP packet wrapped with an IP header, and the completed packet transmitted to the next hop or destination.

Referring to FIG. 6, the following particular steps may be employed in the above-described method for generating TARP packets.

- S20. A background loop operation applies an algorithm that determines the generation of decoy IP packets. The loop is interrupted when a data stream containing IP packets is received for transmission.
- S21. The received IP packets are grouped into a set consisting of messages with a constant IP destination address. The set is further broken down to coincide with a maximum size of an interleave window. The set is encrypted, and interleaved into a set of payloads destined to become TARP packets.
- S22. The TARP address corresponding to the IP address is determined from a lookup table and stored to generate the TARP header. An initial TTL count is generated and stored in the

header. The TTL count may be random with minimum and maximum values or it may be fixed or determined by some other parameter.

- S23. The window sequence numbers and interleave sequence numbers are recorded in the TARP headers of each packet.
- 5 • S24. One TARP router address is randomly chosen for each TARP packet and the IP address corresponding to it stored for use in the clear IP header. The link key corresponding to this router is identified and used to encrypt TARP packets containing interleaved and encrypted data and TARP headers.
- S25. A clear IP header with the first hop router's real IP address is generated and added to each of the encrypted TARP packets and the resulting packets.

Referring to FIG. 7, the following particular steps may be employed in the above-described method for receiving TARP packets.

- 15 • S40. A background loop operation is performed which applies an algorithm which determines the generation of decoy IP packets. The loop is interrupted when an encrypted TARP packet is received.
- S42. The TARP packet may be probed to authenticate the packet before attempting to decrypt it using the link key.
- 20 • S43. The TARP packet is decrypted with the appropriate link key to expose the destination TARP address and an indication of whether the packet is a decoy packet or part of a real message.
- S44. If the packet is a decoy packet, the perishable decoy counter is incremented.
- S45. Based on the decoy generation/dropping algorithm and the perishable decoy counter value, if the packet is a decoy packet, the receiver may choose to throw it away.
- 25 • S46. The TARP packets are cached until all packets forming an interleave window are received.

- S47. Once all packets of an interleave window are received, the packets are deinterleaved.
- S48. The packets block of combined packets defining the interleave window is then decrypted using the session key.
- S49. The decrypted block is then divided using the window sequence data and the IP<sub>T</sub> headers are converted into normal IP<sub>C</sub> headers. The window sequence numbers are integrated in the IP<sub>C</sub> headers.
- S50. The packets are then handed up to the IP layer processes.

**1. SCALABILITY ENHANCEMENTS**

The IP agility feature described above relies on the ability to transmit IP address changes to all TARP routers. The embodiments including this feature will be referred to as “boutique” embodiments due to potential limitations in scaling these features up for a large network, such as the Internet. (The “boutique” embodiments would, however, be robust for use in smaller networks, such as small virtual private networks, for example). One problem with the boutique embodiments is that if IP address changes are to occur frequently, the message traffic required to update all routers sufficiently quickly creates a serious burden on the Internet when the TARP router and/or client population gets large. The bandwidth burden added to the networks, for example in ICMP packets, that would be used to update all the TARP routers could overwhelm the Internet for a large scale implementation that approached the scale of the Internet. In other words, the boutique system’s scalability is limited.

A system can be constructed which trades some of the features of the above embodiments to provide the benefits of IP agility without the additional messaging burden. This is accomplished by IP address-hopping according to shared algorithms that govern IP addresses used between links participating in communications sessions between nodes such as TARP nodes. (Note that the IP hopping technique is also applicable to the boutique embodiment.) The IP agility feature discussed with respect to the boutique system can be modified so that it becomes decentralized under this scalable regime and governed by the above-described shared

algorithm. Other features of the boutique system may be combined with this new type of IP-agility.

The new embodiment has the advantage of providing IP agility governed by a local algorithm and set of IP addresses exchanged by each communicating pair of nodes. This local  
5 governance is session-independent in that it may govern communications between a pair of nodes, irrespective of the session or end points being transferred between the directly communicating pair of nodes.

In the scalable embodiments, blocks of IP addresses are allocated to each node in the network. (This scalability will increase in the future, when Internet Protocol addresses are  
10 increased to 128-bit fields, vastly increasing the number of distinctly addressable nodes). Each node can thus use any of the IP addresses assigned to that node to communicate with other nodes in the network. Indeed, each pair of communicating nodes can use a plurality of source IP addresses and destination IP addresses for communicating with each other.

Each communicating pair of nodes in a chain participating in any session stores two  
15 blocks of IP addresses, called netblocks, and an algorithm and randomization seed for selecting, from each netblock, the next pair of source/destination IP addresses that will be used to transmit the next message. In other words, the algorithm governs the sequential selection of IP-address pairs, one sender and one receiver IP address, from each netblock. The combination of algorithm, seed, and netblock (IP address block) will be called a "hopblock." A router issues separate  
20 transmit and receive hopblocks to its clients. The send address and the receive address of the IP header of each outgoing packet sent by the client are filled with the send and receive IP addresses generated by the algorithm. The algorithm is "clocked" (indexed) by a counter so that each time a pair is used, the algorithm turns out a new transmit pair for the next packet to be sent.

The router's receive hopblock is identical to the client's transmit hopblock. The router  
25 uses the receive hopblock to predict what the send and receive IP address pair for the next expected packet from that client will be. Since packets can be received out of order, it is not possible for the router to predict with certainty what IP address pair will be on the next sequential packet. To account for this problem, the router generates a range of predictions

encompassing the number of possible transmitted packet send/receive addresses, of which the next packet received could leap ahead. Thus, if there is a vanishingly small probability that a given packet will arrive at the router ahead of 5 packets transmitted by the client before the given packet, then the router can generate a series of 6 send/receive IP address pairs (or "hop window")

5 to compare with the next received packet. When a packet is received, it is marked in the hop window as such, so that a second packet with the same IP address pair will be discarded. If an out-of-sequence packet does not arrive within a predetermined timeout period, it can be requested for retransmission or simply discarded from the receive table, depending upon the protocol in use for that communications session, or possibly by convention.

10 When the router receives the client's packet, it compares the send and receive IP addresses of the packet with the next N predicted send and receive IP address pairs and rejects the packet if it is not a member of this set. Received packets that do not have the predicted source/destination IP addresses falling within the window are rejected, thus thwarting possible

15 hackers. (With the number of possible combinations, even a fairly large window would be hard to fall into at random.) If it is a member of this set, the router accepts the packet and processes it further. This link-based IP-hopping strategy, referred to as "IHOP," is a network element that stands on its own and is not necessarily accompanied by elements of the boutique system described above. If the routing agility feature described in connection with the boutique embodiment is combined with this link-based IP-hopping strategy, the router's next step would

20 be to decrypt the TARP header to determine the destination TARP router for the packet and determine what should be the next hop for the packet. The TARP router would then forward the packet to a random TARP router or the destination TARP router with which the source TARP router has a link-based IP hopping communication established.

25 Figure 8 shows how a client computer 801 and a TARP router 811 can establish a secure session. When client 801 seeks to establish an IHOP session with TARP router 811, the client 801 sends "secure synchronization" request ("SSYN") packet 821 to the TARP router 811. This SYN packet 821 contains the client's 801 authentication token, and may be sent to the router 811 in an encrypted format. The source and destination IP numbers on the packet 821 are the client's

801 current fixed IP address, and a “known” fixed IP address for the router 811. (For security purposes, it may be desirable to reject any packets from outside of the local network that are destined for the router’s known fixed IP address.) Upon receipt and validation of the client’s 801 SSYN packet 821, the router 811 responds by sending an encrypted “secure synchronization acknowledgment” (“SSYN ACK”) 822 to the client 801. This SSYN ACK 822 will contain the transmit and receive hopblocks that the client 801 will use when communicating with the TARP router 811. The client 801 will acknowledge the TARP router’s 811 response packet 822 by generating an encrypted SSYN ACK ACK packet 823 which will be sent from the client’s 801 fixed IP address and to the TARP router’s 811 known fixed IP address. The client 801 will simultaneously generate a SSYN ACK ACK packet; this SSYN ACK packet, referred to as the Secure Session Initiation (SSI) packet 824, will be sent with the first {sender, receiver} IP pair in the client’s transmit table 921 (FIG. 9), as specified in the transmit hopblock provided by the TARP router 811 in the SSYN ACK packet 822. The TARP router 811 will respond to the SSI packet 824 with an SSI ACK packet 825, which will be sent with the first {sender, receiver} IP pair in the TARP router’s transmit table 923. Once these packets have been successfully exchanged, the secure communications session is established, and all further secure communications between the client 801 and the TARP router 811 will be conducted via this secure session, as long as synchronization is maintained. If synchronization is lost, then the client 801 and TARP router 802 may re-establish the secure session by the procedure outlined in Figure 8 and described above.

While the secure session is active, both the client 901 and TARP router 911 (FIG. 9) will maintain their respective transmit tables 921, 923 and receive tables 922, 924, as provided by the TARP router during session synchronization 822. It is important that the sequence of IP pairs in the client’s transmit table 921 be identical to those in the TARP router’s receive table 924; similarly, the sequence of IP pairs in the client’s receive table 922 must be identical to those in the router’s transmit table 923. This is required for the session synchronization to be maintained. The client 901 need maintain only one transmit table 921 and one receive table 922 during the course of the secure session. Each sequential packet sent by the client 901 will employ the next

{send, receive} IP address pair in the transmit table, regardless of TCP or UDP session. The TARP router 911 will expect each packet arriving from the client 901 to bear the next IP address pair shown in its receive table.

5 Since packets can arrive out of order, however, the router 911 can maintain a "look ahead" buffer in its receive table, and will mark previously-received IP pairs as invalid for future packets; any future packet containing an IP pair that is in the look-ahead buffer but is marked as previously received will be discarded. Communications from the TARP router 911 to the client 901 are maintained in an identical manner; in particular, the router 911 will select the next IP address pair from its transmit table 923 when constructing a packet to send to the client 901, and  
10 the client 901 will maintain a look-ahead buffer of expected IP pairs on packets that it is receiving. Each TARP router will maintain separate pairs of transmit and receive tables for each client that is currently engaged in a secure session with or through that TARP router.

15 While clients receive their hopblocks from the first server linking them to the Internet, routers exchange hopblocks. When a router establishes a link-based IP-hopping communication regime with another router, each router of the pair exchanges its transmit hopblock. The transmit hopblock of each router becomes the receive hopblock of the other router. The communication between routers is governed as described by the example of a client sending a packet to the first router.

20 While the above strategy works fine in the IP milieu, many local networks that are connected to the Internet are Ethernet systems. In Ethernet, the IP addresses of the destination devices must be translated into hardware addresses, and vice versa, using known processes ("address resolution protocol," and "reverse address resolution protocol"). However, if the link-based IP-hopping strategy is employed, the correlation process would become explosive and burdensome. An alternative to the link-based IP hopping strategy may be employed within an  
25 Ethernet network. The solution is to provide that the node linking the Internet to the Ethernet (call it the border node) use the link-based IP-hopping communication regime to communicate with nodes outside the Ethernet LAN. Within the Ethernet LAN, each TARP node would have a single IP address which would be addressed in the conventional way. Instead of comparing the



{sender, receiver} IP address pairs to authenticate a packet, the intra-LAN TARP node would use one of the IP header extension fields to do so. Thus, the border node uses an algorithm shared by the intra-LAN TARP node to generate a symbol that is stored in the free field in the IP header, and the intra-LAN TARP node generates a range of symbols based on its prediction of the next expected packet to be received from that particular source IP address. The packet is rejected if it does not fall into the set of predicted symbols (for example, numerical values) or is accepted if it does. Communications from the intra-LAN TARP node to the border node are accomplished in the same manner, though the algorithm will necessarily be different for security reasons. Thus, each of the communicating nodes will generate transmit and receive tables in a similar manner to that of Figure 9; the intra-LAN TARP nodes transmit table will be identical to the border node's receive table, and the intra-LAN TARP node's receive table will be identical to the border node's transmit table.

The algorithm used for IP address-hopping can be any desired algorithm. For example, the algorithm can be a given pseudo-random number generator that generates numbers of the range covering the allowed IP addresses with a given seed. Alternatively, the session participants can assume a certain type of algorithm and specify simply a parameter for applying the algorithm. For example the assumed algorithm could be a particular pseudo-random number generator and the session participants could simply exchange seed values.

Note that there is no permanent physical distinction between the originating and destination terminal nodes. Either device at either end point can initiate a synchronization of the pair. Note also that the authentication/synchronization-request (and acknowledgment) and hopblock-exchange may all be served by a single message so that separate message exchanges may not be required.

As another extension to the stated architecture, multiple physical paths can be used by a client, in order to provide link redundancy and further thwart attempts at denial of service and traffic monitoring. As shown in Figure 10, for example, client 1001 can establish three simultaneous sessions with each of three TARP routers provided by different ISPs 1011, 1012, 1013. As an example, the client 1001 can use three different telephone lines 1021, 1022, 1023 to

connect to the ISPs, or two telephone lines and a cable modem, etc. In this scheme, transmitted packets will be sent in a random fashion among the different physical paths. This architecture provides a high degree of communications redundancy, with improved immunity from denial-of-service attacks and traffic monitoring.

5

## **2. FURTHER EXTENSIONS**

The following describes various extensions to the techniques, systems, and methods described above. As described above, the security of communications occurring between computers in a computer network (such as the Internet, an Ethernet, or others) can be enhanced by using seemingly random source and destination Internet Protocol (IP) addresses for data packets transmitted over the network. This feature prevents eavesdroppers from determining which computers in the network are communicating with each other while permitting the two communicating computers to easily recognize whether a given received data packet is legitimate or not. In one embodiment of the above-described systems, an IP header extension field is used to authenticate incoming packets on an Ethernet.

10

15

Various extensions to the previously described techniques described herein include: (1) use of hopped hardware or "MAC" addresses in broadcast type network; (2) a self-synchronization technique that permits a computer to automatically regain synchronization with a sender; (3) synchronization algorithms that allow transmitting and receiving computers to quickly re-establish synchronization in the event of lost packets or other events; and (4) a fast-packet rejection mechanism for rejecting invalid packets. Any or all of these extensions can be combined with the features described above in any of various ways.

20

### **A. Hardware Address Hopping**

Internet protocol-based communications techniques on a LAN—or across any dedicated physical medium—typically embed the IP packets within lower-level packets, often referred to as "frames." As shown in FIG. 11, for example, a first Ethernet frame 1150 comprises a frame header 1101 and two embedded IP packets IP1 and IP2, while a second Ethernet frame 1160 comprises a different frame header 1104 and a single IP packet IP3. Each frame header generally includes a source hardware address 1101A and a destination hardware address 1101B;

25

other well-known fields in frame headers are omitted from FIG. 11 for clarity. Two hardware nodes communicating over a physical communication channel insert appropriate source and destination hardware addresses to indicate which nodes on the channel or network should receive the frame.

5 It may be possible for a nefarious listener to acquire information about the contents of a frame and/or its communicants by examining frames on a local network rather than (or in addition to) the IP packets themselves. This is especially true in broadcast media, such as Ethernet, where it is necessary to insert into the frame header the hardware address of the machine that generated the frame and the hardware address of the machine to which frame is  
10 being sent. All nodes on the network can potentially "see" all packets transmitted across the network. This can be a problem for secure communications, especially in cases where the communicants do not want for any third party to be able to identify who is engaging in the information exchange. One way to address this problem is to push the address-hopping scheme down to the hardware layer. In accordance with various embodiments of the invention, hardware  
15 addresses are "hopped" in a manner similar to that used to change IP addresses, such that a listener cannot determine which hardware node generated a particular message nor which node is the intended recipient.

FIG. 12A shows a system in which Media Access Control ("MAC") hardware addresses are "hopped" in order to increase security over a network such as an Ethernet. While the  
20 description refers to the exemplary case of an Ethernet environment, the inventive principles are equally applicable to other types of communications media. In the Ethernet case, the MAC address of the sender and receiver are inserted into the Ethernet frame and can be observed by anyone on the LAN who is within the broadcast range for that frame. For secure communications, it becomes desirable to generate frames with MAC addresses that are not  
25 attributable to any specific sender or receiver.

As shown in FIG. 12A, two computer nodes 1201 and 1202 communicate over a communication channel such as an Ethernet. Each node executes one or more application programs 1203 and 1218 that communicate by transmitting packets through communication

software 1204 and 1217, respectively. Examples of application programs include video conferencing, e-mail, word processing programs, telephony, and the like. Communication software 1204 and 1217 can comprise, for example, an OSI layered architecture or "stack" that standardizes various services provided at different levels of functionality.

5           The lowest levels of communication software 1204 and 1217 communicate with hardware components 1206 and 1214 respectively, each of which can include one or more registers 1207 and 1215 that allow the hardware to be reconfigured or controlled in accordance with various communication protocols. The hardware components (an Ethernet network interface card, for example) communicate with each other over the communication medium.

10          Each hardware component is typically pre-assigned a fixed hardware address or MAC number that identifies the hardware component to other nodes on the network. One or more interface drivers control the operation of each card and can, for example, be configured to accept or reject packets from certain hardware addresses. As will be described in more detail below, various embodiments of the inventive principles provide for "hopping" different addresses using one or

15          more algorithms and one or more moving windows that track a range of valid addresses to validate received packets. Packets transmitted according to one or more of the inventive principles will be generally referred to as "secure" packets or "secure communications" to differentiate them from ordinary data packets that are transmitted in the clear using ordinary, machine-correlated addresses.

20           One straightforward method of generating non-attributable MAC addresses is an extension of the IP hopping scheme. In this scenario, two machines on the same LAN that desire to communicate in a secure fashion exchange random-number generators and seeds, and create sequences of quasi-random MAC addresses for synchronized hopping. The implementation and synchronization issues are then similar to that of IP hopping.

25           This approach, however, runs the risk of using MAC addresses that are currently active on the LAN—which, in turn, could interrupt communications for those machines. Since an Ethernet MAC address is at present 48 bits in length, the chance of randomly misusing an active MAC address is actually quite small. However, if that figure is multiplied by a large number of

nodes (as would be found on an extensive LAN), by a large number of frames (as might be the case with packet voice or streaming video), and by a large number of concurrent Virtual Private Networks (VPNs), then the chance that a non-secure machine's MAC address could be used in an address-hopped frame can become non-trivial. In short, any scheme that runs even a small risk of interrupting communications for other machines on the LAN is bound to receive resistance from prospective system administrators. Nevertheless, it is technically feasible, and can be implemented without risk on a LAN on which there is a small number of machines, or if all of the machines on the LAN are engaging in MAC-hopped communications.

Synchronized MAC address hopping may incur some overhead in the course of session establishment, especially if there are multiple sessions or multiple nodes involved in the communications. A simpler method of randomizing MAC addresses is to allow each node to receive and process *every* incident frame on the network. Typically, each network interface driver will check the destination MAC address in the header of every incident frame to see if it matches that machine's MAC address; if there is no match, then the frame is discarded. In one embodiment, however, these checks can be disabled, and every incident packet is passed to the TARP stack for processing. This will be referred to as "promiscuous" mode, since every incident frame is processed. Promiscuous mode allows the sender to use completely random, unsynchronized MAC addresses, since the destination machine is guaranteed to process the frame. The decision as to whether the packet was truly intended for that machine is handled by the TARP stack, which checks the source and destination IP addresses for a match in its IP synchronization tables. If no match is found, the packet is discarded; if there is a match, the packet is unwrapped, the inner header is evaluated, and if the inner header indicates that the packet is destined for that machine then the packet is forwarded to the IP stack—otherwise it is discarded.

One disadvantage of purely-random MAC address hopping is its impact on processing overhead; that is, since every incident frame must be processed, the machine's CPU is engaged considerably more often than if the network interface driver is discriminating and rejecting packets unilaterally. A compromise approach is to select either a single fixed MAC address or a

small number of MAC addresses (e.g., one for each virtual private network on an Ethernet) to use for MAC-hopped communications, regardless of the actual recipient for which the message is intended. In this mode, the network interface driver can check each incident frame against one (or a few) pre-established MAC addresses, thereby freeing the CPU from the task of physical-layer packet discrimination. This scheme does not betray any useful information to an interloper on the LAN; in particular, every secure packet can already be identified by a unique packet type in the outer header. However, since all machines engaged in secure communications would either be using the same MAC address, or be selecting from a small pool of predetermined MAC addresses, the association between a specific machine and a specific MAC address is effectively broken.

In this scheme, the CPU will be engaged more often than it would be in non-secure communications (or in synchronized MAC address hopping), since the network interface driver cannot always unilaterally discriminate between secure packets that are destined for that machine, and secure packets from other VPNs. However, the non-secure traffic is easily eliminated at the network interface, thereby reducing the amount of processing required of the CPU. There are boundary conditions where these statements would not hold, of course—e.g., if *all* of the traffic on the LAN is secure traffic, then the CPU would be engaged to the same degree as it is in the purely-random address hopping case; alternatively, if each VPN on the LAN uses a different MAC address, then the network interface can perfectly discriminate secure frames destined for the local machine from those constituting other VPNs. These are engineering tradeoffs that might be best handled by providing administrative options for the users when installing the software and/or establishing VPNs.

Even in this scenario, however, there still remains a slight risk of selecting MAC addresses that are being used by one or more nodes on the LAN. One solution to this problem is to formally assign one address or a range of addresses for use in MAC-hopped communications. This is typically done via an assigned numbers registration authority; e.g., in the case of Ethernet, MAC address ranges are assigned to vendors by the Institute of Electrical and Electronics Engineers (IEEE). A formally-assigned range of addresses would ensure that secure

frames do not conflict with any properly-configured and properly-functioning machines on the LAN.

Reference will now be made to FIGS. 12A and 12B in order to describe the many combinations and features that follow the inventive principles. As explained above, two computer nodes 1201 and 1202 are assumed to be communicating over a network or communication medium such as an Ethernet. A communication protocol in each node (1204 and 1217, respectively) contains a modified element 1205 and 1216 that performs certain functions that deviate from the standard communication protocols. In particular, computer node 1201 implements a first "hop" algorithm 1208X that selects seemingly random source and destination IP addresses (and, in one embodiment, seemingly random IP header discriminator fields) in order to transmit each packet to the other computer node. For example, node 1201 maintains a transmit table 1208 containing triplets of source (S), destination (D), and discriminator fields (DS) that are inserted into outgoing IP packet headers. The table is generated through the use of an appropriate algorithm (e.g., a random number generator that is seeded with an appropriate seed) that is known to the recipient node 1202. As each new IP packet is formed, the next sequential entry out of the sender's transmit table 1208 is used to populate the IP source, IP destination, and IP header extension field (e.g., discriminator field). It will be appreciated that the transmit table need not be created in advance but could instead be created on-the-fly by executing the algorithm when each packet is formed.

At the receiving node 1202, the same IP hop algorithm 1222X is maintained and used to generate a receive table 1222 that lists valid triplets of source IP address, destination IP address, and discriminator field. This is shown by virtue of the first five entries of transmit table 1208 matching the second five entries of receive table 1222. (The tables may be slightly offset at any particular time due to lost packets, misordered packets, or transmission delays). Additionally, node 1202 maintains a receive window W3 that represents a list of valid IP source, IP destination, and discriminator fields that will be accepted when received as part of an incoming IP packet. As packets are received, window W3 slides down the list of valid entries, such that the possible valid entries change over time. Two packets that arrive out of order but are

nevertheless matched to entries within window W3 will be accepted; those falling outside of window W3 will be rejected as invalid. The length of window W3 can be adjusted as necessary to reflect network delays or other factors.

Node 1202 maintains a similar transmit table 1221 for creating IP packets and frames  
5 destined for node 1201 using a potentially different hopping algorithm 1221X, and node 1201 maintains a matching receive table 1209 using the same algorithm 1209X. As node 1202 transmits packets to node 1201 using seemingly random IP source, IP destination, and/or discriminator fields, node 1201 matches the incoming packet values to those falling within window W1 maintained in its receive table. In effect, transmit table 1208 of node 1201 is  
10 synchronized (i.e., entries are selected in the same order) to receive table 1222 of receiving node 1202. Similarly, transmit table 1221 of node 1202 is synchronized to receive table 1209 of node 1201. It will be appreciated that although a common algorithm is shown for the source, destination and discriminator fields in FIG. 12A (using, e.g., a different seed for each of the three fields), an entirely different algorithm could in fact be used to establish values for each of these  
15 fields. It will also be appreciated that one or two of the fields can be "hopped" rather than all three as illustrated.

In accordance with another aspect of the invention, hardware or "MAC" addresses are hopped instead of or in addition to IP addresses and/or the discriminator field in order to improve security in a local area or broadcast-type network. To that end, node 1201 further maintains a  
20 transmit table 1210 using a transmit algorithm 1210X to generate source and destination hardware addresses that are inserted into frame headers (e.g., fields 1101A and 1101B in FIG. 11) that are synchronized to a corresponding receive table 1224 at node 1202. Similarly, node 1202 maintains a different transmit table 1223 containing source and destination hardware addresses that is synchronized with a corresponding receive table 1211 at node 1201. In this  
25 manner, outgoing hardware frames appear to be originating from and going to completely random nodes on the network, even though each recipient can determine whether a given packet is intended for it or not. It will be appreciated that the hardware hopping feature can be



implemented at a different level in the communications protocol than the IP hopping feature (e.g., in a card driver or in a hardware card itself to improve performance).

FIG. 12B shows three different embodiments or modes that can be employed using the aforementioned principles. In a first mode referred to as "promiscuous" mode, a common hardware address (e.g., a fixed address for source and another for destination) or else a completely random hardware address is used by all nodes on the network, such that a particular packet cannot be attributed to any one node. Each node must initially accept all packets containing the common (or random) hardware address and inspect the IP addresses or discriminator field to determine whether the packet is intended for that node. In this regard, either the IP addresses or the discriminator field or both can be varied in accordance with an algorithm as described above. As explained previously, this may increase each node's overhead since additional processing is involved to determine whether a given packet has valid source and destination hardware addresses.

In a second mode referred to as "promiscuous per VPN" mode, a small set of fixed hardware addresses are used, with a fixed source/destination hardware address used for all nodes communicating over a virtual private network. For example, if there are six nodes on an Ethernet, and the network is to be split up into two private virtual networks such that nodes on one VPN can communicate with only the other two nodes on its own VPN, then two sets of hardware addresses could be used: one set for the first VPN and a second set for the second VPN. This would reduce the amount of overhead involved in checking for valid frames since only packets arriving from the designated VPN would need to be checked. IP addresses and one or more discriminator fields could still be hopped as before for secure communication within the VPN. Of course, this solution compromises the anonymity of the VPNs (i.e., an outsider can easily tell what traffic belongs in which VPN, though he cannot correlate it to a specific machine/person). It also requires the use of a discriminator field to mitigate the vulnerability to certain types of DoS attacks. (For example, without the discriminator field, an attacker on the LAN could stream frames containing the MAC addresses being used by the VPN; rejecting those

frames could lead to excessive processing overhead. The discriminator field would provide a low-overhead means of rejecting the false packets.)

In a third mode referred to as "hardware hopping" mode, hardware addresses are varied as illustrated in FIG. 12A, such that hardware source and destination addresses are changed constantly in order to provide non-attributable addressing. Variations on these embodiments are of course possible, and the invention is not intended to be limited in any respect by these illustrative examples.

### **B. Extending the Address Space**

Address hopping provides security and privacy. However, the level of protection is limited by the number of addresses in the blocks being hopped. A hopblock denotes a field or fields modulated on a packet-wise basis for the purpose of providing a VPN. For instance, if two nodes communicate with IP address hopping using hopblocks of 4 addresses (2 bits) each, there would be 16 possible address-pair combinations. A window of size 16 would result in most address pairs being accepted as valid most of the time. This limitation can be overcome by using a discriminator field in addition to or instead of the hopped address fields. The discriminator field would be hopped in exactly the same fashion as the address fields and it would be used to determine whether a packet should be processed by a receiver.

Suppose that two clients, each using four-bit hopblocks, would like the same level of protection afforded to clients communicating via IP hopping between two A blocks (24 address bits eligible for hopping). A discriminator field of 20 bits, used in conjunction with the 4 address bits eligible for hopping in the IP address field, provides this level of protection. A 24-bit discriminator field would provide a similar level of protection if the address fields were not hopped or ignored. Using a discriminator field offers the following advantages: (1) an arbitrarily high level of protection can be provided, and (2) address hopping is unnecessary to provide protection. This may be important in environments where address hopping would cause routing problems.

### C. Synchronization Techniques

It is generally assumed that once a sending node and receiving node have exchanged algorithms and seeds (or similar information sufficient to generate quasi-random source and destination tables), subsequent communication between the two nodes will proceed smoothly.

5 Realistically, however, two nodes may lose synchronization due to network delays or outages, or other problems. Consequently, it is desirable to provide means for re-establishing synchronization between nodes in a network that have lost synchronization.

10 One possible technique is to require that each node provide an acknowledgment upon successful receipt of each packet and, if no acknowledgment is received within a certain period of time, to re-send the unacknowledged packet. This approach, however, drives up overhead costs and may be prohibitive in high-throughput environments such as streaming video or audio, for example.

15 A different approach is to employ an automatic synchronizing technique that will be referred to herein as "self-synchronization." In this approach, synchronization information is embedded into each packet, thereby enabling the receiver to re-synchronize itself upon receipt of a single packet if it determines that it has lost synchronization with the sender. (If communications are already in progress, and the receiver determines that it is still in sync with the sender, then there is no need to re-synchronize.) A receiver could detect that it was out of synchronization by, for example, employing a "dead-man" timer that expires after a certain  
20 period of time, wherein the timer is reset with each valid packet. A time stamp could be hashed into the public sync field (see below) to preclude packet-retry attacks.

In one embodiment, a "sync field" is added to the header of each packet sent out by the sender. This sync field could appear in the clear or as part of an encrypted portion of the packet. Assuming that a sender and receiver have selected a random-number generator (RNG) and seed  
25 value, this combination of RNG and seed can be used to generate a random-number sequence (RNS). The RNS is then used to generate a sequence of source/destination IP pairs (and, if desired, discriminator fields and hardware source and destination addresses), as described above. It is not necessary, however, to generate the entire sequence (or the first N-1 values) in order to

generate the Nth random number in the sequence; if the sequence index N is known, the random value corresponding to that index can be directly generated (see below). Different RNGs (and seeds) with different fundamental periods could be used to generate the source and destination IP sequences, but the basic concepts would still apply. For the sake of simplicity, the following discussion will assume that IP source and destination address pairs (only) are hopped using a single RNG sequencing mechanism.

In accordance with a “self-synchronization” feature, a sync field in each packet header provides an index (i.e., a sequence number) into the RNS that is being used to generate IP pairs. Plugging this index into the RNG that is being used to generate the RNS yields a specific random number value, which in turn yields a specific IP pair. That is, an IP pair can be generated directly from knowledge of the RNG, seed, and index number; it is not necessary, in this scheme, to generate the entire sequence of random numbers that precede the sequence value associated with the index number provided.

Since the communicants have presumably previously exchanged RNGs and seeds, the only new information that must be provided in order to generate an IP pair is the sequence number. If this number is provided by the sender in the packet header, then the receiver need only plug this number into the RNG in order to generate an IP pair – and thus verify that the IP pair appearing in the header of the packet is valid. In this scheme, if the sender and receiver lose synchronization, the receiver can immediately re-synchronize upon receipt of a single packet by simply comparing the IP pair in the packet header to the IP pair generated from the index number. Thus, synchronized communications can be resumed upon receipt of a single packet, making this scheme ideal for multicast communications. Taken to the extreme, it could obviate the need for synchronization tables entirely; that is, the sender and receiver could simply rely on the index number in the sync field to validate the IP pair on each packet, and thereby eliminate the tables entirely.

The aforementioned scheme may have some inherent security issues associated with it — namely, the placement of the sync field. If the field is placed in the outer header, then an interloper could observe the values of the field and their relationship to the IP stream. This could

potentially compromise the algorithm that is being used to generate the IP-address sequence, which would compromise the security of the communications. If, however, the value is placed in the inner header, then the sender must decrypt the inner header before it can extract the sync value and validate the IP pair; this opens up the receiver to certain types of denial-of-service (DoS) attacks, such as packet replay. That is, if the receiver must decrypt a packet before it can validate the IP pair, then it could potentially be forced to expend a significant amount of processing on decryption if an attacker simply retransmits previously valid packets. Other attack methodologies are possible in this scenario.

10 A possible compromise between algorithm security and processing speed is to split up the sync value between an inner (encrypted) and outer (unencrypted) header. That is, if the sync value is sufficiently long, it could potentially be split into a rapidly-changing part that can be viewed in the clear, and a fixed (or very slowly changing) part that must be protected. The part that can be viewed in the clear will be called the "public sync" portion and the part that must be protected will be called the "private sync" portion.

15 Both the public sync and private sync portions are needed to generate the complete sync value. The private portion, however, can be selected such that it is fixed or will change only occasionally. Thus, the private sync value can be stored by the recipient, thereby obviating the need to decrypt the header in order to retrieve it. If the sender and receiver have previously agreed upon the frequency with which the private part of the sync will change, then the receiver  
20 can selectively decrypt a single header in order to extract the new private sync if the communications gap that has led to lost synchronization has exceeded the lifetime of the previous private sync. This should not represent a burdensome amount of decryption, and thus should not open up the receiver to denial-of-service attack simply based on the need to occasionally decrypt a single header.

25 One implementation of this is to use a hashing function with a one-to-one mapping to generate the private and public sync portions from the sync value. This implementation is shown in FIG. 13, where (for example) a first ISP 1302 is the sender and a second ISP 1303 is the receiver. (Other alternatives are possible from FIG. 13.) A transmitted packet comprises a public

or “outer” header 1305 that is not encrypted, and a private or “inner” header 1306 that is encrypted using for example a link key. Outer header 1305 includes a public sync portion while inner header 1306 contains the private sync portion. A receiving node decrypts the inner header using a decryption function 1307 in order to extract the private sync portion. This step is necessary only if the lifetime of the currently buffered private sync has expired. (If the currently-buffered private sync is still valid, then it is simply extracted from memory and “added” (which could be an inverse hash) to the public sync, as shown in step 1308.) The public and decrypted private sync portions are combined in function 1308 in order to generate the combined sync 1309. The combined sync (1309) is then fed into the RNG (1310) and compared to the IP address pair (1311) to validate or reject the packet.

An important consideration in this architecture is the concept of “future” and “past” where the public sync values are concerned. Though the sync values, themselves, should be random to prevent spoofing attacks, it may be important that the receiver be able to quickly identify a sync value that has already been sent — even if the packet containing that sync value was never actually received by the receiver. One solution is to hash a time stamp or sequence number into the public sync portion, which could be quickly extracted, checked, and discarded, thereby validating the public sync portion itself.

In one embodiment, packets can be checked by comparing the source/destination IP pair generated by the sync field with the pair appearing in the packet header. If (1) they match, (2) the time stamp is valid, and (3) the dead-man timer has expired, then re-synchronization occurs; otherwise, the packet is rejected. If enough processing power is available, the dead-man timer and synchronization tables can be avoided altogether, and the receiver would simply resynchronize (e.g., validate) on every packet.

The foregoing scheme may require large-integer (e.g., 160-bit) math, which may affect its implementation. Without such large-integer registers, processing throughput would be affected, thus potentially affecting security from a denial-of-service standpoint. Nevertheless, as large-integer math processing features become more prevalent, the costs of implementing such a feature will be reduced.

#### D. Other Synchronization Schemes

As explained above, if  $W$  or more consecutive packets are lost between a transmitter and receiver in a VPN (where  $W$  is the window size), the receiver's window will not have been updated and the transmitter will be transmitting packets not in the receiver's window. The sender and receiver will not recover synchronization until perhaps the random pairs in the window are repeated by chance. Therefore, there is a need to keep a transmitter and receiver in synchronization whenever possible and to re-establish synchronization whenever it is lost.

A "checkpoint" scheme can be used to regain synchronization between a sender and a receiver that have fallen out of synchronization. In this scheme, a checkpoint message comprising a random IP address pair is used for communicating synchronization information. In one embodiment, two messages are used to communicate synchronization information between a sender and a recipient:

1. SYNC\_REQ is a message used by the sender to indicate that it wants to synchronize; and
2. SYNC\_ACK is a message used by the receiver to inform the transmitter that it has been synchronized.

According to one variation of this approach, both the transmitter and receiver maintain three checkpoints (see FIG. 14):

1. In the transmitter, ckpt\_o ("checkpoint old") is the IP pair that was used to re-send the last SYNC\_REQ packet to the receiver. In the receiver, ckpt\_o ("checkpoint old") is the IP pair that receives repeated SYNC\_REQ packets from the transmitter.
2. In the transmitter, ckpt\_n ("checkpoint new") is the IP pair that will be used to send the next SYNC\_REQ packet to the receiver. In the receiver, ckpt\_n ("checkpoint new") is the IP pair that receives a new SYNC\_REQ packet from the transmitter and which causes the receiver's window to be re-aligned, ckpt\_o set to ckpt\_n, a new ckpt\_n to be generated and a new ckpt\_r to be generated.
3. In the transmitter, ckpt\_r is the IP pair that will be used to send the next SYNC\_ACK packet to the receiver. In the receiver, ckpt\_r is the IP pair that receives a new

SYNC\_ACK packet from the transmitter and which causes a new ckpt\_n to be generated. Since SYNC\_ACK is transmitted from the receiver ISP to the sender ISP, the transmitter ckpt\_r refers to the ckpt\_r of the receiver and the receiver ckpt\_r refers to the ckpt\_r of the transmitter (see FIG. 14).

- 5 When a transmitter initiates synchronization, the IP pair it will use to transmit the next data packet is set to a predetermined value and when a receiver first receives a SYNC\_REQ, the receiver window is updated to be centered on the transmitter's next IP pair. This is the primary mechanism for checkpoint synchronization.

10 Synchronization can be initiated by a packet counter (e.g., after every N packets transmitted, initiate a synchronization) or by a timer (every S seconds, initiate a synchronization) or a combination of both. See FIG. 15. From the transmitter's perspective, this technique operates as follows: (1) Each transmitter periodically transmits a "sync request" message to the receiver to make sure that it is in sync. (2) If the receiver is still in sync, it sends back a "sync ack" message. (If this works, no further action is necessary). (3) If no "sync ack" has been  
15 received within a period of time, the transmitter retransmits the sync request again. If the transmitter reaches the next checkpoint without receiving a "sync ack" response, then synchronization is broken, and the transmitter should stop transmitting. The transmitter will continue to send sync\_reqs until it receives a sync\_ack , at which point transmission is reestablished.

20 From the receiver's perspective, the scheme operates as follows: (1) when it receives a "sync request" request from the transmitter, it advances its window to the next checkpoint position (even skipping pairs if necessary), and sends a "sync ack" message to the transmitter. If sync was never lost, then the "jump ahead" really just advances to the next available pair of addresses in the table (i.e., normal advancement).

25 If an interloper intercepts the "sync request" messages and tries to interfere with communication by sending new ones, it will be ignored if the synchronization has been established or it it will actually help to re-establish synchronization.



A window is realigned whenever a re-synchronization occurs. This realignment entails updating the receiver's window to straddle the address pairs used by the packet transmitted immediately after the transmission of the SYNC\_REQ packet. Normally, the transmitter and receiver are in synchronization with one another. However, when network events occur, the receiver's window may have to be advanced by many steps during resynchronization. In this case, it is desirable to move the window ahead without having to step through the intervening random numbers sequentially. (This feature is also desirable for the auto-sync approach discussed above).

#### E. Random Number Generator with a Jump-Ahead capability

An attractive method for generating randomly hopped addresses is to use identical random number generators in the transmitter and receiver and advance them as packets are transmitted and received. There are many random number generation algorithms that could be used. Each one has strengths and weaknesses for address hopping applications.

Linear congruential random number generators (LCRs) are fast, simple and well characterized random number generators that can be made to jump ahead  $n$  steps efficiently. An LCR generates random numbers  $X_1, X_2, X_3 \dots X_k$  starting with seed  $X_0$  using a recurrence

$$X_i = (a X_{i-1} + b) \text{ mod } c, \quad (1)$$

where  $a, b$  and  $c$  define a particular LCR. Another expression for  $X_i$ ,

$$X_i = ((a^i(X_0 + b) - b) / (a - 1)) \text{ mod } c \quad (2)$$

enables the jump-ahead capability. The factor  $a^i$  can grow very large even for modest  $i$  if left unfettered. Therefore some special properties of the modulo operation can be used to control the size and processing time required to compute (2). (2) can be rewritten as:

$$X_i = (a^i (X_0(a-1) + b) - b) / (a-1) \text{ mod } c. \quad (3)$$

It can be shown that:

$$(a^i (X_0(a-1) + b) - b) / (a-1) \text{ mod } c = ((a^i \text{ mod } ((a-1)c) (X_0(a-1) + b) - b) / (a-1)) \text{ mod } c \quad (4).$$

$(X_0(a-1) + b)$  can be stored as  $(X_0(a-1) + b) \text{ mod } c$ ,  $b$  as  $b \text{ mod } c$  and compute  $a^i \text{ mod } ((a-1)c)$  (this requires  $O(\log(i))$  steps).

A practical implementation of this algorithm would jump a fixed distance,  $n$ , between synchronizations; this is tantamount to synchronizing every  $n$  packets. The window would commence  $n$  IP pairs from the start of the previous window. Using  $X_j^w$ , the random number at the  $j^{\text{th}}$  checkpoint, as  $X_0$  and  $n$  as  $i$ , a node can store  $a^n \text{ mod } ((a-1)c)$  once per LCR and set

$$5 \quad X_{j+1}^w = X_{n(j+1)} = ((a^n \text{ mod } ((a-1)c) (X_j^w (a-1) + b) - b) / (a-1)) \text{ mod } c, \quad (5)$$

to generate the random number for the  $j+1^{\text{th}}$  synchronization. Using this construction, a node could jump ahead an arbitrary (but fixed) distance between synchronizations in a constant amount of time (independent of  $n$ ).

10 Pseudo-random number generators, in general, and LCRs, in particular, will eventually repeat their cycles. This repetition may present vulnerability in the IP hopping scheme. An adversary would simply have to wait for a repeat to predict future sequences. One way of coping with this vulnerability is to create a random number generator with a known long cycle. A random sequence can be replaced by a new random number generator before it repeats. LCRs can be constructed with known long cycles. This is not currently true of many random number  
15 generators.

Random number generators can be cryptographically insecure. An adversary can derive the RNG parameters by examining the output or part of the output. This is true of LCGs. This vulnerability can be mitigated by incorporating an encryptor, designed to scramble the output as part of the random number generator. The random number generator prevents an adversary from  
20 mounting an attack—e.g., a known plaintext attack—against the encryptor.

#### **F. Random Number Generator Example**

Consider a RNG where  $a=31, b=4$  and  $c=15$ . For this case equation (1) becomes:

$$X_i = (31 X_{i-1} + 4) \text{ mod } 15. \quad (6)$$

25 If one sets  $X_0=1$ , equation (6) will produce the sequence 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 0, 4, 8, 12. This sequence will repeat indefinitely. For a jump ahead of 3 numbers in this sequence  $a^n = 31^3 = 29791$ ,  $c*(a-1) = 15*30 = 450$  and  $a^n \text{ mod } ((a-1)c) = 31^3 \text{ mod } (15*30) = 29791 \text{ mod } (450) = 91$ . Equation (5) becomes:

$$((91 (X_i 30 + 4) - 4) / 30) \text{ mod } 15 \quad (7).$$

Table 1 shows the jump ahead calculations from (7) . The calculations start at 5 and jump ahead 3.

TABLE 1

I	$X_i$	$(X_i \cdot 30 + 4)$	$91(X_i \cdot 30 + 4) - 4$	$((91(X_i \cdot 30 + 4) - 4) / 30)$	$X_{i+3}$
1	5	154	14010	467	2
4	2	64	5820	194	14
7	14	424	38580	1286	11
10	11	334	30390	1013	8
13	8	244	22200	740	5

### G. Fast Packet Filter

Address hopping VPNs must rapidly determine whether a packet has a valid header and thus requires further processing, or has an invalid header (a hostile packet) and should be immediately rejected. Such rapid determinations will be referred to as "fast packet filtering." This capability protects the VPN from attacks by an adversary who streams hostile packets at the receiver at a high rate of speed in the hope of saturating the receiver's processor (a so-called "denial of service" attack). Fast packet filtering is an important feature for implementing VPNs on shared media such as Ethernet.

Assuming that all participants in a VPN share an unassigned "A" block of addresses, one possibility is to use an experimental "A" block that will never be assigned to any machine that is not address hopping on the shared medium. "A" blocks have a 24 bits of address that can be hopped as opposed to the 8 bits in "C" blocks. In this case a hopblock will be the "A" block. The use of the experimental "A" block is a likely option on an Ethernet because:

1. The addresses have no validity outside of the Ethernet and will not be routed out to a valid outside destination by a gateway.
2. There are  $2^{24}$  (~16 million) addresses that can be hopped within each "A" block. This yields >280 trillion possible address pairs making it very unlikely that an adversary would guess a

valid address. It also provides acceptably low probability of collision between separate VPNs (all VPNs on a shared medium independently generate random address pairs from the same "A" block).

3. The packets will not be received by someone on the Ethernet who is not on a VPN (unless the machine is in promiscuous mode) minimizing impact on non-VPN computers.

The Ethernet example will be used to describe one implementation of fast packet filtering. The ideal algorithm would quickly examine a packet header, determine whether the packet is hostile, and reject any hostile packets or determine which active IP pair the packet header matches. The problem is a classical associative memory problem. A variety of techniques have been developed to solve this problem (hashing, B-trees etc). Each of these approaches has its strengths and weaknesses. For instance, hash tables can be made to operate quite fast in a statistical sense, but can occasionally degenerate into a much slower algorithm. This slowness can persist for a period of time. Since there is a need to discard hostile packets quickly at all times, hashing would be unacceptable.

#### H. Presence Vector Algorithm

A presence vector is a bit vector of length  $2^n$  that can be indexed by  $n$ -bit numbers (each ranging from 0 to  $2^n-1$ ). One can indicate the presence of  $k$   $n$ -bit numbers (not necessarily unique), by setting the bits in the presence vector indexed by each number to 1. Otherwise, the bits in the presence vector are 0. An  $n$ -bit number,  $x$ , is one of the  $k$  numbers if and only if the  $x^{\text{th}}$  bit of the presence vector is 1. A fast packet filter can be implemented by indexing the presence vector and looking for a 1, which will be referred to as the "test."

For example, suppose one wanted to represent the number 135 using a presence vector. The 135<sup>th</sup> bit of the vector would be set. Consequently, one could very quickly determine whether an address of 135 was valid by checking only one bit: the 135<sup>th</sup> bit. The presence vectors could be created in advance corresponding to the table entries for the IP addresses. In effect, the incoming addresses can be used as indices into a long vector, making comparisons very fast. As each RNG generates a new address, the presence vector is updated to reflect the

information. As the window moves, the presence vector is updated to zero out addresses that are no longer valid.

There is a trade-off between efficiency of the test and the amount of memory required for storing the presence vector(s). For instance, if one were to use the 48 bits of hopping addresses as an index, the presence vector would have to be 35 terabytes. Clearly, this is too large for practical purposes. Instead, the 48 bits can be divided into several smaller fields. For instance, one could subdivide the 48 bits into four 12-bit fields (see FIG. 16). This reduces the storage requirement to 2048 bytes at the expense of occasionally having to process a hostile packet. In effect, instead of one long presence vector, the decomposed address portions must match all four shorter presence vectors before further processing is allowed. (If the first part of the address portion doesn't match the first presence vector, there is no need to check the remaining three presence vectors).

A presence vector will have a 1 in the  $y^{\text{th}}$  bit if and only if one or more addresses with a corresponding field of  $y$  are active. An address is active only if each presence vector indexed by the appropriate sub-field of the address is 1.

Consider a window of 32 active addresses and 3 checkpoints. A hostile packet will be rejected by the indexing of one presence vector more than 99% of the time. A hostile packet will be rejected by the indexing of all 4 presence vectors more than 99.9999995% of the time. On average, hostile packets will be rejected in less than 1.02 presence vector index operations.

The small percentage of hostile packets that pass the fast packet filter will be rejected when matching pairs are not found in the active window or are active checkpoints. Hostile packets that serendipitously match a header will be rejected when the VPN software attempts to decrypt the header. However, these cases will be extremely rare. There are many other ways this method can be configured to arbitrate the space/speed tradeoffs.

### **I. Further Synchronization Enhancements**

A slightly modified form of the synchronization techniques described above can be employed. The basic principles of the previously described checkpoint synchronization scheme remain unchanged. The actions resulting from the reception of the checkpoints are, however,

slightly different. In this variation, the receiver will maintain between OoO (“Out of Order”) and  $2 \times \text{WINDOW\_SIZE} + \text{OoO}$  active addresses ( $1 \leq \text{OoO} \leq \text{WINDOW\_SIZE}$  and  $\text{WINDOW\_SIZE} \geq 1$ ). OoO and WINDOW\_SIZE are engineerable parameters, where OoO is the minimum number of addresses needed to accommodate lost packets due to events in the network or out of order arrivals and WINDOW\_SIZE is the number of packets transmitted before a SYNC\_REQ is issued. FIG. 17 depicts a storage array for a receiver’s active addresses.

The receiver starts with the first  $2 \times \text{WINDOW\_SIZE}$  addresses loaded and active (ready to receive data). As packets are received, the corresponding entries are marked as “used” and are no longer eligible to receive packets. The transmitter maintains a packet counter, initially set to 0, containing the number of data packets transmitted since the last *initial* transmission of a SYNC\_REQ for which SYNC\_ACK has been received. When the transmitter packet counter equals WINDOW\_SIZE, the transmitter generates a SYNC\_REQ and does its initial transmission. When the receiver receives a SYNC\_REQ corresponding to its current CKPT\_N, it generates the next WINDOW\_SIZE addresses and starts loading them in order starting at the first location after the last active address wrapping around to the beginning of the array after the end of the array has been reached. The receiver’s array might look like FIG. 18 when a SYNC\_REQ has been received. In this case a couple of packets have been either lost or will be received out of order when the SYNC\_REQ is received.

FIG. 19 shows the receiver’s array after the new addresses have been generated. If the transmitter does not receive a SYNC\_ACK, it will re-issue the SYNC\_REQ at regular intervals. When the transmitter receives a SYNC\_ACK, the packet counter is decremented by WINDOW\_SIZE. If the packet counter reaches  $2 \times \text{WINDOW\_SIZE} - \text{OoO}$  then the transmitter ceases sending data packets until the appropriate SYNC\_ACK is finally received. The transmitter then resumes sending data packets. Future behavior is essentially a repetition of this initial cycle. The advantages of this approach are:

1. There is no need for an efficient jump ahead in the random number generator,
2. No packet is ever transmitted that does not have a corresponding entry in the receiver side

3. No timer based re-synchronization is necessary. This is a consequence of 2.
4. The receiver will always have the ability to accept data messages transmitted within OoO messages of the most recently transmitted message.

#### **J. Distributed Transmission Path Variant**

5 Another embodiment incorporating various inventive principles is shown in FIG. 20. In this embodiment, a message transmission system includes a first computer 2001 in communication with a second computer 2002 through a network 2011 of intermediary computers. In one variant of this embodiment, the network includes two edge routers 2003 and 2004 each of which is linked to a plurality of Internet Service Providers (ISPs) 2005 through 10 2010. Each ISP is coupled to a plurality of other ISPs in an arrangement as shown in FIG. 20, which is a representative configuration only and is not intended to be limiting. Each connection between ISPs is labeled in FIG. 20 to indicate a specific physical transmission path (e.g., AD is a physical path that links ISP A (element 2005) to ISP D (element 2008)). Packets arriving at each edge router are selectively transmitted to one of the ISPs to which the router is attached on the 15 basis of a randomly or quasi-randomly selected basis.

As shown in FIG. 21, computer 2001 or edge router 2003 incorporates a plurality of link transmission tables 2100 that identify, for each potential transmission path through the network, valid sets of IP addresses that can be used to transmit the packet. For example, AD table 2101 contains a plurality of IP source/destination pairs that are randomly or quasi-randomly generated. 20 When a packet is to be transmitted from first computer 2001 to second computer 2002, one of the link tables is randomly (or quasi-randomly) selected, and the next valid source/destination address pair from that table is used to transmit the packet through the network. If path AD is randomly selected, for example, the next source/destination IP address pair (which is pre-determined to transmit between ISP A (element 2005) and ISP B (element 2008)) is used to 25 transmit the packet. If one of the transmission paths becomes degraded or inoperative, that link table can be set to a "down" condition as shown in table 2105, thus preventing addresses from being selected from that table. Other transmission paths would be unaffected by this broken link.

### 3. CONTINUATION-IN-PART IMPROVEMENTS

The following describes various improvements and features that can be applied to the embodiments described above. The improvements include: (1) a load balancer that distributes packets across different transmission paths according to transmission path quality; (2) a DNS proxy server that transparently creates a virtual private network in response to a domain name inquiry; (3) a large-to-small link bandwidth management feature that prevents denial-of-service attacks at system chokepoints; (4) a traffic limiter that regulates incoming packets by limiting the rate at which a transmitter can be synchronized with a receiver; and (5) a signaling synchronizer that allows a large number of nodes to communicate with a central node by partitioning the communication function between two separate entities. Each is discussed separately below.

#### A. Load Balancer

Various embodiments described above include a system in which a transmitting node and a receiving node are coupled through a plurality of transmission paths, and wherein successive packets are distributed quasi-randomly over the plurality of paths. See, for example, FIGS. 20 and 21 and accompanying description. The improvement extends this basic concept to encompass distributing packets across different paths in such a manner that the loads on the paths are generally balanced according to transmission link quality.

In one embodiment, a system includes a transmitting node and a receiving node that are linked via a plurality of transmission paths having potentially varying transmission quality. Successive packets are transmitted over the paths based on a weight value distribution function for each path. The rate that packets will be transmitted over a given path can be different for each path. The relative "health" of each transmission path is monitored in order to identify paths that have become degraded. In one embodiment, the health of each path is monitored in the transmitter by comparing the number of packets transmitted to the number of packet acknowledgements received. Each transmission path may comprise a physically separate path (e.g., via dial-up phone line, computer network, router, bridge, or the like), or may comprise logically separate paths contained within a broadband communication medium (e.g., separate



channels in an FDM, TDM, CDMA, or other type of modulated or unmodulated transmission link).

When the transmission quality of a path falls below a predetermined threshold and there are other paths that can transmit packets, the transmitter changes the weight value used for that path, making it less likely that a given packet will be transmitted over that path. The weight will preferably be set no lower than a minimum value that keeps nominal traffic on the path. The weights of the other available paths are altered to compensate for the change in the affected path. When the quality of a path degrades to where the transmitter is turned off by the synchronization function (i.e., no packets are arriving at the destination), the weight is set to zero. If all transmitters are turned off, no packets are sent.

Conventional TCP/IP protocols include a "throttling" feature that reduces the transmission rate of packets when it is determined that delays or errors are occurring in transmission. In this respect, timers are sometimes used to determine whether packets have been received. These conventional techniques for limiting transmission of packets, however, do not involve multiple transmission paths between two nodes wherein transmission across a particular path relative to the others is changed based on link quality.

According to certain embodiments, in order to damp oscillations that might otherwise occur if weight distributions are changed drastically (e.g., according to a step function), a linear or an exponential decay formula can be applied to gradually decrease the weight value over time that a degrading path will be used. Similarly, if the health of a degraded path improves, the weight value for that path is gradually increased.

Transmission link health can be evaluated by comparing the number of packets that are acknowledged within the transmission window (see embodiments discussed above) to the number of packets transmitted within that window and by the state of the transmitter (i.e., on or off). In other words, rather than accumulating general transmission statistics over time for a path, one specific implementation uses the "windowing" concepts described above to evaluate transmission path health.

The same scheme can be used to shift virtual circuit paths from an "unhealthy" path to a "healthy" one, and to select a path for a new virtual circuit.

FIG. 22A shows a flowchart for adjusting weight values associated with a plurality of transmission links. It is assumed that software executing in one or more computer nodes executes the steps shown in FIG. 22A. It is also assumed that the software can be stored on a computer-readable medium such as a magnetic or optical disk for execution by a computer.

Beginning in step 2201, the transmission quality of a given transmission path is measured. As described above, this measurement can be based on a comparison between the number of packets transmitted over a particular link to the number of packet acknowledgements received over the link (e.g., per unit time, or in absolute terms). Alternatively, the quality can be evaluated by comparing the number of packets that are acknowledged within the transmission window to the number of packets that were transmitted within that window. In yet another variation, the number of missed synchronization messages can be used to indicate link quality. Many other variations are of course possible.

In step 2202, a check is made to determine whether more than one transmitter (e.g., transmission path) is turned on. If not, the process is terminated and resumes at step 2201.

In step 2203, the link quality is compared to a given threshold (e.g., 50%, or any arbitrary number). If the quality falls below the threshold, then in step 2207 a check is made to determine whether the weight is above a minimum level (e.g., 1%). If not, then in step 2209 the weight is set to the minimum level and processing resumes at step 2201. If the weight is above the minimum level, then in step 2208 the weight is gradually decreased for the path, then in step 2206 the weights for the remaining paths are adjusted accordingly to compensate (e.g., they are increased).

If in step 2203 the quality of the path was greater than or equal to the threshold, then in step 2204 a check is made to determine whether the weight is less than a steady-state value for that path. If so, then in step 2205 the weight is increased toward the steady-state value, and in step 2206 the weights for the remaining paths are adjusted accordingly to compensate (e.g., they

are decreased). If in step 2204 the weight is not less than the steady-state value, then processing resumes at step 2201 without adjusting the weights.

The weights can be adjusted incrementally according to various functions, preferably by changing the value gradually. In one embodiment, a linearly decreasing function is used to adjust the weights; according to another embodiment, an exponential decay function is used. Gradually changing the weights helps to damp oscillators that might otherwise occur if the probabilities were abruptly.

Although not explicitly shown in FIG. 22A the process can be performed only periodically (e.g., according to a time schedule), or it can be continuously run, such as in a background mode of operation. In one embodiment, the combined weights of all potential paths should add up to unity (e.g., when the weighting for one path is decreased, the corresponding weights that the other paths will be selected will increase).

Adjustments to weight values for other paths can be prorated. For example, a decrease of 10% in weight value for one path could result in an evenly distributed increase in the weights for the remaining paths. Alternatively, weightings could be adjusted according to a weighted formula as desired (e.g., favoring healthy paths over less healthy paths). In yet another variation, the difference in weight value can be amortized over the remaining links in a manner that is proportional to their traffic weighting.

FIG. 22B shows steps that can be executed to shut down transmission links where a transmitter turns off. In step 2210, a transmitter shut-down event occurs. In step 2211, a test is made to determine whether at least one transmitter is still turned on. If not, then in step 2215 all packets are dropped until a transmitter turns on. If in step 2211 at least one transmitter is turned on, then in step 2212 the weight for the path is set to zero, and the weights for the remaining paths are adjusted accordingly.

FIG. 23 shows a computer node 2301 employing various principles of the above-described embodiments. It is assumed that two computer nodes of the type shown in FIG. 23 communicate over a plurality of separate physical transmission paths. As shown in FIG. 23, four transmission paths X1 through X4 are defined for communicating between the two nodes. Each

node includes a packet transmitter 2302 that operates in accordance with a transmit table 2308 as described above. (The packet transmitter could also operate without using the IP-hopping features described above, but the following description assumes that some form of hopping is employed in conjunction with the path selection mechanism.). The computer node also includes  
5 a packet receiver 2303 that operates in accordance with a receive table 2309, including a moving window *W* that moves as valid packets are received. Invalid packets having source and destination addresses that do not fall within window *W* are rejected.

As each packet is readied for transmission, source and destination IP addresses (or other discriminator values) are selected from transmit table 2308 according to any of the various  
10 algorithms described above, and packets containing these source/destination address pairs, which correspond to the node to which the four transmission paths are linked, are generated to a transmission path switch 2307. Switch 2307, which can comprise a software function, selects from one of the available transmission paths according to a weight distribution table 2306. For example, if the weight for path X1 is 0.2, then every fifth packet will be transmitted on path X1.  
15 A similar regime holds true for the other paths as shown. Initially, each link's weight value can be set such that it is proportional to its bandwidth, which will be referred to as its "steady-state" value.

Packet receiver 2303 generates an output to a link quality measurement function 2304 that operates as described above to determine the quality of each transmission path. (The input  
20 to packet receiver 2303 for receiving incoming packets is omitted for clarity). Link quality measurement function 2304 compares the link quality to a threshold for each transmission link and, if necessary, generates an output to weight adjustment function 2305. If a weight adjustment is required, then the weights in table 2306 are adjusted accordingly, preferably according to a gradual (e.g., linearly or exponentially declining) function. In one embodiment,  
25 the weight values for all available paths are initially set to the same value, and only when paths degrade in quality are the weights changed to reflect differences.

Link quality measurement function 2304 can be made to operate as part of a synchronizer function as described above. That is, if resynchronization occurs and the receiver detects that

synchronization has been lost (e.g., resulting in the synchronization window  $W$  being advanced out of sequence), that fact can be used to drive link quality measurement function 2304. According to one embodiment, load balancing is performed using information garnered during the normal synchronization, augmented slightly to communicate link health from the receiver to the transmitter. The receiver maintains a count,  $MESS\_R(W)$ , of the messages received in synchronization window  $W$ . When it receives a synchronization request ( $SYNC\_REQ$ ) corresponding to the end of window  $W$ , the receiver includes counter  $MESS\_R$  in the resulting synchronization acknowledgement ( $SYNC\_ACK$ ) sent back to the transmitter. This allows the transmitter to compare messages sent to messages received in order to assess the health of the link.

If synchronization is completely lost, weight adjustment function 2305 decreases the weight value on the affected path to zero. When synchronization is regained, the weight value for the affected path is gradually increased to its original value. Alternatively, link quality can be measured by evaluating the length of time required for the receiver to acknowledge a synchronization request. In one embodiment, separate transmit and receive tables are used for each transmission path.

When the transmitter receives a  $SYNC\_ACK$ , the  $MESS\_R$  is compared with the number of messages transmitted in a window ( $MESS\_T$ ). When the transmitter receives a  $SYNC\_ACK$ , the traffic probabilities will be examined and adjusted if necessary.  $MESS\_R$  is compared with the number of messages transmitted in a window ( $MESS\_T$ ). There are two possibilities:

1. If  $MESS\_R$  is less than a threshold value,  $THRESH$ , then the link will be deemed to be unhealthy. If the transmitter was turned off, the transmitter is turned on and the weight  $P$  for that link will be set to a minimum value  $MIN$ . This will keep a trickle of traffic on the link for monitoring purposes until it recovers. If the transmitter was turned on, the weight  $P$  for that link will be set to:

$$P' = \alpha \times MIN + (1 - \alpha) \times P \quad (1)$$

Equation 1 will exponentially damp the traffic weight value to  $MIN$  during sustained periods of degraded service.

2. If MESS\_R for a link is greater than or equal to THRESH, the link will be deemed healthy. If the weight P for that link is greater than or equal to the steady state value S for that link, then P is left unaltered. If the weight P for that link is less than THRESH then P will be set to:

5 
$$P' = \beta \times S + (1 - \beta) \times P \quad (2)$$

where  $\beta$  is a parameter such that  $0 < \beta \leq 1$  that determines the damping rate of P.

Equation 2 will increase the traffic weight to S during sustained periods of acceptable service in a damped exponential fashion.

10 A detailed example will now be provided with reference to FIG. 24. As shown in FIG. 24, a first computer 2401 communicates with a second computer 2402 through two routers 2403 and 2404. Each router is coupled to the other router through three transmission links. As described above, these may be physically diverse links or logical links (including virtual private networks).

15 Suppose that a first link L1 can sustain a transmission bandwidth of 100 Mb/s and has a window size of 32; link L2 can sustain 75 Mb/s and has a window size of 24; and link L3 can sustain 25 Mb/s and has a window size of 8. The combined links can thus sustain 200Mb/s. The steady state traffic weights are 0.5 for link L1; 0.375 for link L2, and 0.125 for link L3. MIN=1Mb/s, THRESH =0.8 MESS\_T for each link,  $\alpha=.75$  and  $\beta=.5$ . These traffic weights will remain stable until a link stops for synchronization or reports a number of packets received less  
20 than its THRESH. Consider the following sequence of events:

1. Link L1 receives a SYNC\_ACK containing a MESS\_R of 24, indicating that only 75% of the MESS\_T (32) messages transmitted in the last window were successfully received. Link 1 would be below THRESH (0.8). Consequently, link L1's traffic weight value would be reduced to 0.12825, while link L2's traffic weight value would be increased to 0.65812 and link L3's  
25 traffic weight value would be increased to 0.217938.

2. Link L2 and L3 remained healthy and link L1 stopped to synchronize. Then link L1's traffic weight value would be set to 0, link L2's traffic weight value would be set to 0.75, and link L3's traffic weight value would be set to 0.25.

5 3. Link L1 finally received a SYNC\_ACK containing a MESS\_R of 0 indicating that none of the MESS\_T (32) messages transmitted in the last window were successfully received. Link L1 would be below THRESH. Link L1's traffic weight value would be increased to .005, link L2's traffic weight value would be decreased to 0.74625, and link L3's traffic weight value would be decreased to 0.24875.

10 4. Link L1 received a SYNC\_ACK containing a MESS\_R of 32 indicating that 100% of the MESS\_T (32) messages transmitted in the last window were successfully received. Link L1 would be above THRESH. Link L1's traffic weight value would be increased to 0.2525, while link L2's traffic weight value would be decreased to 0.560625 and link L3's traffic weight value would be decreased to .186875.

15 5. Link L1 received a SYNC\_ACK containing a MESS\_R of 32 indicating that 100% of the MESS\_T (32) messages transmitted in the last window were successfully received. Link L1 would be above THRESH. Link L1's traffic weight value would be increased to 0.37625; link L2's traffic weight value would be decreased to 0.4678125, and link L3's traffic weight value would be decreased to 0.1559375.

20 6. Link L1 remains healthy and the traffic probabilities approach their steady state traffic probabilities.

**B. Use of a DNS Proxy to Transparently Create Virtual Private Networks**

A second improvement concerns the automatic creation of a virtual private network (VPN) in response to a domain-name server look-up function.

25 Conventional Domain Name Servers (DNSs) provide a look-up function that returns the IP address of a requested computer or host. For example, when a computer user types in the web name "Yahoo.com," the user's web browser transmits a request to a DNS, which converts the name into a four-part IP address that is returned to the user's browser and then used by the browser to contact the destination web site.

This conventional scheme is shown in FIG. 25. A user's computer 2501 includes a client application 2504 (for example, a web browser) and an IP protocol stack 2505. When the user enters the name of a destination host, a request DNS REQ is made (through IP protocol stack 2505) to a DNS 2502 to look up the IP address associated with the name. The DNS returns the IP address DNS RESP to client application 2504, which is then able to use the IP address to communicate with the host 2503 through separate transactions such as PAGE REQ and PAGE RESP.

In the conventional architecture shown in FIG. 25, nefarious listeners on the Internet could intercept the DNS REQ and DNS RESP packets and thus learn what IP addresses the user was contacting. For example, if a user wanted to set up a secure communication path with a web site having the name "Target.com," when the user's browser contacted a DNS to find the IP address for that web site, the true IP address of that web site would be revealed over the Internet as part of the DNS inquiry. This would hamper anonymous communications on the Internet.

One conventional scheme that provides secure virtual private networks over the Internet provides the DNS server with the public keys of the machines that the DNS server has the addresses for. This allows hosts to retrieve automatically the public keys of a host that the host is to communicate with so that the host can set up a VPN without having the user enter the public key of the destination host. One implementation of this standard is presently being developed as part of the FreeS/WAN project(RFC 2535).

The conventional scheme suffers from certain drawbacks. For example, any user can perform a DNS request. Moreover, DNS requests resolve to the same value for all users.

According to certain aspects of the invention, a specialized DNS server traps DNS requests and, if the request is from a special type of user (e.g., one for which secure communication services are defined), the server does not return the true IP address of the target node, but instead automatically sets up a virtual private network between the target node and the user. The VPN is preferably implemented using the IP address "hopping" features of the basic invention described above, such that the true identity of the two nodes cannot be determined even if packets during the communication are intercepted. For DNS requests that are determined



to not require secure services (e.g., an unregistered user), the DNS server transparently “passes through” the request to provide a normal look-up function and return the IP address of the target web server, provided that the requesting host has permissions to resolve unsecured sites. Different users who make an identical DNS request could be provided with different results.

5 FIG. 26 shows a system employing various principles summarized above. A user’s computer 2601 includes a conventional client (e.g., a web browser) 2605 and an IP protocol stack 2606 that preferably operates in accordance with an IP hopping function 2607 as outlined above. A modified DNS server 2602 includes a conventional DNS server function 2609 and a DNS proxy 2610. A gatekeeper server 2603 is interposed between the modified DNS server and  
10 a secure target site 2704. An “unsecure” target site 2611 is also accessible via conventional IP protocols.

According to one embodiment, DNS proxy 2610 intercepts all DNS lookup functions from client 2605 and determines whether access to a secure site has been requested. If access to a secure site has been requested (as determined, for example, by a domain name extension, or by  
15 reference to an internal table of such sites), DNS proxy 2610 determines whether the user has sufficient security privileges to access the site. If so, DNS proxy 2610 transmits a message to gatekeeper 2603 requesting that a virtual private network be created between user computer 2601 and secure target site 2604. In one embodiment, gatekeeper 2603 creates “hopblocks” to be used by computer 2601 and secure target site 2604 for secure communication. Then, gatekeeper 2603  
20 communicates these to user computer 2601. Thereafter, DNS proxy 2610 returns to user computer 2601 the resolved address passed to it by the gatekeeper (this address could be different from the actual target computer) 2604, preferably using a secure administrative VPN. The address that is returned need not be the actual address of the destination computer.

Had the user requested lookup of a non-secure web site such as site 2611, DNS proxy  
25 would merely pass through to conventional DNS server 2609 the look-up request, which would be handled in a conventional manner, returning the IP address of non-secure web site 2611. If the user had requested lookup of a secure web site but lacked credentials to create such a connection, DNS proxy 2610 would return a “host unknown” error to the user. In this manner,

different users requesting access to the same DNS name could be provided with different look-up results.

Gatekeeper 2603 can be implemented on a separate computer (as shown in FIG. 26) or as a function within modified DNS server 2602. In general, it is anticipated that gatekeeper 2703 facilitates the allocation and exchange of information needed to communicate securely, such as using "hopped" IP addresses. Secure hosts such as site 2604 are assumed to be equipped with a secure communication function such as an IP hopping function 2608.

It will be appreciated that the functions of DNS proxy 2610 and DNS server 2609 can be combined into a single server for convenience. Moreover, although element 2602 is shown as combining the functions of two servers, the two servers can be made to operate independently.

FIG. 27 shows steps that can be executed by DNS proxy server 2610 to handle requests for DNS look-up for secure hosts. In step 2701, a DNS look-up request is received for a target host. In step 2702, a check is made to determine whether access to a secure host was requested. If not, then in step 2703 the DNS request is passed to conventional DNS server 2609, which looks up the IP address of the target site and returns it to the user's application for further processing.

In step 2702, if access to a secure host was requested, then in step 2704 a further check is made to determine whether the user is authorized to connect to the secure host. Such a check can be made with reference to an internally stored list of authorized IP addresses, or can be made by communicating with gatekeeper 2603 (e.g., over an "administrative" VPN that is secure). It will be appreciated that different levels of security can also be provided for different categories of hosts. For example, some sites may be designated as having a certain security level, and the security level of the user requesting access must match that security level. The user's security level can also be determined by transmitting a request message back to the user's computer requiring that it prove that it has sufficient privileges.

If the user is not authorized to access the secure site, then a "host unknown" message is returned (step 2705). If the user has sufficient security privileges, then in step 2706 a secure VPN is established between the user's computer and the secure target site. As described above,

this is preferably done by allocating a hopping regime that will be carried out between the user's computer and the secure target site, and is preferably performed transparently to the user (i.e., the user need not be involved in creating the secure link). As described in various embodiments of this application, any of various fields can be "hopped" (e.g., IP source/destination addresses; a field in the header; etc.) in order to communicate securely.

Some or all of the security functions can be embedded in gatekeeper 2603, such that it handles all requests to connect to secure sites. In this embodiment, DNS proxy 2610 communicates with gatekeeper 2603 to determine (preferably over a secure administrative VPN) whether the user has access to a particular web site. Various scenarios for implementing these features are described by way of example below:

Scenario #1: Client has permission to access target computer, and gatekeeper has a rule to make a VPN for the client. In this scenario, the client's DNS request would be received by the DNS proxy server 2610, which would forward the request to gatekeeper 2603. The gatekeeper would establish a VPN between the client and the requested target. The gatekeeper would provide the address of the destination to the DNS proxy, which would then return the resolved name as a result. The resolved address can be transmitted back to the client in a secure administrative VPN.

Scenario #2: Client does not have permission to access target computer. In this scenario, the client's DNS request would be received by the DNS proxy server 2610, which would forward the request to gatekeeper 2603. The gatekeeper would reject the request, informing DNS proxy server 2610 that it was unable to find the target computer. The DNS proxy 2610 would then return a "host unknown" error message to the client.

Scenario #3: Client has permission to connect using a normal non-VPN link, and the gatekeeper does not have a rule to set up a VPN for the client to the target site. In this scenario, the client's DNS request is received by DNS proxy server 2610, which would check its rules and determine that no VPN is needed. Gatekeeper 2603 would then inform the DNS proxy server to forward the request to conventional DNS server 2609, which would resolve the request and return the result to the DNS proxy server and then back to the client.

Scenario #4: Client does not have permission to establish a normal/non-VPN link, and the gatekeeper does not have a rule to make a VPN for the client to the target site. In this scenario, the DNS proxy server would receive the client's DNS request and forward it to gatekeeper 2603. Gatekeeper 2603 would determine that no special VPN was needed, but that the client is not authorized to communicate with non-VPN members. The gatekeeper would reject the request, causing DNS proxy server 2610 to return an error message to the client.

### **C. Large Link to Small Link Bandwidth Management**

One feature of the basic architecture is the ability to prevent so-called "denial of service" attacks that can occur if a computer hacker floods a known Internet node with packets, thus preventing the node from communicating with other nodes. Because IP addresses or other fields are "hopped" and packets arriving with invalid addresses are quickly discarded, Internet nodes are protected against flooding targeted at a single IP address.

In a system in which a computer is coupled through a link having a limited bandwidth (e.g., an edge router) to a node that can support a much higher-bandwidth link (e.g., an Internet Service Provider), a potential weakness could be exploited by a determined hacker. Referring to FIG. 28, suppose that a first host computer 2801 is communicating with a second host computer 2804 using the IP address hopping principles described above. The first host computer is coupled through an edge router 2802 to an Internet Service Provider (ISP) 2803 through a low bandwidth link (LOW BW), and is in turn coupled to second host computer 2804 through parts of the Internet through a high bandwidth link (HIGH BW). In this architecture, the ISP is able to support a high bandwidth to the internet, but a much lower bandwidth to the edge router 2802.

Suppose that a computer hacker is able to transmit a large quantity of dummy packets addressed to first host computer 2801 across high bandwidth link HIGH BW. Normally, host computer 2801 would be able to quickly reject the packets since they would not fall within the acceptance window permitted by the IP address hopping scheme. However, because the packets must travel across low bandwidth link LOW BW, the packets overwhelm the lower bandwidth link before they are received by host computer 2801. Consequently, the link to host computer 2801 is effectively flooded before the packets can be discarded.

According to one inventive improvement, a "link guard" function 2805 is inserted into the high-bandwidth node (e.g., ISP 2803) that quickly discards packets destined for a low-bandwidth target node if they are not valid packets. Each packet destined for a low-bandwidth node is cryptographically authenticated to determine whether it belongs to a VPN. If it is not a valid VPN packet, the packet is discarded at the high-bandwidth node. If the packet is authenticated as belonging to a VPN, the packet is passed with high preference. If the packet is a valid non-VPN packet, it is passed with a lower quality of service (e.g., lower priority).

In one embodiment, the ISP distinguishes between VPN and non-VPN packets using the protocol of the packet. In the case of IPSEC [rfc 2401], the packets have IP protocols 420 and 421. In the case of the TARP VPN, the packets will have an IP protocol that is not yet defined. The ISP's link guard, 2805, maintains a table of valid VPNs which it uses to validate whether VPN packets are cryptographically valid. According to one embodiment, packets that do not fall within any hop windows used by nodes on the low-bandwidth link are rejected, or are sent with a lower quality of service. One approach for doing this is to provide a copy of the IP hopping tables used by the low-bandwidth nodes to the high-bandwidth node, such that both the high-bandwidth and low-bandwidth nodes track hopped packets (e.g., the high-bandwidth node moves its hopping window as valid packets are received). In such a scenario, the high-bandwidth node discards packets that do not fall within the hopping window before they are transmitted over the low-bandwidth link. Thus, for example, ISP 2903 maintains a copy 2910 of the receive table used by host computer 2901. Incoming packets that do not fall within this receive table are discarded. According to a different embodiment, link guard 2805 validates each VPN packet using a keyed hashed message authentication code (HMAC) [rfc 2104].

According to another embodiment, separate VPNs (using, for example, hopblocks) can be established for communicating between the low-bandwidth node and the high-bandwidth node (i.e., packets arriving at the high-bandwidth node are converted into different packets before being transmitted to the low-bandwidth node).

As shown in FIG. 29, for example, suppose that a first host computer 2900 is communicating with a second host computer 2902 over the Internet, and the path includes a high

5  
10  
15  
20

bandwidth link HIGH BW to an ISP 2901 and a low bandwidth link LOW BW through an edge router 2904. In accordance with the basic architecture described above, first host computer 2900 and second host computer 2902 would exchange hopblocks (or a hopblock algorithm) and would be able to create matching transmit and receive tables 2905, 2906, 2912 and 2913. Then in accordance with the basic architecture, the two computers would transmit packets having seemingly random IP source and destination addresses, and each would move a corresponding hopping window in its receive table as valid packets were received.

Suppose that a nefarious computer hacker 2903 was able to deduce that packets having a certain range of IP addresses (e.g., addresses 100 to 200 for the sake of simplicity) are being transmitted to ISP 2901, and that these packets are being forwarded over a low-bandwidth link. Hacker computer 2903 could thus "flood" packets having addresses falling into the range 100 to 200, expecting that they would be forwarded along low bandwidth link LOW BW, thus causing the low bandwidth link to become overwhelmed. The fast packet reject mechanism in first host computer 3000 would be of little use in rejecting these packets, since the low bandwidth link was effectively jammed before the packets could be rejected. In accordance with one aspect of the improvement, however, VPN link guard 2911 would prevent the attack from impacting the performance of VPN traffic because the packets would either be rejected as invalid VPN packets or given a lower quality of service than VPN traffic over the lower bandwidth link. A denial-of-service flood attack could, however, still disrupt non-VPN traffic.

According to one embodiment of the improvement, ISP 2901 maintains a separate VPN with first host computer 2900, and thus translates packets arriving at the ISP into packets having a different IP header before they are transmitted to host computer 2900. The cryptographic keys used to authenticate VPN packets at the link guard 2911 and the cryptographic keys used to encrypt and decrypt the VPN packets at host 2902 and host 2901 can be different, so that link guard 2911 does not have access to the private host data; it only has the capability to authenticate those packets.

According to yet a third embodiment, the low-bandwidth node can transmit a special message to the high-bandwidth node instructing it to shut down all transmissions on a particular

IP address, such that only hopped packets will pass through to the low-bandwidth node. This embodiment would prevent a hacker from flooding packets using a single IP address. According to yet a fourth embodiment, the high-bandwidth node can be configured to discard packets transmitted to the low-bandwidth node if the transmission rate exceeds a certain predetermined threshold for any given IP address; this would allow hopped packets to go through. In this respect, link guard 2911 can be used to detect that the rate of packets on a given IP address are exceeding a threshold rate; further packets addressed to that same IP address would be dropped or transmitted at a lower priority (e.g., delayed).

**D. Traffic Limiter**

In a system in which multiple nodes are communicating using "hopping" technology, a treasonous insider could internally flood the system with packets. In order to prevent this possibility, one inventive improvement involves setting up "contracts" between nodes in the system, such that a receiver can impose a bandwidth limitation on each packet sender. One technique for doing this is to delay acceptance of a checkpoint synchronization request from a sender until a certain time period (e.g., one minute) has elapsed. Each receiver can effectively control the rate at which its hopping window moves by delaying "SYNC ACK" responses to "SYNC\_REQ" messages.

A simple modification to the checkpoint synchronizer will serve to protect a receiver from accidental or deliberate overload from an internally treasonous client. This modification is based on the observation that a receiver will not update its tables until a SYNC\_REQ is received on hopped address CKPT\_N. It is a simple matter of deferring the generation of a new CKPT\_N until an appropriate interval after previous checkpoints.

Suppose a receiver wished to restrict reception from a transmitter to 100 packets a second, and that checkpoint synchronization messages were triggered every 50 packets. A compliant transmitter would not issue new SYNC\_REQ messages more often than every 0.5 seconds. The receiver could delay a non-compliant transmitter from synchronizing by delaying the issuance of CKPT\_N for 0.5 second after the last SYNC\_REQ was accepted.

In general, if M receivers need to restrict N transmitters issuing new SYNC\_REQ messages after every W messages to sending R messages a second in aggregate, each receiver could defer issuing a new CKPT\_N until  $M \times N \times W / R$  seconds have elapsed since the last SYNC\_REQ has been received and accepted. If the transmitter exceeds this rate between a pair of checkpoints, it will issue the new checkpoint before the receiver is ready to receive it, and the SYNC\_REQ will be discarded by the receiver. After this, the transmitter will re-issue the SYNC\_REQ every T1 seconds until it receives a SYNC\_ACK. The receiver will eventually update CKPT\_N and the SYNC\_REQ will be acknowledged. If the transmission rate greatly exceeds the allowed rate, the transmitter will stop until it is compliant. If the transmitter exceeds the allowed rate by a little, it will eventually stop after several rounds of delayed synchronization until it is in compliance. Hacking the transmitter's code to not shut off only permits the transmitter to lose the acceptance window. In this case it can recover the window and proceed only after it is compliant again.

Two practical issues should be considered when implementing the above scheme:

1. The receiver rate should be slightly higher than the permitted rate in order to allow for statistical fluctuations in traffic arrival times and non-uniform load balancing.

2. Since a transmitter will rightfully continue to transmit for a period after a SYNC\_REQ is transmitted, the algorithm above can artificially reduce the transmitter's bandwidth. If events prevent a compliant transmitter from synchronizing for a period (e.g. the network dropping a SYNC\_REQ or a SYNC\_ACK) a SYNC\_REQ will be accepted later than expected. After this, the transmitter will transmit fewer than expected messages before encountering the next checkpoint. The new checkpoint will not have been activated and the transmitter will have to retransmit the SYNC\_REQ. This will appear to the receiver as if the transmitter is not compliant. Therefore, the next checkpoint will be accepted late from the transmitter's perspective. This has the effect of reducing the transmitter's allowed packet rate until the transmitter transmits at a packet rate below the agreed upon rate for a period of time.

To guard against this, the receiver should keep track of the times that the last C SYNC\_REQs were received and accepted and use the minimum of  $M \times N \times W / R$  seconds after the



last SYNC\_REQ has been received and accepted,  $2xMxNxW/R$  seconds after next to the last SYNC\_REQ has been received and accepted,  $CxMxNxW/R$  seconds after  $(C-1)^{th}$  to the last SYNC\_REQ has been received, as the time to activate CKPT\_N. This prevents the receiver from inappropriately limiting the transmitter's packet rate if at least one out of the last C SYNC\_REQs was processed on the first attempt.

FIG. 30 shows a system employing the above-described principles. In FIG. 30, two computers 3000 and 3001 are assumed to be communicating over a network N in accordance with the "hopping" principles described above (e.g., hopped IP addresses, discriminator values, etc.). For the sake of simplicity, computer 3000 will be referred to as the receiving computer and computer 3001 will be referred to as the transmitting computer, although full duplex operation is of course contemplated. Moreover, although only a single transmitter is shown, multiple transmitters can transmit to receiver 3000.

As described above, receiving computer 3000 maintains a receive table 3002 including a window W that defines valid IP address pairs that will be accepted when appearing in incoming data packets. Transmitting computer 3001 maintains a transmit table 3003 from which the next IP address pairs will be selected when transmitting a packet to receiving computer 3000. (For the sake of illustration, window W is also illustrated with reference to transmit table 3003). As transmitting computer moves through its table, it will eventually generate a SYNC\_REQ message as illustrated in function 3010. This is a request to receiver 3000 to synchronize the receive table 3002, from which transmitter 3001 expects a response in the form of a CKPT\_N (included as part of a SYNC\_ACK message). If transmitting computer 3001 transmits more messages than its allotment, it will prematurely generate the SYNC\_REQ message. (If it has been altered to remove the SYNC\_REQ message generation altogether, it will fall out of synchronization since receiver 3000 will quickly reject packets that fall outside of window W, and the extra packets generated by transmitter 3001 will be discarded).

In accordance with the improvements described above, receiving computer 3000 performs certain steps when a SYNC\_REQ message is received, as illustrated in FIG. 30. In step 3004, receiving computer 3000 receives the SYNC\_REQ message. In step 3005, a check is

made to determine whether the request is a duplicate. If so, it is discarded in step 3006. In step 3007, a check is made to determine whether the SYNC\_REQ received from transmitter 3001 was received at a rate that exceeds the allowable rate R (i.e., the period between the time of the last SYNC\_REQ message). The value R can be a constant, or it can be made to fluctuate as desired.

5 If the rate exceeds R, then in step 3008 the next activation of the next CKPT\_N hopping table entry is delayed by W/R seconds after the last SYNC\_REQ has been accepted.

Otherwise, if the rate has not been exceeded, then in step 3109 the next CKPT\_N value is calculated and inserted into the receiver's hopping table prior to the next SYNC\_REQ from the transmitter 3101. Transmitter 3101 then processes the SYNC\_REQ in the normal manner.

#### 10 **E. Signaling Synchronizer**

In a system in which a large number of users communicate with a central node using secure hopping technology, a large amount of memory must be set aside for hopping tables and their supporting data structures. For example, if one million subscribers to a web site occasionally communicate with the web site, the site must maintain one million hopping tables, thus using up valuable computer resources, even though only a small percentage of the users may actually be using the system at any one time. A desirable solution would be a system that permits a certain maximum number of simultaneous links to be maintained, but which would "recognize" millions of registered users at any one time. In other words, out of a population of a million registered users, a few thousand at a time could simultaneously communicate with a central server, without requiring that the server maintain one million hopping tables of appreciable size.

15

20

One solution is to partition the central node into two nodes: a signaling server that performs session initiation for user log-on and log-off (and requires only minimally sized tables), and a transport server that contains larger hopping tables for the users. The signaling server listens for the millions of known users and performs a fast-packet reject of other (bogus) packets. When a packet is received from a known user, the signaling server activates a virtual private link (VPL) between the user and the transport server, where hopping tables are allocated and maintained. When the user logs onto the signaling server, the user's computer is provided with

25

hop tables for communicating with the transport server, thus activating the VPL. The VPLs can be torn down when they become inactive for a time period, or they can be torn down upon user log-out. Communication with the signaling server to allow user log-on and log-off can be accomplished using a specialized version of the checkpoint scheme described above.

5 FIG. 31 shows a system employing certain of the above-described principles. In FIG. 31, a signaling server 3101 and a transport server 3102 communicate over a link. Signaling server 3101 contains a large number of small tables 3106 and 3107 that contain enough information to authenticate a communication request with one or more clients 3103 and 3104. As described in more detail below, these small tables may advantageously be constructed as a special case of the  
10 synchronizing checkpoint tables described previously. Transport server 3102, which is preferably a separate computer in communication with signaling server 3101, contains a smaller number of larger hopping tables 3108, 3109, and 3110 that can be allocated to create a VPN with one of the client computers.

According to one embodiment, a client that has previously registered with the system  
15 (e.g., via a system administration function, a user registration procedure, or some other method) transmits a request for information from a computer (e.g., a web site). In one variation, the request is made using a "hopped" packet, such that signaling server 3101 will quickly reject invalid packets from unauthorized computers such as hacker computer 3105. An "administrative" VPN can be established between all of the clients and the signaling server in  
20 order to ensure that a hacker cannot flood signaling server 3101 with bogus packets. Details of this scheme are provided below.

Signaling server 3101 receives the request 3111 and uses it to determine that client 3103 is a validly registered user. Next, signaling server 3101 issues a request to transport server 3102 to allocate a hopping table (or hopping algorithm or other regime) for the purpose of creating a  
25 VPN with client 3103. The allocated hopping parameters are returned to signaling server 3101 (path 3113), which then supplies the hopping parameters to client 3103 via path 3114, preferably in encrypted form.

Thereafter, client 3103 communicates with transport server 3102 using the normal hopping techniques described above. It will be appreciated that although signaling server 3101 and transport server 3102 are illustrated as being two separate computers, they could of course be combined into a single computer and their functions performed on the single computer. Alternatively, it is possible to partition the functions shown in FIG. 31 differently from as shown without departing from the inventive principles.

One advantage of the above-described architecture is that signaling server 3101 need only maintain a small amount of information on a large number of potential users, yet it retains the capability of quickly rejecting packets from unauthorized users such as hacker computer 3105. Larger data tables needed to perform the hopping and synchronization functions are instead maintained in a transport server 3102, and a smaller number of these tables are needed since they are only allocated for "active" links. After a VPN has become inactive for a certain time period (e.g., one hour), the VPN can be automatically torn down by transport server 3102 or signaling server 3101.

A more detailed description will now be provided regarding how a special case of the checkpoint synchronization feature can be used to implement the signaling scheme described above.

The signaling synchronizer may be required to support many (millions) of standing, low bandwidth connections. It therefore should minimize per-VPL memory usage while providing the security offered by hopping technology. In order to reduce memory usage in the signaling server, the data hopping tables can be completely eliminated and data can be carried as part of the SYNC\_REQ message. The table used by the server side (receiver) and client side (transmitter) is shown schematically as element 3106 in FIG. 31.

The meaning and behaviors of CKPT\_N, CKPT\_O and CKPT\_R remain the same from the previous description, except that CKPT\_N can receive a combined data and SYNC\_REQ message or a SYNC\_REQ message without the data.

The protocol is a straightforward extension of the earlier synchronizer. Assume that a client transmitter is on and the tables are synchronized. The initial tables can be generated "out

of band.” For example, a client can log into a web server to establish an account over the Internet. The client will receive keys etc encrypted over the Internet. Meanwhile, the server will set up the signaling VPN on the signaling server.

Assuming that a client application wishes to send a packet to the server on the client’s standing signaling VPL:

1. The client sends the message marked as a data message on the inner header using the transmitter’s CKPT\_N address. It turns the transmitter off and starts a timer T1 noting CKPT\_O. Messages can be one of three types: DATA, SYNC\_REQ and SYNC\_ACK. In the normal algorithm, some potential problems can be prevented by identifying each message type as part of the encrypted inner header field. In this algorithm, it is important to distinguish a data packet and a SYNC\_REQ in the signaling synchronizer since the data and the SYNC\_REQ come in on the same address.

2. When the server receives a data message on its CKPT\_N, it verifies the message and passes it up the stack. The message can be verified by checking message type and other information (i.e user credentials) contained in the inner header. It replaces its CKPT\_O with CKPT\_N and generates the next CKPT\_N. It updates its transmitter side CKPT\_R to correspond to the client’s receiver side CKPT\_R and transmits a SYNC\_ACK containing CKPT\_O in its payload.

3. When the client side receiver receives a SYNC\_ACK on its CKPT\_R with a payload matching its transmitter side CKPT\_O and the transmitter is off, the transmitter is turned on and the receiver side CKPT\_R is updated. If the SYNC\_ACK’s payload does not match the transmitter side CKPT\_O or the transmitter is on, the SYNC\_ACK is simply discarded.

4. T1 expires: If the transmitter is off and the client’s transmitter side CKPT\_O matches the CKPT\_O associated with the timer, it starts timer T1 noting CKPT\_O again, and a SYNC\_REQ is sent using the transmitter’s CKPT\_O address. Otherwise, no action is taken.

5. When the server receives a SYNC\_REQ on its CKPT\_N, it replaces its CKPT\_O with CKPT\_N and generates the next CKPT\_N. It updates its transmitter side CKPT\_R to correspond

to the client's receiver side CKPT\_R and transmits a SYNC\_ACK containing CKPT\_O in its payload.

6. When the server receives a SYNC\_REQ on its CKPT\_O, it updates its transmitter side CKPT\_R to correspond to the client's receiver side CKPT\_R and transmits a SYNC\_ACK  
5 containing CKPT\_O in its payload.

FIG. 32 shows message flows to highlight the protocol. Reading from top to bottom, the client sends data to the server using its transmitter side CKPT\_N. The client side transmitter is turned off and a retry timer is turned off. The transmitter will not transmit messages as long as the transmitter is turned off. The client side transmitter then loads CKPT\_N into CKPT\_O and  
10 updates CKPT\_N. This message is successfully received and a passed up the stack. It also synchronizes the receiver i.e, the server loads CKPT\_N into CKPT\_O and generates a new CKPT\_N, it generates a new CKPT\_R in the server side transmitter and transmits a SYNC\_ACK containing the server side receiver's CKPT\_O the server. The SYNC\_ACK is successfully received at the client. The client side receiver's CKPT\_R is updated, the transmitter is turned on  
15 and the retry timer is killed. The client side transmitter is ready to transmit a new data message.

Next, the client sends data to the server using its transmitter side CKPT\_N. The client side transmitter is turned off and a retry timer is turned off. The transmitter will not transmit messages as long as the transmitter is turned off. The client side transmitter then loads CKPT\_N into CKPT\_O and updates CKPT\_N. This message is lost. The client side timer expires and as a  
20 result a SYNC\_REQ is transmitted on the client side transmitter's CKPT\_O (this will keep happening until the SYNC\_ACK has been received at the client). The SYNC\_REQ is successfully received at the server. It synchronizes the receiver i.e, the server loads CKPT\_N into CKPT\_O and generates a new CKPT\_N, it generates an new CKPT\_R in the server side transmitter and transmits a SYNC\_ACK containing the server side receiver's CKPT\_O the  
25 server. The SYNC\_ACK is successfully received at the client. The client side receiver's CKPT\_R is updated, the transmitter is turned off and the retry timer is killed. The client side transmitter is ready to transmit a new data message.

There are numerous other scenarios that follow this flow. For example, the SYNC\_ACK could be lost. The transmitter would continue to re-send the SYNC\_REQ until the receiver synchronizes and responds.

5 The above-described procedures allow a client to be authenticated at signaling server 3201 while maintaining the ability of signaling server 3201 to quickly reject invalid packets, such as might be generated by hacker computer 3205. In various embodiments, the signaling synchronizer is really a derivative of the synchronizer. It provides the same protection as the hopping protocol, and it does so for a large number of low bandwidth connections.

0479.85672

**CLAIMS**

1. A method of transmitting data packets between a first computer and a second computer, wherein the first computer and the second computer are linked via a plurality of separate transmission paths, the method comprising the steps of:

5 (1) assigning a weight value to each of the plurality of transmission paths, wherein each respective weight value represents the relative number of packets that a respective transmission path will transmit;

(2) for each data packet that is to be transmitted from the first computer to the second computer, selecting one of the plurality of transmission paths on the basis of each respective  
10 transmission path's assigned weight value;

(3) measuring the transmission quality for each of the plurality of transmission paths; and

(4) adjusting downwardly to a non-zero value the assigned weight value for a transmission path for which the transmission quality has declined.

2. The method of claim 1, wherein step (4) comprises the step of gradually decreasing  
15 over time the assigned weight value in relation to weight values assigned to the remaining transmission paths.

3. The method of claim 2, wherein step (4) comprises the step of gradually decreasing the assigned weight value according to an incrementally decreasing function.

4. The method of claim 2, wherein step (4) comprises the step of gradually decreasing  
20 the assigned weight value according to an exponentially decaying function.

5. The method of claim 1, wherein step (3) comprises the step of determining that one or more packets transmitted to the second computer was not acknowledged by the second computer.

6. The method of claim 1, wherein step (3) comprises the step of evaluating the contents  
25 of a synchronization packet that maintains synchronization with a moving window of valid values.

7. The method of claim 1, further comprising the step of inserting into each data packet a source and destination IP address pair that is selected according to a pseudo-random sequence.



8. The method of claim 1, wherein step (4) comprises the step of adjusting downwardly the assigned weight value for a transmission path only if the transmission quality has declined below a predetermined threshold.

9. The method of claim 1, further comprising the step of adjusting upwardly the assigned weight value that was adjusted in step (4) if it is later determined that the transmission quality has improved.

10. The method of claim 1, further comprising the step of adjusting upwardly the weight values of the remaining transmission links in an amount that compensates for the downwardly adjusted weight value.

11. The method of claim 10, wherein the step of adjusting upwardly comprises the step of equally distributing the amount that was downwardly adjusted across the remaining transmission links.

12. The method of claim 1, further comprising the step of adjusting downwardly to zero the assigned weight value for any transmission link whose quality has degraded below a preset threshold.

13. The method of claim 1, wherein steps (2) through (4) are repeated periodically.

14. A first computer that transmits data packets to a second computer over a plurality of separate transmission paths, wherein the first computer performs the steps of:

(1) assigning a weight value to each of the plurality of transmission paths, wherein each respective weight value represents the relative number of packets that a respective transmission path will transmit;

(2) for each data packet that is to be transmitted to the second computer, selecting one of the plurality of transmission paths on the basis of each respective transmission path's assigned weight value;

(3) measuring the transmission quality for each of the plurality of transmission paths; and

(4) adjusting downwardly to a non-zero value the assigned weight value for a transmission path for which the transmission quality has declined.

15. The first computer of claim 14, wherein the first computer gradually decreases over time the assigned weight value in relation to weight values assigned to the remaining transmission paths.

5 16. The first computer of claim 15, wherein the first computer gradually decreases the assigned weight value according to an incrementally decreasing function.

17. The first computer of claim 15, wherein the first computer gradually decreases the assigned weight value according to an exponentially decaying function.

10 18. The first computer of claim 14, wherein the first computer measures the transmission quality by determining that one or more packets transmitted to the second computer was not acknowledged by the second computer.

19. The first computer of claim 14, wherein the first computer measures the transmission quality by evaluating the contents of a synchronization packet that maintains synchronization with a moving window of valid values.

15 20. The first computer of claim 14, wherein the first computer inserts into each data packet a source and destination IP address pair that is selected according to a pseudo-random sequence.

21. The first computer of claim 14, wherein the first computer adjusts downwardly the assigned weight value for any transmission path only if the transmission quality has declined below a predetermined threshold.

20 22. The first computer of claim 14, wherein the first computer adjusts upwardly the assigned weight value that was adjusted in step (4) if it is later determined that the transmission quality has improved.

25 23. The first computer of claim 14, wherein the first computer adjusts upwardly the weight values of the remaining transmission links in an amount that compensates for the downwardly adjusted weight value.

24. The first computer of claim 23, wherein the first computer upwardly adjusts probabilities across the remaining transmission links in an amount equal to the downwardly adjusted weight value.

25. The first computer of claim 14, wherein the first computer adjusts downwardly to zero the assigned weight value for any transmission link whose quality has degraded below a preset threshold.

5 26. The first computer of claim 14, wherein the first computer repeats steps (2) through (4) periodically.

27. A system comprising the first computer of claim 14 and a second computer constructed in accordance with the first computer of claim 14.

1 28. A method of transparently creating a virtual private network (VPN) between a client computer and a target computer, comprising the steps of:

10 (1) generating from the client computer a Domain Name Service (DNS) request that requests an IP address corresponding to a domain name associated with the target computer;

(2) determining whether the DNS request transmitted in step (1) is requesting access to a secure web site; and

15 (3) in response to determining that the DNS request in step (2) is requesting access to a secure target web site, automatically initiating the VPN between the client computer and the target computer.

2 29. The method of claim 28, wherein steps (2) and (3) are performed at a DNS server separate from the client computer.

3 30. The method of claim 28, further comprising the step of:

20 (4) in response to determining that the DNS request in step (2) is not requesting access to a secure target web site, resolving the IP address for the domain name and returning the IP address to the client computer.

4 31. The method of claim 28, wherein step (3) comprises the step of, prior to automatically initiating the VPN between the client computer and the target computer, determining whether the client computer is authorized to establish a VPN with the target computer and, if not so authorized, returning an error from the DNS request.

5 32. The method of claim 28, wherein step (3) comprises the step of, prior to automatically initiating the VPN between the client computer and the target computer,

77 

determining whether the client computer is authorized to resolve addresses of non secure target computers and, if not so authorized, returning an error from the DNS request.

33. The method of claim 28, wherein step (3) comprises the step of establishing the VPN by creating an IP address hopping scheme between the client computer and the target computer.

5 34. The method of claim 28, wherein step (3) comprises the step of using a gatekeeper computer that allocates VPN resources for communicating between the client computer and the target computer.

10 35. The method of claim 28, wherein step (2) is performed in a DNS proxy server that passes through the request to a DNS server if it is determined in step (3) that access is not being requested to a secure target web site.

36. The method of claim 32, wherein step (3) comprises the step of transmitting a message to the client computer to determine whether the client computer is authorized to establish the VPN target computer.

15 37. A system that transparently creates a virtual private network (VPN) between a client computer and a secure target computer, comprising:

a DNS proxy server that receives a request from the client computer to look up an IP address for a domain name, wherein the DNS proxy server returns the IP address for the requested domain name if it is determined that access to a non-secure web site has been requested, and wherein the DNS proxy server generates a request to create the VPN between the client computer and the secure target computer if it is determined that access to a secure web site has been requested; and

a gatekeeper computer that allocates resources for the VPN between the client computer and the secure web computer in response to the request by the DNS proxy server.

25 38. The system of claim 37, wherein the gatekeeper computer creates the VPN by establishing an IP address hopping regime that is used to pseudorandomly change IP addresses in packets transmitted between the client computer and the secure target computer.

12  
39. The system of claim 37, wherein the gatekeeper computer determines whether the client computer has sufficient security privileges to create the VPN and, if the client computer lacks sufficient security privileges, rejecting the request to create the VPN.

5 40. A method of preventing data packets received from a high bandwidth link from flooding a low bandwidth link, comprising the steps of:

(1) receiving data packets from the high bandwidth link that are ostensibly addressed to a computer residing on the low-bandwidth link;

(2) for each data packet, determining whether the data packet is validly addressed to the computer on the low-bandwidth link;

10 (3) in response to determining that the data packet is not validly addressed to the computer on the low-bandwidth link, rejecting the data packet; and

(4) in response to determining that the data packet is validly addressed to the computer on the low-bandwidth link, forwarding the data packet to the computer over the low-bandwidth link.

15 41. The method of claim 40, wherein step (3) comprises the step of comparing a value in a header of each data packet to a set of valid values maintained for the computer on the low-bandwidth link.

42. The method of claim 41, wherein step (3) comprises the step of comparing a value in a header of each data packet to a moving window of valid values.

20 43. The method of claim 42, wherein step (3) comprises the step of comparing the IP address in the header of each data packet to a moving window of valid IP addresses, wherein the moving window is also maintained by the computer on the low-bandwidth link.

25 44. The method of claim 40, wherein step (3) comprises the step of reducing a priority level of the packet in relation to other data packets, wherein the priority level determines whether a particular data packet will be transmitted before another data packet having a different priority level.

45. The method of claim 40, wherein step (3) comprises the step of performing a cryptographic check on each data packet to determine whether each data packet is validly addressed.

46. The method of claim 40, wherein step (3) comprises the step of receiving a message from the computer on the low-bandwidth link to stop accepting messages having a particular characteristic.

5 47. The method of claim 46, wherein step (3) comprises the step of receiving a message from the computer on the low-bandwidth link to stop accepting messages addressed to a particular IP address.

48. The method of claim 40, wherein step (3) comprises the step of determining that a packet transmission rate has been exceeded for a given packet parameter.

10 49. The method of claim 48, wherein step (3) comprises the step of determining that a packet transmission rate has been exceeded for a given IP destination address.

50. In a system having a low bandwidth data link, a first computer coupled to the low bandwidth data link, and a high bandwidth data link, an improvement comprising:

15 a second computer coupled between the low bandwidth data link and the high bandwidth data link, wherein the second computer receives data packets from the high bandwidth data link and, if they are addressed to the first computer, routes them to the first computer over the low bandwidth data link,

wherein the second computer prevents invalid data packets ostensibly addressed to the first computer from being transmitted over the low bandwidth data link.

20 51. The system of claim 50, wherein the second computer prevents invalid data packets from being transmitted over the low bandwidth data link by comparing a discriminator field in a header of each data packet to a table of valid discriminator fields maintained for the first computer.

52. The system of claim 50, wherein the second computer compares an Internet Protocol (IP) address in a header of each data packet to a table of valid IP addresses.

25 53. The system of claim 52, wherein the second computer compares the IP address in the header of each data packet to a moving window of valid IP addresses, wherein the moving window is also maintained by the first computer.

54. The system of claim 50, wherein the second computer reduces a priority level of a data packet in relation to other data packets, wherein the priority level determines whether a particular data packet will be transmitted before another data packet having a different priority level.

5 55. The system of claim 50, wherein the second computer performs a cryptographic check on each data packet to determine whether each data packet is validly addressed.

56. The system of claim 50, wherein the second computer receives a message from the first computer that causes the second computer to stop accepting messages having a particular characteristic.

10 57. The system of claim 56, wherein the second computer receiving a message from the first computer to stop accepting messages addressed to a particular IP address.

58. The system of claim 50, wherein the second computer rejects invalid packets by determining that a packet transmission rate has been exceeded for a given packet parameter.

15 59. The system of claim 58, wherein the second computer determines that a packet transmission rate has been exceeded for a given IP destination address.

20 60. In a system comprising a first computer that transmits data packets to a second computer over a network according to a scheme by which at least one field in a series of data packets is periodically changed according to a sequence known by the first and second computers, and wherein the second computer periodically receives a synchronization request from the first computer to maintain synchronization of the sequence between the first and second computers, a method comprising the steps of:

(1) receiving at the first computer the synchronization request from the second computer;

(2) determining whether the synchronization request was received in less than a predetermined interval;

25 (3) in response to determining that the synchronization request was received in less than the predetermined interval, ignoring the synchronization request; and

(4) in response to determining that the synchronization request was not received in less than the predetermined interval, providing the synchronization response to the first computer.

61. The method of claim 60, wherein step (3) comprises the step of delaying the acceptance of a SYNC\_REQ for  $W/R$  seconds, where  $W$  is the number of data packets between synchronization requests according to an agreed schedule, and  $R$  is the agreed rate at which synchronization requests should be received according to the agreed schedule.

5 62. The method of claim 60, further comprising the step of determining whether the synchronization request is a duplicate of a previously received synchronization request and, if it is a duplicate, discarding it.

63. The method of claim 60, wherein step (4) comprises the step of providing a response that includes a new checkpoint for synchronizing a window in a hopping table.

10 64. A computer that receives data packets from a second computer over a network according to a scheme by which at least one field in a series of data packets is periodically changed according to a known sequence, wherein the second computer periodically transmits a synchronization request to maintain synchronization of the sequence, wherein the computer performs the steps of:

- 15 (1) receiving the synchronization request from the second computer;
- (2) determining whether the synchronization request was received in less than a predetermined interval;
- (3) in response to determining that the synchronization request was received in less than a predetermined interval ignoring the synchronization request; and
- 20 (4) in response to determining that the synchronization request was not received in less than a predetermined interval, providing the response to the first computer.

25 65. The computer of claim 64, wherein the computer delays the acceptance of a SYNC\_REQ in step (3) for  $W/R$  seconds, where  $W$  is the number of data packets between synchronization requests according to an agreed schedule, and  $R$  is the agreed rate at which synchronization requests should be received according to the agreed schedule.

66. The computer of claim 64, wherein the computer further performs the step of determining whether the synchronization request is a duplicate of a previously received synchronization request and, if it is a duplicate, discarding it.



<sup>13</sup>  
~~67~~. A method of establishing communication between one of a plurality of client computers and a central computer that maintains a plurality of authentication tables each corresponding to one of the client computers, the method comprising the steps of:

5 (1) in the central computer, receiving from one of the plurality of client computers a request to establish a connection;

(2) authenticating, with reference to one of the plurality of authentication tables, that the request received in step (1) is from an authorized client;

(3) responsive to a determination that the request is from an authorized client, allocating resources to establish a virtual private link between the client and a second computer; and

10 (4) communicating between the authorized client and the second computer using the virtual private link.

<sup>14</sup>  
~~68~~. The method of claim <sup>13</sup>~~67~~, wherein step (4) comprises the step of communicating according to a scheme by which at least one field in a series of data packets is periodically changed according to a known sequence.

15 <sup>15</sup>  
~~69~~. The method of claim <sup>14</sup>~~68~~, wherein step (4) comprises the step of comparing an Internet Protocol (IP) address in a header of each data packet to a table of valid IP addresses maintained in a table in the second computer.

<sup>16</sup>  
~~70~~. The method of claim <sup>15</sup>~~69~~, wherein step (4) comprises the step of comparing the IP address in the header of each data packet to a moving window of valid IP addresses, and  
20 rejecting data packets having IP addresses that do not fall within the moving window.

<sup>17</sup>  
~~71~~. The method of claim <sup>13</sup>~~67~~, wherein step (2) comprises the step of using a checkpoint data structure that maintains synchronization of a periodically changing parameter known by the central computer and the client computer to authenticate the client.

Add R1  
Add C1

74

**ABSTRACT**

A plurality of computer nodes communicate using seemingly random Internet Protocol source and destination addresses. Data packets matching criteria defined by a moving window of valid addresses are accepted for further processing, while those that do not meet the criteria are quickly rejected. Improvements to the basic design include (1) a load balancer that distributes packets across different transmission paths according to transmission path quality; (2) a DNS proxy server that transparently creates a virtual private network in response to a domain name inquiry; (3) a large-to-small link bandwidth management feature that prevents denial-of-service attacks at system chokepoints; (4) a traffic limiter that regulates incoming packets by limiting the rate at which a transmitter can be synchronized with a receiver; and (5) a signaling synchronizer that allows a large number of nodes to communicate with a central node by partitioning the communication function between two separate entities.

## JOINT DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As the below named inventors, we hereby declare that:

Our residences, post office addresses and citizenships are as stated below next to our names:

We believe we are the original, first and joint inventors of the subject matter which is claimed and for which a patent is sought on the invention entitled:

### IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

the specification of which

is attached hereto.

was filed on \_\_\_\_\_ as Application Serial Number \_\_\_\_\_ and was amended on \_\_\_\_\_ (if applicable).

We hereby state that we have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

We acknowledge the duty to disclose information which is material to patentability in accordance with Title 37, Code of Federal Regulations, §1.56.

### Prior Foreign Application(s)

We hereby claim foreign priority benefits under Title 35, United States Code, §119(a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below any foreign application(s) for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Country	Application Number	Date of Filing (day, month, year)	Date of Issue (day, month, year)	Priority Claimed Under 35 U.S.C. 119	Certified Copy Attached
				Yes / No	Yes / No

### Prior United States Application(s)

We hereby claim the benefit under 35 U.S.C. 119(e) of any United States provisional application(s) listed below:

Application Number(s)	Filing Date (MM/DD/YYYY)	<input type="checkbox"/> Additional provisional application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto.
60/106,261	10/30/98	
60/137,704	6/7/99	

We hereby claim the benefit under Title 35, United States Code, §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, §112, We acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Application Serial Number	Date of Filing (Day, Month, Year)	Status Patented, Pending, Abandoned
09/429,643	10/29/99	Pending

**Power of Attorney**

And we hereby appoint, both jointly and severally, as our attorneys with full power of substitution and revocation, to prosecute this application and transact all business in the U.S. Patent and Trademark Office connected herewith as well as before any office or agency of a foreign country or any international organization in connection with any foreign counterpart application claiming priority to this application, including the power to appoint agents and local representatives in connection with such foreign applications, the following attorneys of Banner & Witcoff, their registration numbers being listed after their names:

Robert Altherr, Reg. No. 31,810, Donald W. Banner, Reg. No. 17,037; Edward F. McKie, Jr., Reg. No. 17,335; William W. Beckett, Reg. No. 18,262; Dale H. Hoscheit, Reg. No. 19,090; Joseph M. Potenza, Reg. No. 28,175; James A. Niegowski, Reg. No. 28,331; Joseph M. Skerpon, Reg. No. 29,864; Thomas L. Peterson, Reg. No. 30,969; Nina L. Medlock, Reg. No. 29,673; William J. Fisher, Reg. No. 32,133; Thomas H. Jackson, Reg. No. 29,808; Franklin D. Wolffe, Reg. No. 19,724; Susan A. Wolffe, Reg. No. 33,568; Daniel E. Fisher, Reg. No. 34,162; Kevin A. Wolff, Reg. No. 42,233 and Bradley C. Wright, Reg. No. 38,061.

All correspondence and telephone communications should be addressed to:

Banner & Witcoff, Ltd.  
 Eleventh Floor  
 1001 G Street, N.W.  
 Washington, D.C. 20001-4597  
 Tel. No. (202) 508-9100

We hereby declare that all statements made herein of our 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 18 U.S.C. 1001 and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature \_\_\_\_\_ Date \_\_\_\_\_

Full Name of Joint Inventor MUNGER Edmund Colby  
 Family Name First Given Name Second Given Name

Residence 1101 Opaca Court, Crownsville, Maryland 21032

Citizenship U.S.

Post Office Address 1101 Opaca Court, Crownsville, Maryland 21032

Signature \_\_\_\_\_ Date \_\_\_\_\_

Full Name of Joint Inventor SCHMIDT Douglas Charles  
Family Name First Given Name Second Given Name

Residence 230 Oak Court, Severna Park, Maryland 21146

Citizenship U.S.

Post Office Address 230 Oak Court, Severna Park, Maryland 21146

Signature *Robert D. Short* Date 2/14/00

Full Name of Joint Inventor SHORT Robert Dunham, III  
Family Name First Given Name Second Given Name

Residence 38710 Goose Creek Lane, Leesburg, Virginia 20175

Citizenship U.S.

Post Office Address 38710 Goose Creek Lane, Leesburg, Virginia 20175

Signature *Victor Larson* Date 2/14/2000

Full Name of Joint Inventor LARSON Victor  
Family Name First Given Name Second Given Name

Residence 12026 Lisa Marie Court, Fairfax, Virginia 22033

Citizenship U.S.

Post Office Address 12026 Lisa Marie Court, Fairfax, Virginia 22033

Signature  Date 3/14/2000

Full Name of Joint Inventor WILLIAMSON Michael  
Family Name First Given Name Second Given Name

Residence 26203 Ocala Circle, South Riding, Virginia 20152

Citizenship U.S.

Post Office 26203 Ocala Circle, South Riding, Virginia 20152

Address 26203 Ocala Circle, South Riding, Virginia 20152

1001 G STREET, N.W. WASHINGTON, D.C. 20001-4597 (202) 508-9100

LAW OFFICES  
BANNER & WITCOFF, LTD.  
1001 G STREET, N.W.  
WASHINGTON, D.C. 20001-4597  
(202) 508-9100

**JOINT DECLARATION AND POWER OF ATTORNEY  
FOR PATENT APPLICATION**

As the below named inventors, we hereby declare that:

Our residences, post office addresses and citizenships are as stated below next to our names:

We believe we are the original, first and joint inventors of the subject matter which is claimed and for which a patent is sought on the invention entitled:

**IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS  
WITH ASSURED SYSTEM AVAILABILITY**

the specification of which

is attached hereto.

was filed on \_\_\_\_\_ as Application Serial Number \_\_\_\_\_ and was amended on \_\_\_\_\_ (if applicable).

We hereby state that we have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

We acknowledge the duty to disclose information which is material to patentability in accordance with Title 37, Code of Federal Regulations, § 1.56.

**Prior Foreign Application(s)**

We hereby claim foreign priority benefits under Title 35, United States Code, § 119(a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below any foreign application(s) for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Country	Application Number	Date of Filing (day, month, year)	Date of Issue (day, month, year)	Priority Claimed Under 35 U.S.C. 119	Certified Copy Attached
				Yes / No	Yes / No

**Prior United States Application(s)**

We hereby claim the benefit under 35 U.S.C. 119(e) of any United States provisional application(s) listed below:

Application Number(s)	Filing Date (MM/DD/YYYY)	<input type="checkbox"/> Additional provisional application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto.
60/106,261	10/30/98	
60/137,704	6/7/99	

We hereby claim the benefit under Title 35, United States Code, §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, §112, We acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Application Serial Number	Date of Filing (Day, Month, Year)	Status Patented, Pending, Abandoned
09/429,643	10/29/99	Pending

**Power of Attorney**

And we hereby appoint, both jointly and severally, as our attorneys with full power of substitution and revocation, to prosecute this application and transact all business in the U.S. Patent and Trademark Office connected herewith as well as before any office or agency of a foreign country or any international organization in connection with any foreign counterpart application claiming priority to this application, including the power to appoint agents and local representatives in connection with such foreign applications, the following attorneys of Banner & Witcoff, their registration numbers being listed after their names:

Robert Altherr, Reg. No. 31,810, Donald W. Banner, Reg. No. 17,037; Edward F. McKie, Jr., Reg. No. 17,335; William W. Beckett, Reg. No. 18,262; Dale H. Hoscheit, Reg. No. 19,090; Joseph M. Potenza, Reg. No. 28,175; James A. Niegowski, Reg. No. 28,331; Joseph M. Skerpon, Reg. No. 29,864; Thomas L. Peterson, Reg. No. 30,969; Nina L. Medlock, Reg. No. 29,673; William J. Fisher, Reg. No. 32,133; Thomas H. Jackson, Reg. No. 29,808; Franklin D. Wolffe, Reg. No. 19,724; Susan A. Wolffe, Reg. No. 33,568; Daniel E. Fisher, Reg. No. 34,162; Kevin A. Wolff, Reg. No. 42,233 and Bradley C. Wright, Reg. No. 38,061.

All correspondence and telephone communications should be addressed to:

Banner & Witcoff, Ltd.  
 Eleventh Floor  
 1001 G Street, N.W.  
 Washington, D.C. 20001-4597  
 Tel. No. (202) 508-9100

We hereby declare that all statements made herein of our 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 18 U.S.C. 1001 and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature \_\_\_\_\_ Date \_\_\_\_\_

Full Name of Joint Inventor MUNGER Edmund Colby  
 Family Name First Given Name Second Given Name

Residence 1101 Opaca Court, Crownsville, Maryland 21032

Citizenship U.S.

Post Office Address 1101 Opaca Court, Crownsville, Maryland 21032



Signature Dy C Schmitt Date 2/14/00

Full Name of Joint Inventor SCHMIDT Douglas Charles  
Family Name First Given Name Second Given Name

Residence 230 Oak Court, Severna Park, Maryland 21146

Citizenship U.S.

Post Office Address 230 Oak Court, Severna Park, Maryland 21146

Signature \_\_\_\_\_ Date \_\_\_\_\_

Full Name of Joint Inventor SHORT Robert Dunham, III  
Family Name First Given Name Second Given Name

Residence 38710 Goose Creek Lane, Leesburg, Virginia 20175

Citizenship U.S.

Post Office Address 38710 Goose Creek Lane, Leesburg, Virginia 20175

Signature \_\_\_\_\_ Date \_\_\_\_\_

Full Name of Joint Inventor LARSON Victor  
Family Name First Given Name Second Given Name

Residence 12026 Lisa Marie Court, Fairfax, Virginia 22033

Citizenship U.S.

Post Office Address 12026 Lisa Marie Court, Fairfax, Virginia 22033

Signature \_\_\_\_\_ Date \_\_\_\_\_

Full Name of Joint Inventor WILLIAMSON Michael  
Family Name First Given Name Second Given Name

Residence 26203 Ocala Circle, South Riding, Virginia 20152

Citizenship U.S.

Post Office Address 26203 Ocala Circle, South Riding, Virginia 20152

11/11/08 10:00 AM

LAW OFFICES  
BANNER & WITCOFF, LTD.  
1001 G STREET, N.W.  
WASHINGTON, D.C. 20001-4597  
(202) 508-9100

### JOINT DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As the below named inventors, we hereby declare that:

Our residences, post office addresses and citizenships are as stated below next to our names:

We believe we are the original, first and joint inventors of the subject matter which is claimed and for which a patent is sought on the invention entitled:

#### IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

the specification of which

is attached hereto.

was filed on \_\_\_\_\_ as Application Serial Number \_\_\_\_\_ and was amended on \_\_\_\_\_ (if applicable).

We hereby state that we have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

We acknowledge the duty to disclose information which is material to patentability in accordance with Title 37, Code of Federal Regulations, §1.56.

#### Prior Foreign Application(s)

We hereby claim foreign priority benefits under Title 35, United States Code, §119(a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below any foreign application(s) for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Country	Application Number	Date of Filing	Date of Issue	Priority Claimed Under 35 U.S.C. 119	Consistent Copy Available
				Yes / No	Yes / No

#### Prior United States Application(s)

We hereby claim the benefit under 35 U.S.C. 119(e) of any United States provisional application(s) listed below:

Application Number	Filing Date	<input type="checkbox"/> Additional provisional application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto.
60/106,261	10/30/98	
60/137,704	6/7/99	

02/15/00 14:22 FAX

003

Attorney Docket No. 00479.85672

We hereby claim the benefit under Title 35, United States Code, §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, §112, We acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application and the national or PCT International filing date of this application:

09/429,643	10/29/99	Pending

**Power of Attorney**

And we hereby appoint, both jointly and severally, as our attorneys with full power of substitution and revocation, to prosecute this application and transact all business in the U.S. Patent and Trademark Office connected herewith as well as before any office or agency of a foreign country or any international organization in connection with any foreign counterpart application claiming priority to this application, including the power to appoint agents and local representatives in connection with such foreign applications, the following attorneys of Banner & Witcoff, their registration numbers being listed after their names:

Robert Alther, Reg. No. 31,810, Donald W. Barner, Reg. No. 17,037; Edward F. McKie, Jr., Reg. No. 17,335; William W. Beckett, Reg. No. 18,262; Dale H. Hoscheit, Reg. No. 19,090; Joseph M. Potenza, Reg. No. 28,175; James A. Niegowski, Reg. No. 28,331; Joseph M. Skerpon, Reg. No. 29,864; Thomas L. Peterson, Reg. No. 30,969; Nina L. Medlock, Reg. No. 29,673; William J. Fisher, Reg. No. 32,133; Thomas H. Jackson, Reg. No. 29,808; Franklin D. Wolffe, Reg. No. 19,724; Susan A. Wolffe, Reg. No. 33,568; Daniel E. Fisher, Reg. No. 34,162; Kevin A. Wolff, Reg. No. 42,233 and Bradley C. Wright, Reg. No. 38,061.

All correspondence and telephone communications should be addressed to:

Banner & Witcoff, Ltd.  
Eleventh Floor  
1001 G Street, N.W.  
Washington, D.C. 20001-4597  
Tel. No. (202) 508-9100

We hereby declare that all statements made herein of our 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 18 U.S.C. 1001 and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature *Edmund Colby Munger* Date 15 FEB 2000

Full Name of Joint Inventor MUNGER Edmund Colby

Family Name First Given Name Second Given Name

Residence 1101 Opaca Court, Crownsville, Maryland 21032

Citizenship U.S.

Post Office Address 1101 Opaca Court, Crownsville, Maryland 21032

02/15/00 14:23 FAX

0004

Attorney Docket No. 00479.85672

Signature \_\_\_\_\_ Date \_\_\_\_\_

Full Name of Joint Inventor SCHMIDT Douglas Charles  
Family Name First Given Name Second Given Name

Residence 230 Oak Court, Severna Park, Maryland 21146

Citizenship U.S.

Post Office Address 230 Oak Court, Severna Park, Maryland 21146

Signature \_\_\_\_\_ Date \_\_\_\_\_

Full Name of Joint Inventor SHORT Robert Dunham, III  
Family Name First Given Name Second Given Name

Residence 38710 Goose Creek Lane, Leesburg, Virginia 20175

Citizenship U.S.

Post Office Address 38710 Goose Creek Lane, Leesburg, Virginia 20175

Signature \_\_\_\_\_ Date \_\_\_\_\_

Full Name of Joint Inventor LABSON Victor  
Family Name First Given Name Second Given Name

Residence 12026 Lisa Marie Court, Fairfax, Virginia 22033

Citizenship U.S.

Post Office Address 12026 Lisa Marie Court, Fairfax, Virginia 22033





in Data Sheet



UNITED STATES DEPARTMENT OF COMMERCE  
Patent and Trademark Office

Address: COMMISSIONER OF PATENTS AND TRADEMARKS  
Washington, D.C. 20231

<b>SERIAL NUMBER</b> 09/504,783	<b>FILING DATE</b> 02/15/2000 <b>RULE</b> -	<b>CLASS</b> 709	<b>GROUP ART UNIT</b> 2755	<b>ATTORNEY DOCKET NO.</b> 00479.85672
------------------------------------	--	---------------------	-------------------------------	---

**APPLICANTS**

Edmund Colby Murger, Crownsville, MD ;  
 Douglas Charles Schmidt, Severna Park, MD ;  
 Robert Dunham Shott III, Leesburg, VA ;  
 Victor Larson, Fairfax, VA ;  
 -Michael Williamson, South Riding, VA ;

**\* CONTINUING DATA \*\*\*\*\***

THIS APPLN CLAIMS BENEFIT OF 60/106,261 10/30/1998  
 WHICH CLAIMS BENEFIT OF 60/137,704 06/07/1999  
 WHICH IS A CIP OF 09/129,643 10/29/1999  
 WHICH CLAIMS BENEFIT OF 60/137,704 06/07/1999

*Yes Kc*

**\*\* FOREIGN APPLICATIONS \*\* \*\*\*\*\***

**IF REQUIRED, FOREIGN FILING LICENSE GRANTED \*\***

\*\* 04/28/2000

Foreign Priority claimed <input type="checkbox"/> yes <input checked="" type="checkbox"/> no	<b>STATE OR COUNTRY</b> MD	<b>SHEETS DRAWING</b> 35	<b>TOTAL CLAIMS</b> 71	<b>INDEPENDENT CLAIMS</b> 9
35 USC 119 (a-d) conditions met <input type="checkbox"/> yes <input checked="" type="checkbox"/> no <input type="checkbox"/> Met after Allowance				
Verified and Acknowledged Examiner's Signature: _____ Initials: _____				

**ADDRESS**

Banner & Witcoff, Ltd  
 1001 G Street, NW  
 Washington, DC 20001-4597

**TITLE**

File network protocol for secure communications with assured system availability

<b>FILING FEE RECEIVED</b> 2076	FEES: Authority has been given in Paper No. _____ to charge/credit DEPOSIT ACCOUNT No. _____ for following:	<input type="checkbox"/> All Fees
		<input type="checkbox"/> 1.16 Fees ( Filing ) <input type="checkbox"/> 1.17 Fees ( Processing Ext. of time ) <input type="checkbox"/> 1.18 Fees ( Issue ) <input type="checkbox"/> Other _____ <input type="checkbox"/> Credit

PATENT APPLICATION SERIAL NO. \_\_\_\_\_

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

02/24/2000 NVILLARI 00000027 09504783

01 FC:101	690.00 OP
02 FC:102	468.00 OP
03 FC:103	918.00 OP

PTO-1556  
(5/87)

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



**PATENT APPLICATION FEE DETERMINATION RECORD**

Effective December 29, 1999

Application or Docket Number

09/004, 783

**CLAIMS AS FILED - PART I**

(Column 1) (Column 2)

FOR	NUMBER FILED	NUMBER EXTRA
BASIC FEE		
TOTAL CLAIMS	71 minus 20 = *	51
INDEPENDENT CLAIMS	9 minus 3 = *	6
MULTIPLE DEPENDENT CLAIM PRESENT		

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

**CLAIMS AS AMENDED - PART II**

(Column 1) (Column 2) (Column 3)

AMENDMENT A	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
<i>A</i>			
Total	* 17 Minus	** 71	=
Independent	* 2 Minus	*** 9	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM			

AMENDMENT B	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
<i>B</i>			
Total	* 27 Minus	** 71	=
Independent	* <del>2</del> Minus	*** 9	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM			

AMENDMENT C	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
<i>C</i>			
Total	* 37 Minus	** 27	= 10
Independent	* 8 Minus	*** 4	= 4
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM			

\* 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.

SMALL ENTITY TYPE  OR

RATE	FEE
	345.00
X\$ 9=	
X39=	
+130=	
TOTAL	

OTHER THAN SMALL ENTITY

RATE	FEE
	690.00
X\$18=	918
X78=	468
+260=	
TOTAL	2,076

SMALL ENTITY OR

RATE	ADDITIONAL FEE
X\$ 9=	
X39=	
+130=	
TOTAL ADDIT. FEE	

OTHER THAN SMALL ENTITY

RATE	ADDITIONAL FEE
X\$18=	
X78=	
+260=	
TOTAL ADDIT. FEE	

RATE	ADDITIONAL FEE
X\$ 9=	
X39=	
+130=	
TOTAL ADDIT. FEE	

RATE	ADDITIONAL FEE
X\$18=	
X78=	
+260=	
TOTAL ADDIT. FEE	

RATE	ADDITIONAL FEE
X\$ 9=	
X39=	
+130=	
TOTAL ADDIT. FEE	

RATE	ADDITIONAL FEE
X\$18=	180
X78=	336
+260=	
TOTAL ADDIT. FEE	516

09/504793

**FEE CALCULATION SHEET  
(FOR USE WITH FORM PTO-875)**

APPLICANT(S)

**CLAIMS**

	AS FILED		AFTER 1st AMENDMENT		AFTER 2nd AMENDMENT								
	IND.	DEP.	IND.	DEP.	IND.	DEP.		IND.	DEP.	IND.	DEP.	IND.	DEP.
1	/						51		/				
2		/					52		/				
3		/					53		/				
4		/					54		/				
5		/					55		/				
6		/					56		/				
7		/					57		/				
8		/					58		/				
9		/					59		/				
10		/					60	/					
11		/					61		/				
12		/					62		/				
13		/					63		/				
14	/						64	/					
15		/					65		/				
16		/					66		/				
17		/					67	/					
18		/					68		/				
19		/					69		/				
20		/					70		/				
21		/					71		/				
22		/					72						
23		/					73						
24		/					74						
25		/					75						
26		/					76						
27		/					77						
28	/						78						
29		/					79						
30		/					80						
31		/					81						
32		/					82						
33		/					83						
34		/					84						
35		/					85						
36		/					86						
37	/						87						
38		/					88						
39		/					89						
40	/						90						
41		/					91						
42		/					92						
43		/					93						
44		/					94						
45		/					95						
46		/					96						
47		/					97						
48		/					98						
49		/					99						
50	/						100						
TOTAL IND.							TOTAL IND.	9					
TOTAL DEP.							TOTAL DEP.	62					
TOTAL CLAIMS							TOTAL CLAIMS	71					

PTO-1360 (3-78)

\*MAY BE USED FOR ADDITIONAL CLAIMS OR AMENDMENTS

U.S. DEPARTMENT OF COMMERCE  
Patent and Trademark Office

## INTERNATIONAL SEARCH REPORT

International application No.  
PCT/US00/08219

<b>A. CLASSIFICATION OF SUBJECT MATTER</b>		
IPC(7) : G06F 11/00 US CL : 713/201		
According to International Patent Classification (IPC) or to both national classification and IPC		
<b>B. FIELDS SEARCHED</b>		
Minimum documentation searched (classification system followed by classification symbols)		
U.S. : 713/201,200,202; 340/825.31,825.34; 380/255;		
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 US PATENT FILE; WEST; JPAB; EPAB; DWPI; TDBD;		
<b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b>		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,805,801 A (HOLLOWAY ET AL) 08 SEPTEMBER 1998, Entire document.	1-25
Y	US 5,796,942 A (ESBENSEN) 18 AUGUST 1998, Entire document.	1-25
Y,P	US 5,905,859 A (HOLLOWAY ET AL) 18 MAY 1999, Entire document.	1-25
Y	US 5,892,903 A (KLAUS) 06 APRIL 1999, Entire document.	1-25
A	US 5,537,099 A (LIANG) 16 JULY 1996, Entire document.	1-25
A	US 5,278,901 A (SHIEH ET AL) 11 JANUARY 1994, Entire document.	1-25
<input checked="" type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.		
* *A* *E* *L* *O* *P*	Special categories of cited documents: document defining the general state of the art which is not considered to be of particular relevance earlier document published on or after the international filing date 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) document referring to an oral disclosure, use, exhibition or other means document published prior to the international filing date but later than the priority date claimed	*T* *X* *Y* *Z*
		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 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 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		Date of mailing of the international search report
20 JULY 2000		22 AUG 2000 <i>Rajendra Zogar</i>
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231		Authorized officer
Facsimile No. (703) 305-3230		NADEEM IQBAL
		Telephone No. (703) 308-5228

INTERNATIONAL SEARCH REPORT

International application No.  
PCT/US00/08219

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A,P	US 5,991,881 A (CONKLIN ET AL) 23 NOVEMBER 1999, Entire document.	1-25

INTERNATIONAL SEARCH REPORT

International application No.  
PCT/SE 00/02565

A. CLASSIFICATION OF SUBJECT MATTER

IPC7: H04L 12/46, H04L 12/56, H04L 9/00  
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)

IPC7: H04L, G09F, H04Q

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5898830 A (R.E.WESINGER, JR. ET AL), 27 April 1999 (27.04.99), column 3, line 47 - column 4, line 52, figure 1, claims 1-10, abstract, cited in Application --	1-20
A	C. HUITEMA: An Experiment in DNS BasedIP Routing. K B Labs Kashpureff Boling Laboratories, Inc., Network Working Group, rfc 1383, INRIA dec. 1992. <a href="http://www.kblabs.com/lab/lib/rfcs/1300/rfc1383.txt.html">http://www.kblabs.com/lab/lib/rfcs/1300/rfc1383.txt.html</a> --	1-20
A	WO 9859470 A2 (TELEFONAKTIEBOLAGET LM ERICSSON (PUBL)), 30 December 1998 (30.12.98), page 1, line 13 - page 3, line 16, figures 1-2, claims 1-12 --	1, 20

Further documents are listed in the continuation of Box C.  See patent family annex.

* Special categories of cited documents:	"I" 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
"A" document defining the general state of the art which is not considered to be of particular relevance	"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
"E" earlier application or patent but published on or after the international filing date	"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
"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)	"&" document member of the same patent family
"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 claim.ed	

Date of the actual completion of the international search <b>17 April 2001</b>	Date of mailing of the international search report <b>18 -04- 2001</b>
---	---

Name and mailing address of the ISA/ Swedish Patent Office Box 5055, S-102 42 STOCKHOLM Facsimile No. +46 8 666 02 86	Authorized officer  Roger Bou Faisal/LR Telephone No. +46 8 782 25 00
--	--

INTERNATIONAL SEARCH REPORT

International application No.  
PCT/SE 00/02565

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 9726731 A1 (RAPTOR SYSTEMS, INC.), 24 July 1997 (24.07.97), see whole document  -----  -----	1,20

Form PCT/ISA/210 (continuation of second sheet) (July 1998)

**INTERNATIONAL SEARCH REPORT**  
Information on patent family members

25/02/01

International application No.  
PCT/SE 00/02565

Patent document cited in search report			Publication date	Patent family member(s)		Publication date
US	5898830	A	27/04/99	US	6052788 A	18/04/00
WO	9859470	A2	30/12/98	AU	8052398 A	04/01/99
				SE	9702385 A	24/12/98
WO	9726731	A1	24/07/97	AU	2242697 A	11/08/97

Form PCT/ISA/210 (patent family annex) (July 1998)



09/504783  
 02/15/00  
 709 225  
 Class: 709  
 Subclass: 225  
 ISSUE CLASSIFICATION

PATENT NUMBER  
 6502135  
 6502135

U.S. UTILITY Patent Application

O.I.P.E. PATENT DATE  
 SCANNED: [Signature] DEC 31 2007

APPLICATION NO.	CONT/PRIOR	CLASS	SUBCLASS	ART UNIT	EXAMINER
09/504783	D	709	225	2153	[Signature]

APPLICANTS: [Signature] Mungler, [Signature] Schmidt, [Signature] Short, [Signature] Larson  
 RSD

Mobile network protocol for secure communications with assured system availability

Certificate

SEP 09 2003

PTO-6040  
12/99

of Correction

ISSUING CLASSIFICATION			
ORIGINAL		CROSS REFERENCE(S)	
CLASS	SUBCLASS	CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)
709	225	709	229 245
INTERNATIONAL CLASSIFICATION			
G06F 15/173			

Continued on Issue Slip Inside File Jacket

<input type="checkbox"/> <b>TERMINAL DISCLAIMER</b>	<b>DRAWINGS</b> Sheets Drwg. 35 Figs. Drwg. 35 Print Fig. 2			<b>CLAIMS ALLOWED</b> Total Claims 17 Print Claim for O.G. 1	
	The term of this patent subsequent to (date) has been disclaimed.			<b>NOTICE OF ALLOWANCE MAILED</b> 7-3-02	
<input type="checkbox"/> The term of this patent shall not extend beyond the expiration date of U.S. Patent No.	KRISNA LIM PRIMARY EXAMINER 7/1/02			<b>ISSUE FEE</b> (w) Amount Due \$1280.- Date Paid 9-30-02	
	The term of this patent shall not extend beyond the expiration date of this patent.			<b>ISSUE BATCH NUMBER</b> 7-12-02	

**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. 8/99)  
 FILED WITH:  DISK (CRF)  FICHE  CD-ROM (Attached in pocket on right inside flap)

Formal drawings sheet 55  
 [Signature]



SEARCHED			
Class	Sub.	Date	Exmr.
709 713	249 22B 201	3/9/02	KC
709	225 229 245	3/10/02	KC
updated above 7/1/02 KC			

SEARCH NOTES (INCLUDING SEARCH STRATEGY)		
	Date	Exmr.
Easy	3/9/02	KC
Easy	6/28/02	KC

INTERFERENCE SEARCHED			
Class	Sub.	Date	Exmr.
709	225 229 245	7/1/02	KC

ISSUE SLIP STAPLE HERE (for additional cross references)

POSITION	INITIALS	ID NO.	DATE
FEE DETERMINATION	WJ	7539	02-26-00
O.P.E. CLASSIFIER	M.T.V	5A	03-06-00
FORMALITY REVIEW			
RESPONSE FORMALITY REVIEW	CG	WAL65	4-28-00

INDEX OF CLAIMS

- ✓ ..... Rejected
- = ..... Allowed
- (Through numeral)... Canceled
- + ..... Restricted
- N ..... Non-elected
- I ..... Interference
- A ..... Appeal
- O ..... Objected

Claim	Final	Original	Date
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28	✓		
29	✓		
30	✓		
31	✓		
32	✓		
33	✓		
34	✓		
35	✓		
36	✓		
37	✓		
38	0		
39	0		
40			
41			
42			
43			
44			
45			
46			
47			
48			
49			
50			

Claim	Final	Original	Date
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
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	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			

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

(LEFT INSIDE)

G-2700

0210 #2  
2754

**PATENT APPLICATION**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

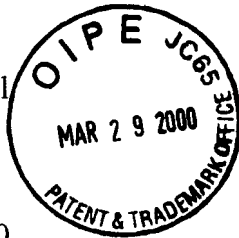
**RECEIVED**

In re application of

Edmund C. Munger et al

Appln. No. 09/504,783

Filed: February 15, 2000



Attorney Docket No.: 00479.85672

Group Art Unit:

Examiner:

JUL 26 2000

Group 2700

For: **IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY**

**INFORMATION DISCLOSURE STATEMENT**  
**UNDER 37 CFR §§ 1.97 and 1.98**

Assistant Commissioner for Patents  
Washington, D.C. 20231

Sir:

In accordance with the duty of disclosure under 37 CFR §1.56, Applicants hereby notify the U.S. Patent and Trademark Office of the documents which are listed on the attached Form PTO-1449 and/or listed herein and which the Examiner may deem relevant to patentability of the claims of the above-identified application.

1. U.S. Patent 4,933,846.
2. U.S. Patent 5,842,040.
3. Reiter, Michael K. and Rubin, Aviel D. (AT&T Labs - Research), "Crowds: Anonymity for Web Transactions", pages 1-23.
4. Dolev, Shlomi and Ostrovsky, Rafail, "Efficient Anonymous Multicast and Reception" (Extended Abstract), 16 pages.
5. Rubin, Aviel D., Geer, Daniel, and Ranum, Marcus J. (Wiley Computer Publishing), "Web Security Sourcebook", pages 82-94.

INFORMATION DISCLOSURE STATEMENT

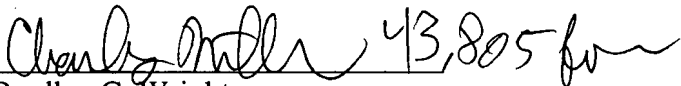
09/504,783

00479.85672

The present Information Disclosure Statement is being filed before the mailing date of the first Office Action on the merits, and therefore no certification under 37 CFR §1.97(e) or fee under 37 CFR §1.17(p) is required.

The submission of the listed documents is not intended as an admission that any such document constitutes prior art against the claims of the present application. Applicants do not waive any right to take any action that would be appropriate to antedate or otherwise remove any listed document as a competent reference against the claims of the present application.

Respectfully submitted,

  
Bradley C. Wright  
Reg. No. 38,061

Banner & Witcoff, Ltd.  
Eleventh Floor  
1001 G Street, N.W.  
Washington, D.C. 20001-4597  
(202) 508-9100

Dated: 3/28/00

INTERNATIONAL SEARCH REPORT

International Application No  
PCT/US 99/25325

COPY

A. CLASSIFICATION OF SUBJECT MATTER  
IPC 7 H04L29/06

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)  
IPC 7 H04L

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 practical, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>FASBENDER A ET AL: "VARIABLE AND SCALABLE SECURITY: PROTECTION OF LOCATION INFORMATION IN MOBILE IP" IEEE VEHICULAR TECHNOLOGY CONFERENCE, US, NEW YORK, IEEE, vol. CONF. 46, 1996, pages 963-967, XP000593113 ISBN: 0-7803-3158-3 the whole document</p> <p>-----</p>	1-67

Further documents are listed in the continuation of box C.

Patent family members are listed in 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 but 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

20 July 2000

Date of mailing of the international search report

27/07/2000

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
Fax: (+31-70) 340-3016

Authorized officer

Canosa Aresté, C

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In Re Application of:

Edmund Colby Munger, et al.

Serial No. 09/504,783

Filed: February 15, 2000

For: IMPROVEMENTS TO AN AGILE NETWORK  
PROTOCOL FOR SECURE COMMUNICATIONS  
WITH ASSURED SYSTEM AVAILABILITY



Group Art Unit:

Examiner:

Attorney Docket: 00479.85672

RECEIVED  
SEP 27 2000  
100 MAIL ROOM

**INFORMATION DISCLOSURE STATEMENT**

Commissioner of the U.S. Patent and Trademark Office  
Washington, D.C. 20231

Sir:

In accordance with 37 C.F.R. 1.97 and 1.98, enclosed is a PTO Form-1449 listing art for consideration by the Examiner and a copy of the identified document. The International Searching Authority cited this document for corresponding International Application Nos. PCT/US99/25323 and PCT/US99/25325 on July 27, 2000, and was not previously cited in the subject application. Copies of the International Search Report listing the relevant art are attached.

The accompanying Information Disclosure Statement is being filed before the mailing date of the first Office Action on the merits, and therefore no certification or fee is believed to be required. However, if a fee is required, please charge our Deposit Account No. 19-0733.

The submission of this document is not intended as an admission that any such documents constitutes prior art against the claims of the present application.

**Information Disclosure Statement**

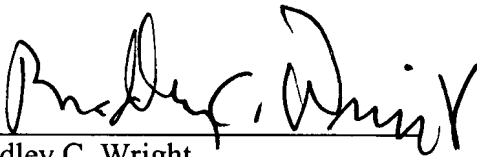
Serial No. 09/504,783

Applicants do not waive any right to take any action that would be appropriate to antedate or otherwise remove any listed documents as a competent reference against the claims of the present application.

Consideration of this information is respectfully requested.

Respectfully submitted,

Date: September 25, 2000

By:   
Bradley C. Wright  
Registration No. 38,061

Banner & Witcoff, Ltd.  
1001 G Street, N.W., Eleventh Floor  
Washington, D.C. 20001-4597  
(202) 508-9100

BCW:pp



#3  
2153  
W Meredith  
PATENT 6/13/01

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	:	
Edmund Colby Munger et al.	:	
Application No. 09/504,783	:	Group Art Unit: 2100
Filed: February 15, 2000	:	Examiner: G. Burgess
For: IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY	:	Atty Docket: 00479.85672

RECEIVED  
JUN 11 2001  
Technology Center 2100

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT

Assistant Commissioner for Patents  
Washington, D.C. 20231

Sir:

Pursuant to the duty of disclosure under 37 CFR §§ 1.56 and 1.97-1.98, the document listed on the attached Form PTO-1449 is being brought to the attention of the Examiner in charge of the above-identified application. A copy of the document is enclosed.

The Examiner is respectfully requested to initial the space adjacent the document entry on the Form PTO-1449, and to return a copy of the initialed Form PTO-1449 to confirm that the document has been considered and has been officially made of record in this application.

If the Examiner has any questions or wishes to discuss this application, the Examiner is invited to telephone the undersigned representative at the number set forth below.



Any fees required for consideration of this paper is authorized to be charged to our Deposit

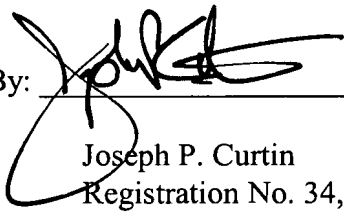
Account No. 19-0733.

Respectfully submitted,

BANNER & WITCOFF, LTD.

Date: June 8, 2001

1001 G Street N.W.  
11th Floor  
Washington, D.C. 20001  
(202) 508-9100

By:   
Joseph P. Curtin  
Registration No. 34,571

17-6



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER OF PATENTS AND TRADEMARKS  
Washington, D.C. 20231  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/504,783	02/15/2000	Edmund Colby Munger	00479.85672	8308

7590 12/28/2001

Banner & Witcoff, Ltd  
1001 G Street, NW  
Washington, DC 20001-4597

EXAMINER

LIM, KRISNA

ART UNIT	PAPER NUMBER
2153	

DATE MAILED: 12/28/2001 #4

Please find below and/or attached an Office communication concerning this application or proceeding.

1-1-G

<b>Office Action Summary</b>	<b>Application No.</b> 09/504,783	<b>Applicant(s)</b> MUNGER ET AL.	
	<b>Examiner</b> Krishna Lim	<b>Art Unit</b> 2153	

-- 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 1 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 \_\_\_\_\_ .
- 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) 1-71 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) \_\_\_\_\_ is/are rejected.
- 7)  Claim(s) \_\_\_\_\_ is/are objected to.
- 8)  Claim(s) 1-71 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).
- 11)  The proposed drawing correction filed on \_\_\_\_\_ is: a)  approved b)  disapproved by the Examiner.  
If approved, corrected drawings are required in reply to this Office action.
- 12)  The oath or declaration is objected to by the Examiner.

**Priority under 35 U.S.C. §§ 119 and 120**

- 13)  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.
- 14)  Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).
  - a)  The translation of the foreign language provisional application has been received.
- 15)  Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

**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) Paper No(s) \_\_\_\_\_ .
- 4)  Interview Summary (PTO-413) Paper No(s). \_\_\_\_\_ .
- 5)  Notice of Informal Patent Application (PTO-152)
- 6)  Other: \_\_\_\_\_ .

Art Unit: 2153

1. Claims 1-71 are presented for examination.
2. Restriction to one of the following inventions is required under 35 U.S.C. § 121:
  - I. Claims 1-27 and 60-66, drawn to a system for transmitting data packet between computers, comprising: a) assigning a weight value ..., b) selecting one of ... transmission paths ..., c) measuring the transmission quality ..., d) adjusting downwardly to a non-zero value ..., classified in Class 709, subclass 241.
  - II. Claims 28-39 and 67-71, drawn to a system for transparently creating a virtual private network (VPN) between a client computer and a target computer, comprising: a) generating from the client computer a DNS request ..., b) determining whether the DNS ..., c) determining that the DNS ..., classified in Class 709, subclass 249.
  - III. Claims 40-59, drawn to a system of preventing data packet received from high bandwidth link from flooding a low bandwidth link, comprising: a) receiving data packet from the high bandwidth link ..., b) determining whether the data packet is validly addressed ..., c) determining whether the data packet is not validly addressed ..., classified in Class 370, subclass 351.
3. Inventions I and II are related as subcombinations disclosed as usable together in a single combination. The subcombinations are distinct from each other if they are shown to be separately usable. In the instant case, invention I has separate utility such as a system for transmitting data packet between computers lacks of: a) generating from the client computer a DNS request ..., b) determining whether the DNS ..., c)

Art Unit: 2153

determining that the DNS ..... See MPEP § 805.05(d).

4. Inventions I and III are related as subcombinations disclosed as usable together in a single combination. The subcombinations are distinct from each other if they are shown to be separately usable. In the instant case, invention I has separate utility such as a system for transmitting data packet between computers lacks of: a) receiving data packet from the high bandwidth link ..., b) determining whether the data packet is validly addressed ..., c) determining whether the data packet is not validly addressed ... See MPEP § 805.05(d).

5. Inventions II and III are related as subcombinations disclosed as usable together in a single combination. The subcombinations are distinct from each other if they are shown to be separately usable. In the instant case, invention II has separate utility such as a system for transparently creating a virtual private network (VPN) between a client computer and a target computer lacks of: a) receiving data packet from the high bandwidth link ..., b) determining whether the data packet is validly addressed ..., c) determining whether the data packet is not validly addressed .... See MPEP § 805.05(d).

6. These inventions are distinct for the reasons given above, and the search required for each Group is different and not co-extensive for examination purpose.

7. For example, the searches for the four inventions would not be co-extensive because these groups would require different searches on PTO's classification class and subclass as following:

1) The Group I search (claims 1-27 and 6-66) would require use of search class 709, subclass 241 ( which would not required for the groups II and III).

Art Unit: 2153

2) The Group II search (claims 28-39 and 67-71) would require use of search class 709, subclass 249 ( which would not required for the groups I and III).

3) The Group III search (claims 40-59) would require use of search class 370, subclass 351 ( which would not required for the groups I and II).

7. Applicant is advised that the response to this requirement to be complete must include an election of the invention to be examined even though the requirement be traversed.

8. Applicant is reminded that the required for response to this requirement is **30 days, not one month.**

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Examiner Krisna Lim whose telephone number is (703) 305-9672. The examiner can normally be reached on Monday-Friday from 7:00 to 3:30.

The fax phone numbers for the organization where this application or proceeding is assigned is are as following:

(703) 746-7238 [After Final Communication]

or

(703) 746-7239 [Official Communication]

(703) 746-7240 [For Status inquires, draft communication]

and/or

(703) 306-5631, (703) 306-5632 or (703) 306-5633 for [Customer Service Numbers]

Art Unit: 2153

Any inquiry of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is (703) 305-3900.

All Internet e-mail communication will be made of record in the application file. PTO employees do not engage in Internet communications where there exists a possibility that sensitive information could be identified or exchanged unless the record includes a properly signed express waiver of the confidentiality requirement of 35 U.S.C. 122. This is more clearly set forth in the Interim Internet Usage Policy published in the Office Gazette of the Patent and Trademark on February 25, 1997 at 1195 OG 89.

kl

December 22, 2001



**KRISNA LIM  
PRIMARY EXAMINER**



PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	:	
Edmund Colby Munger et al.	:	
Application No. 09/504,783	:	Group Art Unit: 2100
Filed: February 15, 2000	:	Examiner: G. Burgess
For: IMPROVEMENTS TO AN AGILE	:	Atty Docket: 00479.85672
NETWORK PROTOCOL FOR	:	
SECURE COMMUNICATIONS	:	
WITH ASSURED SYSTEM	:	
AVAILABILITY	:	

#5/A  
1-31-02  
RECEIVED  
JAN 31 2002  
Technology Center 2100

AMENDMENT AND RESPONSE UNDER 37 U.S.C. § 1.111

Assistant Commissioner for Patents  
Washington, D.C. 20231

Sir:

In response to the Office Action mailed December 28, 2001, Applicants submit the following response. Any fees required for consideration of this paper is authorized to be charged to our Deposit Account No. 19-0733.

IN THE CLAIMS:

Please cancel claims 1-27 and 40-66.



**Remarks**

Applicants are in receipt of the Office Action mailed December 28, 2001. The Office Action restricts the claims into the following groups:

Group I: Claims 1-27 and 60-66

Group II: Claims 28-39 and 67-71

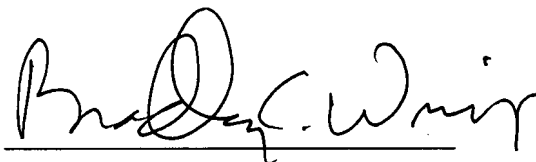
Group III: Claims 40-59

Applicants elect Group II without traverse. If the Examiner has any questions or wishes to discuss this application, the Examiner is invited to telephone the undersigned representative at the number set forth below.

Respectfully submitted,

BANNER & WITCOFF, LTD.

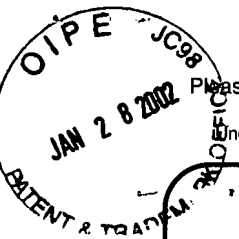
By:



Bradley C. Wright  
Registration No. 38,061

<sup>28</sup>  
Date: January ~~24~~, 2002

101 G Street N.W.  
11th Floor  
Washington, D.C. 20001  
(202) 508-9100



Please type a plus sign (+) inside this box →

PTO/SB/21 (08-00)

Approved for use through 10/31/2002. OMB 0651-0031

U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

RECEIVED  
JAN 31 2002  
Technology Center 2100

<b>TRANSMITTAL FORM</b> <i>(to be used for all correspondence after initial filing)</i>	<b>Application Number</b>	09/504,783	
	<b>Filing Date</b>	February 15, 2000	
	<b>First Named Inventor</b>	Edmund Colby Munger	
	<b>Group Art Unit</b>	2100	
	<b>Examiner Name</b>	G. Burgess <b>2153</b>	
<b>Total Number of Pages in This Submission</b>	2	<b>Attorney Docket Number</b>	00479.85672

ENCLOSURES <i>(check all that apply)</i>		
<input type="checkbox"/> Fee Transmittal Form <input type="checkbox"/> Fee Attached <input checked="" type="checkbox"/> Amendment / Response <input type="checkbox"/> After Final <input type="checkbox"/> Affidavits/declaration(s) <input type="checkbox"/> Extension of Time Request <input type="checkbox"/> Express Abandonment Request <input type="checkbox"/> Information Disclosure Statement <input type="checkbox"/> Certified Copy of Priority Document(s) <input type="checkbox"/> Response to Missing Parts/ Incomplete Application <input type="checkbox"/> Response to Missing Parts under 37 CFR 1.52 or 1.53	<input type="checkbox"/> Assignment Papers <i>(for an Application)</i> <input type="checkbox"/> Drawing(s) <input type="checkbox"/> Licensing-related Papers <input type="checkbox"/> Petition <input type="checkbox"/> Petition to Convert to a Provisional Application <input type="checkbox"/> Power of Attorney, Revocation Change of Correspondence Address <input type="checkbox"/> Terminal Disclaimer <input type="checkbox"/> Request for Refund <input type="checkbox"/> CD, Number of CD(s) _____	<input type="checkbox"/> After Allowance Communication to Group <input type="checkbox"/> Appeal Communication to Board of Appeals and Interferences <input type="checkbox"/> Appeal Communication to Group <i>(Appeal Notice, Brief, Reply Brief)</i> <input type="checkbox"/> Proprietary Information <input type="checkbox"/> Status Letter <input type="checkbox"/> Other Enclosure(s) <i>(please identify below):</i>
Remarks		

RECEIVED  
JAN 30 2002  
Technology Center 2600

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT	
Firm or Individual name	Bradley C. Wright, Reg. No. 38,061
Signature	<i>Bradley C. Wright</i> , Reg. No. 49,024
Date	January 28, 2002

CERTIFICATE OF MAILING			
I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on this date: <input type="text"/>			
Typed or printed name	<input type="text"/>		
Signature	<input type="text"/>	Date	<input type="text"/>

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be send to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.



PATENT

#4

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	:	
Edmund Colby Munger et al.	:	
Application No. 09/504,783	:	Group Art Unit: 2153
Filed: February 15, 2000	:	Examiner: Krisna Lim
For: IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY	:	Atty Docket: 00479.85672

RECEIVED  
FEB 06 2002  
Technology Center 2100

SUPPLEMENTAL AMENDMENT AND RESPONSE UNDER 37 U.S.C. § 1.111

Assistant Commissioner for Patents  
Washington, D.C. 20231

Sir:

A response to the Office Action mailed December 28, 2001 was filed on January 28, 2002 with incorrect examiner information. Attached is a courtesy copy of the paper as filed. This response has the correct examiner information.

In response to the Office Action mailed December 28, 2001, Applicants submit the following response. Any fees required for consideration of this paper is authorized to be charged to our Deposit Account No. 19-0733.

IN THE CLAIMS:

Please cancel claims 1-27 and 40-66. /

**Remarks**

Applicants are in receipt of the Office Action mailed December 28, 2001. The Office Action restricts the claims into the following groups:

- Group I: Claims 1-27 and 60-66
- Group II: Claims 28-39 and 67-71 ✓
- Group III: Claims 40-59

Applicants elect Group II without traverse. If the Examiner has any questions or wishes to discuss this application, the Examiner is invited to telephone the undersigned representative at the number set forth below.

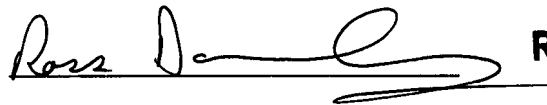
Respectfully submitted,

BANNER & WITCOFF, LTD.

Date: January 30, 2002

101 G Street N.W.  
11th Floor  
Washington, D.C. 20001  
(202) 508-9100

By:



Bradley C. Wright  
Registration No. 38,061

**U.S.P.T.O.**  
**Reg. No. 49,024**



PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

RECEIVED  
FEB 06 2002  
Technology Center 2100

In re Application of:	:	
	:	
Edmund Colby Munger et al.	:	
	:	
Application No. 09/504,783	:	Group Art Unit: 2100
	:	
Filed: February 15, 2000	:	Examiner: G. Burgess
	:	
For: IMPROVEMENTS TO AN AGILE	:	Atty Docket: 00479.85672
NETWORK PROTOCOL FOR	:	
SECURE COMMUNICATIONS	:	
WITH ASSURED SYSTEM	:	
AVAILABILITY	:	

AMENDMENT AND RESPONSE UNDER 37 U.S.C. § 1.111

Assistant Commissioner for Patents  
Washington, D.C. 20231

Sir:

In response to the Office Action mailed December 28, 2001, Applicants submit the following response. Any fees required for consideration of this paper is authorized to be charged to our Deposit Account No. 19-0733.

IN THE CLAIMS:

Please cancel claims 1-27 and 40-66.

**Remarks**

Applicants are in receipt of the Office Action mailed December 28, 2001. The Office Action restricts the claims into the following groups:

Group I: Claims 1-27 and 60-66

Group II: Claims 28-39 and 67-71

Group III: Claims 40-59

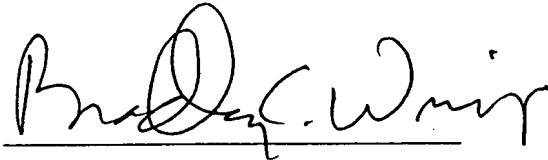
Applicants elect Group II without traverse. If the Examiner has any questions or wishes to discuss this application, the Examiner is invited to telephone the undersigned representative at the number set forth below.

Respectfully submitted,

BANNER & WITCOFF, LTD.

<sup>28</sup>  
Date: January 24, 2002

By:



101 G Street N.W.  
11th Floor  
Washington, D.C. 20001  
(202) 508-9100

Bradley C. Wright  
Registration No. 38,061

*Handwritten signature and date: 2/27/02*

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Edmund Colby MUNGER *et al.*

Application No. 09/504,783

Filed: February 15, 2000

For: IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

Group Art Unit: 2153

Examiner: K. Lim

Atty Docket: 00479.85672

*Official*

**RECEIVED**  
02/27/02

PRELIMINARY AMENDMENT

Assistant Commissioner for Patents  
Washington, D.C. 20231

Sir:

Applicants submit the following amendment and request its entry prior to examination of the claims. The Office is authorized to charge any required fees for consideration of this paper to our Deposit Account No. 19-0733.

IN THE CLAIMS:

Please add the following new claims:

72. A method for establishing an encrypted channel between a client and a target computer, comprising the steps of:

- (i) intercepting a DNS request sent by the client; and

*B1*

- B1
- (ii) based on the DNS request, establishing the encrypted channel between the client and the target.
73. The method of claim 72, wherein step (ii) comprises steps of:
- determining whether the client is authorized to access the target;
  - when the client is authorized to access the target, initiating the encrypted channel; and
  - when the client is not authorized to access the target, sending an error message to the client.
74. The method of claim 73, wherein step b) comprises sending encrypted channel parameters to the client.
75. The method of claim 72, wherein step (ii) occurs in a communication protocol independently of an application program.
76. The method of claim 72, wherein step (i) comprises a DNS proxy server intercepting the DNS request sent by the client.
77. The method of claim 72, wherein step (ii) comprises establishing the encrypted channel responsive to intercepting a DNS request for a domain name comprising a predetermined domain name extension.
78. A method for establishing an encrypted channel between a client and a secure host, comprising the step of automatically creating the encrypted channel upon intercepting a DNS request for a domain name comprising a predetermined domain name extension.
79. The method of claim 78, wherein the creating step is performed in a communication protocol independently of an application program.



B1

80. A method for establishing an encrypted channel between a client and a secure host, comprising the step of automatically creating the encrypted channel in response to detecting a request for access to a predetermined IP address.

81. The method of claim 80, wherein the creating step is performed in a communication protocol independently of an application program.

**Remarks**

Applicants have added new claims 72-81 to more completely claim the disclosed invention.

Support for the new claims may be found at least on pages 57-62.

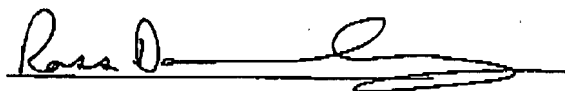
If the Examiner has any questions or wishes to discuss this amendment, the Examiner is invited to telephone the undersigned representative at the number set forth below.

Respectfully submitted,

BANNER & WITCOFF, LTD.

Date: February 5, 2002

By:



Bradley C. Wright  
Registration No. 38,061

**U.S.P.T.O.**  
**Reg. No. 49,024**

11th Floor  
1001 G Street N.W.  
Washington, D.C. 20001  
(202) 508-9100

Approved for use through 10/31/2002. OMB 0851-0031  
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE  
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Official

RECEIVED  
2/5/02 PM

**Certificate of Transmission under 37 CFR 1.8**

I hereby certify that this correspondence is being facsimile transmitted to the Patent and Trademark Office

on February 5, 2002.  
Date

Marilyn M. Davis  
Signature

Marilyn M. Davis  
Typed or printed name of person signing Certificate

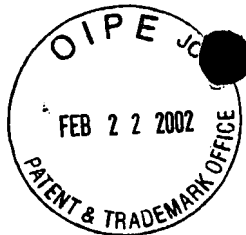
Note: Each paper must have its own certificate of transmission, or this certificate must identify each submitted paper.

To: Examiner K. Lim  
Fax Number: (703) 746-7239

Serial No. 09/504,783  
Filed: February 15, 2000  
Atty. Dkt. 00479.85672

Submission: PRELIMINARY AMENDMENT

Burden Hour Statement: This form is estimated to take 0.03 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.



PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

#9/C  
3-18-02  
m

In re Application of:

Edmund Colby MUNGER *et al.*

Application No. 09/504,783

Group Art Unit: 2153

Filed: February 15, 2000

Examiner: K. Lim

For: IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

Atty Docket: 00479.85672

RECEIVED  
MAR 01 2002  
Technology Center 2100

SECOND PRELIMINARY AMENDMENT

Assistant Commissioner for Patents  
Washington, D.C. 20231

Sir:

Applicants submit the following amendment and request its entry prior to examination of the claims. The Office is authorized to charge any required fees for consideration of this paper to our Deposit Account No. 19-0733.

IN THE CLAIMS:

Please add the following new claims:

82. A data processing device, comprising memory storing a domain name server (DNS) proxy module that intercepts DNS requests sent by a client and, for each intercepted DNS request, performs the steps of:

C1

02/27/2002 JBALINAN 00000096 190733  
01 FC:102 84.00 CH

09/504783  
00000001 90733  
03/18/2002 EMILY 00000001 90733  
01 FC:102 252.00 CH  
02 FC:103 180.00 CH

- (i) determining whether the intercepted DNS request corresponds to a secure server;
- (ii) when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer; and
- (iii) when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.

83. The data processing device of claim 82, wherein step (iii) comprises the steps of:

- (a) determining whether the client is authorized to access the secure server; and
- (b) when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client.

84. The data processing device of claim 83, wherein step (iii) further comprises the step

of:

- (c) when the client is not authorized to access the secure server, returning a host unknown error message to the client.

85. The data processing device of claim 84, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

86. A data processing device, comprising memory storing a domain name server (DNS) proxy module that intercepts DNS requests sent by a client and, for each intercepted DNS request, when the intercepted DNS request corresponds to a secure server, determines whether the client is authorized to access the secure server and, if so, automatically initiates an encrypted channel between the client and the secure server.

87. A computer readable medium storing a domain name server (DNS) proxy module comprised of computer readable instructions that, when executed, cause a data processing device to perform the steps of:

- C1
- (i) intercepting a DNS request sent by a client;
  - (ii) determining whether the intercepted DNS request corresponds to a secure server;
  - (iii) when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer; and
  - (iv) when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.

88. The computer readable medium of claim 87, wherein step (iii) comprises the steps of:

- (a) determining whether the client is authorized to access the secure server; and
- (b) when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client.

89. The computer readable medium of claim 88, wherein step (iii) further comprises the step of:

(c) when the client is not authorized to access the secure server, returning a host unknown error message to the client.

90. The computer readable medium of claim 89, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

91. A computer readable medium comprising computer readable instructions that, when executed, cause a domain name server (DNS) proxy module to intercept DNS requests sent by a client and, for each intercepted DNS request, when the intercepted DNS request corresponds to a secure server, determines whether the client is authorized to access the secure server and, if so, automatically initiates an encrypted channel between the client and the secure server.

---

**Remarks**

Applicants have added new claims 82 - 91 to more completely claim the disclosed invention.

Support for the new claims may be found at least on pages 59-60 and in FIG. 26.

If the Examiner has any questions or wishes to discuss this amendment, the Examiner is invited to telephone the undersigned representative at the number set forth below.

Respectfully submitted,

BANNER & WITCOFF, LTD.

**U.S.P.T.O.**  
**Reg. No. 49,024**

Date: 2/22/02

By: 

Bradley C. Wright  
Registration No. 38,061

11th Floor  
1001 G Street N.W.  
Washington, D.C. 20001  
(202) 508-9100

2153#

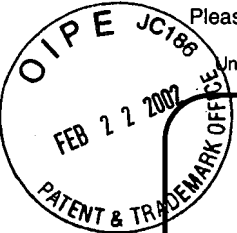
PTO/SB/21 (08-00)

Approved for use through 10/31/2002. OMB 0651-0031

U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

Please type a plus sign (+) inside this box →

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.



<b>TRANSMITTAL FORM</b> <i>(to be used for all correspondence after initial filing)</i>	<b>Application Number</b>	09/504,783
	<b>Filing Date</b>	February 15, 2000
	<b>First Named Inventor</b>	Edmund Colby Munger
	<b>Group Art Unit</b>	2153
	<b>Examiner Name</b>	Krisna Lim
<b>Total Number of Pages in This Submission</b>	<b>Attorney Docket Number</b>	00479.85672

Technology Center 2100  
MAR 01 2002

RECEIVED

**ENCLOSURES (check all that apply)**

<input checked="" type="checkbox"/> Fee Transmittal Form <input type="checkbox"/> Fee Attached <input type="checkbox"/> Amendment / Response <input type="checkbox"/> After Final <input type="checkbox"/> Affidavits/declaration(s) <input type="checkbox"/> Extension of Time Request <input type="checkbox"/> Express Abandonment Request <input checked="" type="checkbox"/> Information Disclosure Statement <input type="checkbox"/> Certified Copy of Priority Document(s) <input type="checkbox"/> Response to Missing Parts/ Incomplete Application <input type="checkbox"/> Response to Missing Parts under 37 CFR 1.52 or 1.53	<input type="checkbox"/> Assignment Papers (for an Application) <input type="checkbox"/> Drawing(s) <input type="checkbox"/> Licensing-related Papers <input type="checkbox"/> Petition <input type="checkbox"/> Petition to Convert to a Provisional Application <input type="checkbox"/> Power of Attorney, Revocation Change of Correspondence Address <input type="checkbox"/> Terminal Disclaimer <input type="checkbox"/> Request for Refund <input type="checkbox"/> CD, Number of CD(s) _____	<input type="checkbox"/> After Allowance Communication Group <input type="checkbox"/> Appeal Communication to Board of Appeals and Interferences <input type="checkbox"/> Appeal Communication to Group (Appeal Notice, Brief, Reply Brief) <input type="checkbox"/> Proprietary Information <input type="checkbox"/> Status Letter <input checked="" type="checkbox"/> Other Enclosure(s) (please identify below): <p style="text-align: center;"><b>Second Preliminary Amendment</b></p>
Remarks		

**SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT**

Firm or Individual name	Bradley C. Wright, Reg. No. 38,061	<b>U.S.P.T.O.</b>
Signature		<b>Reg. No. 49,024</b>
Date	February 22, 2002	

**CERTIFICATE OF MAILING**

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on this date:

Typed or printed name			
Signature		Date	

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be send to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.





# FREE TRANSMITTAL for FY 2002

Patent fees are subject to annual revision.

TOTAL AMOUNT OF PAYMENT (\$) 84

Complete If Known

Application Number	09/504,783
Filing Date	February 15, 2000
First Named Inventor	Edmund Colby Munger
Examiner Name	K. Lim
Group / Art Unit	2153
Attorney Docket No.	000479.85672

RECEIVED  
 MAR 01 2002  
 Technology Center 2100

METHOD OF PAYMENT (check all that apply)	FEE CALCULATION (continued)																																																																																																																																																																														
<input type="checkbox"/> Check <input type="checkbox"/> Credit card <input type="checkbox"/> Money Order <input type="checkbox"/> Other <input type="checkbox"/> None <input checked="" type="checkbox"/> Deposit Account: Deposit Account Number: 19-0733 Deposit Account Name: Banner & Witcoff, Ltd. The Commissioner is authorized to: (check all that apply) <input checked="" type="checkbox"/> Charge fee(s) indicated below <input checked="" type="checkbox"/> Credit any overpayments <input type="checkbox"/> Charge any additional fee(s) during the pendency of this application <input type="checkbox"/> Charge fee(s) indicated below, except for the filing fee to the above-identified deposit account.	<b>3. ADDITIONAL FEES</b> <table border="1"> <thead> <tr> <th>Fee Code</th> <th>Large Entity Fee (\$)</th> <th>Small Entity Fee Code</th> <th>Small Entity Fee (\$)</th> <th>Fee Description</th> <th>Fee Paid</th> </tr> </thead> <tbody> <tr><td>105</td><td>130</td><td>205</td><td>65</td><td>Surcharge - late filing fee or oath</td><td></td></tr> <tr><td>127</td><td>50</td><td>227</td><td>25</td><td>Surcharge - late provisional filing fee or cover sheet.</td><td></td></tr> <tr><td>139</td><td>130</td><td>139</td><td>130</td><td>Non-English specification</td><td></td></tr> <tr><td>147</td><td>2,520</td><td>147</td><td>2,520</td><td>For filing a request for reexamination</td><td></td></tr> <tr><td>112</td><td>920*</td><td>112</td><td>920*</td><td>Requesting publication of SIR prior to Examiner action</td><td></td></tr> <tr><td>113</td><td>1,840*</td><td>113</td><td>1,840*</td><td>Requesting publication of SIR after Examiner action</td><td></td></tr> <tr><td>115</td><td>110</td><td>215</td><td>55</td><td>Extension for reply within first month</td><td></td></tr> <tr><td>116</td><td>400</td><td>216</td><td>200</td><td>Extension for reply within second month</td><td></td></tr> <tr><td>117</td><td>920</td><td>217</td><td>460</td><td>Extension for reply within third month</td><td></td></tr> <tr><td>118</td><td>1,440</td><td>218</td><td>720</td><td>Extension for reply within fourth month</td><td></td></tr> <tr><td>128</td><td>1,960</td><td>228</td><td>980</td><td>Extension for reply within fifth month</td><td></td></tr> <tr><td>119</td><td>320</td><td>219</td><td>160</td><td>Notice of Appeal</td><td></td></tr> <tr><td>120</td><td>320</td><td>220</td><td>160</td><td>Filing a brief in support of an appeal</td><td></td></tr> <tr><td>121</td><td>280</td><td>221</td><td>140</td><td>Request for oral hearing</td><td></td></tr> <tr><td>138</td><td>1,510</td><td>138</td><td>1,510</td><td>Petition to institute a public use proceeding</td><td></td></tr> <tr><td>140</td><td>110</td><td>240</td><td>55</td><td>Petition to revive - unavoidable</td><td></td></tr> <tr><td>141</td><td>1,280</td><td>241</td><td>640</td><td>Petition to revive - unintentional</td><td></td></tr> <tr><td>142</td><td>1,280</td><td>242</td><td>640</td><td>Utility issue fee (or reissue)</td><td></td></tr> <tr><td>143</td><td>460</td><td>243</td><td>230</td><td>Design issue fee</td><td></td></tr> <tr><td>144</td><td>620</td><td>244</td><td>310</td><td>Plant issue fee</td><td></td></tr> <tr><td>122</td><td>130</td><td>122</td><td>130</td><td>Petitions to the Commissioner</td><td></td></tr> <tr><td>123</td><td>50</td><td>123</td><td>50</td><td>Processing fee under 37 CFR 1.17 (q)</td><td></td></tr> <tr><td>126</td><td>180</td><td>126</td><td>180</td><td>Submission of Information Disclosure Stmt</td><td></td></tr> <tr><td>581</td><td>40</td><td>581</td><td>40</td><td>Recording each patent assignment per property (times number of properties)</td><td></td></tr> <tr><td>146</td><td>740</td><td>246</td><td>370</td><td>Filing a submission after final rejection (37 CFR § 1.129(a))</td><td></td></tr> <tr><td>149</td><td>740</td><td>249</td><td>370</td><td>For each additional invention to be examined (37 CFR § 1.129(b))</td><td></td></tr> <tr><td>179</td><td>740</td><td>279</td><td>370</td><td>Request for Continued Examination (RCE)</td><td></td></tr> <tr><td>169</td><td>900</td><td>169</td><td>900</td><td>Request for expedited examination of a design application</td><td></td></tr> </tbody> </table> Other fee (specify) _____ *Reduced by Basic Filing Fee Paid <b>SUBTOTAL (3)</b> (\$ 0)	Fee Code	Large Entity Fee (\$)	Small Entity Fee Code	Small Entity Fee (\$)	Fee Description	Fee Paid	105	130	205	65	Surcharge - late filing fee or oath		127	50	227	25	Surcharge - late provisional filing fee or cover sheet.		139	130	139	130	Non-English specification		147	2,520	147	2,520	For filing a request for reexamination		112	920*	112	920*	Requesting publication of SIR prior to Examiner action		113	1,840*	113	1,840*	Requesting publication of SIR after Examiner action		115	110	215	55	Extension for reply within first month		116	400	216	200	Extension for reply within second month		117	920	217	460	Extension for reply within third month		118	1,440	218	720	Extension for reply within fourth month		128	1,960	228	980	Extension for reply within fifth month		119	320	219	160	Notice of Appeal		120	320	220	160	Filing a brief in support of an appeal		121	280	221	140	Request for oral hearing		138	1,510	138	1,510	Petition to institute a public use proceeding		140	110	240	55	Petition to revive - unavoidable		141	1,280	241	640	Petition to revive - unintentional		142	1,280	242	640	Utility issue fee (or reissue)		143	460	243	230	Design issue fee		144	620	244	310	Plant issue fee		122	130	122	130	Petitions to the Commissioner		123	50	123	50	Processing fee under 37 CFR 1.17 (q)		126	180	126	180	Submission of Information Disclosure Stmt		581	40	581	40	Recording each patent assignment per property (times number of properties)		146	740	246	370	Filing a submission after final rejection (37 CFR § 1.129(a))		149	740	249	370	For each additional invention to be examined (37 CFR § 1.129(b))		179	740	279	370	Request for Continued Examination (RCE)		169	900	169	900	Request for expedited examination of a design application	
Fee Code	Large Entity Fee (\$)	Small Entity Fee Code	Small Entity Fee (\$)	Fee Description	Fee Paid																																																																																																																																																																										
105	130	205	65	Surcharge - late filing fee or oath																																																																																																																																																																											
127	50	227	25	Surcharge - late provisional filing fee or cover sheet.																																																																																																																																																																											
139	130	139	130	Non-English specification																																																																																																																																																																											
147	2,520	147	2,520	For filing a request for reexamination																																																																																																																																																																											
112	920*	112	920*	Requesting publication of SIR prior to Examiner action																																																																																																																																																																											
113	1,840*	113	1,840*	Requesting publication of SIR after Examiner action																																																																																																																																																																											
115	110	215	55	Extension for reply within first month																																																																																																																																																																											
116	400	216	200	Extension for reply within second month																																																																																																																																																																											
117	920	217	460	Extension for reply within third month																																																																																																																																																																											
118	1,440	218	720	Extension for reply within fourth month																																																																																																																																																																											
128	1,960	228	980	Extension for reply within fifth month																																																																																																																																																																											
119	320	219	160	Notice of Appeal																																																																																																																																																																											
120	320	220	160	Filing a brief in support of an appeal																																																																																																																																																																											
121	280	221	140	Request for oral hearing																																																																																																																																																																											
138	1,510	138	1,510	Petition to institute a public use proceeding																																																																																																																																																																											
140	110	240	55	Petition to revive - unavoidable																																																																																																																																																																											
141	1,280	241	640	Petition to revive - unintentional																																																																																																																																																																											
142	1,280	242	640	Utility issue fee (or reissue)																																																																																																																																																																											
143	460	243	230	Design issue fee																																																																																																																																																																											
144	620	244	310	Plant issue fee																																																																																																																																																																											
122	130	122	130	Petitions to the Commissioner																																																																																																																																																																											
123	50	123	50	Processing fee under 37 CFR 1.17 (q)																																																																																																																																																																											
126	180	126	180	Submission of Information Disclosure Stmt																																																																																																																																																																											
581	40	581	40	Recording each patent assignment per property (times number of properties)																																																																																																																																																																											
146	740	246	370	Filing a submission after final rejection (37 CFR § 1.129(a))																																																																																																																																																																											
149	740	249	370	For each additional invention to be examined (37 CFR § 1.129(b))																																																																																																																																																																											
179	740	279	370	Request for Continued Examination (RCE)																																																																																																																																																																											
169	900	169	900	Request for expedited examination of a design application																																																																																																																																																																											
<b>FEE CALCULATION</b> <b>1. BASIC FILING FEE</b> <table border="1"> <thead> <tr> <th>Large Entity Fee Code</th> <th>Large Entity Fee (\$)</th> <th>Small Entity Fee Code</th> <th>Small Entity Fee (\$)</th> <th>Fee Description</th> <th>Fee Paid</th> </tr> </thead> <tbody> <tr><td>101</td><td>740</td><td>201</td><td>370</td><td>Utility filing fee</td><td></td></tr> <tr><td>106</td><td>330</td><td>206</td><td>165</td><td>Design filing fee</td><td></td></tr> <tr><td>107</td><td>510</td><td>207</td><td>255</td><td>Plant filing fee</td><td></td></tr> <tr><td>108</td><td>740</td><td>208</td><td>370</td><td>Reissue filing fee</td><td></td></tr> <tr><td>114</td><td>160</td><td>214</td><td>80</td><td>Provisional filing fee</td><td></td></tr> </tbody> </table> SUBTOTAL (1) (\$ 0)	Large Entity Fee Code	Large Entity Fee (\$)	Small Entity Fee Code	Small Entity Fee (\$)	Fee Description	Fee Paid	101	740	201	370	Utility filing fee		106	330	206	165	Design filing fee		107	510	207	255	Plant filing fee		108	740	208	370	Reissue filing fee		114	160	214	80	Provisional filing fee																																																																																																																																												
Large Entity Fee Code	Large Entity Fee (\$)	Small Entity Fee Code	Small Entity Fee (\$)	Fee Description	Fee Paid																																																																																																																																																																										
101	740	201	370	Utility filing fee																																																																																																																																																																											
106	330	206	165	Design filing fee																																																																																																																																																																											
107	510	207	255	Plant filing fee																																																																																																																																																																											
108	740	208	370	Reissue filing fee																																																																																																																																																																											
114	160	214	80	Provisional filing fee																																																																																																																																																																											
<b>2. EXTRA CLAIM FEES</b> <table border="1"> <thead> <tr> <th>Total Claims</th> <th>Extra Claims</th> <th>Fee from below</th> <th>Fee Paid</th> </tr> </thead> <tbody> <tr> <td>37</td> <td>0</td> <td>18</td> <td>180</td> </tr> <tr> <td>108</td> <td>148</td> <td>84</td> <td>336</td> </tr> <tr> <td></td> <td></td> <td></td> <td>516</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Large Entity Fee Code</th> <th>Large Entity Fee (\$)</th> <th>Small Entity Fee Code</th> <th>Small Entity Fee (\$)</th> <th>Fee Description</th> <th>Fee Paid</th> </tr> </thead> <tbody> <tr><td>103</td><td>18</td><td>203</td><td>9</td><td>Claims in excess of 20</td><td></td></tr> <tr><td>102</td><td>84</td><td>202</td><td>42</td><td>Independent claims in excess of 3</td><td></td></tr> <tr><td>104</td><td>280</td><td>204</td><td>140</td><td>Multiple dependent claim, if not paid</td><td></td></tr> <tr><td>109</td><td>84</td><td>209</td><td>42</td><td>** Reissue independent claims over original patent</td><td></td></tr> <tr><td>110</td><td>18</td><td>210</td><td>9</td><td>** Reissue claims in excess of 20 and over original patent</td><td></td></tr> </tbody> </table> SUBTOTAL (2) (\$ 84)	Total Claims	Extra Claims	Fee from below	Fee Paid	37	0	18	180	108	148	84	336				516	Large Entity Fee Code	Large Entity Fee (\$)	Small Entity Fee Code	Small Entity Fee (\$)	Fee Description	Fee Paid	103	18	203	9	Claims in excess of 20		102	84	202	42	Independent claims in excess of 3		104	280	204	140	Multiple dependent claim, if not paid		109	84	209	42	** Reissue independent claims over original patent		110	18	210	9	** Reissue claims in excess of 20 and over original patent																																																																																																																												
Total Claims	Extra Claims	Fee from below	Fee Paid																																																																																																																																																																												
37	0	18	180																																																																																																																																																																												
108	148	84	336																																																																																																																																																																												
			516																																																																																																																																																																												
Large Entity Fee Code	Large Entity Fee (\$)	Small Entity Fee Code	Small Entity Fee (\$)	Fee Description	Fee Paid																																																																																																																																																																										
103	18	203	9	Claims in excess of 20																																																																																																																																																																											
102	84	202	42	Independent claims in excess of 3																																																																																																																																																																											
104	280	204	140	Multiple dependent claim, if not paid																																																																																																																																																																											
109	84	209	42	** Reissue independent claims over original patent																																																																																																																																																																											
110	18	210	9	** Reissue claims in excess of 20 and over original patent																																																																																																																																																																											
**or number previously paid, if greater; For Reissues, see above																																																																																																																																																																															

SUBMITTED BY		Complete (if applicable)	
Name (Print/Type)	Bradley C. Wright	Registration No. Attorney/Agent	28,061
Signature	<i>Rosa De...</i>	Telephone	(202) 508-9160
		Date	February 22, 2002

**WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.**

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.



PATENT APPLICATION

#10

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application Of

Edmond Colby Munger et al.

Serial No.: 09/504,783

Filed: February 15, 2000

For: IMPROVEMENTS TO AN AGILE  
NETWORK PROTOCOL FOR  
SECURE COMMUNICATIONS  
WITH ASSURED SYSTEM  
AVAILABILITY

Group Art Unit: 2153

Examiner: K. Lim

Atty. Dkt. No. 00479.85672

RECEIVED

MAR 01 2002

Technology Center 2100

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT

Assistant Commissioner for Patents and Trademarks  
Washington, D.C. 20231

Sir:

In accordance with 37 C.F.R. § 1.97(c) and 1.98, enclosed is a PTO Form-1449 listing art for consideration by the Examiner and a copy of each of the identified documents. It is believed no fee is required to make this a complete and timely filing. However, if a fee is required, please charge our Deposit Account No. 19-0733.

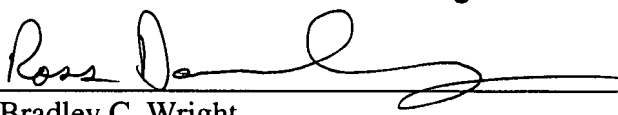
Consideration of this information is respectfully requested.

The submission of the listed document is not intended as an admission that any such document constitutes prior art against the claims of the present application. Applicant does not waive any right to take any action that would be appropriate to antedate or otherwise remove any listed document as a competent reference against the claims of the present application.

Respectfully submitted,

**BANNER & WITCOFF, LTD.**

**U.S.P.T.O.  
Reg. No. 49,024**

By:   
for Bradley C. Wright  
Registration No. 38,061

1001 G. Street, N.W.  
Washington, D.C. 20001-4597  
(202) 508-9100  
Dated: 2/22/02

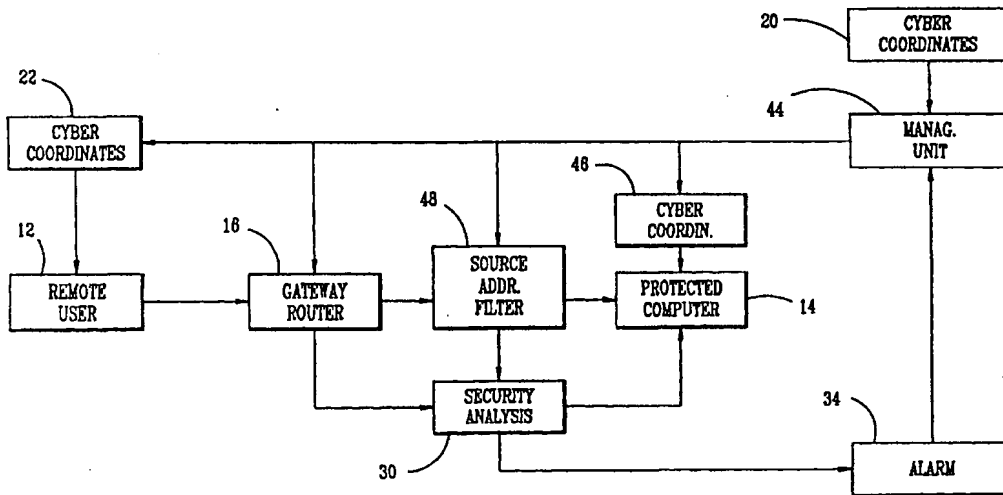
BCW/RAD/mmd



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(51) International Patent Classification 7 : G06F 11/00</p>	<p>A1</p>	<p>(11) International Publication Number: <b>WO 00/70458</b>  (43) International Publication Date: 23 November 2000 (23.11.00)</p>
<p>(21) International Application Number: PCT/US00/08219 (22) International Filing Date: 15 May 2000 (15.05.00)  (30) Priority Data: 60/134,547 17 May 1999 (17.05.99) US  (71) Applicant: COMSEC CORPORATION [US/US]; 10217 Cedar Pond Drive, Vienna, VA 22182 (US).  (72) Inventor: SHEYMOV, Victor, I.; 10217 Cedar Pond Drive, Vienna, VA 22182 (US).  (74) Agent: SIXBEY, Daniel, W.; Nixon Peabody LLP, Suite 800, 8180 Greensboro Drive, McLean, VA 22102 (US).</p>		<p>(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, 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, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).</p> <p>Published With international search report.</p>

(54) Title: METHOD OF COMMUNICATIONS AND COMMUNICATION NETWORK INTRUSION PROTECTION METHODS AND INTRUSION ATTEMPT DETECTION SYSTEM



(57) Abstract

The intrusion protection method and system for a communication network provides address agility wherein the cyber coordinates of a target host (14) are changed both on a determined time schedule and when an intrusion attempt is detected. The system includes a management unit (18) which generates a random sequence of cyber coordinates and maintains a series of tables containing the current and next set of cyber coordinates. These cyber coordinates are distributed to authorized users (12) under an encryption process to prevent unauthorized access.

*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						

METHOD OF COMMUNICATIONS AND  
COMMUNICATION NETWORK INTRUSION PROTECTION METHODS AND  
INTRUSION ATTEMPT DETECTION SYSTEM

5           This application is a continuation-in-part application of U.S. Serial No. 60/134,547  
filed May 17, 1999.

Background Art

10           Historically, every technology begins its evolution focusing mainly on performance  
parameters, and only at a certain developmental stage does it address the security aspects of  
its applications. Computer and communications networks follow this pattern in a classic  
way. For instance, first priorities in development of the Internet were reliability,  
survivability, optimization of the use of communications channels, and maximization of their  
15           speed and capacity. With a notable exception of some government systems, communications  
security was not an early high priority, if at all. Indeed, with a relatively low number of  
users at initial stages of Internet development, as well as with their exclusive nature,  
problems of potential cyber attacks would have been almost unnatural to address,  
considering the magnitude of other technical and organizational problems to overcome at  
20           that time. Furthermore, one of the ideas of the Internet was "democratization" of  
communications channels and of access to information, which is almost contradictory to the  
concept of security. Now we are faced with a situation, which requires adequate levels of  
security in communications while preserving already achieved "democratization" of  
communications channels and access to information.

25           All the initial objectives of the original developers of the Internet were achieved with  
results spectacular enough to almost certainly surpass their expectations. One of the most  
remarkable results of the Internet development to date is the mentioned "democratization".  
However in its unguarded way "democratization" apparently is either premature to a certain  
percentage of the Internet users, or contrary to human nature, or both. The fact remains that  
30           this very percentage of users presents a serious threat to the integrity of national critical  
infrastructure, to privacy of information, and to further advance of commerce by utilization

of the Internet capabilities. At this stage it seems crucial to address security issues but, as usual, it is desirable to be done within already existing structures and technological conventions.

Existing communications protocols, while streamlining communications, still lack  
5 underlying entropy sufficient for security purposes. One way to increase entropy, of course, is encryption as illustrated by U.S. Patent No. 5,742,666 to Finley. Here each node in the Internet encrypts the destination address with a code which only the next node can unscramble.

Encryption alone has not proven to be a viable security solution for many  
10 communications applications. Even within its core purpose, encryption still retains certain security problems, including distribution and safeguarding of the keys. Besides, encryption represents a "ballast", substantially reducing information processing speed and transfer time. These factors discourage its use in many borderline cases.

Another way is the use of the passwords. This method has been sufficient against  
15 humans, but it is clearly not working against computers. Any security success of the password-based security is temporary at best. Rapid advances in computing power make even the most sophisticated password arrangement a short-term solution.

Recent studies clearly indicate that the firewall technology, as illustrated by U.S.  
20 Patent No. 5,898,830 to Wesinger et al., also does not provide a sufficient long-term solution to the security problem. While useful to some extent, it cannot alone withstand the modern levels of intrusion cyber attacks.

On the top of everything else, none of the existing security methods, including  
25 encryption, provides protection against denial of service attacks. Protection against denial of service attacks has become a critical aspect of communication system security. All existing log-on security systems, including those using encryption, are practically defenseless against such attacks. Given a malicious intent of a potential attacker, it is reasonable to assume that, even having failed with an intrusion attempt, the attacker is still capable of doing harm by disabling the system with a denial of service attack. Since existing systems by definition have to deal with every log-on attempt, legitimate or not, it is certain  
30 that these systems cannot defend themselves against a denial of service attack.

The deficiencies of existing security methods for protecting communications systems leads to the conclusion that a new generation of cyber protection technology is needed to achieve acceptable levels of security in network communications.

5 Summary of the Invention

It is a primary object of the present invention to provide a novel and improved method of communications, and a novel and improved communication network intrusion protection method and systems and novel and improved intrusion attempt detection method and systems, adapted for use with a wide variety of communication networks including Internet based computers, corporate and organizational computer networks (LANs), e-commerce systems, wireless computer communications networks, telephone dial-up systems, wireless dial-up systems, wireless telephone and computer communications systems, cellular and satellite telephone systems, mobile telephone and mobile communications systems, cable based systems and computer databases, as well as protection of network nodes such as routers, switches, gateways, bridges, and frame relays.

Another object of the present invention is to provide a novel and improved communication network intrusion protection method and system which provides address agility combined with a limited allowable number of log-on attempts.

20 Yet another object of the present invention is to provide a novel and improved intrusion protection method for a wide variety of communication and other devices which may be accessed by a number, address code, and/or access code. This number, address code, and/or access code is periodically changed and the new number, address code, or access code is provided only to authorized users. The new number, address code, or access code may be provided to a computer or a device for the authorized user and not be accessible to others. This identifier causes the user's computer to transmit the otherwise unknown and inaccessible number, address code, and/or access code.

25 A still further object of the present invention is to provide a novel and improved communication network intrusion protection method and system wherein a plurality of different cyber coordinates must be correctly provided before access is granted to a protected communications unit or a particular piece of information. If all or some cyber coordinates



are not correctly provided, access is denied, an alarm situation is instigated and the affected cyber coordinates may be instantly changed.

For the purposes of this invention cyber coordinates are defined as a set of statements determining location of an object (such as a computer) or a piece of information (such as a computer file) in cyber space. Cyber coordinates include but are not limited to private or public protocol network addresses such as an IP address in the Internet, a computer port number or designator, a computer or database directory, a file name or designator, a telephone number, an access number and/or code, etc.

These and other objects of the present invention are achieved by providing a communication network intrusion protection method and system where a potential intruder must first guess where a target computer such as a host workstation is in cyber space and to predict where the target computer such as a workstation will next be located in cyber space. This is achieved by changing a cyber coordinate (the address) or a plurality of cyber coordinates for the computers such as workstations on a determined or random time schedule and making an unscheduled cyber coordinates change when the system detects an intrusion attempt. A limited number of log-on attempts may be permitted before an intrusion attempt is confirmed and the cyber coordinates are changed. A management unit is provided for generating a random sequence of cyber coordinates and which maintains a series of tables containing current and the next set of addresses. These addresses are distributed to authorized parties, usually with use of an encryption process.

The present invention further provides for a piece of information, a computer or a database intrusion protection method and system where a potential intruder must first guess where a target piece of information such as a computer file or a directory is in cyber space and to predict where the target piece of information will be next in cyber space. This is achieved by changing a cyber coordinate or a plurality of cyber coordinates for the piece of information on a determined or random time schedule and making an unscheduled cyber coordinates change when the system detects an intrusion attempt. A limited number of log-on attempts may be permitted before an intrusion attempt is confirmed and the coordinates changed. A management unit is provided for generating a random sequence of cyber coordinates and which maintains a series of tables containing current and the next set of cyber coordinates. These coordinates are distributed to authorized parties, usually by means

of an encryption process.

The intrusion attempt detection methods and systems are provided to the protected devices and pieces of information as described above by means of categorizing a log-on attempt when all or some of the correct cyber coordinates are not present as an intrusion attempt and by instigating an alarm situation.

#### Brief Description of the Drawings

Figure 1 is a block diagram of the communication network protection system of the present invention;

Figure 2 is a flow diagram showing the operation of the system of Figure 1;

Figure 3 is a block diagram of a second embodiment of the communication network protection system of the present invention;

Figure 4 is a flow diagram showing the operation of the system of Figure 3;

Figure 5 is a block diagram of a third embodiment of the communication network protection system of the present invention;

Figure 6 is a flow diagram showing the operation of the system of Figure 5; and

Figure 7 is a block diagram of a fourth embodiment of the communication network protection system of the present invention.

#### Description of the Preferred Embodiments

Existing communications systems use fixed coordinates in cyber space for the communications source and communications receiver. Commonly accepted terminology for the Internet refers to these cyber coordinates as source and destination IP addresses. For

purposes of an unauthorized intrusion into these communication systems, the situation of a cyber attack might be described in military terms as shooting at a stationary target positioned at known coordinates in cyber space. Obviously, a moving target is more secure than the stationary one, and a moving target with coordinates unknown to the intruder is more secure yet. The method of the present invention takes advantage of the cyber space environment and the fact that the correlation between the physical coordinates of computers or other communication devices and their cyber coordinates is insignificant.

While it is difficult to change the physical coordinates of computers or other communications devices, their cyber coordinates (cyber addresses) can be changed much easier, and in accordance with the present invention, may be variable and changing over time. In addition to varying the cyber coordinates over time, the cyber coordinates can immediately be changed when an attempted intrusion is sensed. Furthermore, making the current cyber coordinates available to only authorized parties makes a computer or other communications device a moving target with cyber coordinates unknown to potential attackers. In effect, this method creates a device which perpetually moves in cyber space.

Considering first the method of the present invention as applied to computers and computer networks, the computer's current cyber address may serve also as its initial log-on password with a difference that this initial log-on password is variable. A user, however, has to deal only with a computer's permanent identifier, which is, effectively its assigned "name" within a corresponding network. Any permanent identifier system can be used, and an alphabetic "name" system seems to be reasonably user-friendly. One of such arrangements would call for using a computer's alphabetic Domain Name System, as a cyber address permanent identifier, while subjecting its numeric, or any other cyber address to a periodic change with regular or irregular intervals. This separation will make the security system transparent to the user, who will have to deal only with the alphabetic addresses. In effect, the user's computer would contain an "address book" where the alphabetic addresses are permanent, and the corresponding variable addresses are more complex and periodically updated by a network's management. While a user is working with other members of the network on the name or the alphabetic address basis, the computer conducts communications based on the corresponding variable numeric or other addresses assigned for that particular time.

A variable address system can relatively easily be made to contain virtually any level of entropy, and certainly enough entropy to defy most sophisticated attacks. Obviously, the level of protection is directly related to the level of entropy contained in the variable address system and to the frequency of the cyber address change.

5 This scenario places a potential attacker in a very difficult situation when he has to find the target before launching an attack. If a restriction on a number of allowable log-on tries is implemented, it becomes more difficult for an attacker to find the target than to actually attack it. This task of locating the target can be made difficult if a network's cyber address system contains sufficient entropy. This difficulty is greatly increased if the security  
10 system also limits the number of allowable log-on tries, significantly raising the entropy density.

For the purpose of this invention, entropy density is defined as entropy per one attempt to guess a value of a random variable.

Figure 1 illustrates a simple computer intrusion protection system 10 which operates  
15 in accordance with the method of the present invention. Here, a remote user's computer 12 is connected to a protected computer 14 by a gateway router or bridge 16. A management system 18 periodically changes the address for the computer 14 by providing a new address from a cyber address book 20 which stores a plurality of cyber addresses. Each new cyber address is provided by the management system 18 to the router 16 and to a user computer  
20 address book 22. The address book 22 contains both the alphabetic destination address for the computer 14 which is available to the user and the variable numeric cyber address which is not available to the user. When the user wants to transmit a packet of information with the alphabetic address for the computer 14, this alphabetic address is automatically substituted for the current numerical cyber address and used in the packet.

25 With the reference to Figures 1 and 2, when a packet is received by the gateway router or bridge 16 as indicated at 24, the cyber address is checked by the gateway router or bridge at 26, and if the destination address is correct, the packet is passed at 28 to the computer 14. If the destination address is not correct, the packet is directed to a security analysis section 30 which, at 32 determines if the packet is retransmitted with a correct  
30 address within a limited number of log-in attempts. If this occurs, the security analysis section transmits the packet to the computer 14 at 28. However, if no correct address is

received within the allowed limited number of log-in attempts, the packet is not transmitted to the computer 14 and the security analysis section activates an alarm section 34 at 36 which in turn causes the management section to immediately operate at 38 to change the cyber address.

5           Sophisticated cyber attacks often include intrusion through computer ports other than the port intended for a client log-on. If a system principally described in connection with Figures 1 and 2 is implemented, the port vulnerability still represents an opening for an attack from within the network, that is if an attacker has even a low-level authorized access to a particular computer and thus knows its current variable address.

10           Computer ports can be protected in a way similar to protection of the computer itself. In this case port assignment for the computer becomes variable and is changed periodically in a manner similar to that described in connection with Figures 1 and 2. Then, a current assignment of a particular port is communicated only to appropriate parties and is not known to others. At the same time, similarly to methods described, a computer user would deal  
15 with permanent port assignments, which would serve as the ports' permanent "names".

          This arrangement in itself may not be sufficient, however, to reliably protect against a port attack using substantial computing power because of a possible insufficient entropy density. Such a protection can be achieved by implementing an internal computer "port router" which would serve essentially the same role for port identifiers as the common  
20 gateway router or bridge 16 serves for computer destination addresses.

          With reference to Figures 3 and 4 wherein like reference numerals are used for components and operations which are the same as those previously described in connection with Figures 1 and 2, a port router 40 is provided prior to the protected computer 14, and this port router is provided with a port number or designator by the management unit 18. This  
25 port number or designator is also provided to the user address book 22 and will be changed when the cyber address is changed, or separately. Thus, with reference to Figure 4, once the cyber address has been cleared at 26, the port number or designator is examined at 42. If the port number is also correct, the data packet will be passed to the computer 14 at 28. If the port number is initially incorrect, the packet is directed to the security analysis section 30  
30 which at 32 determines if the packet is retransmitted with the correct port number within the limited number of log-in attempts.

The port protection feature can be used independently of other features of the system. It can effectively protect nodes of the infrastructure such as routers, gateways, bridges, and frame relays from unauthorized access. This can protect systems from an attacker staging a cyber attack from such nodes.

5 The method and system of the present invention may be adapted to provide security for both Internet based computer networks and private computer networks such as LANs.

Internet structure allows the creation of an Internet based Private Cyber Network (PCN) among a number of Internet-connected computers. The main concern for using the Internet for this purpose as an alternative to the actual private networks with dedicated  
10 communication channels is security of Internet-based networks.

The present invention facilitates establishment of adequate and controllable level of security for the PCNs. Furthermore, this new technology provides means for flexible structure of a PCN, allowing easy and practically instant changes in its membership. Furthermore, it allows preservation of adequate security in an environment where a computer  
15 could be a member of multiple PCNs with different security requirements. Utilizing the described concept, a protected computer becomes a "moving target" for the potential intruders where its cyber coordinates are periodically changed and the new coordinates are communicated on a "need to know" basis only to the other members of the PCN authorized to access this computer along with appropriate routers and gateways. This change of cyber  
20 coordinates can be performed either by previous arrangement or by communicating future addresses to the authorized members prior to the change. Feasible frequency of such a change can range from a low extreme of a stationary system changing cyber coordinates only upon detection of a cyber attack to an extremely high frequency such as with every packet. The future coordinates can be transmitted either encrypted or unencrypted. Furthermore,  
25 each change of position of each PCN member can be made random in terms of both its current cyber coordinates and the time of the coordinates change. These parameters of a protected PCN member's cyber moves are known only to the PCN management, other PCN members with authorization to communicate with this particular member, and appropriate gateways and routers. PCN management would implement and coordinate periodic cyber  
30 coordinates changes for all members of the PCN. While the PCN management is the logical party to make all the notification of the cyber coordinates changes, in certain instances it

could be advantageous to shift a part of this task to a PCN member computer itself. With certain limitations, the routers and gateways with the “need to know” the current address of the protected computer are located in cyber space in the general vicinity of the protected computer. In such instances the protected computer could be in a better position to make the mentioned notifications of nearby routers and gateways.

The address changes could be done simultaneously for all the members of the PCN, or separately, particularly if security requirements for the members substantially differ. The latter method is advantageous, for instance, if some of the computers within the PCN are much more likely than others to be targeted by potential intruders. A retail banking PCN could be an example of such an arrangement where the bank’s computer is much more likely to be attacked than a customer’s computer. It should be noted that, while in certain cases some members of the PCN may not require any protection at all, it still is prudent to provide it as long as the computer belongs to a protected PCN. The correct “signature” of the current “return address” would serve as additional authenticity verification. In the above example of the retail banking, while many customers’ computers may not require any protection, assigning variable addresses to them would serve as an additional assurance to the bank that every log-on is authorized. In fact, this system automatically provides two-tier security. In order to reach a protected computer, the client computer has to know the server computer current cyber address in the first place. Then, even if a potential intruder against odds “hits” the correct current address the information packet is screened for the correct “signature” or return address. If that signature does not belong to the list of the PCN’s current addresses, the packet is rejected. In high security instances this should trigger an unscheduled address change of the protected computer.

With the reference to Figures 5 and 6 which illustrate this two-tier security system, a network management unit 44 provides different unique cyber coordinates to the address books for each computer in the system (two computers 12 and 14 with address books 22 and 46 respectively being shown). Now when the computer 12 sends a data packet to the computer 14, the gateway router or bridge 16, first checks for the correct current destination address for the computer 14 at 26 in the manner previously described. If the destination address is correct, a source address sensor 48 checks at 50 to determine if the correct source address (i.e. return address) for the computer 12 is also present. If both correct addresses are

present, the data packet is passed to the computer 14 at 28, but if the correct source address is not present, the data packet is passed to the security analysis section 30 where at 32 where it is determined if a correct source address is received within the acceptable number of log-on tries. If the correct return address is not received, an alarm situation is activated at 36 and the network management system operates at 38 to change the cyber address of the computer 14

In addition to the penetration (hacking) detection and protection, the system above provides real-time detection of a cyber attack and protection against "flooding" denial of service attacks. A gateway router or bridge 16 filters all the incorrectly addressed packets thus protecting against "flooding". Further yet, since the "address book" of the protected network contains only trusted destinations, this system also protects against instructive viruses or worms if such are present or introduced into the network. For the purpose of this invention, an instructive virus or worm is defined as a foreign unit of software introduced into a computer system so it sends certain computer data to otherwise unauthorized parties outside of the system.

Elements of the system described above are: a gateway router or bridge 16, a computer protection unit, and a management unit. A gateway router or bridge represents an element of collective defense for the network, while the source address filter and the "port router" and filter represent a unit of individual defense for a member computer. This individual defense unit (server unit) can be implemented either as a standalone computer, as a card in the protected computer, as software in the protected computer, or imbedded into the protected computer operating system. For further improvement of the overall security, port assignments can be generated autonomously from the management unit thus creating a "two keys" system in a cryptographic sense. This would allow for security to still be in place even if a security breach happened at the security management level.

The method and system of the present invention minimize human involvement in the system. The system can be configured in such a way that computer users deal only with simple identifiers or names permanently assigned to every computer in the network. All the real (current) cyber coordinates can be stored separately and be inaccessible to the user, and could be available to the appropriate computers only. This approach both enhances security and makes this security system transparent to the user. The user deals only with the simple



alphabetic side of the "address book", and is not bothered with the inner workings of the security system. A telephone equivalent of this configuration is an electronic white pages residing in a computerized telephone set, which is automatically updated by the telephone company. The user just has to find a name, and push the "connect" button while the  
5 telephone set does the rest of the task.

A numeric cyber address system, based on the Internet host number could be relatively easily utilized for the discussed security purposes, however a limitation exists for this address system in its current form represented by the IPv.4 protocol. This limitation is posed by the fact that the address is represented by a 32-bit number. 32-bit format does not  
10 contain sufficient entropy in the address system to enable establishment of adequate security. This is a particularly serious limitation in regard to securing an entire network. The availability of the network numbers are limited to the extent that not only entropy, but a simple permanently assigned number is becoming more and more difficult to obtain with the rapid expansion of the Internet.

If this address system is to be used for the security purposes, than the format of the  
15 host number should be adequately expanded to create sufficient size of the address numbers field in the system. If this is done, than the corresponding address in the Domain Name System (DNS) could be conveniently used as permanent identifier for a particular computer and the Internet host number would be variable, creating a moving regime of a protected  
20 computer. Currently being implemented IPv.6 (IPNG) protocol solves this problem by providing sufficient entropy.

Another way to achieve the same goal is to use the DNS address as a variable for security purposes. This way, the traditional Internet DNS address system would not be affected and no change in format is required. The relevant part of the protected computer's  
25 DNS address would become a variable, utilizing more characters than the alphabet, with a very large number of variations, also creating sufficient level of entropy.

Yet another way to implement the same method is to utilize the geographic zone-based system. While its utilization is somewhat similar to the DNS system, it offers some practical advantages for security use. Naturally, when a computer is protected by a security  
30 system, it is still essential to preserve the communication redundancy of the Internet communications. However, the redundancy may suffer if only a limited number of the

5 routers and gateways are informed of the protected computer current cyber address. This effect could be particularly important with the members of a particular protected network vastly remote in geographic terms. The necessary notification of a large number of the routers and gateways can also become problematic, not only technically, but also because it can decrease the level of security. In this sense a geographic zone-based system offers advantages since the variable part of the computer's cyber address could be made to involve only certain geographic locale while initial routing of the information packet could be done by the traditional method. After the packet has been moved to the general vicinity of the addressee computer, it would get into the area of the "informed" routers and gateways. This scheme would simplify the notification process of the routers as well as improve security by limiting the number of the "need to know" parties. It is important to recognize that, after the "general" part of the cyber address caused the information packet to arrive in a cyber vicinity of the addressee, virtually any, even private, address system can be used for the rest of the delivery. This would further increase the level of underlying entropy in the system.

10  
15 While certain specific address systems have been discussed, it is an important quality of the present invention that it can be implemented with virtually any address system.

Corporate and organizational computer networks such as LANs or, at least those in closed configurations, do not possess as much vulnerability to cyber attacks as Internet-based networks. However, even in these cases, their remote access security is a subject of concern. This is especially visible when a private network (PN) contains information of different levels of confidentiality with access restricted to appropriate parties. In other words, along with other generally accessible organizational information, an organizational PN can contain information restricted to certain limited groups. Enforcement of these restrictions requires a remote access security system. Usually these security systems employ a password-based scheme of one type or another and, perhaps, a firewall. However, reliance on passwords may not be entirely justified since the passwords can be lost or stolen, giving a malicious insider with a low access level a reasonable chance of access to information intended only for higher levels of access. Furthermore, in some cases use of cracking techniques from such a position is not entirely out of the question. Such an occurrence can relatively easily defeat both the password and the firewall. This would prevent a LAN from a cyber attack launched from within the network.

20  
25  
30

The present invention provides adequate security to such PCNs without reliance on the passwords and to limit access to only appropriate computers. Then, the task of overall information access security practically would be narrowed down to control of physical access to a particular computer, usually a less complicated feat.

5 Similarly to the systems described for Internet-based networks, a "closed" LAN as well as an Internet-based LAN can be protected by implementation of periodic changes of the members' network addresses and communicating those changes to the appropriate parties. This way, the lowest access level computers would have the lowest rate of address change. The rate of the address change would increase with the level of access. This system  
10 would ensure that all the PCN computers with legitimate access to a particular computer within the PCN would be informed of its location. Furthermore, it will ensure that the current location of a computer with restricted information would be unknown to the parties without the legitimate access clearance. For instance, a superior's computer would be able to access his subordinate's computer but not vice versa.

15 Also similarly to the systems described for the PCNs, a PCN computer would contain an "address book" where the user can see and use only the permanent side of it with identifiers of all computers accessible to him while the actual communication functions are performed by the computer using the variable side of the "address book" periodically updated by the PN management. To further enhance security, in addition to the computer  
20 address system management, the PCN Administrator can implement an automatic security monitoring system where all wrongly addressed log-on attempts would be registered and analyzed for security purposes.

Thus the method and system of the present invention would allow reliable protection  
25 against unauthorized remote access to information from within a PN while providing a great deal of flexibility, where the granted access can be revised easily and quickly.

A greatly enhanced intrusion protection system and method can be achieved by combining the operating systems of Figures 1-6. Now an arriving data packet would first be screened by a gateway router or a similar device for a correct destination address. If the destination address is correct, the packet is passed for further processing. If the destination  
30 address is incorrect, the alarm is triggered and the packet is passed to the network security managing unit for security analysis.

The packet with correct destination address is then screened for a correct source address. If the source address is correct, the packet is passed to the receiver computer. If the source address is incorrect, the alarm is triggered and the packet is passed to the network security managing unit for security analysis.

5 Then, the packet with a correct destination address and a correct source address is screened for a correct allowed port coordinate such as port number. If the port coordinate is correct, the packet is passed for further processing. If the port coordinate is incorrect, the alarm is triggered and the packet is passed to the network security managing unit for security analysis.

10 Finally, the packet with a correct destination and source addresses and a correct port designator is screened for data integrity by application of authentication check such as a checksum. If the authentication check is passed, the packet is passed to the addressee computer. If the authentication check is failed, the alarm is triggered and the packet is passed to the network security managing unit for security analysis.

15 The security managing unit analyses all the alarms and makes decisions on necessary unscheduled changes of addresses for appropriate network servers. Also, it can notify law enforcement and pass appropriate data on to it.

Figure 7 illustrates an enhanced computer intrusion protection system indicated generally at 52 for one or more network computers 54. A gateway router or a bridge 58  
20 includes a destination address filter 60 which receives data packets which pass in through a load distribution switch 62. A non-interrogatable network address book 64 stores current network server addresses for the destination address filter 60, and the destination address filter checks each data packet to determine if a legitimate destination address is present.

Packets with legitimate destination addresses are forwarded to a source address filter  
25 66, while packets with illegitimate destination addresses are sent to a security analysis section 68 in a management unit 70.

When a preset traffic load level is reached indicating that an attempt at flooding is being made, the destination address filter causes the load distribution switch 62 to distribute traffic to one or more parallel gateway routers or bridges which collectively forward  
30 legitimate traffic and dump the flooding traffic. An alternative arrangement would call for the load distribution function to be done irrespective of the load, utilizing all the parallel

gateways all the time. A source address table 74 stores accessible server's designators and corresponding current addresses for all system servers which may legitimately have access to the computer or computers 54. These addresses are accessed by the source address filter which determines whether or not an incoming data packet with the proper destination address originates from a source with a legitimate source address entered in the source address table 74. If the source address is determined to be legitimate, the data packet is passed to a port address filter 76. Data packets with an illegitimate source address are directed to the security analysis section 68. Alternatively, source address screening can be done at the gateway router or bridge 58 first prior to port filter 76.

A port protection table 78 includes the current port assignments for the computer or computers 54, and these port assignments are accessed by the port designator filter 76 which then determines if an incoming data packet contains legitimate port designation. If it does, it is passed to an actual address translator 80 which forwards the data packet to the specific computer or computers 54 which are to receive the packet. If an illegitimate port address is found by the port address filter 76, the data packet is transmitted to the security analysis section 68.

The management unit 70 is under the control of a security administrator 82. A network membership master file 84 stores a master list of legitimate server's designators along with respective authorized access lists and corresponding current cyber coordinates. The security administrator can update the master list by adding or removing authorized access for every protected computer. An access authorization unit 86 distributes the upgraded relevant portions of the master lists to the address books of the respective authorized servers.

A random character generator 88 generates random characters for use in forming current port designators, and provides these characters to a port designator forming block 90. This port designator forming block forms the next set of network current port designators in conjunction with the master list and these are incorporated for transmission by a port table block 92. Alternatively, port designators can be formed in the computer unit instead of the management unit.

Similarly, a random character generator 94 generates random characters for use in forming current server addresses, and provides these characters to a server address forming

block 96. This server address forming block forms the next set of current network server addresses, and an address table 98 assigns addresses to servers designated on the master list.

A coordinator/dispatcher block 100 coordinates scheduled move of network servers to their next current addresses, provides the next set of network addresses for appropriate servers and routers and coordinates unscheduled changes of addresses on command from the security analysis unit 68. The coordinator/dispatcher block 100 may be connected to an encode/decode block 102 which decodes received address book upgrades from input 104 and encodes new port and server destination addresses to be sent to authorized servers in the system over output 106. Where encoding of new cyber coordinates is used, each authorized computer in the network will have a similar encoding/decoding unit.

The security analysis unit 68 analyses received illegitimate data packets and detects attack attempts. If needed, the security analysis unit orders the coordinator/dispatcher block 100 to provide an unscheduled address change and diverts the attack data packets to an investigation unit 108. This investigation unit simulates the target server keeping a dialog alive with the attacker to permit security personnel to engage and follow the progress of the attacker while tracing the origin of the attack.

Providing security against intrusion for e-commerce systems presents a unique problem, for an important peculiarity of an e-commerce system is that its address must be publicly known. This aspect represents a contradiction to the requirement of the address being known to authorized parties only. However, the only information intended for the general public usually relates to a company catalog and similar material. The rest of the information on a merchant's network is usually considered private and thus should be protected. Using this distinction, a merchant's e-commerce site should be split into two parts: public and private. The public part is set up on a public "catalog" server with a fixed IP address and should contain only information intended for the general public. The rest of the corporate information should be placed in a separate network and protected as described in relation to Figures 1-7.

When a customer has completed shopping and made purchasing decisions concerning the terms and price of the sale, pertinent for the transaction, information is placed in a separate register. This register is periodically swept by a server handling financial transactions ("financial" server), which belongs to the protected corporate network. In fact,

the “catalog” server does not know the current address of the financial transactions server. Thus, even if an intruder penetrates the “catalog” server, the damage is limited to the contents of the catalog and the intruder cannot get an entry to the protected corporate network.

5           The financial server, having received pending transaction data, contacts the customer, offering a short-term temporary access for finalizing the transaction. In other words, the customer is allowed access just long enough to communicate pertinent financial data such as a credit card number and to receive a transaction confirmation at which point the session is terminated, the customer is diverted back to the catalog server and the financial server is  
10 moved to a new cyber address thus making obtained knowledge of its location during the transaction obsolete.

          Dial-up communications systems, in respect to their infrastructure channels susceptibility to transmission intercept by unrelated parties, can be separated into two broad categories: easily interceptable, such as cellular and satellite telephone systems and relatively  
15 protected such as conventional land-line based telephone systems. Relatively protected systems such as conventional land-line based telephone systems can be protected in the following way. Phone numbers, assigned by a telephone company to a dial-up telephone-based private network serve as the members’ computer addresses. As described previously, such a private network can be protected from unauthorized remote access by implementing  
20 periodic changes in the addresses, i.e. telephone numbers assigned to the members for transmission by the network along with other designators such as access codes and communicating the changed numbers to the appropriate parties.

          For the conventional land-line dial-up telephone systems, while the “last mile” connection remains constant, the assigned telephone number is periodically changed, making  
25 the corresponding computer a moving target for a potential attacker. In this case the telephone company serves as the security system manager. It assigns the current variable telephone numbers to the members of a protected, private network, performs notification of all the appropriate parties, and changes the members’ current numbers to a new set at an appropriate time. The telephone company switches naturally serve in the role of routers, and  
30 thus they can be programmed to perform surveillance of the system, to detect potential intrusion attacks and to issue appropriate alarms.

Periodically changing the current assigned numbers creates system entropy for a potential intruder, making unauthorized access difficult. Obviously, the implementation of this security system is dependent on availability of sufficient vacant numbers at a particular facility of the telephone company. Furthermore, for a variety of practical reasons it is advisable to keep a just vacated number unassigned for a certain period of time. All this may require additional number capacity at the telephone company facility in order to enable it to provide remote access security to a larger number of personal networks while preserving a comfortable level of system entropy.

If the mentioned additional capacity is not available, or a still higher level of entropy is desired, it could be artificially increased by adding an access code to the assigned number. This would amount to adding virtual capacity to the system, and would make a combination of the phone number and access code an equivalent of a computer's telephone address. In effect, this would make a dialed number larger than the conventional format. This method makes a virtual number capacity practically unlimited and, since the process is handled by computers without human involvement, it should not put any additional burden on a user. With or without a virtual number capacity, utilization of this method allows the intrusion attempts to be easily identified by their wrong number and/or code. At the same time, implementation of this system might require some changes in dialing protocols as well as additional capabilities of the telephone switching equipment.

Entropy density can be increased by limiting the number of allowable connection attempts. Similarly to the method described previously, telephone company switching equipment can be made to perform a role of an outside security barrier for the private network. In this case wrongly addressed connection attempts should be analyzed in order to detect possible "sweeping". If such an attempt is detected, tracing the origin of the attempt and notifying the appropriate phone company should not present a problem even with the existing technology.

The simplest form of private network protection under the proposed method and system is when at a predetermined time all the members of a particular network are switched to the new "telephone book" of the network. However, in some cases required level of security for some members of the same private network could substantially differ, or they may face different levels of security risk. In such cases frequency of the phone number



change could be set individually with appropriate notification of the other members of the network. This differentiation enables the telephone company to offer differentiated levels of security protection to its customers even within the same private network.

5 A telephone company can also offer its customers protected voice private networks which would provide a higher level of privacy protection than the presently used "unlisted numbers." In this configuration the customers' telephone sets are equipped with a computerized dialing device with remotely upgradeable memory which would allow each member of a protected voice network to contain the network "telephone book" and that book is periodically updated by the telephone company.

10 The telephone company would periodically change the assigned telephone numbers of a protected network to a new set of current numbers. These new numbers would be communicated to the members of a protected voice network through updating their computerized dialing devices.

15 As a derivative of the described system, an updateable electronic telephone directory system can be also implemented. In this case a customer's phone set would include a computerized dialing device with electronic memory containing a conventional telephone directory and a personal directory as well. This telephone directory can be periodically updated on-line by the telephone company.

20 Easily interceptable systems such as cellular and satellite telephone systems, in addition to the protection described above, can be protected from "cloning" when their signals can be intercepted and the "identity" of the phone can be cloned for gaining unauthorized access and use of the system by unauthorized parties.

25 Mobile telephone and mobile communications systems are protected in a manner similar to networks or land based telephone systems. In this instance, the novel and improved method of changing cyber coordinates is designed to reliably protect mobile phone systems from unauthorized use commonly known as cloning as well as to make intercept of wireless communications more difficult than it is at present. With this system the static wireless phone number or other similar identifier is not used for identification and authorization. Instead, a set of private identifiers is generated known only to the phone company and base stations  
30 controlling mobile phone calls and used to continually update the mobile phone and base station directories with current valid identifiers. This approach provides vastly superior

protection over current methods requiring that each call be intercepted in order to track and keep current with changing identifiers. Immediate detection of unauthorized attempts to use a cloned phone is realized and law enforcement may be notified in near real time for appropriate action.

5 Other electronic devices using wireless communications can be protected by the methods and systems described above.

Finally, computers often contain databases with a variety of information. That information in a database often has wide-ranging levels of sensitivity or commercial value. This creates a situation when large computers serve multiple users with vastly different levels  
10 of access. Furthermore, even within the same level of access, security considerations require compartmentalization of information when each user has to have access to only a small portion of the database.

The existing systems try to solve this situation by utilizing passwords and internal firewalls. As it was mentioned earlier, password-based systems and firewalls are not  
15 sufficient against computerized attacks. In practical terms it means that a legitimate user with a low level of access, utilizing hacking techniques from his station, potentially can break into even the most restricted areas of the database.

This problem can be solved by using the method of the present invention. A piece of information such as a file or a directory in a computer exists in cyber space. Accordingly, it  
20 has its cyber address, usually expressed as a directory and/or a file name which defines its position in a particular computer file system. This, in effect, represents the cyber coordinates of that piece of information within a computer.

As described earlier, information security can be provided if a system manager periodically changes the directories and/or file names in the system, i.e. the cyber addresses  
25 of the information, and notifies only appropriate parties of the current file names. This method would ensure that each user computer knows locations of only files to which it has legitimate access. Furthermore, a user would not even know of existence of the files to which he has no access.

To further strengthen the system and make it user-friendly, the user would have a  
30 personal directory similar to an address book, where only permanent directory and/or file names are accessible to him, while the variable side of the "address book" would be

accessible only to the system manager and upgraded periodically. In this arrangement variable directory and/or file names can contain any required level of entropy, further increasing resistance to attacks from within the system. Additionally, an internal “router” or “filter” can also perform information security monitoring functions, detect intrusion attempts  
5 and issue appropriate alarms in real time.

Obviously, in order to ensure information security in such arrangement any computer-wide search by keywords or subject should be disabled and substituted with a search within specific clients’ “address books”.

The systems and methods described above allow for creation of a feasible  
10 infrastructure protection system such as a national or international infrastructure protection system. When detected at specific points cyber attacks are referred to such a system for further analysis and a possible action by law enforcement authorities.

I claim:

1. A method for protecting a communications device which is connected to a communications system against an unauthorized intrusion which includes:

- 5 providing the communications device with at least one identifier,  
providing the at least one identifier for use in accessing the communications device to entities authorized to access said communications device,  
sensing the presence or absence of said identifier before granting access to said communications device,  
10 providing access to said communications device when the use of said at least one correct identifier is sensed  
denying access to said communications device and providing said communications device with at least one new identifier when the absence of the correct at least one identifier is sensed during an attempt to access said communications device, and providing said at least  
15 one new identifier to entities authorized to access said communications device.

2. The method of claim 1 which includes periodically changing the at least one identifier and providing the changed at least one identifier to the entities authorized to access said communications device.

20

3. The method of claim 1 which includes providing said communications device with a plurality of separate identifiers,

sensing the presence or absence of all of said plurality of identifiers before granting access to said communications device,

25

providing access to said communications device when the use of all of said identifiers is sensed, and

denying access to said communications device and providing said communications device with a new plurality of identifiers to replace the previous plurality of identifiers when the absence of any one of the correct identifiers is sensed.

30

4. The method of claim 3 which includes periodically changing said plurality of

separate identifiers and providing the changed identifiers to the entities authorized to access said communications device.

5           5.       The method of claim 1 which includes permitting a predetermined number of attempts to access said communications device with a correct at least one identifier after the absence of the correct at least one identifier is sensed before providing said communications device with at least one new identifier,

                  and providing access to said communications device if the correct at least one identifier is sensed during the predetermined number of attempts to access.

10

                  6.       The method of claim 2 wherein said communications system is a telephone system and said communications device is a telephone.

                  7.       The method of claim 1 wherein said communications system is a computer network with said entities authorized to access said communications device being authorized computers having access to said computer network, said communications device including at least one host computer having access to said computer network.

                  8.       The method of claim 7 which includes periodically changing the at least one identifier for the host computer and providing the changed at least one identifier to the authorized computers.

                  9.       The method of claim 7 which includes providing the authorized computers with an unchangeable, accessible address for the host computer which is used by the authorized computer to activate and transmit the at least one identifier for the host computer when the authorized computer initiates access to the host computer.

                  10.      The method of claim 8 which includes providing each authorized computer with an authorized computer identifier,

30                   providing the host computer with a destination identifier,

                  causing each authorized computer to access said host computer with at least a host

computer destination identifier and the authorized computer identifier,

sensing the presence or absence of both said host computer destination identifier and an authorized computer identifier before granting access to said host computer,

5 providing access to said host computer when the use of both a correct host computer destination identifier and an authorized computer identifier is sensed, and

denying access to said host computer and providing said host computer with a new host computer destination identifier when the absence of either a correct host computer destination identifier or a correct authorized computer identifier is sensed.

10 11. The method of claim 10 which includes permitting a predetermined number of attempts to access said host computer with both a correct host computer destination identifier and an authorized computer identifier after the absence of a correct host computer destination identifier or an authorized computer identifier is sensed before providing said host computer with a new host computer destination identifier, and

15 providing access to said host computer if correct host computer destination and authorized computer identifier are sensed during the predetermined number of attempts to access the host computer.

20 12. The method of claim 11 which includes storing said host computer destination identifier as an inaccessible identifier in said authorized computers, and providing said authorized computers with an unchangeable, accessible host computer address, which will activate and transmit the host computer destination identifier when an authorized computer initiates access to the host computer.

25 13. The method of claim 8 which includes providing said host computer with a host computer destination identifier and a host computer port identifier,

causing each authorized computer to access said host computer with at least the host computer destination identifier and the host computer port identifier,

30 sensing the presence or absence of both said host computer destination identifier and said host computer port identifier before granting access to said host computer,

providing access to said host computer when the use of both a correct host computer

destination identifier and a correct host computer port identifier are sensed, and

denying access to said host computer and providing said host computer with a new destination identifier and port identifier when the absence of either or both of a correct host computer destination or port identifier is sensed.

5

14. The method of claim 13 which includes permitting a predetermined number of attempts to access said host computer with both a correct host computer destination and port identifier when either or both an incorrect host computer destination or port identifier is sensed before providing said host computer with a new destination and port identifier, and

10 providing access to said host computer if both correct host computer destination and port identifiers are sensed during the predetermined number of attempts to access said host computer.

15 15. The method of claim 14 which includes storing said host computer destination and port identifiers as inaccessible identifiers in said authorized computers and providing said authorized computers with an unchangeable, accessible host computer address which will activate and transmit the host computer destination and port identifiers when an authorized computer initiates access to said host computer.

20 16. An intrusion protection method for protecting a host computer connected to a computer communications system which includes one or more authorized computers having access to said computer communications system which are authorized to access said host computer which includes:

25 providing each authorized computer with an authorized computer identifying address, providing said host computer with a host computer destination identifier and a host computer port identifier,

providing said host computer destination identifier and said host computer port identifier to said authorized computers,

30 causing each authorized computer to access said host computer with the host computer destination and port identifiers and said authorized computer identifying address,

sensing the presence or absence of said host computer destination and port identifiers

and said authorized computer identifying address before granting access to said host computer,

providing access to said host computer when the use of correct computer destination and port identifiers and a correct authorized computer identifying address is sensed, and

5 denying immediate access to said host computer when the absence of any one or more of the correct host computer destination and port identifiers or the authorized computer identifying address is sensed.

17. The method of claim 16 which includes periodically changing the host computer destination and port identifiers and providing these changes to the authorized computers.

18. The method of claim 17 which includes storing said host computer destination and port identifiers as inaccessible identifiers in said authorized computer and providing said authorized computers with an unchangeable, accessible host computer address which will activate and transmit the host computer destination and port identifiers when an authorized computer initiates access to said host computer.

19. The method of claim 16 which includes changing the host computer destination and port identifiers when access is denied to said host computer after at least one access attempt has been made and providing these changed identifiers to the authorized computers.

20. The method of claim 16 which includes permitting a predetermined number of attempts to access said host computer with correct host computer destination and port identifiers and a correct authorized computer identifying address after the absence of at least a correct one of said identifiers and authorized computer identifying address is sensed by the host computer and

providing access to said host computer if correct host computer destination and port identifiers and a correct authorized computer identifying address are sensed during the predetermined number of attempts to access said host computer.



21. The method of claim 19 which includes storing said host computer destination and port identifiers as inaccessible identifiers in said authorized computer and providing said authorized computers with an unchangeable, accessible host computer address which will activate and cause transmission of the host computer destination and port identifiers when an  
5 authorized computer initiates access to said host computer.

22. The method of claim 20 which includes changing the host computer destination and port identifiers when access is denied to said host computer after at least one access attempt has been made and providing these changed identifiers to the authorized  
10 computers.

23. The method of claim 22 which includes storing said host computer destination and port identifiers as inaccessible identifiers in said authorized computer and providing said authorized computers with an unchangeable, accessible host computer address which will  
15 activate and cause transmission of the host computer destination and port identifiers when an authorized computer initiates access to said host computer.

24. A method of communication with a remote entity over a communication system which includes  
20 providing the remote entity with at least one remote entity cyber coordinate identifier, providing the remote entity cyber coordinate identifier to one or more base entities authorized to communicate with said remote entity, periodically changing the remote entity cyber coordinate identifier to a new remote entity cyber coordinate identifier and  
25 providing the new remote entity cyber coordinate identifier to said one or more base entities.

25. The method of claim 24 which includes changing the remote entity cyber coordinate identifier to a new cyber coordinate identifier in response to an attempt to  
30 communicate with said remote entity with an incorrect remote entity cyber coordinate identifier and

providing the new remote entity cyber coordinate identifier to said one or more base entities.

FIG. 1

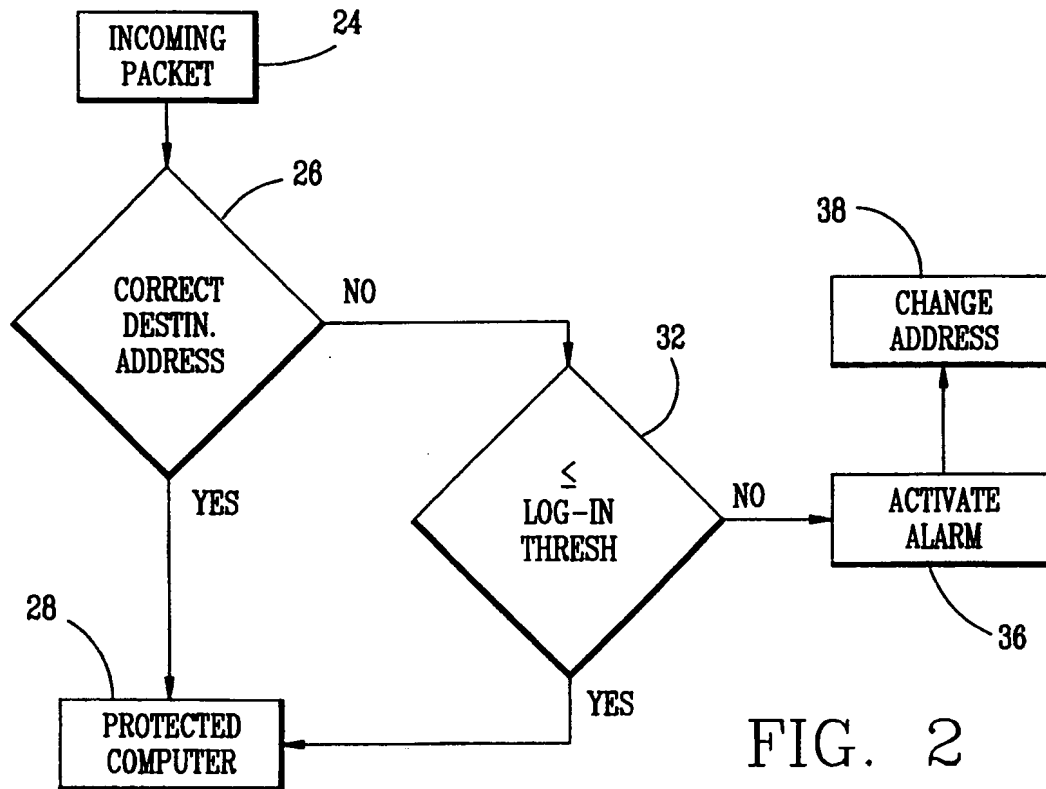
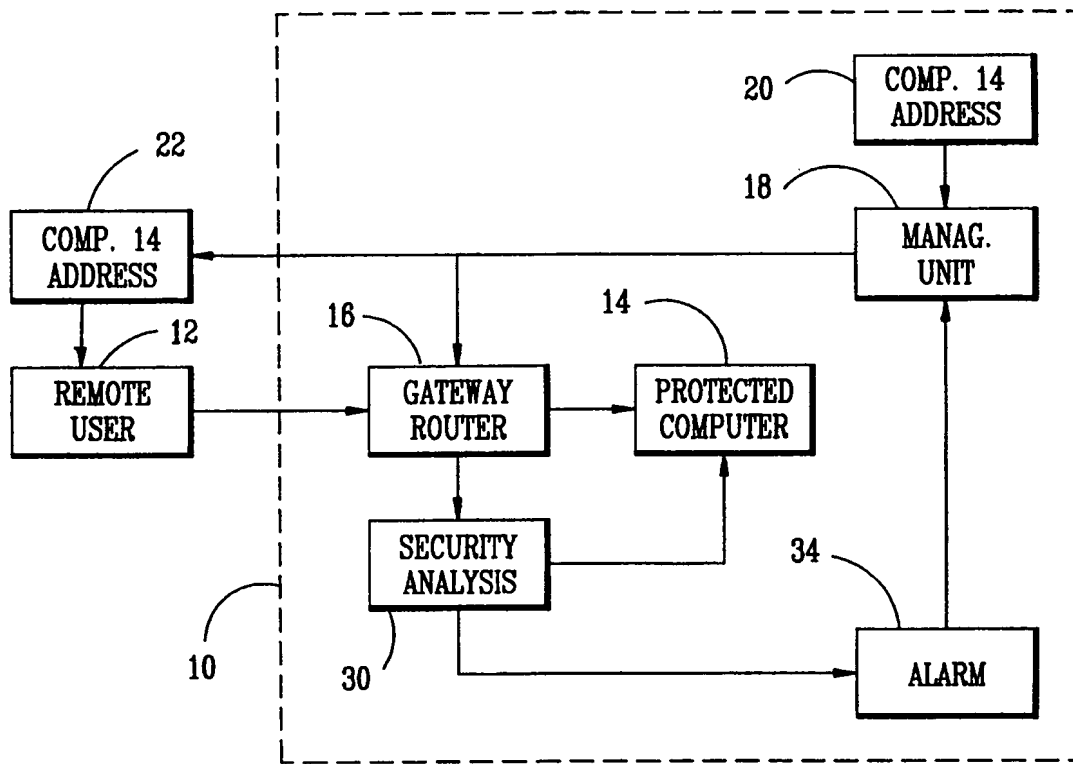


FIG. 2

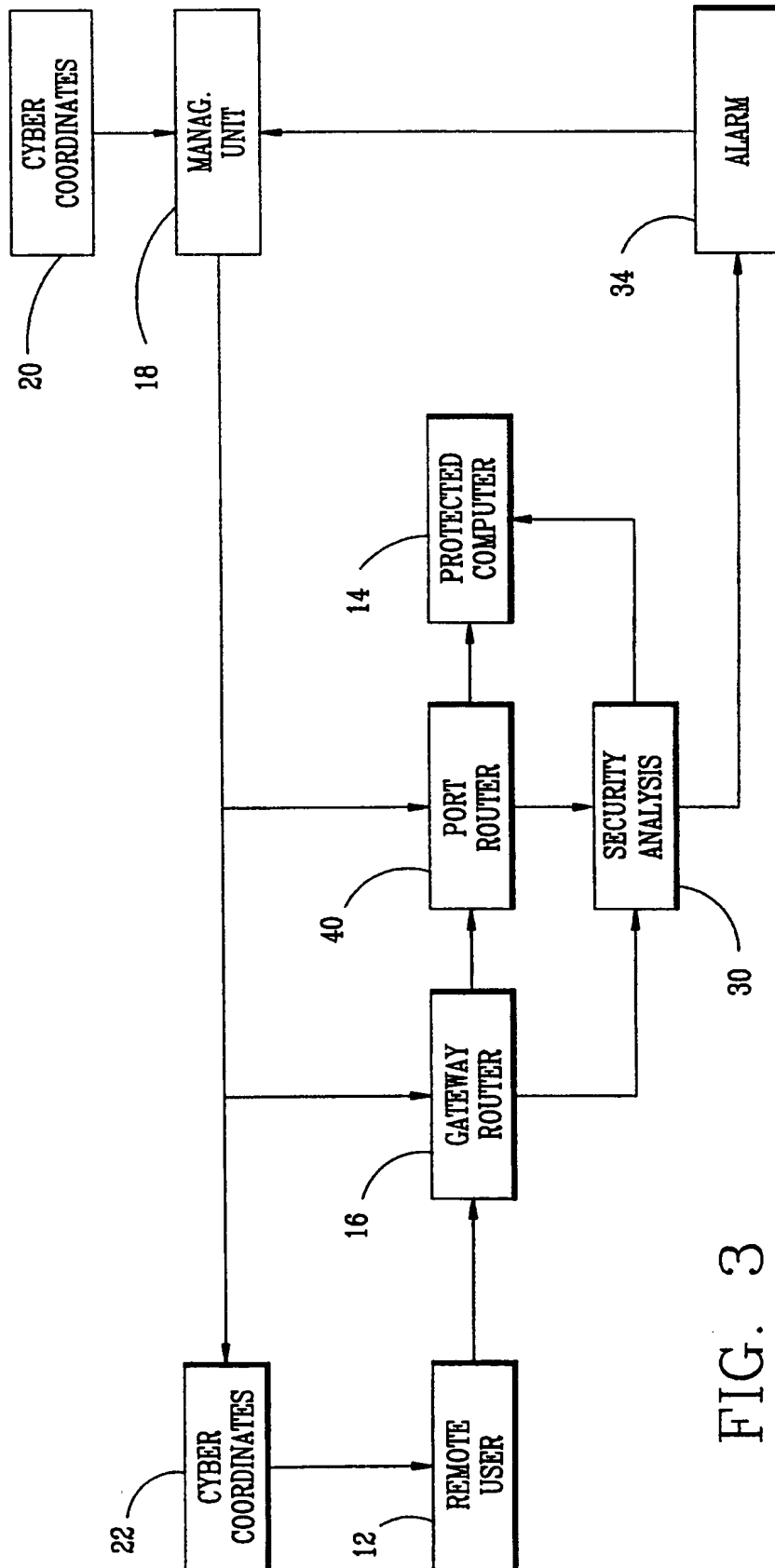
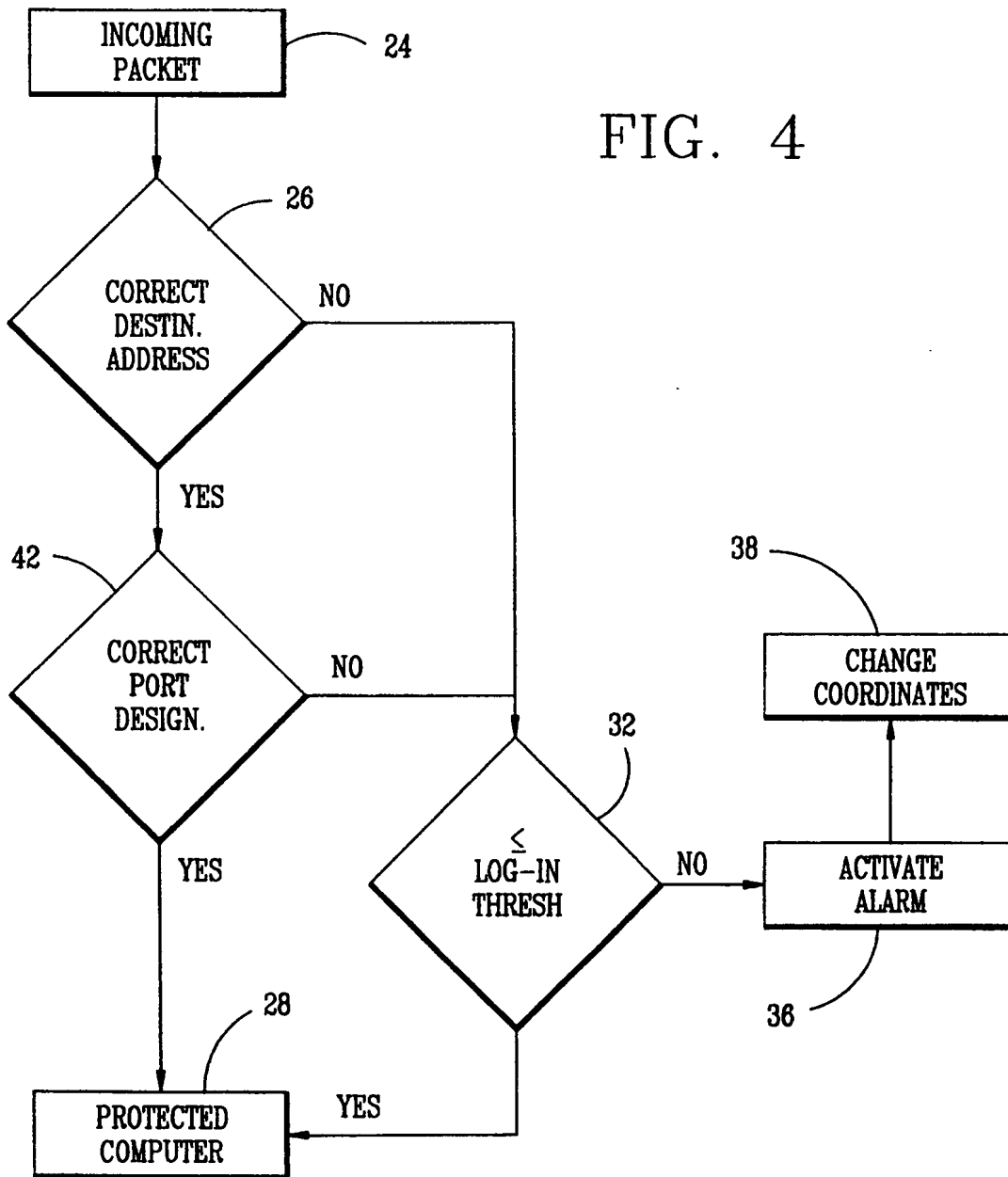


FIG. 3

FIG. 4



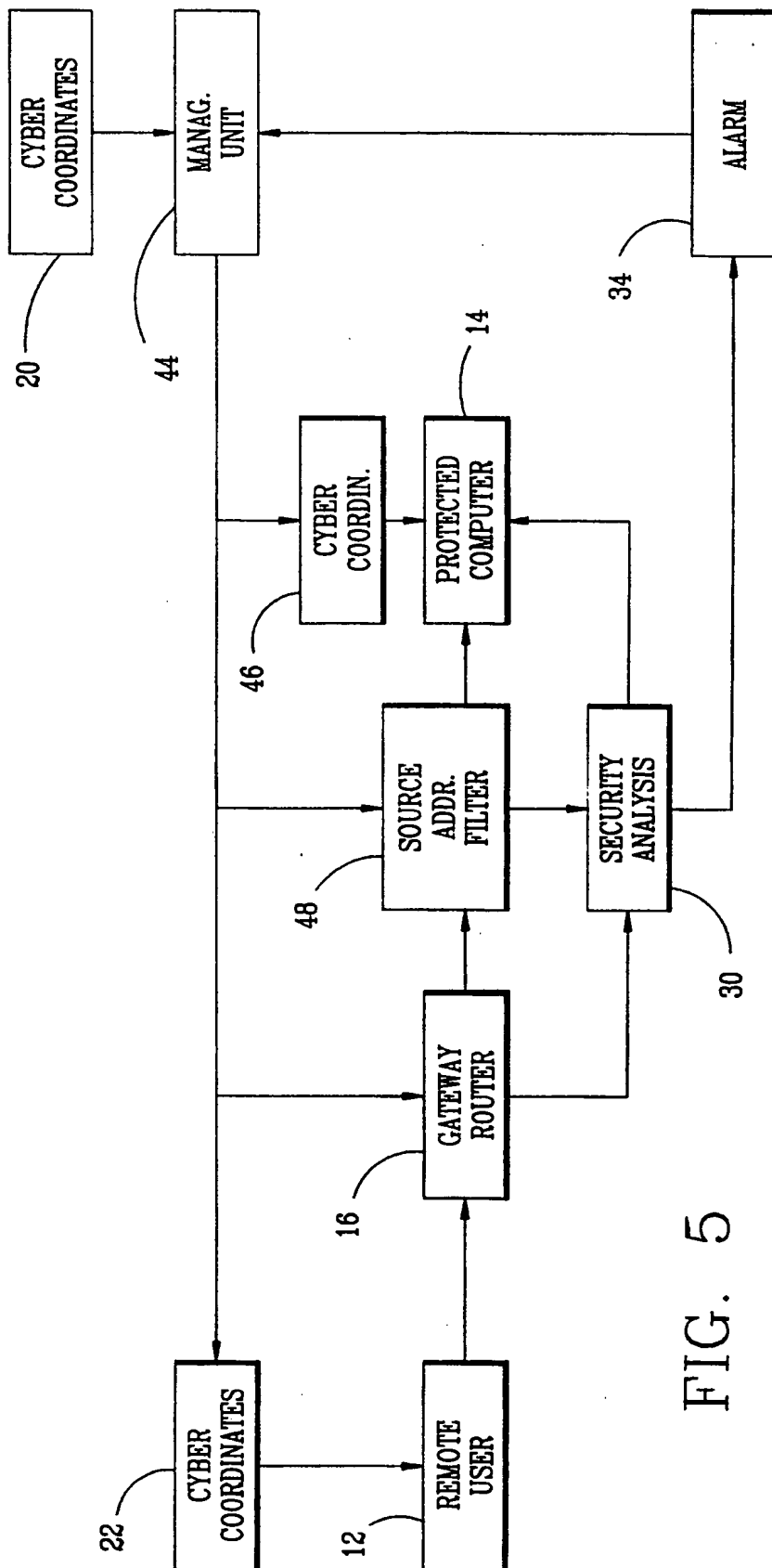


FIG. 5

FIG. 6

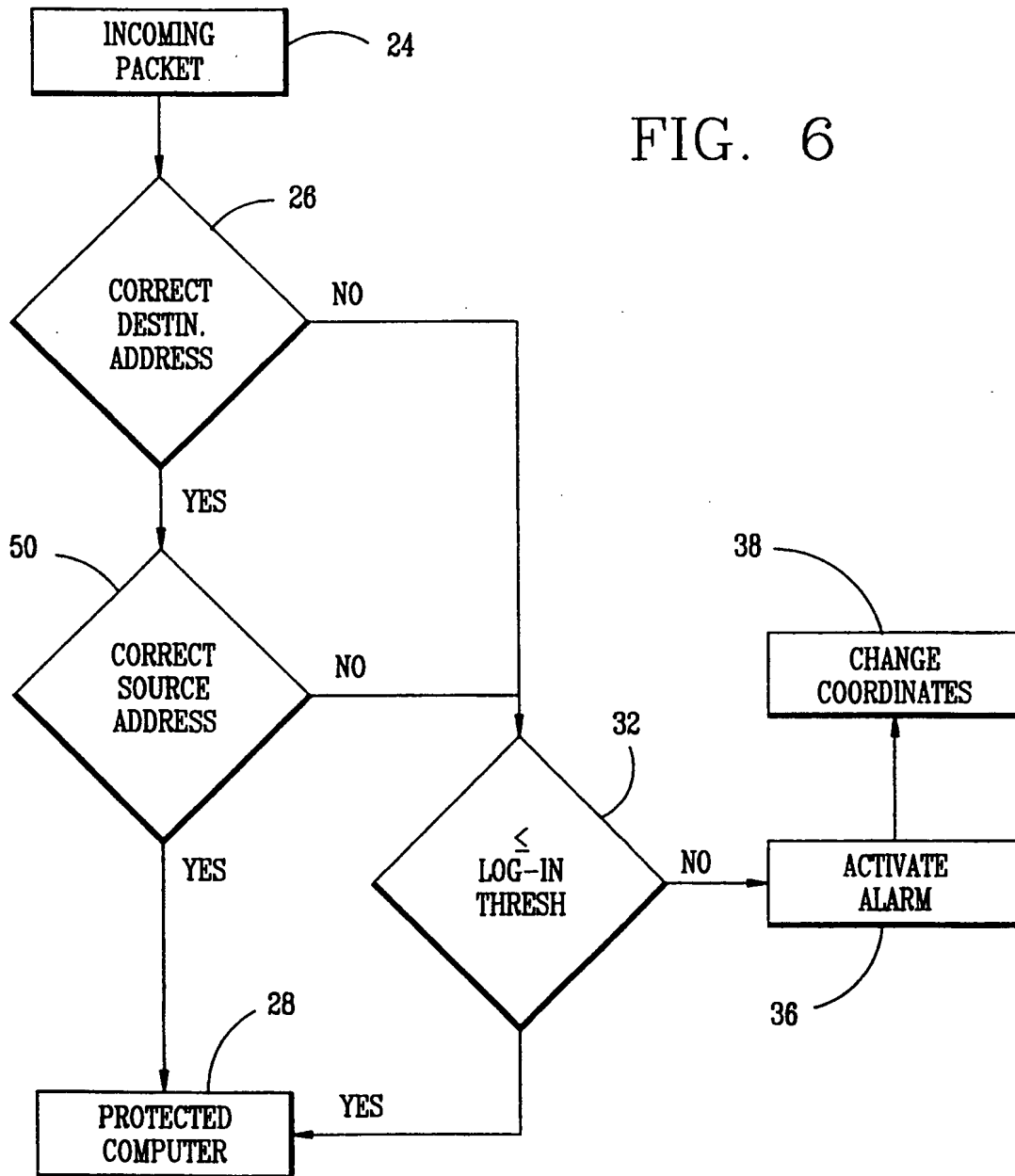
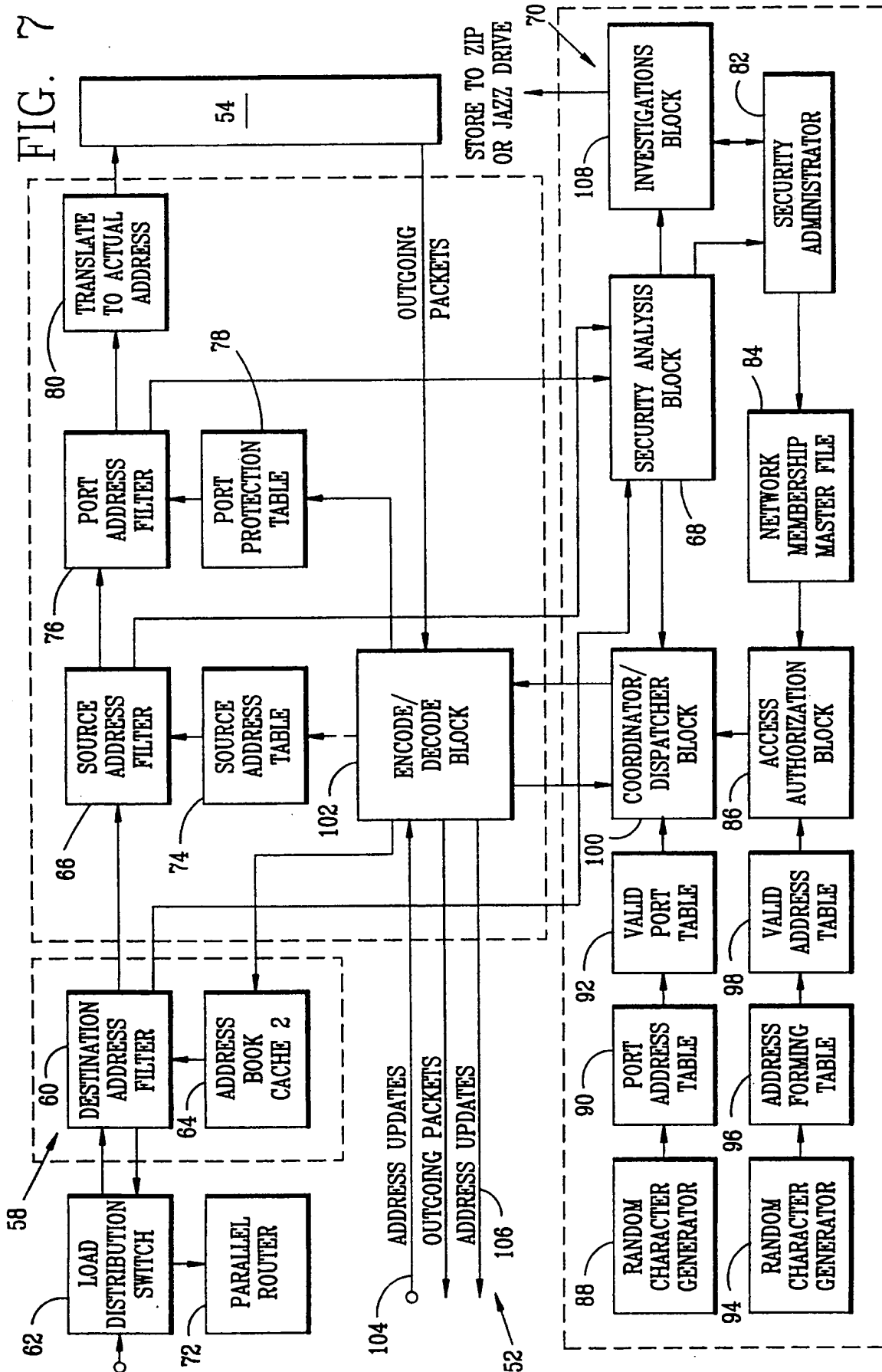


FIG. 7





MG-



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER OF PATENTS AND TRADEMARKS  
Washington, D.C. 20231  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/504,783	02/15/2000	Edmund Colby Munger	00479.85672	8308

7590 03/13/2002

Banner & Witcoff, Ltd  
1001 G Street, NW  
Washington, DC 20001-4597

EXAMINER

LIM, KRISNA

ART UNIT	PAPER NUMBER
2153	

2153

DATE MAILED: 03/13/2002

#8

Please find below and/or attached an Office communication concerning this application or proceeding.

H G

<b>Office Action Summary</b>	Application No. 09/504,783	Applicant(s) MUNGER ET AL.	
	Examiner Krisna Lim	Art Unit 2153	

-- 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 05 February 2002.
- 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) 28-39 and 67-81 is/are pending in the application.
  - 4a) Of the above claim(s) 72-81 is/are withdrawn from consideration.
- 5)  Claim(s) \_\_\_\_\_ is/are allowed.
- 6)  Claim(s) 28-37 and 67-69 is/are rejected.
- 7)  Claim(s) 38,39,70 and 71 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).
- 11)  The proposed drawing correction filed on \_\_\_\_\_ is: a)  approved b)  disapproved by the Examiner.  
If approved, corrected drawings are required in reply to this Office action.
- 12)  The oath or declaration is objected to by the Examiner.

**Priority under 35 U.S.C. §§ 119 and 120**

- 13)  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.
- 14)  Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).
  - a)  The translation of the foreign language provisional application has been received.
- 15)  Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

**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) Paper No(s) 2, 3.
- 4)  Interview Summary (PTO-413) Paper No(s) \_\_\_\_\_.
- 5)  Notice of Informal Patent Application (PTO-152)
- 6)  Other: \_\_\_\_\_.

1. Applicant's election without traverse of Group II (claims 28-39 and 67-71) in Paper No. 5 (filed 1/28/02) is acknowledged.

2. Claims 28-39 and 67-71 are pending for examination, and claims 72-81 are newly added for examination.

3. Restriction to one of the following inventions is required under 35 U.S.C. § 121:

II. Claims 28-39 and 67-71, drawn to a system for transparently creating a virtual private network (VPN) between a client computer and a target computer, comprising: a ) generating from the client computer a DNS request ..., b) determining whether the DNS ..., c) determining that the DNS ..., classified in Class 709, subclass 249.

IV. Claims 72-81, drawn to a method for establishing an encrypted channel between a client and a secure host, comprising the step of automatically creating the encrypted channel upon intercepting a DNS request for a domain name comprising a predetermined domain name extension, classified in Class 713, subclass 201.

4. Inventions II and V are related as subcombinations disclosed as usable together in a single combination. The subcombinations are distinct from each other if they are

shown to be separately usable. In the instant case, invention II has separate utility such as a method of registering a node that does not support Mobile IP with a Home Agent that support Mobile IP lacks the step of automatically creating the encrypted channel upon intercepting a DNS request for a domain name comprising a predetermined domain name extension.

5. These inventions are distinct for the reasons given above, and the search required for each Group is different and not co-extensive for examination purpose.

6. For example, the searches for the four inventions would not be co-extensive because these groups would require different searches on PTO's classification class and subclass as following:

1) The Group II search (claims 28-39 and 67-71) would require use of search class 709, subclass 249 ( which would not required for the group IV).

2) The Group IV search (claims 72-81) would require use of search class 713, subclass 201 ( which would not required for the group II).

7. Newly submitted claims 72-81 are directed to an invention that is independent or distinct from the invention originally claimed as mentioned in paragraphs 3-6 above.

Since applicant has received an action on the merits for the originally presented invention, this invention has been constructively elected by original presentation for

prosecution on the merits. Accordingly, claims 72-81 are withdrawn from consideration as being directed to a non-elected invention. See 37 CFR 1.142(b) and MPEP § 821.03.

8. 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 negated by the manner in which the invention was made.

9. Claims are rejected under 35 U.S.C. § 103(a) as being unpatentable over Boden et al. [U.S. Patent No. 6,330,562 B1] in view Risley et al. [U.S. Patent No. 6,332,158 B1].

10. Boden et al. disclosed (e.g., see Figs. 3A, 3B, 3C and 3D) the invention substantially as claimed. Taking claim 37 as an exemplary claim, the reference disclosed a system for dynamically establishing a virtual private network (VPN) (e.g., see col. 3, lines 14-20) with different security policies and other attributes between a client computer and a target computer, and a system for supporting with dynamically-assigned IP addresses that wished to establish a VPN connection with the locally system and means for maintaining secure connections at the IP level with other VPN

nodes (e.g., see the last 2 lines of the abstract) and means for enabling handling of remote initiating hosts with dynamically assigned IP address.

11. Boden et al., however, did not explicitly detail a DNS proxy server that receives a request from a client computer to look up an IP address for a domain name either returned the IP address for the request domain name or returned an error message. Such feature was clearly taught by Risley et al. (e.g., see an abstract, the teaching of query [www.bessementures.com](http://www.bessementures.com) and the return answer of 180.201.15.250 of Fig. 1A, the teaching Fig. 1B, the detail teaching of Fig. 4, col. 1 (lines 48-50, 55-61), col. 2, col. 5 (lines 45-50)).

Establishing a secure connection between computers with the use of VPN would have been a desired feature in the art as suggested by the Boden et al. (e.g., see col. 1, lines 41-55)). In addition, the system that made it easier to remember, access, and convey the location information in order to access information would have been also a desired feature in the art as suggested by Risley et al. (e.g, see col. 1, lines 46-52). Thus, it would have been obvious to one of ordinary skill in the art to combine the teaching of these two references in order to have an easier to use and secure network connection because the teaching of these two references are complemented each other for easier to use and for securing network connection in a computer network.

12. Claims 38-39 and 70-71 are 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.

13. Claims 28-36 and 67-69 are similar in scope as of claims 37, and therefore claims 28-36 and 67-69 are rejected for the same reasons set forth above for claim 37.

14. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

The references are cited in the Form PTO-892 for the applicant's review.

A shortened statutory period for response to this action is set to expire 3 (three) months and 0 (zero) days from the mail date of this letter. Failure to respond within the period for response will result in **ABANDONMENT** of the application (see 35 U.S.C 133, M.P.E.P 710.02, 710.02(b)).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Examiner Krisna Lim whose telephone number is (703) 305-9672. The examiner can normally be reached on Monday-Friday from 7:00 to 3:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Mr. Glenton Burgess, can be reached at (703) 305-4772. The fax phone numbers for the organization where this application or proceeding is assigned is are as following:

(703) 746-7238 [After Final Communication]

or

(703) 746-7239 [Official Communication]

(703) 746-7240 [For Status inquiries, draft communication]

and/or

(703) 306-5631, (703) 306-5632 or (703) 306-5633 for [Customer Service Numbers]

Any inquiry of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is (703) 305-3900.

All Internet e-mail communication will be made of record in the application file. PTO employees do not engage in Internet communications where there exists a possibility that sensitive information could be identified or exchanged unless the record includes a properly signed express waiver of the confidentiality requirement of 35 U.S.C. 122. This is more clearly set forth in the Interim Internet Usage Policy published in the Office Gazette of the Patent and Trademark on February 25, 1997 at 1195 OG 89.

kl

March 10, 2002



KRISNA LIM  
PRIMARY EXAMINER



**Notice of References Cited**

Application/Control No. 09/504,783	Applicant(s)/Patent Under Reexamination MUNGER ET AL.	
Examiner Krisna Lim	Art Unit 2153	Page 1 of 1

**U.S. PATENT DOCUMENTS**

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-6,330,562 B1	12-2001	Boden et al.	707/10
	B	US-6,332,158 B1	12-2001	Risley et al.	709/219
	C	US-5,878,231	03-1999	Baehr et al.	709/243
	D	US-5,898,830	04-1999	Wesinger et al.	709/225
	E	US-6,226,751 B1	05-2001	Arrow et al.	370/351
	F	US-6,286,047 B1	09-2001	Ramanathan et al.	345/733
	G	US-6,178,505 B1	01-2001	Schneider et al.	713/168
	H	US-6079020	06-2000	Liu	713/201
	I	US-6,016,318	01-2000	Tomoike, Hiroyuki	370/338
	J	US-6,353,614 B1	03-2002	Borella et al.	370/389
	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	

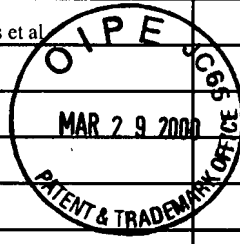
\*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)  
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

PTO-1449 (MPEP)  U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE  INFORMATION DISCLOSURE STATEMENT BY APPLICANT	ATTY. DOCKET NO. 00479.85672	SERIAL NUMBER 09/504,783
	APPLICANTS Edmund C. Munger et al.	
	FILING DATE February 15, 2000	GROUP ART UNIT 2153

**RECEIVED**  
 JUL 26 2000  
 Group 2700

**U.S. PATENT DOCUMENTS**

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE
<i>KL</i>	4,933,846	06/12/90	Humphrey et al.			
<i>KL</i>	5,842,040	11/24/98	Hughes et al.			



**FOREIGN PATENT DOCUMENTS**

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO

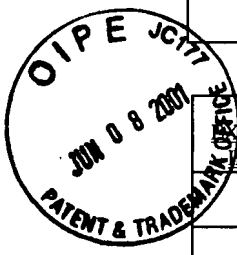
**OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)**

<i>KL</i>	Reiter, Michael K. and Rubin, Aviel D. (AT&T Labs - Research), "Crowds: Anonymity for Web Transactions", pages 1-23.
<i>KL</i>	Dolev, Shlomi and Ostrovsky, Rafail, "Efficient Anonymous Multicast and Reception" (Extended Abstract), 16 pages.
<i>KL</i>	Rubin, Aviel D., Geer, Daniel, and Ranum, Marcus J. (Wiley Computer Publishing), "Web Security Sourcebook", pages 82-94.

EXAMINER	<i>KRISNA LIM</i>	DATE CONSIDERED	<i>11/16/01</i>
----------	-------------------	-----------------	-----------------

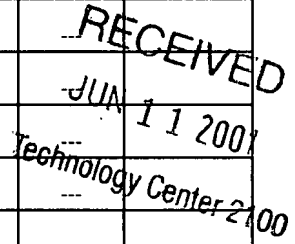
EXAMINER: Initial citation if reference was considered. Draw line through citation if not in conformance to MPEP 609 and not considered. Include copy of this form with next communication to applicant.

PTO-1449 (Modified)  U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE  SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT BY APPLICANT	ATTY. DOCKET NO. 00479.85672	APPLICATION NUMBER 09/504,783
	APPLICANTS  Edmund Colby Munger et al.	
	FILING DATE November 15, 2000	GROUP ART UNIT 2100



U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE
<i>K</i>	6,119,171	9/2000	Alkhatib	<del>---</del>	<del>---</del>	
				---	---	
				---	---	
				---	---	
				---	---	
				---	---	
				---	---	
				---	---	
				---	---	
				---	---	



FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)


EXAMINER	<i>KRISNA LIM</i>	DATE CONSIDERED	<i>11/16/01</i>
EXAMINER: Initial citation if reference was considered. Draw line through citation if not in conformance to MPEP 609 and not considered. Include copy of this form with next communication to applicant.			

PTO-1449 (Modified)  U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE  INFORMATION DISCLOSURE STATEMENT BY APPLICANT	ATTY. DOCKET NO. 00479.85672	SERIAL NUMBER 09/504,783
	APPLICANT Edmund Colby Munger, et al.	
	FILING DATE 2/15/00	GROUP ART UNIT unknown

RECEIVED  
 SEP 21 2000  
 TC 2100 MAIL ROOM

O I P E JC78  
 SEP 25 2000  
 PATENT & TRADEMARK OFFICE

**U.S. PATENT DOCUMENTS**

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILED DATE

**FOREIGN PATENT DOCUMENTS**

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO

**OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)**

<i>K</i>	FASBENDER, KESDOGAN, and KUBITZ: "Variable and Scalable Security: Protection of Location Information in Mobile IP", IEEE publication, 1996, pages 963-967

EXAMINER <i>KRISNA LIM</i>	DATE CONSIDERED <i>11/16/07</i>
----------------------------	---------------------------------

EXAMINER: Initial citation if reference was considered. Draw line through citation if not in conformance to MPEP 609 and not considered. Include copy of this form with next communication to applicant.

Filed: September 25, 2000



#14/D  
mm  
PATENT 6.26.02

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application Of  
Edmund Colby MUNGER *et al.*  
Serial No.: 09/504,783  
Filed: February 15, 2000  
For: IMPROVEMENTS TO AN  
AGILE NETWORK  
PROTOCOL FOR SECURE  
COMMUNICATIONS WITH  
ASSURED SYSTEM  
AVAILABILITY

Group Art Unit: 2153  
Examiner: K. Lim  
Atty. Dkt. No. 00479.85672

RECEIVED  
JUN 24 2002  
Technology Center 2100

AMENDMENT AND RESPONSE UNDER 37 C.F.R. § 1.111

Assistant Commissioner for Patents  
Washington, D.C. 20231

Sir:

In response to the Office Action mailed March 13, 2002, Applicants respectfully request the application be amended as follows. No fee is believed to be due with this Request. However, if a fee is due the Office is authorized to charge any required fees for consideration of this paper to our Deposit Account No. 19-0733.

IN THE CLAIMS

Please cancel claims 72-81.

**Remarks**

Applicants are in receipt of the Office Action mailed March 13, 2002, indicating that claims 28-39 and 67-81 are pending, claims 72-81 are withdrawn from consideration, claims 28-37 and 67-

69 stand rejected, and claims 38, 39, 70 and 71 are objected to. Applicants thank the Examiner for the indication of allowable subject matter in claims 38, 39, 70, and 71.

Submitted concurrently herewith are formal drawings in substitution for the informal drawings submitted with the application as filed. Applicants respectfully request that the official draftsman reviews the formal drawings at his earliest convenience.

### ***Second Preliminary Amendment and IDS***

A Second Preliminary Amendment adding claims 82-91 was submitted on February 22, 2002, but this amendment was not reflected in the Office Action mailed on March 13, 2002. Applicants respectfully request that the Second Preliminary Amendment be entered as of the date of its receipt by the Office, and that the claims submitted in the Second Preliminary Amendment be considered simultaneously with the requested reconsideration of the pending claims.

A Supplemental Information Disclosure Statement was also submitted February 22, 2002, but was not reflected in the Office Action mailed on March 13, 2002. Applicants respectfully request that the references cited in the Supplemental Information Disclosure Statement be considered and acknowledged at the Examiner's earliest convenience.

### ***On the Merits***

The Office Action restricted newly added claims 72-81 (group IV) as being drawn to an independent or distinct invention from the originally claimed invention in claims 28-39 and 67-71 (group II), and constructively elected group II for prosecution on the merits. By the present amendment, Applicants cancel claims 72-81.

The Office Action rejected claims 28-37 and 67-69 under 35 U.S.C. § 103(a) as being unpatentable over *Boden et al.* (U.S. Pat. No. 6,330,562, hereinafter “Boden”) in view of *Risley et al.* (U.S. Pat. No. 6,332,158, hereinafter “Risley”). Applicants respectfully traverse this rejection based on the following arguments.

In order to reject a claim as obvious under § 103(a), three criteria must exist: 1) there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combined reference teachings; 2) there must be a reasonable expectation of success; and 3) the prior art reference(s) must teach or suggest all the claim limitations. See MPEP § 706.02 (j); *In re Vaeck*, 947 F.2d 488 (Fed. Cir. 1991).

First, Applicants submit that there is no motivation or suggestion to combine the Boden and Risley references. Boden discloses a data model for abstracting customer-defined VPN security policy information (Boden, Abstract). The system in Boden addresses the need to enable connection filter rules to be generated and loaded dynamically at negotiation time, due to remote initiating hosts having *dynamically assigned IP addresses*. Boden, col. 2, lines 38-41 (emphasis added). As cited in the Office Action, Boden allows for “dynamically establishing VPN connections with different security policies and other attributes, based solely on an unfixd IP address (e.g. [sic] *a user ID*)....” Boden, col. 3, lines 14-16 (emphasis added). Boden does not disclose establishing a VPN based on a DNS request for an IP address.

Risley discloses a DNS lookup system that allows intelligent correction of domain name searches by providing alternative suggestions of possible intended domain names when a DNS lookup was unsuccessful. Risley, Abstract. That is, when a user submits a domain name query, if the domain name exists, the domain name server (DNS) provides the corresponding machine address

back to the user, as is known in the art. However, if the domain name does not exist, the Risley domain name server returns a machine address for a machine that will help the user identify the desired domain name. Subsequently, the machine to which the user has been redirected suggests possible intended domain names based on heuristics such as common misspellings, phonetic errors, and the like. Risley, Abstract. Risley does not teach or suggest establishing a VPN based on a DNS request, nor establishing any sort of secure communications channel over a network.

The Office Action states that establishing a secure connection between computers with the use of VPN would have been a desired feature in the art as suggested by Boden at col. 1, lines 41-55. However, Boden at col. 1, lines 41-55, discusses a general need for computer security, not a specific suggestion to incorporate the VPN techniques disclosed in Boden, or any other security technique, with a DNS lookup assistant as disclosed by Risley. In addition, there are many ways in which to create a VPN, and Boden at best only discloses a single specific security solution that may be used to establish a VPN. Boden does not include any suggestion or motivation to alter a DNS request scheme to create a VPN (in fact, there is only one instance of the acronym DNS in the entire Boden specification, col. 10, line 3, and no instances of the phrase “domain name service”). Indeed, Boden specifically states that “no verification is made via DNS or similar that [the mapping of ID to IP address] is correct.” *Id.*

The Office Action also states that “the system that made it easier to remember, access, and convey the location information in order to access information would have been also a desired feature in the art as suggested by [Risley col. 1, lines 46-52].” However, Risley at col. 1, lines 46-52, discusses the general notion that users prefer using domain names (e.g., coolsite.com) rather than IP addresses (e.g., 199.227.249.232) when remembering, accessing, and conveying information. Risley does not provide a specific suggestion that its DNS service would benefit from the use of a VPN (or



any other type of security). Risley only discloses that users prefer to use domain names over IP addresses when remembering, accessing, and conveying information, and provides a system for helping a user identify an intended domain name.

The Office Action concludes that it would have been obvious to combine the references in order to have “an easier to use and secure network connection because the teachings of these two references are complemented each other for easier to use and for securing network connection in a computer network.” While two patents may ostensibly complement each other, this does not provide the necessary suggestion to combine the two references. In light of the fact that neither reference includes a specific suggestion to combine the references, the mere fact that two references are complementary does not provide the required suggestion or motivation. Risley does not teach or suggest establishing a VPN using its domain name resolution technique, nor does Boden teach using domain name resolution to establish a VPN.

To allow the combination of Boden and Risley would allow the hindsight combination of almost any two references as long as they had something in common, e.g., they both relate to the Internet. The Federal Circuit has repeatedly stated that the limitations of a claim in a pending application cannot be used as a blueprint to piece together prior art in hindsight, *In re Dembiczak*, 50 U.S.P.Q.2d 1614 (Fed. Cir. 1999), and that the Patent Office should *rigorously* apply the requirement that a teaching or motivation to combine prior art references needs to be provided. *Id.* (emphasis added). Thus, Applicants respectfully submit that that there is no motivation or suggestion to combine Risley, which discloses a modified DNS lookup system, with Boden, which discloses a specific VPN technique.

Second, even if the Boden and Risley references were combined, the combination would not teach or suggest all the limitations of any pending claim. The Office Action uses claim 37 as an exemplary claim, which requires:

a DNS proxy server that receives a request from the client computer to look up an IP address for a domain name, wherein the DNS proxy server returns the IP address for the requested domain name if it is determined that access to a non-secure web site has been requested, and wherein the DNS proxy server generates a request to create the VPN between the client computer and the secure target computer if it is determined that access to a secure web site has been requested; and

a gatekeeper computer that allocates resources for the VPN between the client computer and the secure web computer in response to the request by the DNS proxy server.

At a minimum, neither Boden nor Risley discloses a DNS proxy server that “generates a request to create the VPN between the client computer and the secure target computer if it is determined that access to a secure web site has been requested...” Neither Risley nor Boden teach or suggest triggering the creation of a VPN in response to a DNS request. Instead, Risley discloses a modified DNS lookup, whereby when a DNS request is received that is unsuccessful, Risley redirects the requestor to a domain name resolver to assist the user with locating an intended domain name. Risley does not disclose generating a request to create a VPN, as is required by claim 37, nor does Risley determine whether access to a secure web site has been requested. Likewise, Boden does not disclose these limitation, as is admitted in the Office Action at page 5, para. 11.

In addition, the Office Action does not indicate that either Boden or Risley includes a gatekeeper computer as is required by claim 37.

Based at least on the above arguments, Applicants respectfully traverse the rejection of claim 37 and its dependent claims.

The Office Action also rejected claims 28-36 and 67-69 for the same reasons set forth with respect to claim 37 because the claims are similar in scope. Applicants submit that each claim

presents an individually patentable scope, and that these claims are allowable for at least the same reasons as claim 37.

In addition, with respect to claim 31, none of the cited references teach or suggest, upon determining that a client computer is not authorized to establish a VPN with a secure web site, returning an error from the DNS request.

With respect to claim 32, none of the cited references teach or suggest, upon determining that a client computer is not authorized to resolve addresses of non-secure target computers, returning an error from the DNS request.

With respect to claim 33, none of the cited references teach or suggest establishing the VPN by creating an IP address hopping scheme between the client computer and the target computer. (see, e.g., allowable subject matter in claim 38).

With respect to claim 34, none of the cited references teach or suggest using a gatekeeper computer that allocates VPN resources for communicating between the client computer and the target computer.

With respect to claim 35, none of the cited references teach or suggest that step (2) is performed in a DNS proxy server that passes through the request to a DNS server if it is determined in step (3) that access is not being requested to a secure target web site.

With respect to claim 68, none of the cited references teach or suggest communicating according to a scheme by which at least one field in a series of data packets is periodically changed according to a known sequence.

With respect to claim 69, none of the cited references teach or suggest comparing an Internet Protocol (IP) address in a header of each data packet to a table of valid IP addresses maintained in a table in the second computer.

Based on the aforementioned Applicants respectfully submit that all pending claims are in condition for allowance, and Applicants request that the subject application be reconsidered and passed to issue at the Examiner's earliest possible convenience.

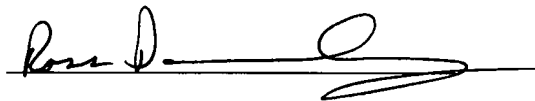
If the Examiner has any questions or wishes to discuss this amendment, the Examiner is invited to telephone the undersigned representative at the number set forth below.

Respectfully submitted,

**BANNER & WITCOFF, LTD.**

Date: June 13, 2002

By:



Bradley C. Wright  
Registration No. 38,061  
1001 G Street N.W., 11<sup>th</sup> Floor  
Washington, D.C. 20001  
(202) 508-9100

**Reg. No. 49,024**



TT 17  
08

PATENT APPLICATION

#12  
RECEIVED  
JUN 24 2002  
Technology Center 2100

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of

Group Art Unit: 2153

Edmond Colby Munger et al.

Examiner: K. Lim

Serial No. 09/504,783

Attorney Docket No. 00479.85672

Filed: February 15, 2000

For: IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

**SUBMISSION OF FORMAL DRAWINGS TO OFFICIAL DRAFTSMAN**

Assistant Commissioner for Patents  
Washington, D.C. 20231


Sir:

Please substitute the attached 35 sheets of formal drawings depicting Figures 1-32 for the informal drawings filed with the patent application on February 15, 2000, in this matter. Applicant respectfully requests the Official Draftsman to review these drawings and advise the undersigned of any objections thereto.

It is believed that no fee is required. However, if a fee is required, please charge our Deposit Account No. 19-0733.

Respectfully submitted,

Date: June 13, 2002

By:   
for Bradley C. Wright  
Registration No. 38,061 **Reg. No. 49,024**

BANNER & WITCOFF, LTD  
1001 G Street, N.W.  
Eleventh Floor  
Washington, D.C. 20001  
(202) 508-9100  
RAD/mmd

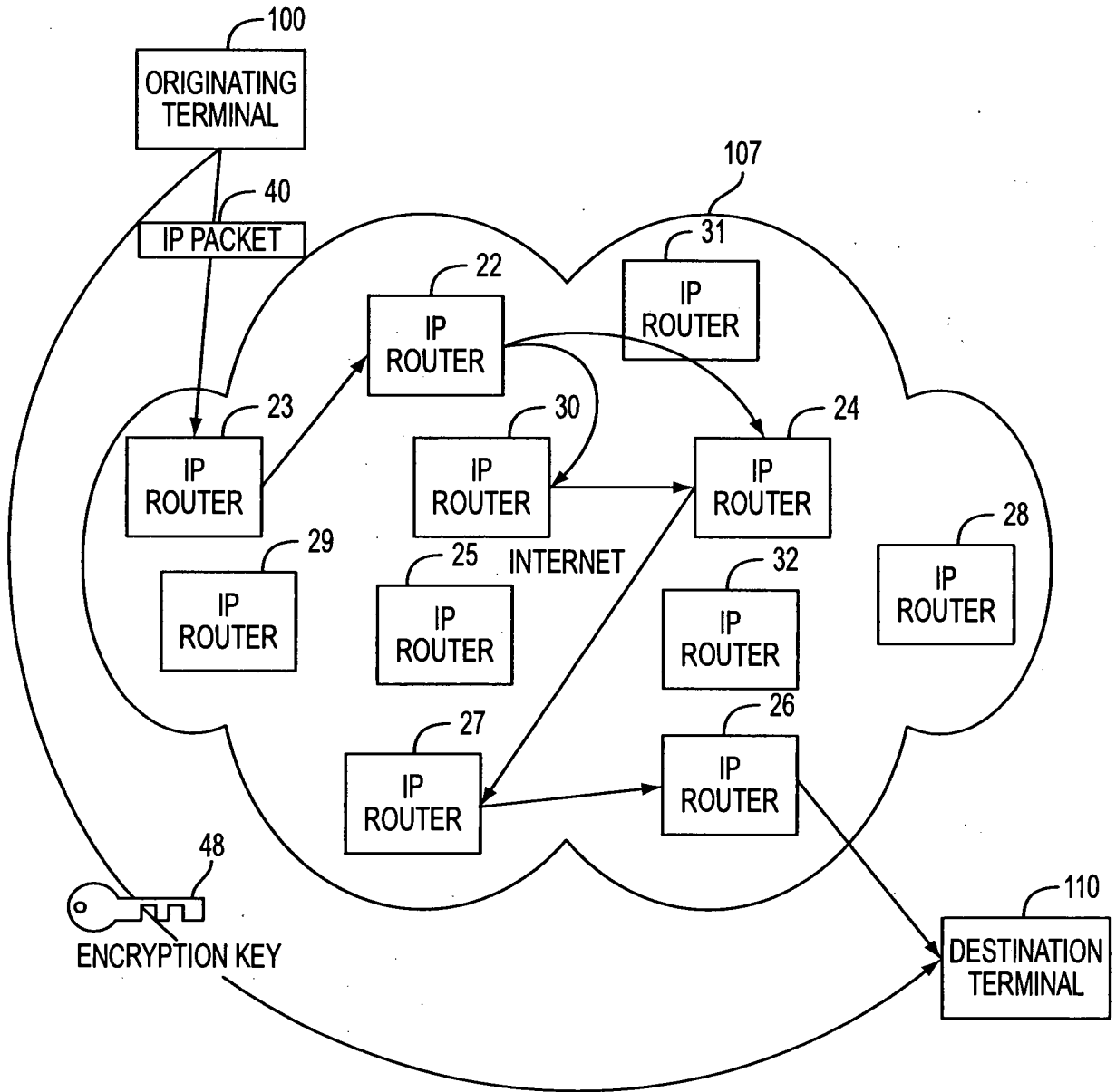


FIG. 1

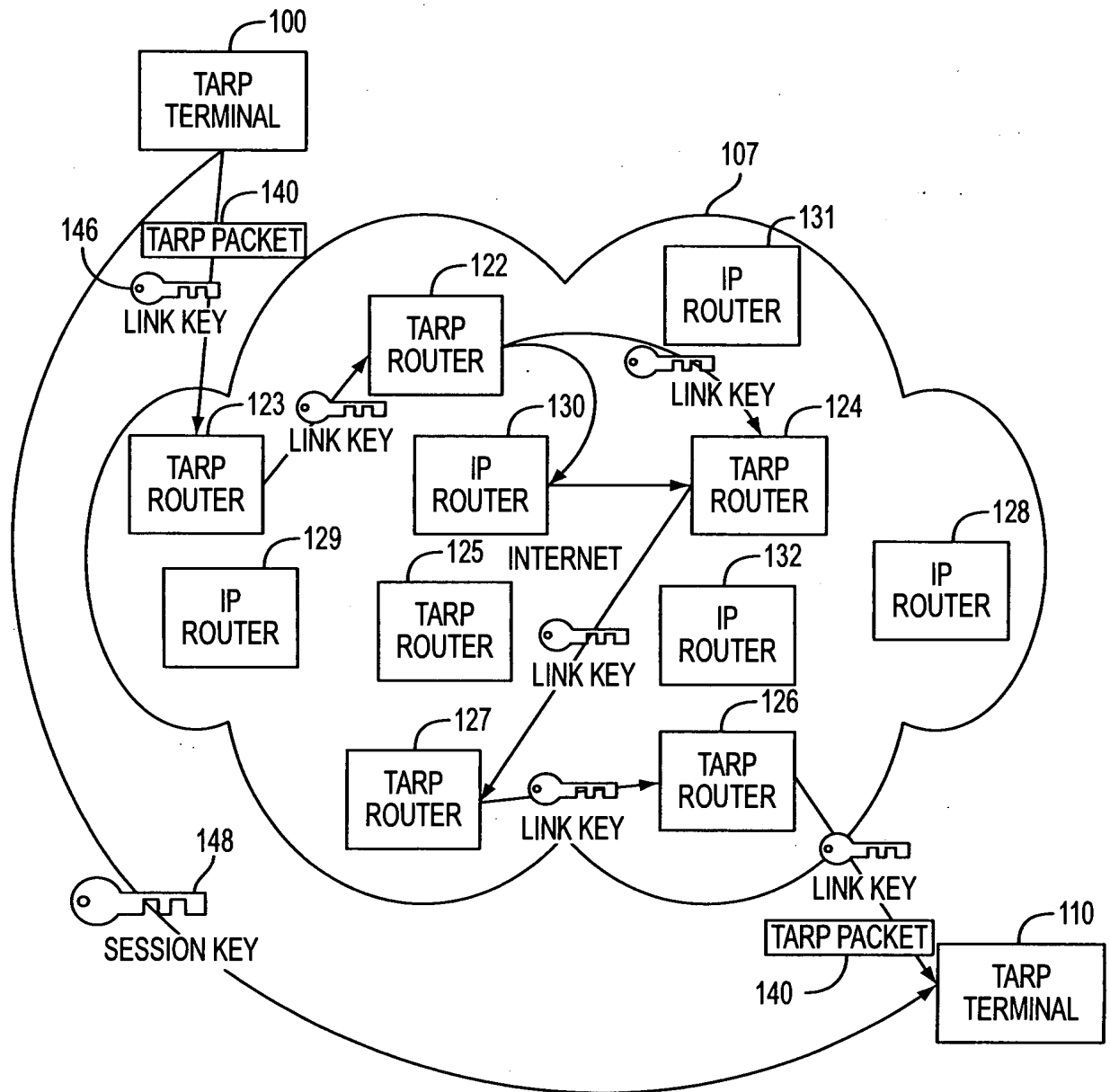


FIG. 2

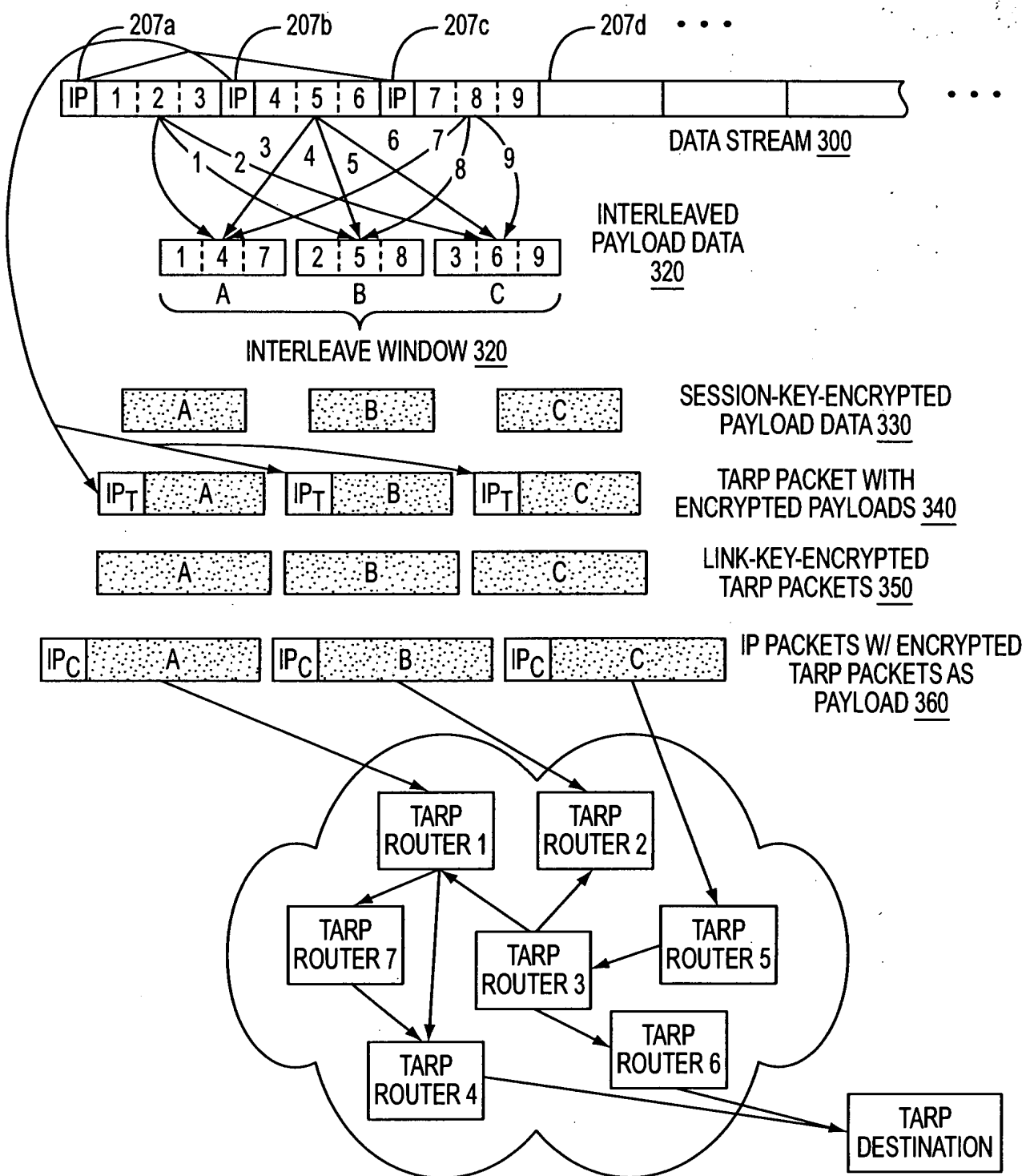


FIG. 3A



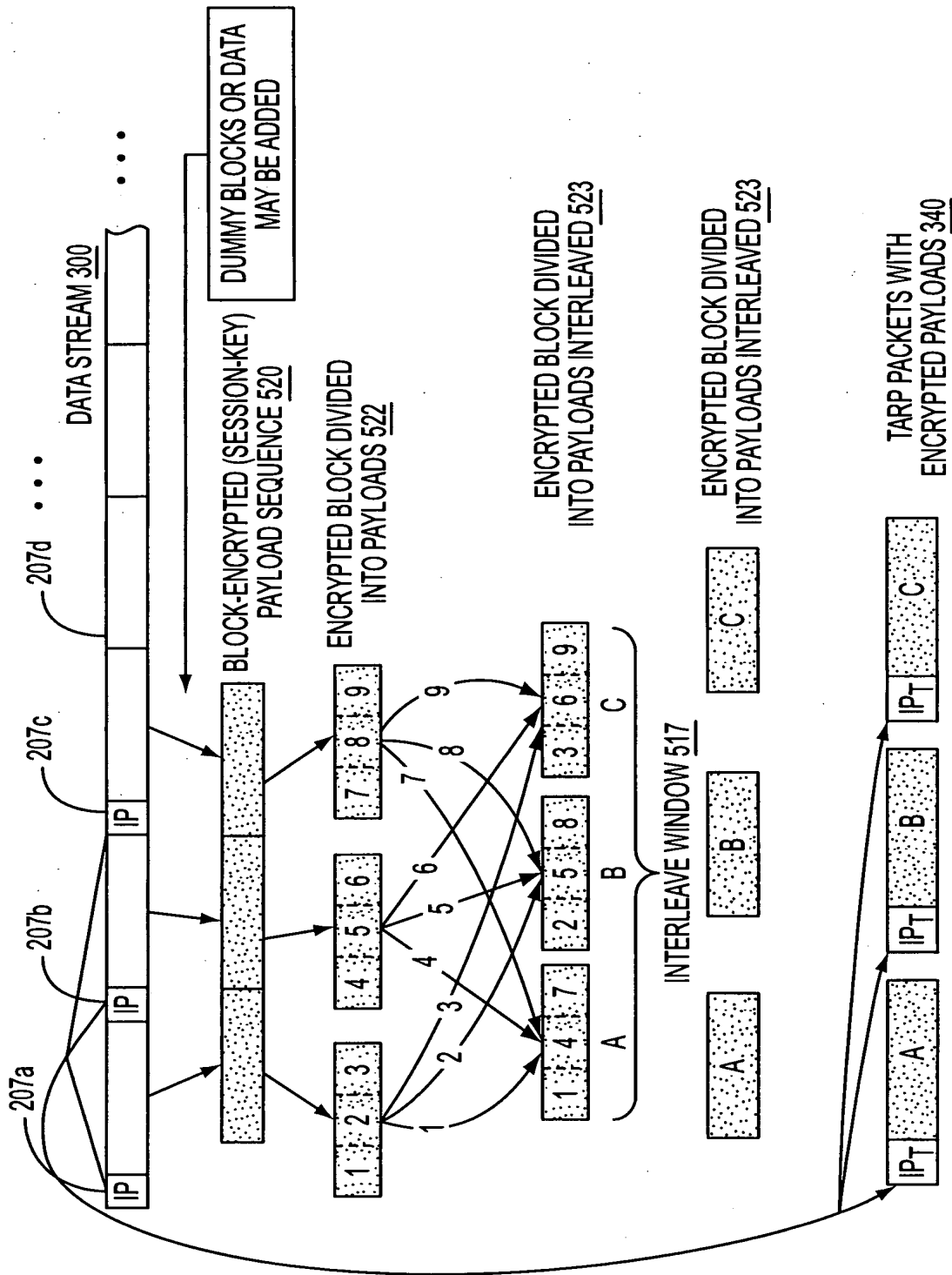


FIG. 3B

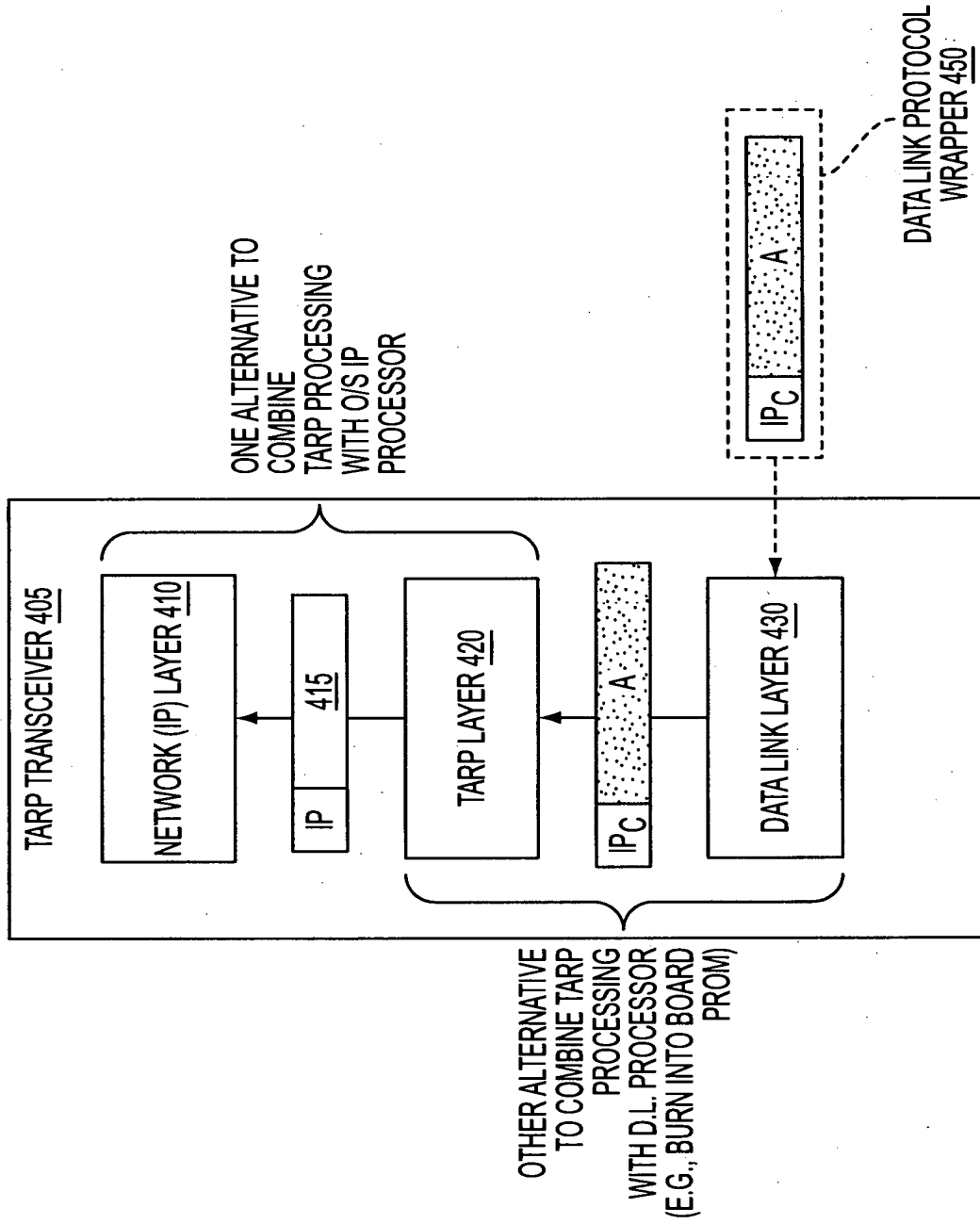


FIG. 4

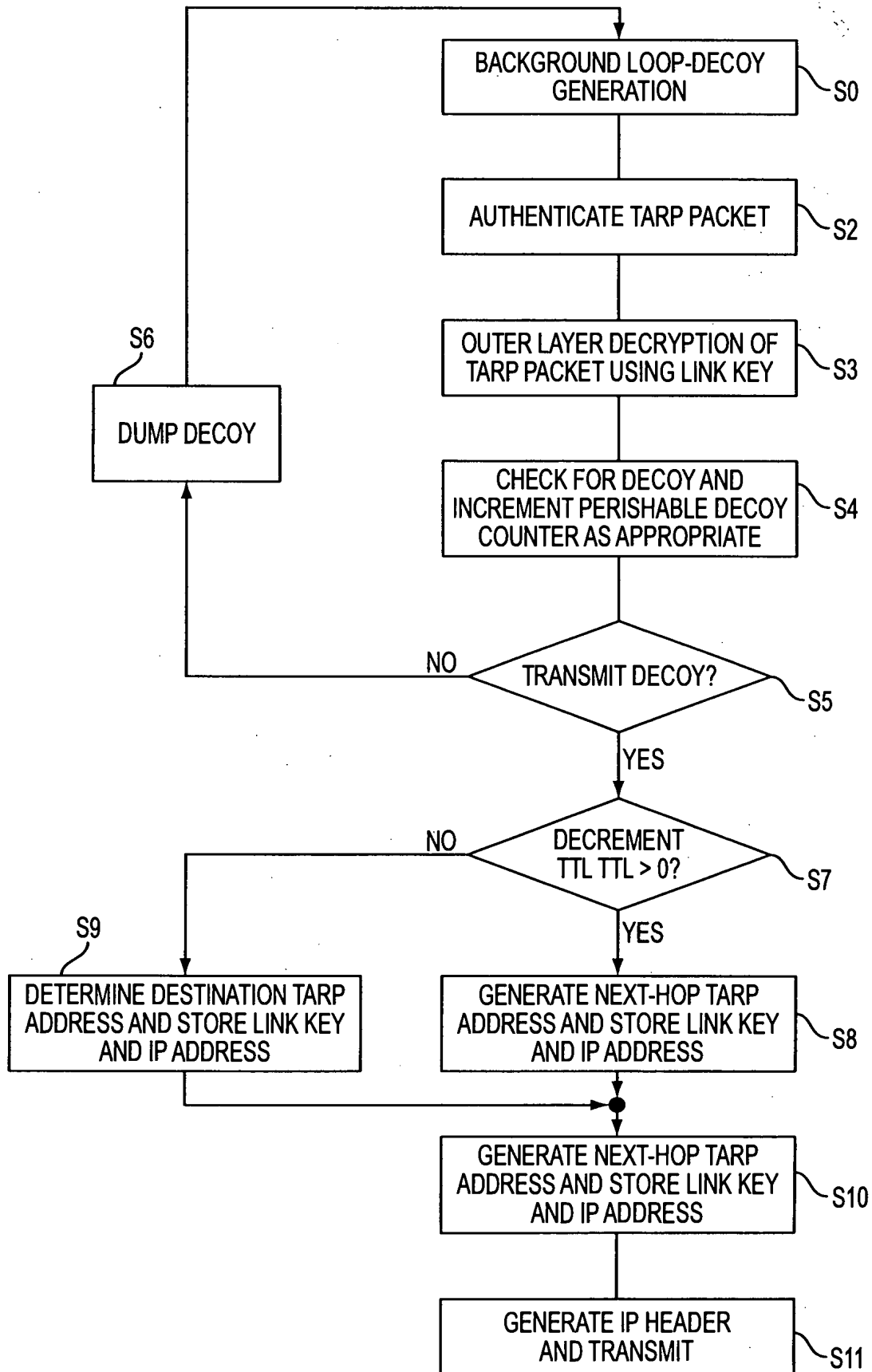


FIG. 5

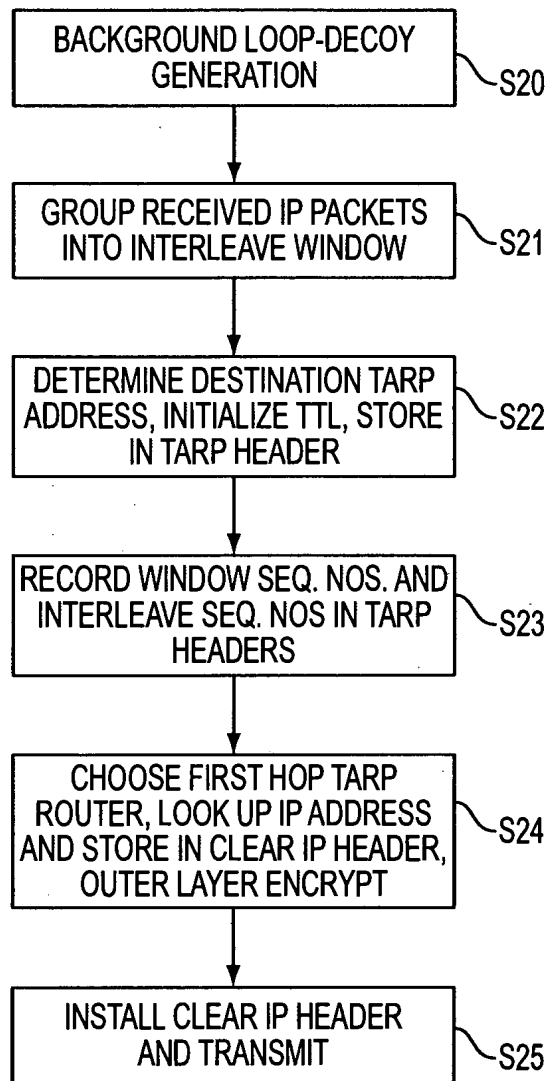


FIG. 6

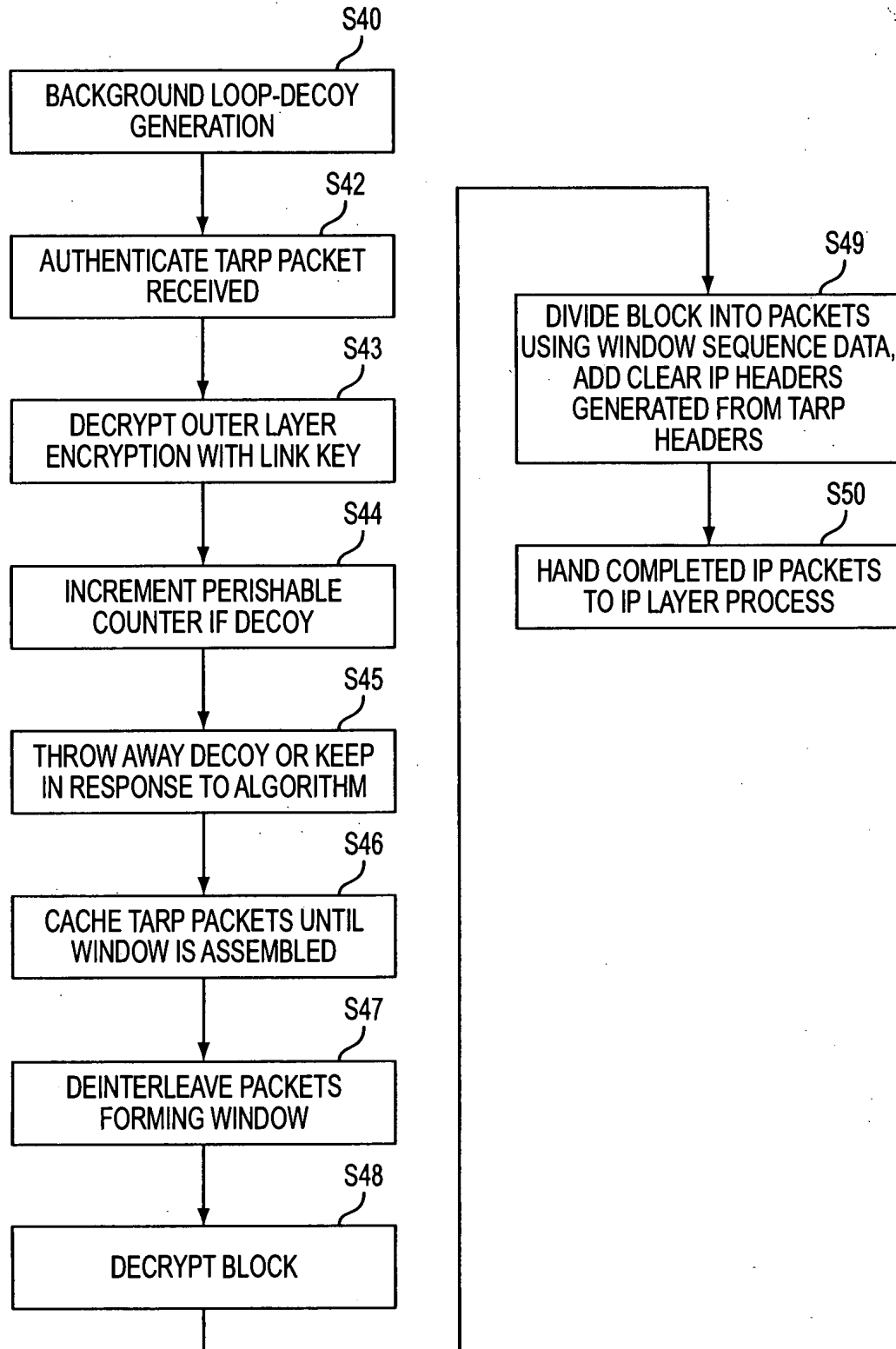


FIG. 7

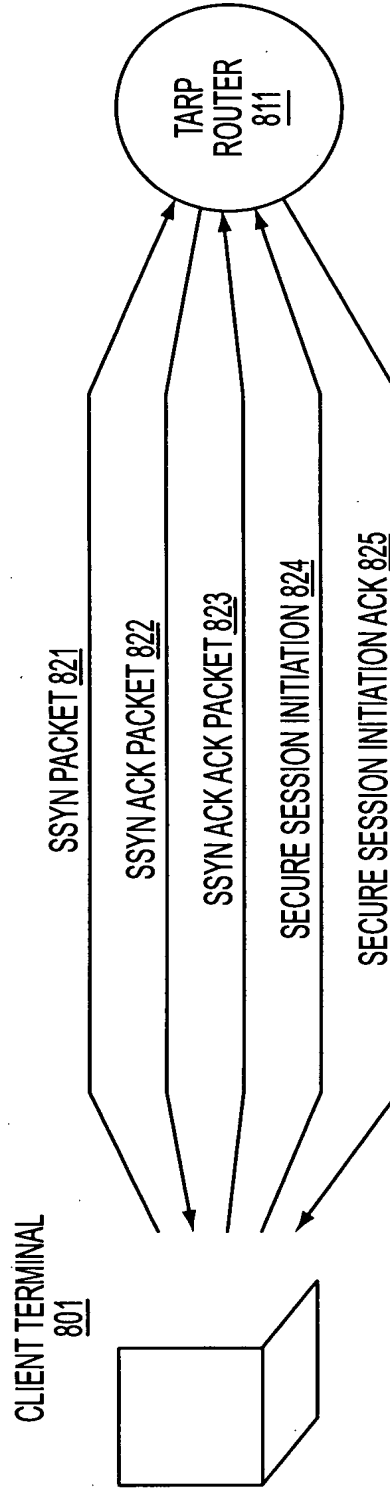
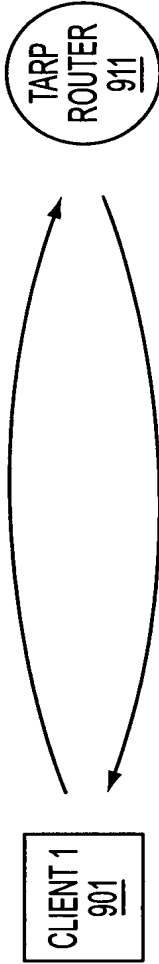


FIG. 8



TRANSMIT TABLE 921

131.218.204.98 ; 131.218.204.65  
 131.218.204.221 ; 131.218.204.97  
 131.218.204.139 ; 131.218.204.186  
 131.218.204.12 ; 131.218.204.55  
 .  
 .  
 .

RECEIVE TABLE 924

131.218.204.98 ; 131.218.204.65  
 131.218.204.221 ; 131.218.204.97  
 131.218.204.139 ; 131.218.204.186  
 131.218.204.12 ; 131.218.204.55  
 .  
 .  
 .

RECEIVE TABLE 922

131.218.204.161 ; 131.218.204.89  
 131.218.204.66 ; 131.218.204.212  
 131.218.204.201 ; 131.218.204.127  
 131.218.204.119 ; 131.218.204.49  
 .  
 .  
 .

TRANSMIT TABLE 923

131.218.204.161 ; 131.218.204.89  
 131.218.204.66 ; 131.218.204.212  
 131.218.204.201 ; 131.218.204.127  
 131.218.204.119 ; 131.218.204.49  
 .  
 .  
 .

FIG. 9

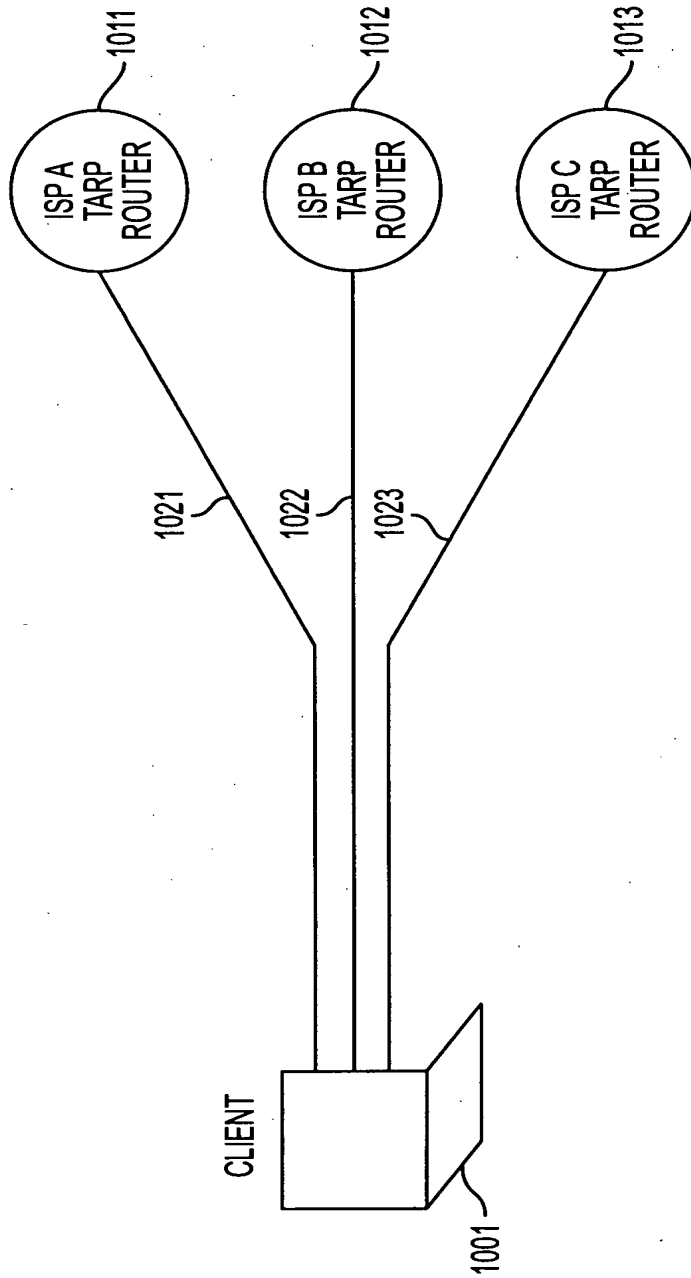


FIG. 10



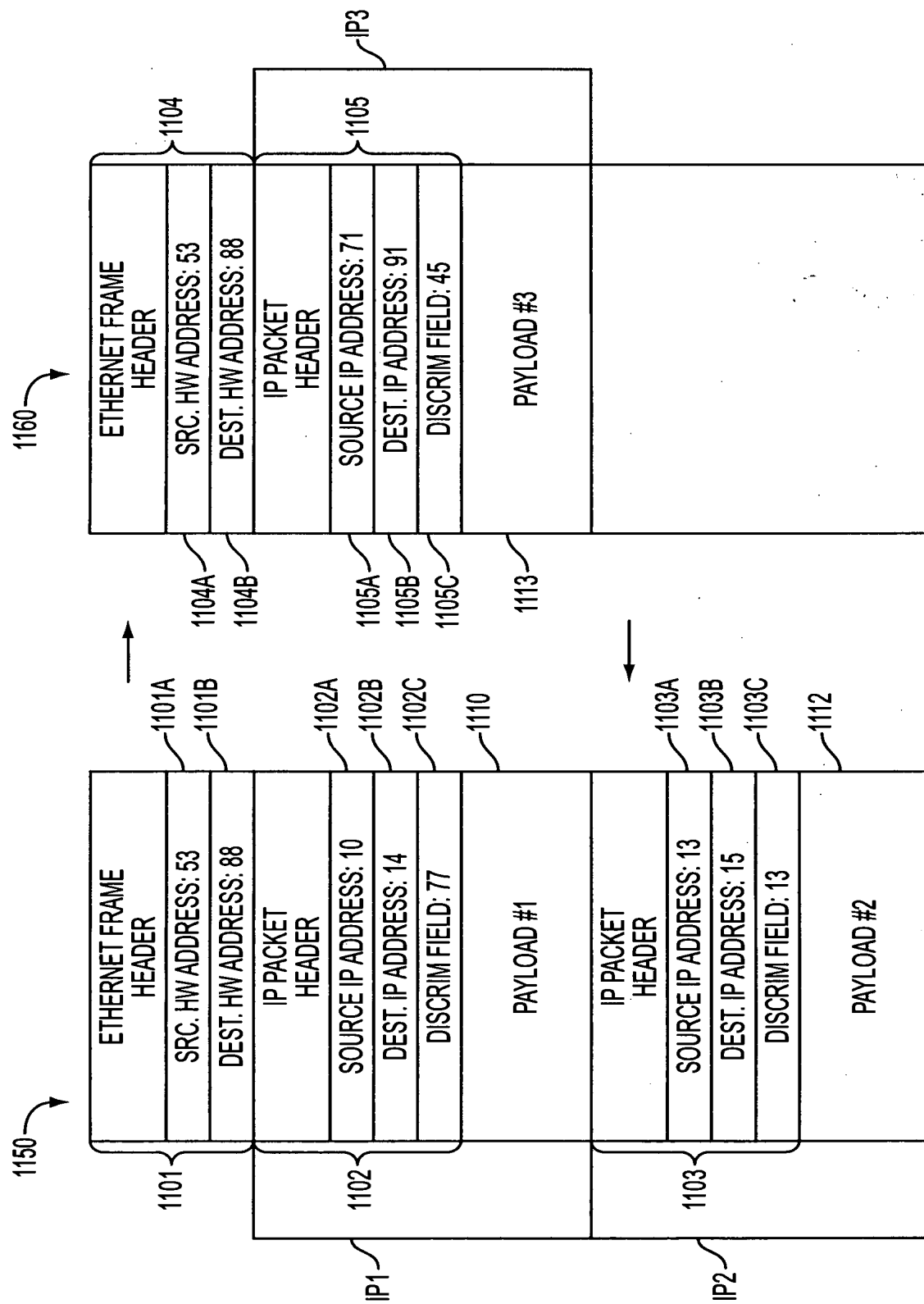


FIG. 11

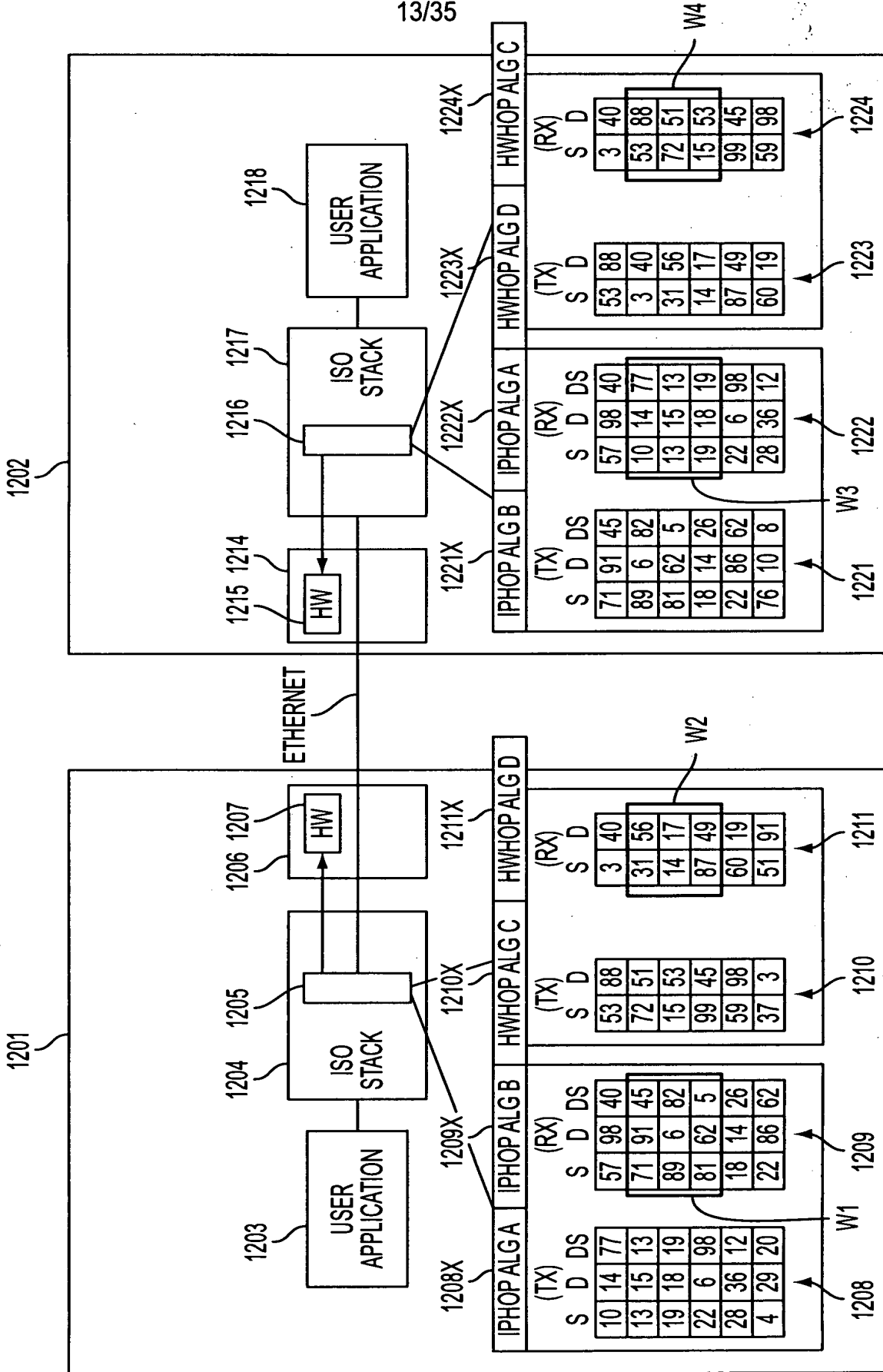


FIG. 12A

MODE OR EMBODIMENT	HARDWARE ADDRESSES	IP ADDRESSES	DISCRIMINATOR FIELD VALUES
1. PROMISCUOUS	SAME FOR ALL NODES OR COMPLETELY RANDOM	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC
2. PROMISCUOUS PER VPN	FIXED FOR EACH VPN	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC
3. HARDWARE HOPPING	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC

FIG. 12B

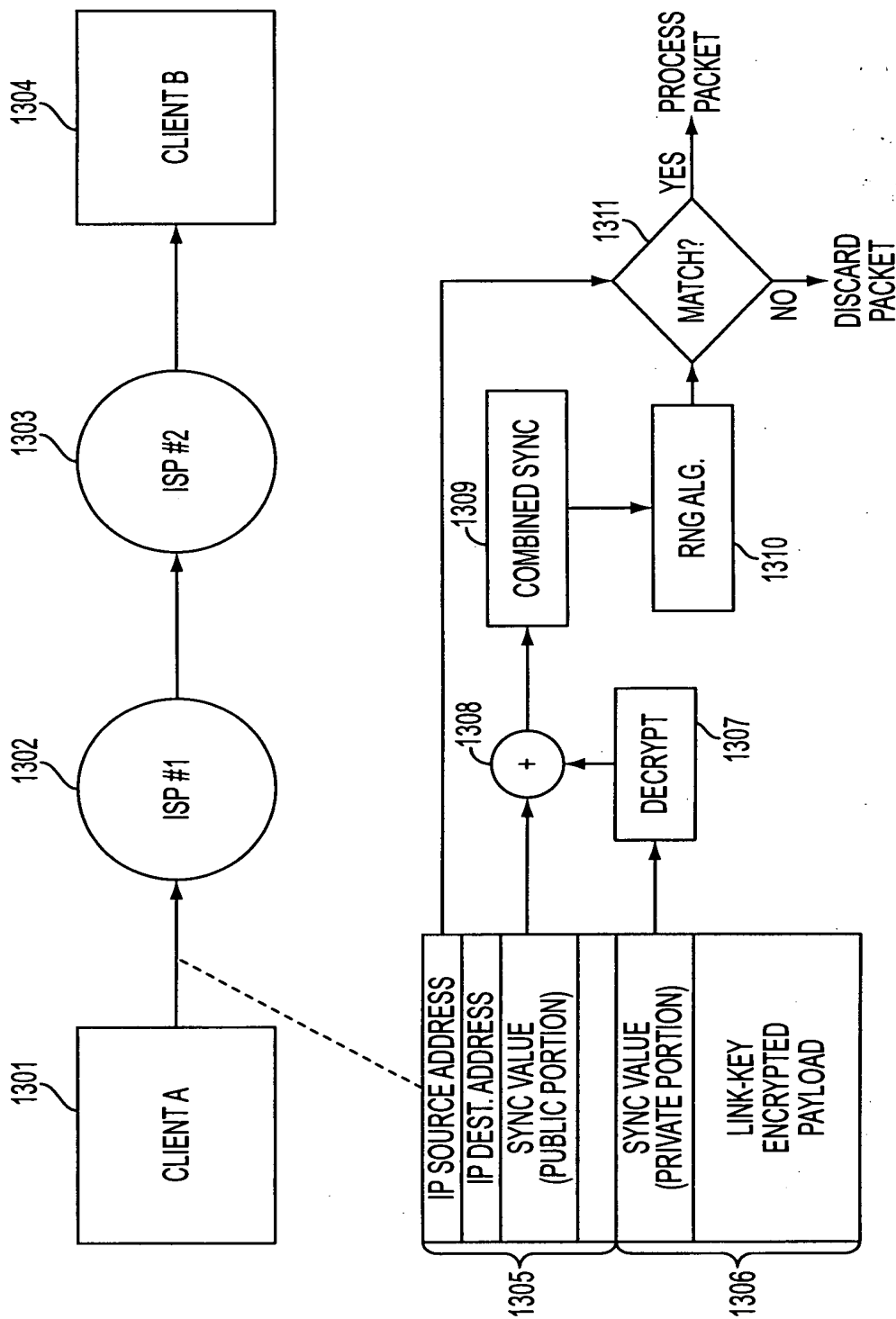


FIG. 13

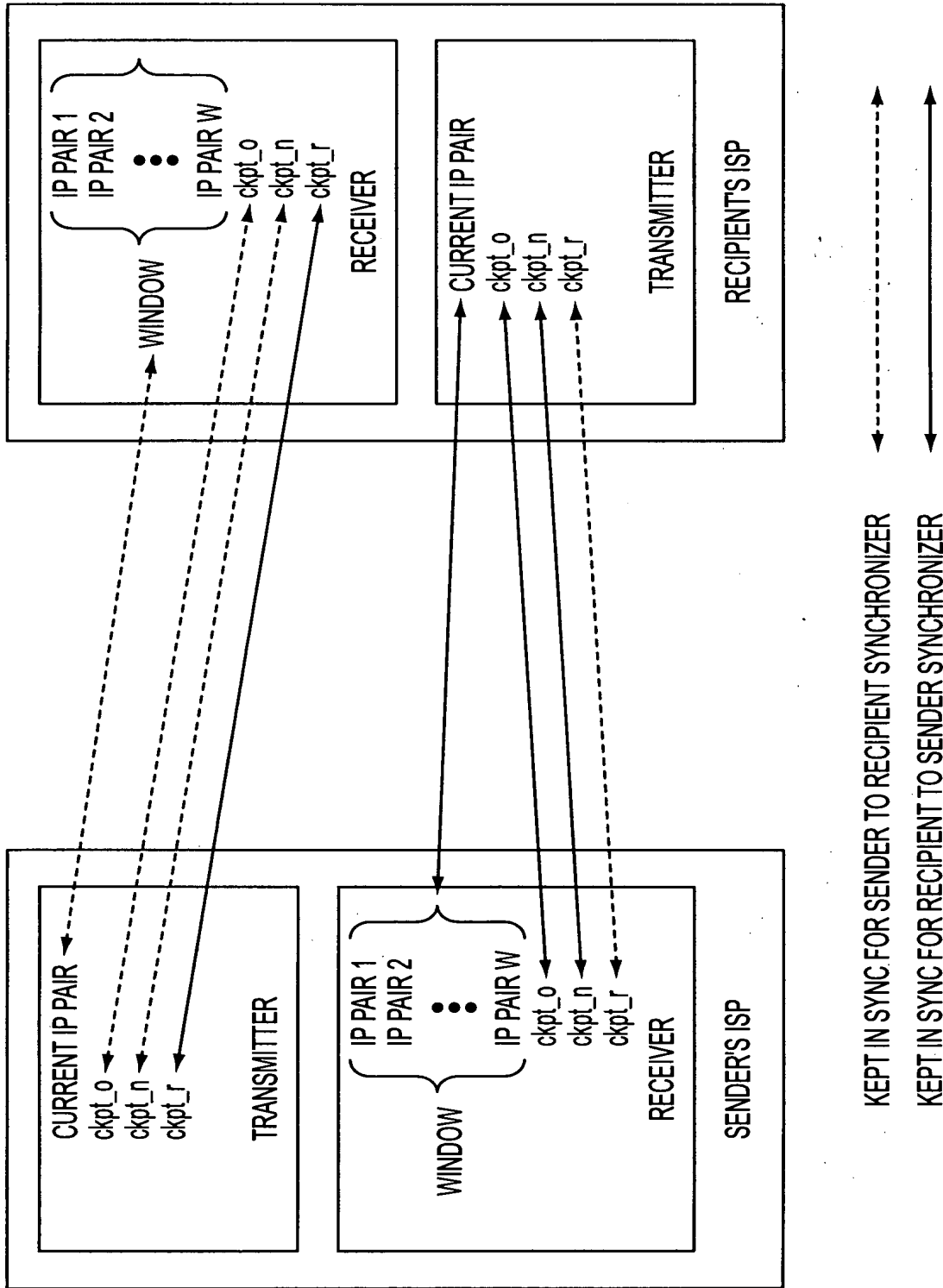


FIG. 14

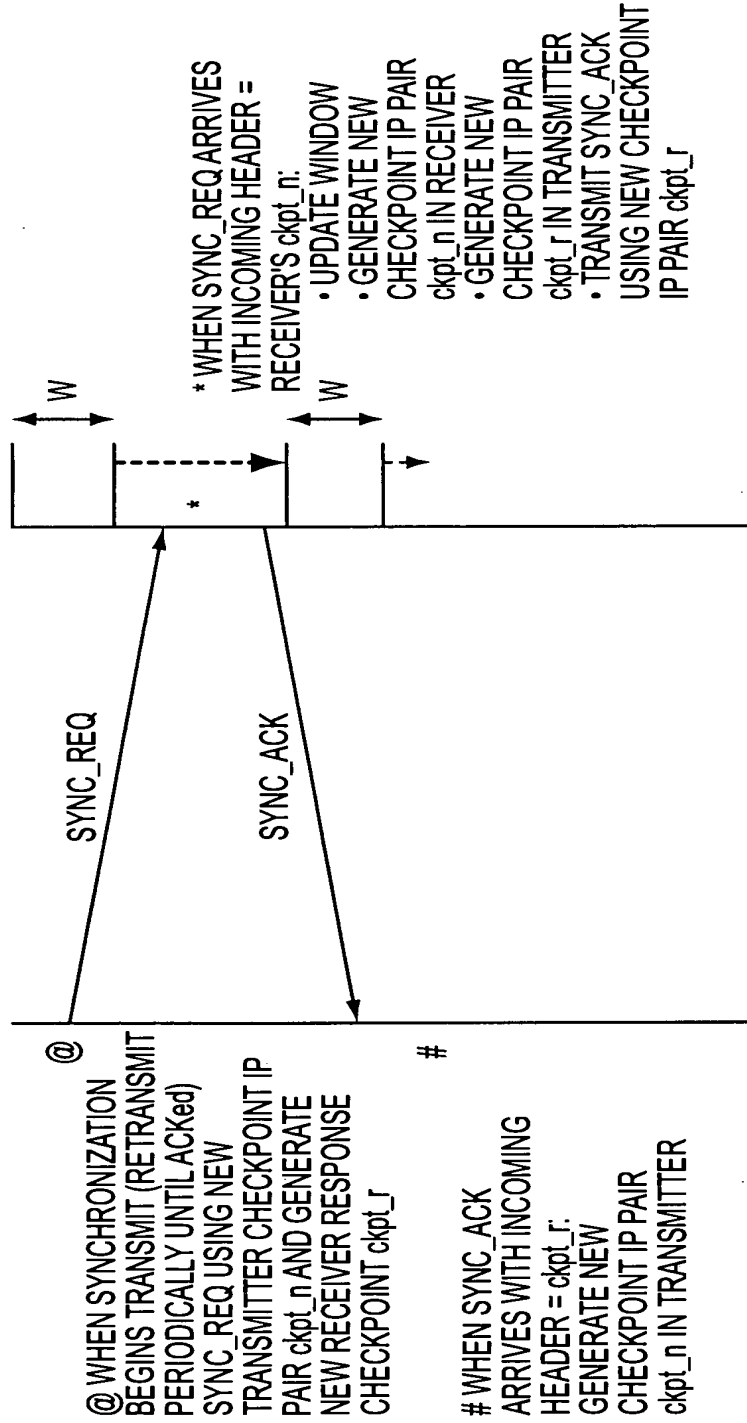


FIG. 15

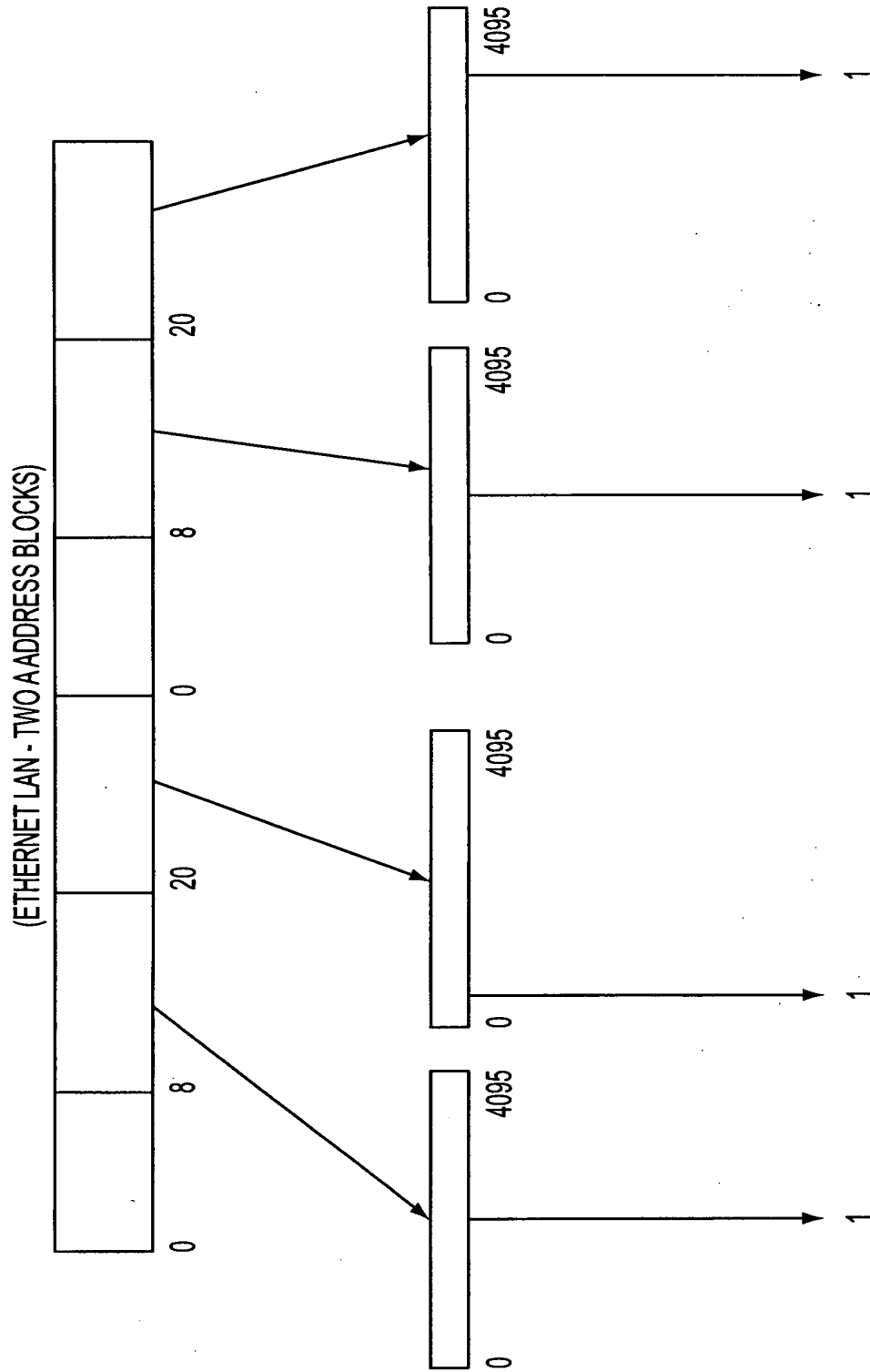


FIG. 16

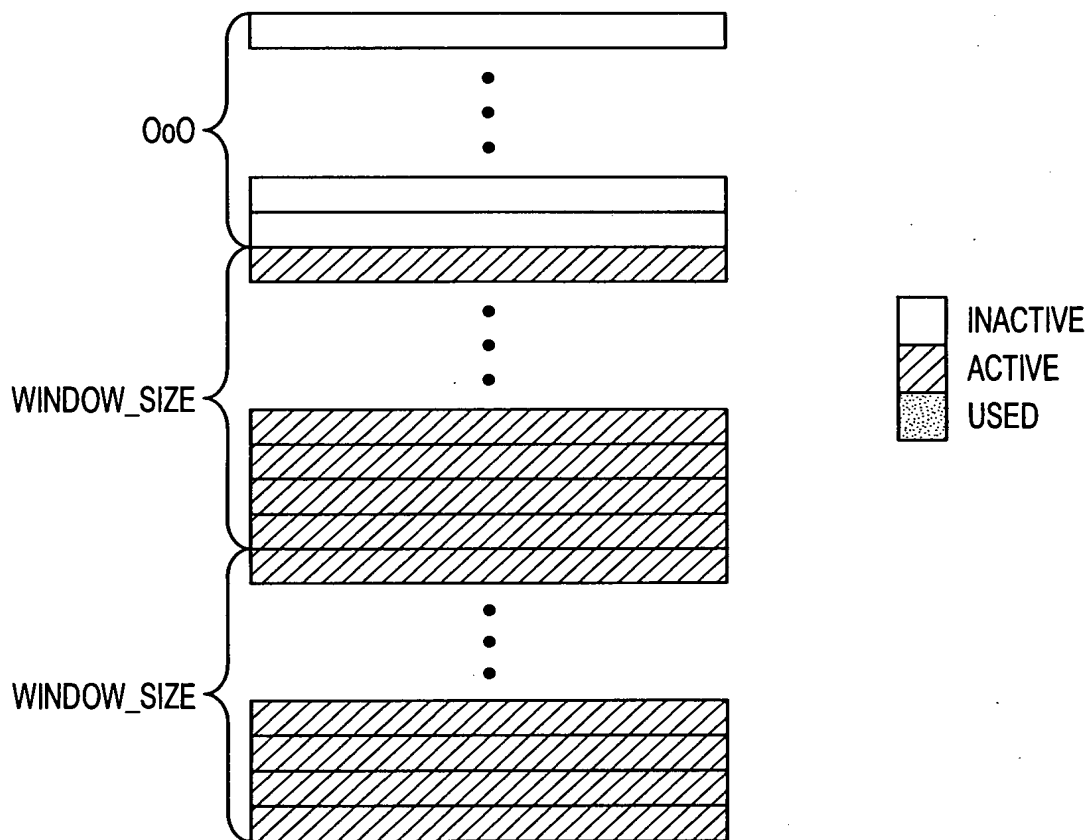


FIG. 17



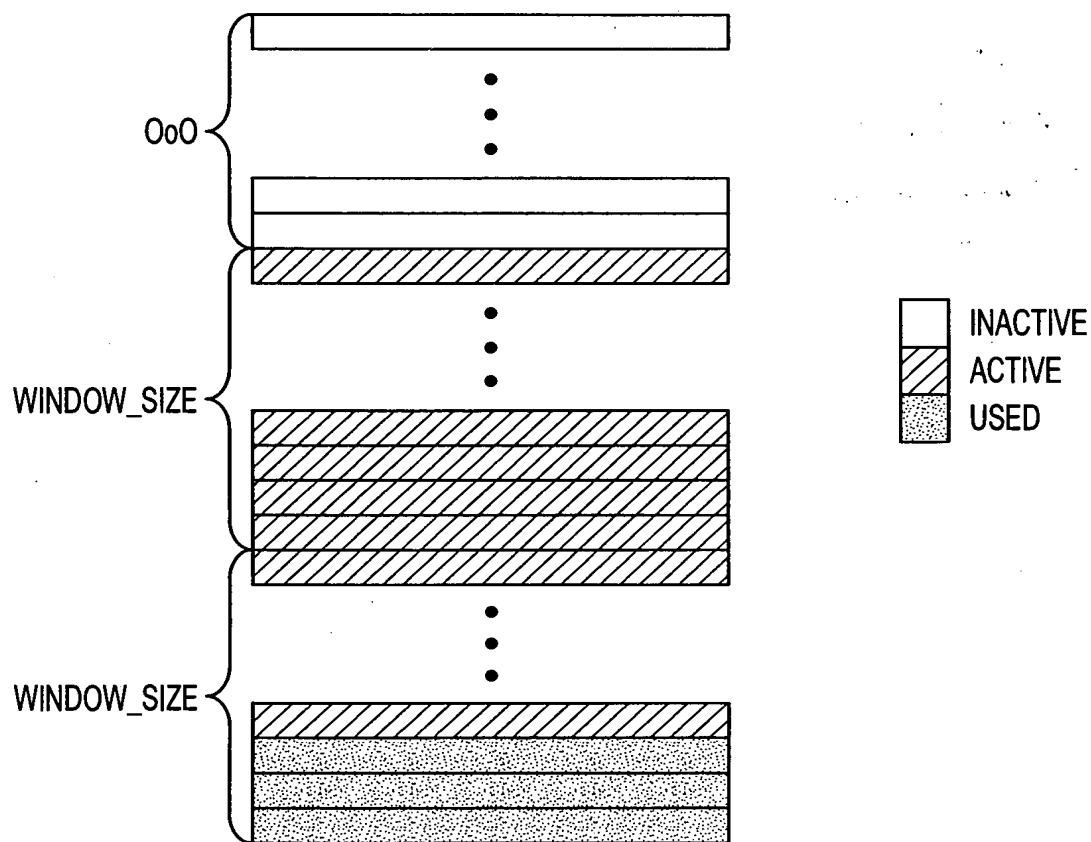


FIG. 18

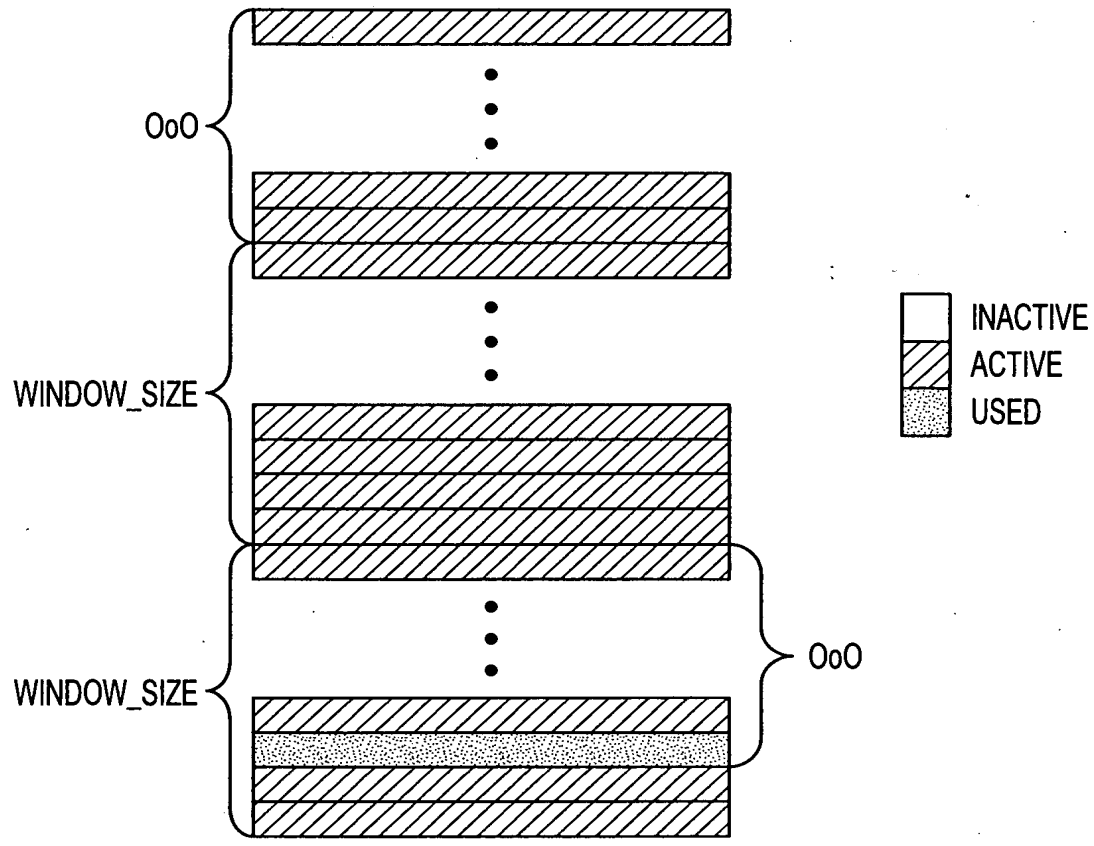


FIG. 19

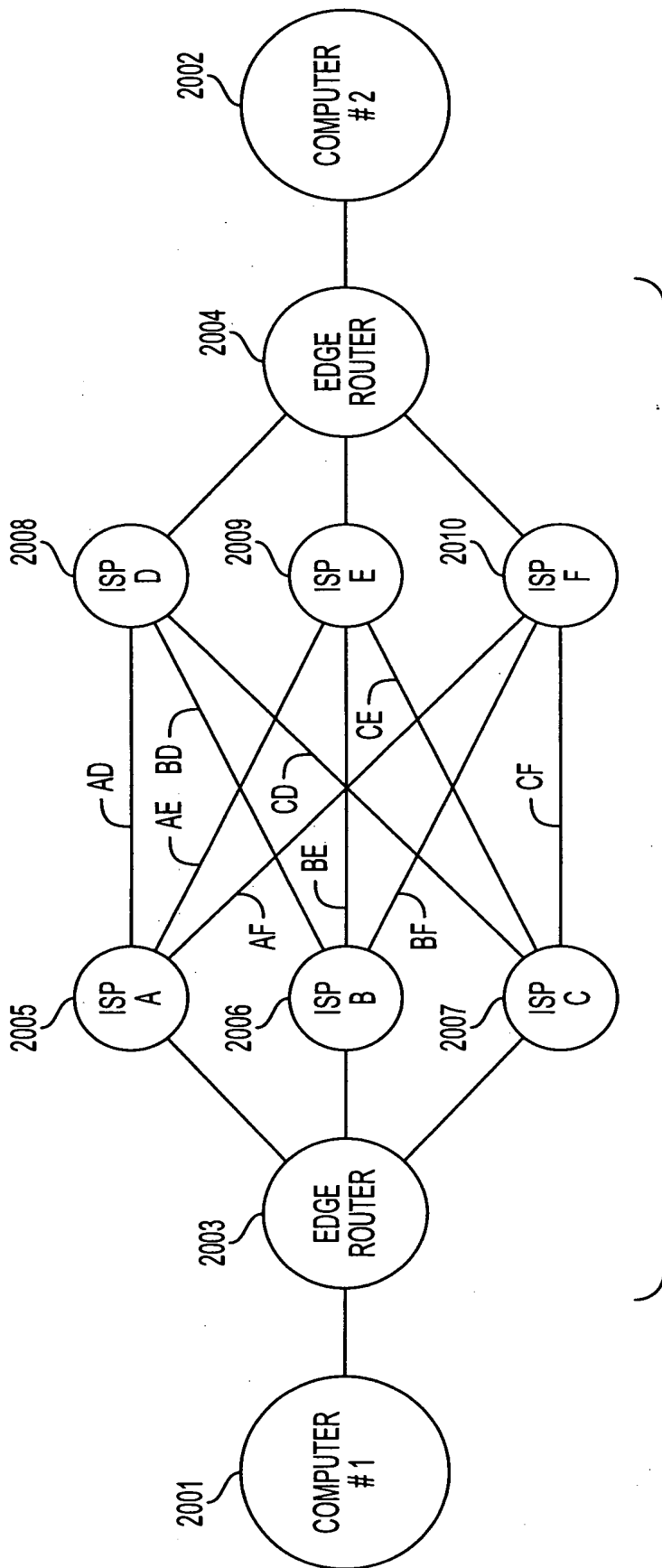


FIG. 20

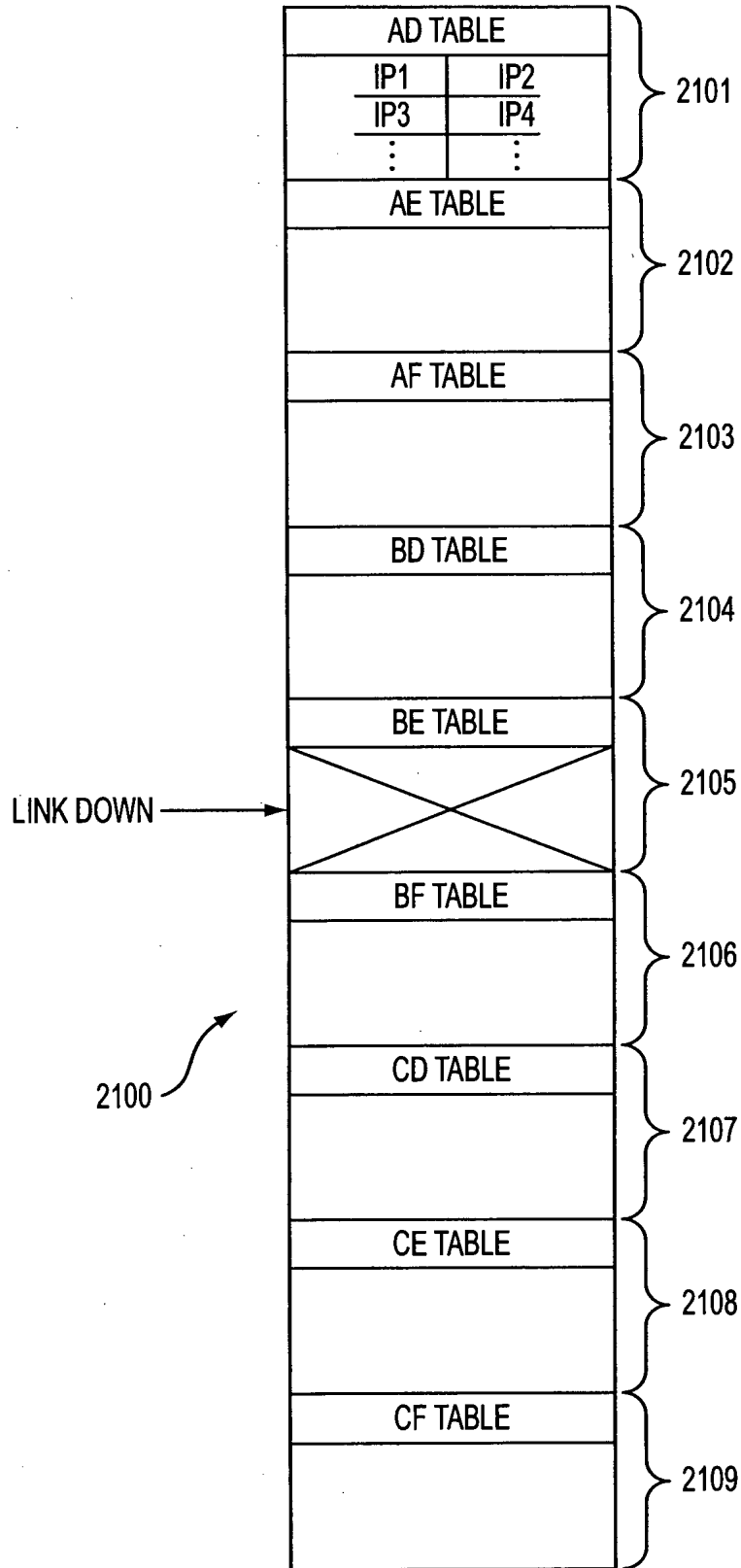


FIG. 21

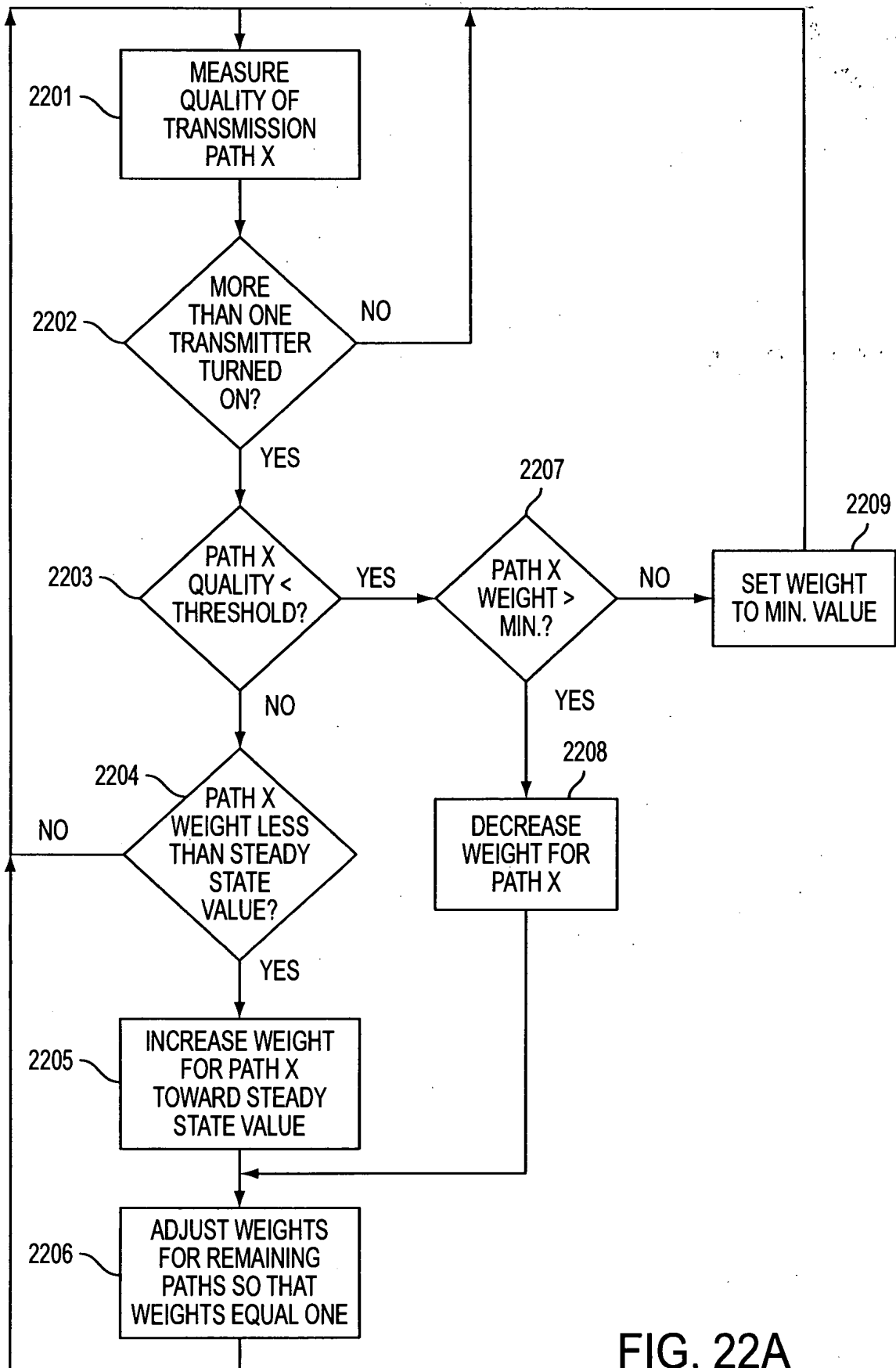


FIG. 22A

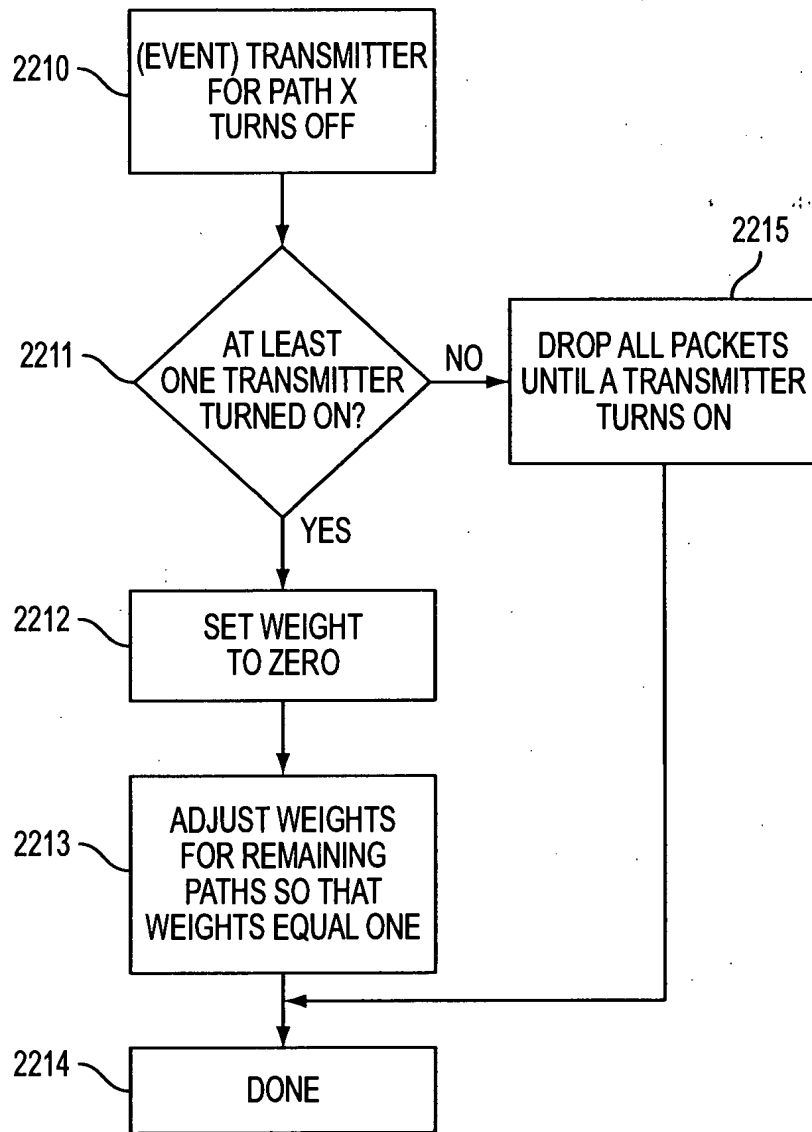


FIG. 22B

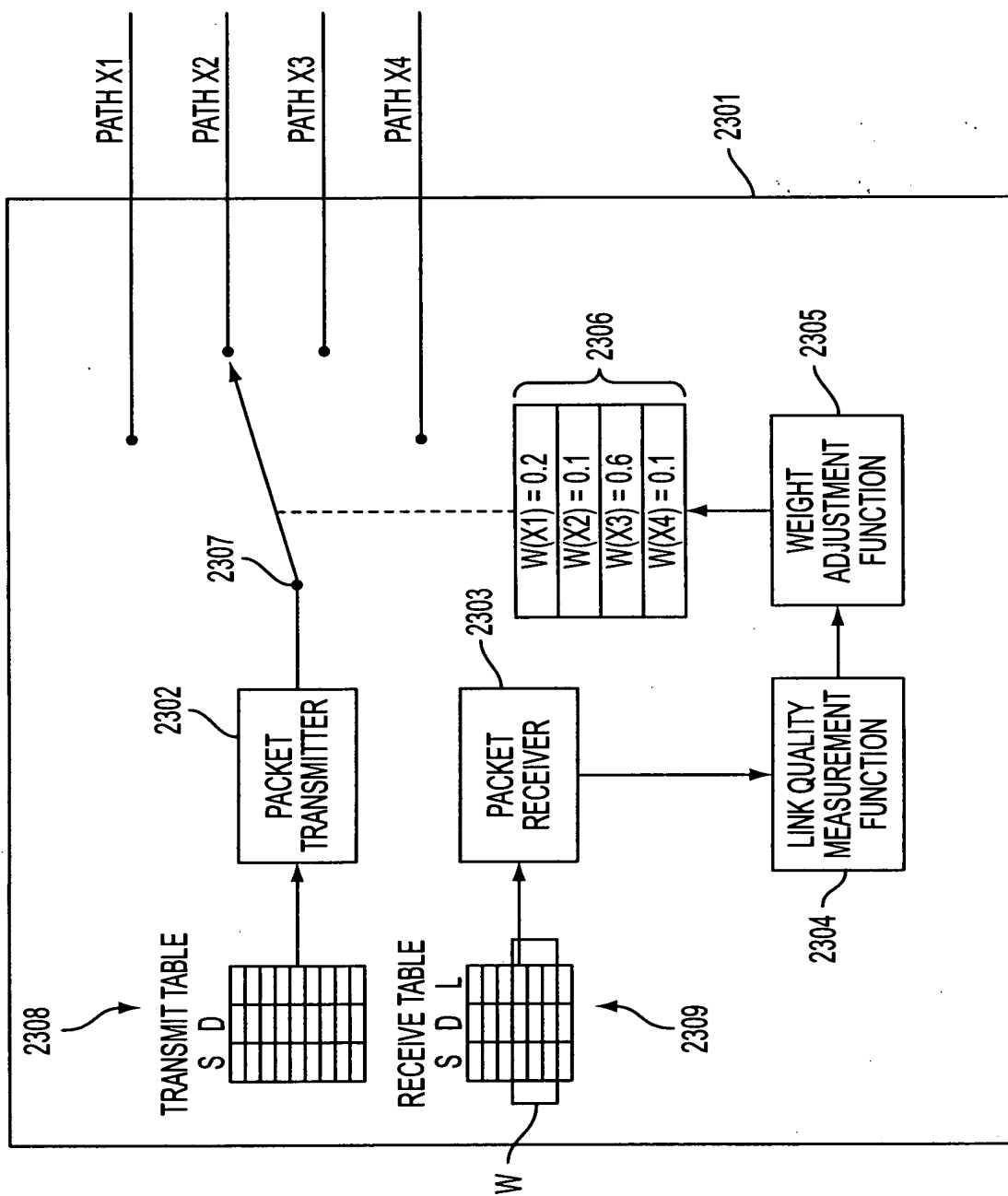


FIG. 23

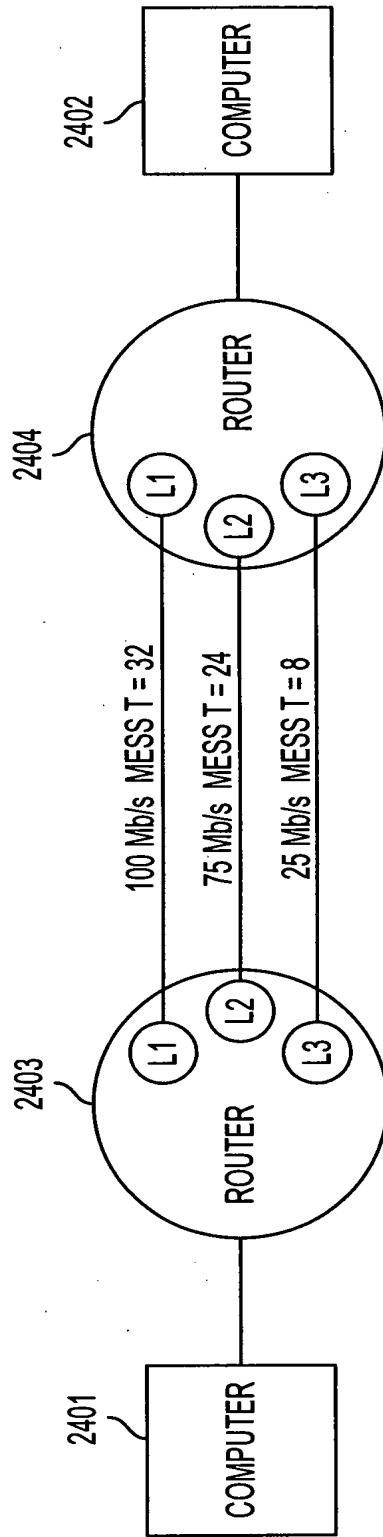


FIG. 24



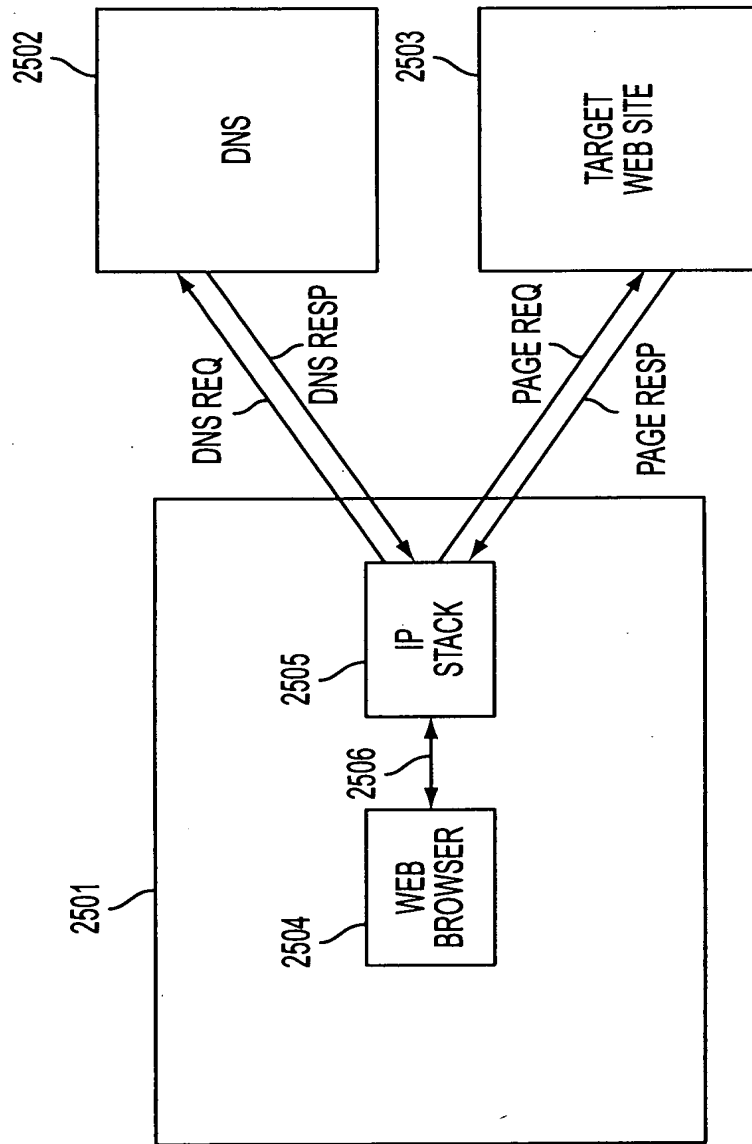


FIG. 25  
(PRIOR ART)

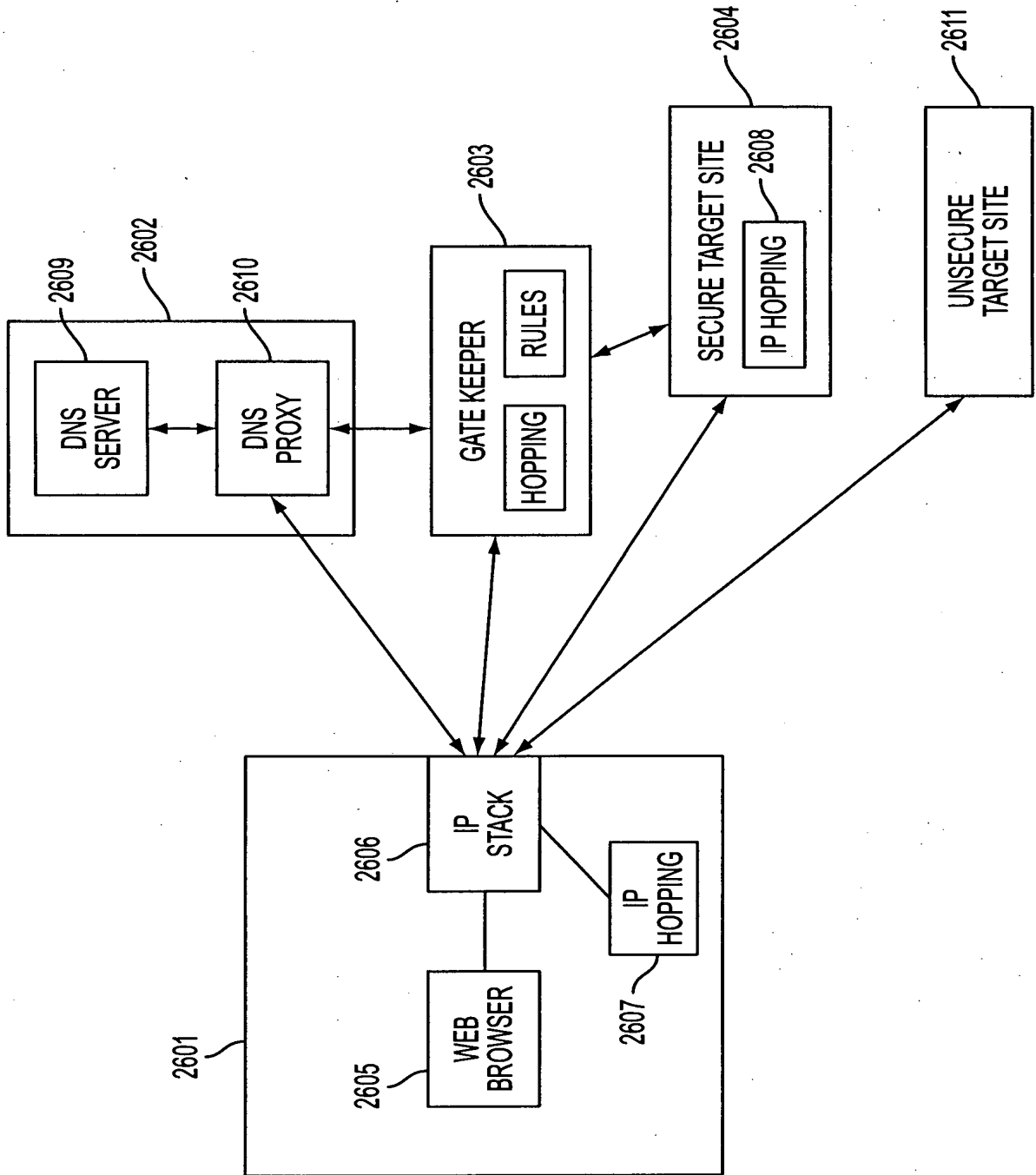


FIG. 26

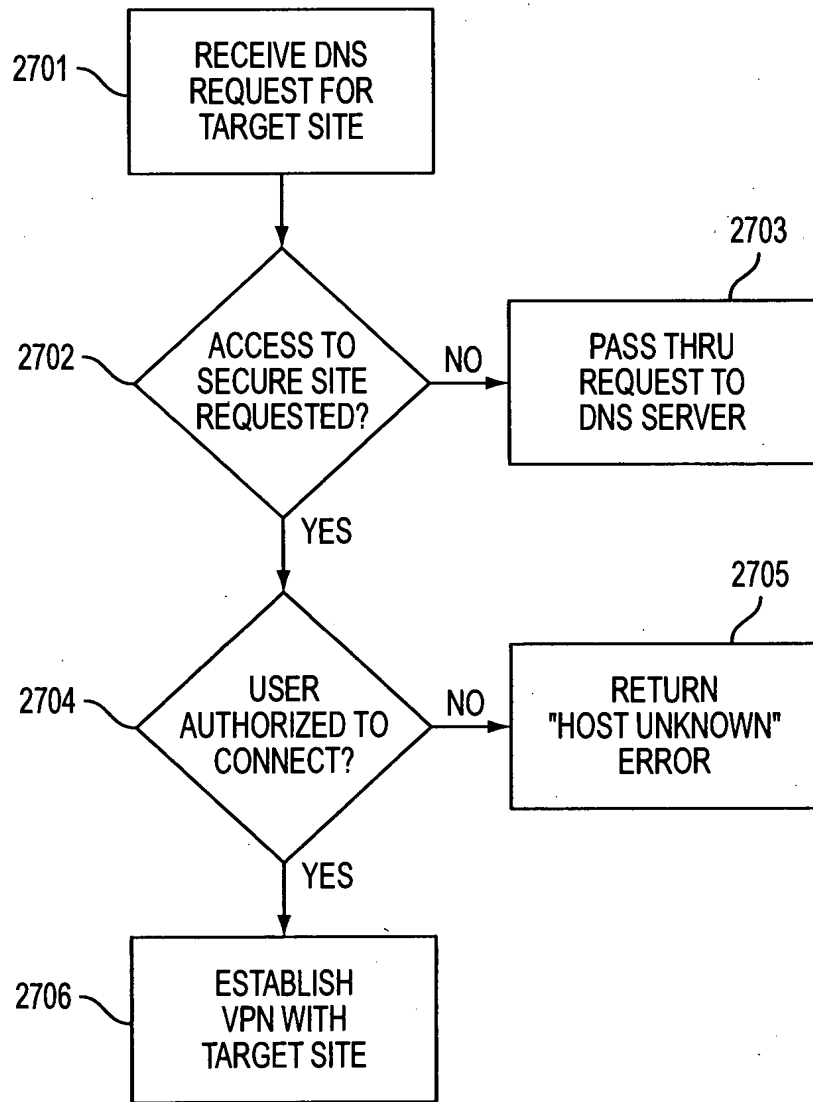


FIG. 27

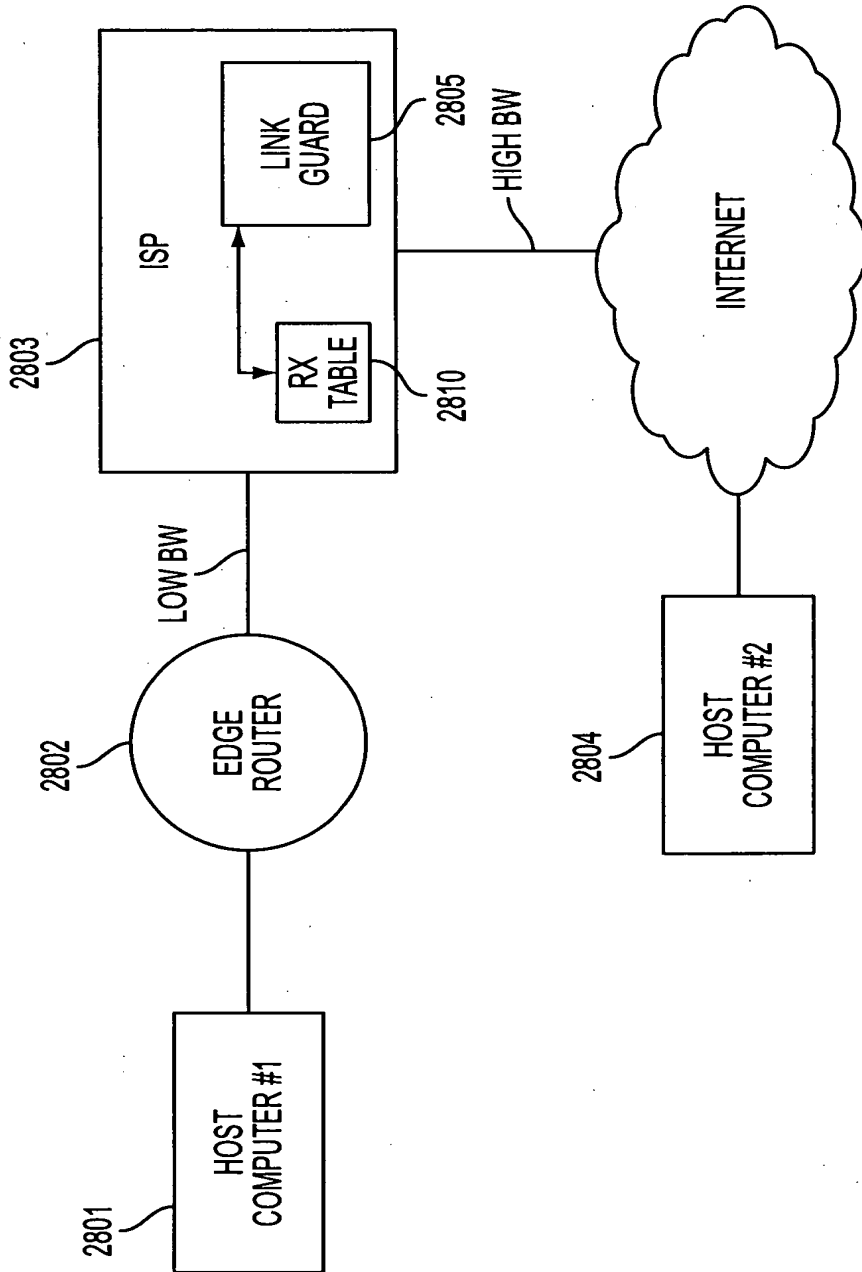


FIG. 28

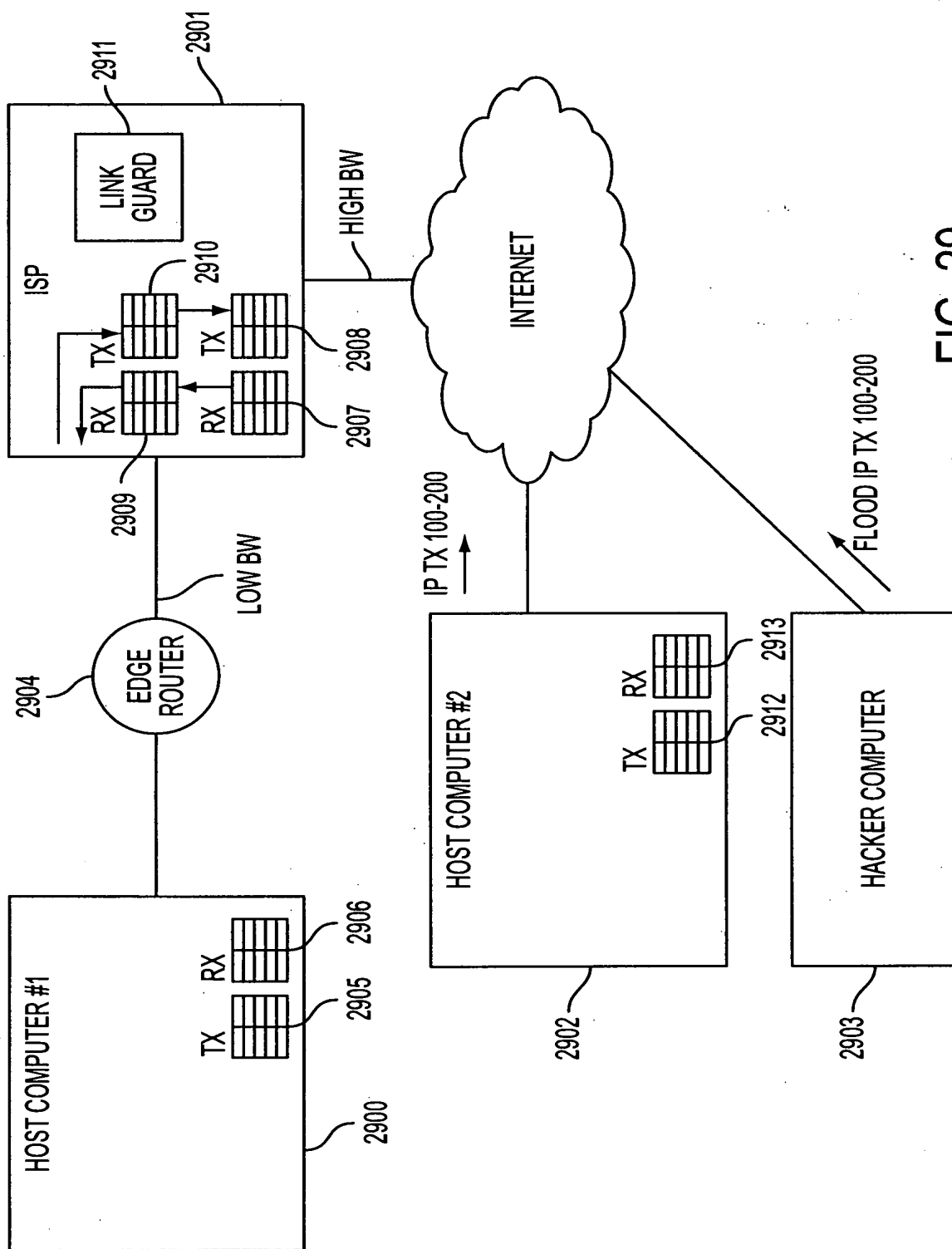


FIG. 29

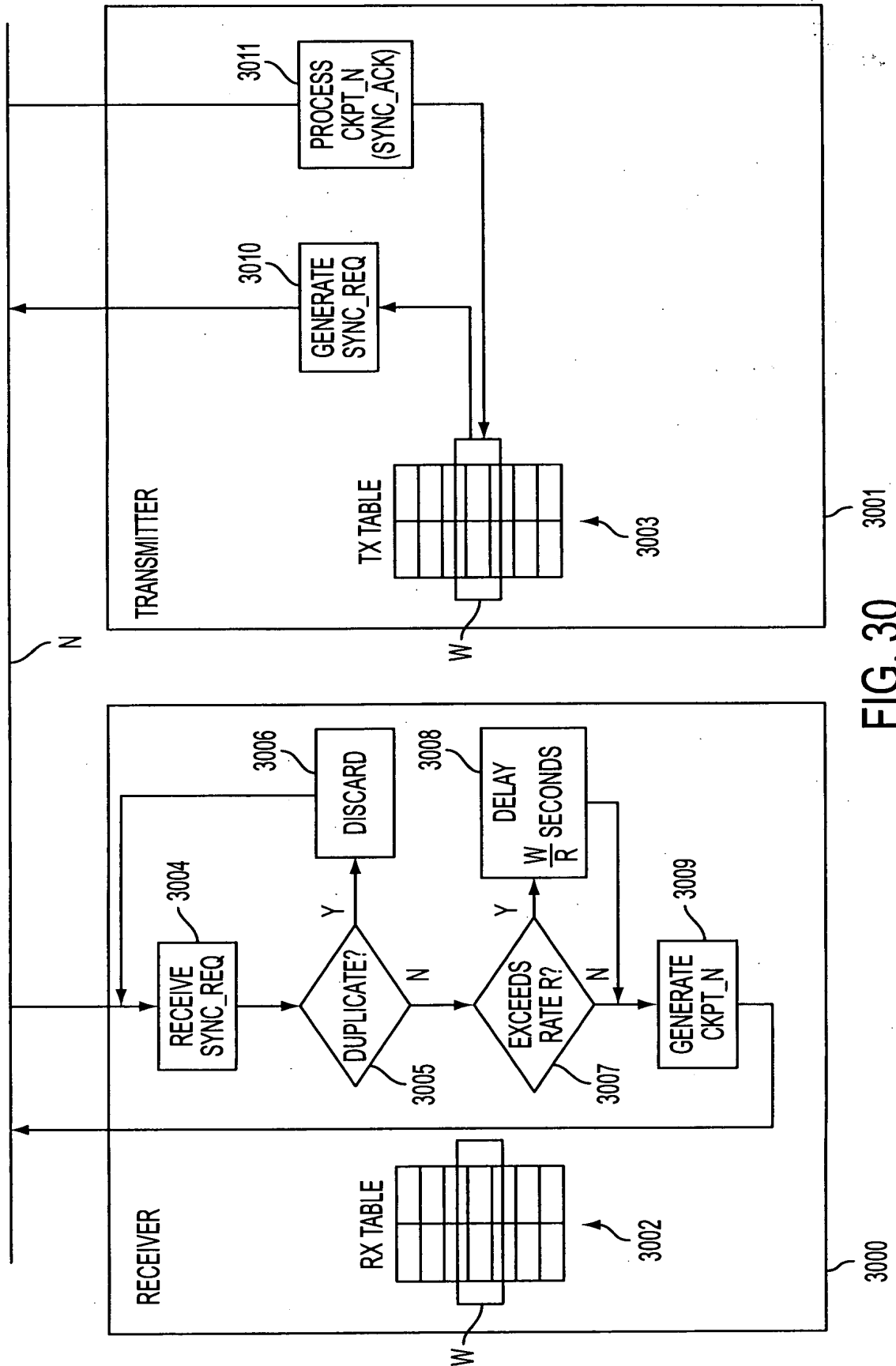


FIG. 30

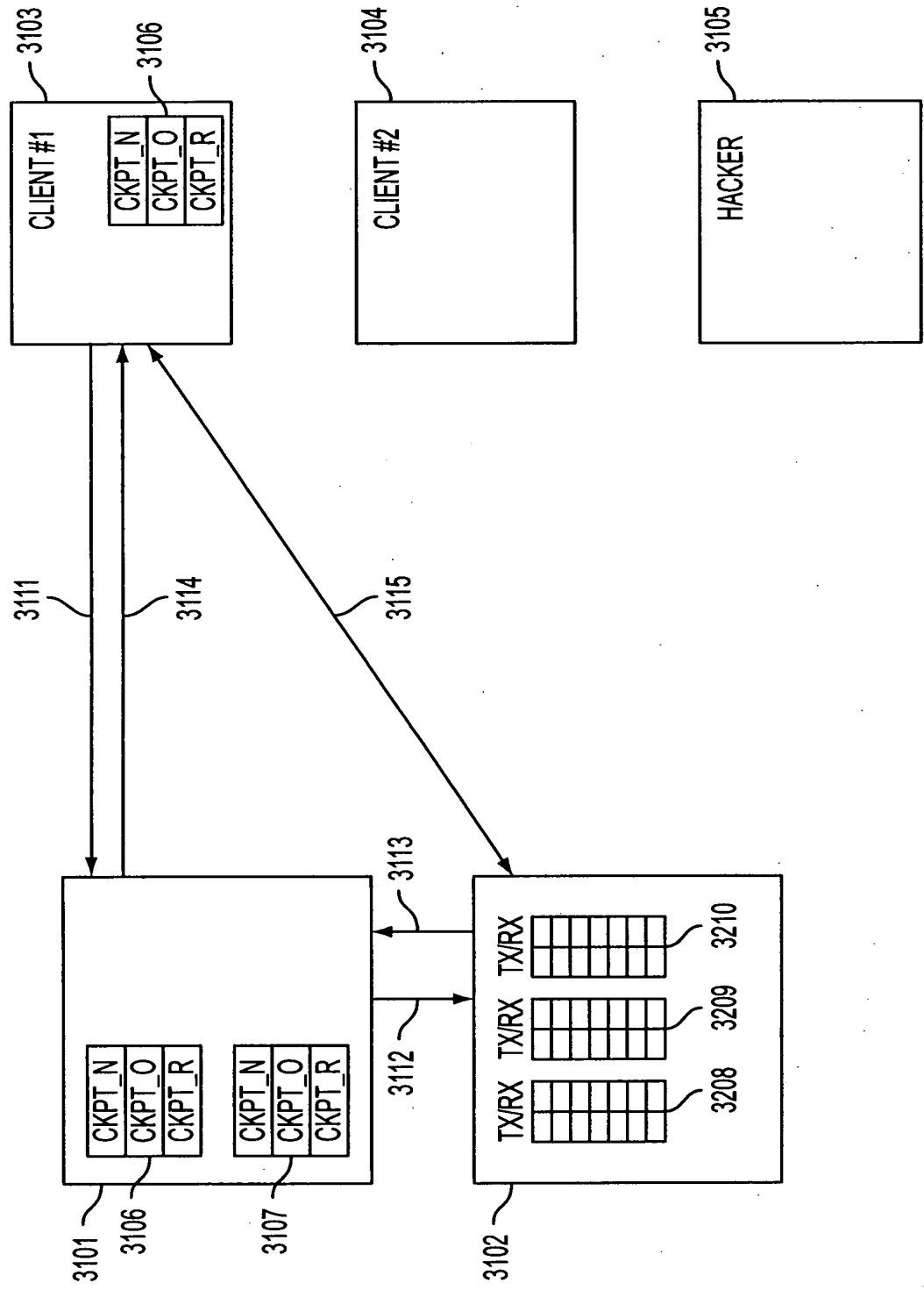


FIG. 31

CLIENT

SERVER

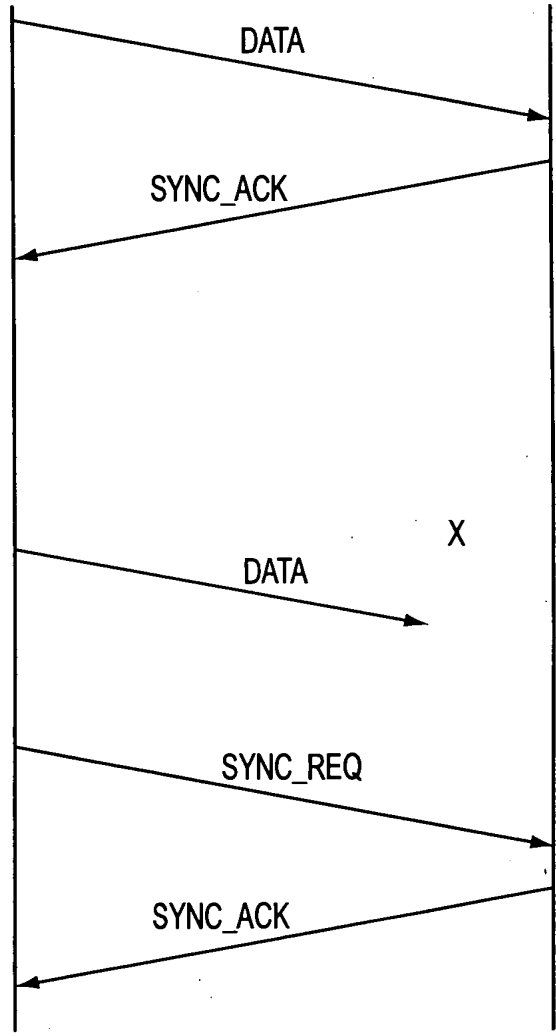
SEND DATA PACKET  
USING CKPT\_N  
CKPT\_O=CKPT\_N  
GENERATE NEW CKPT\_N  
START TIMER, SHUT  
TRANSMITTER OFF

IF CKPT\_O IN SYNC\_ACK  
MATCHES TRANSMITTER'S  
CKPT\_O  
UPDATE RECEIVER'S  
CKPT\_R  
KILL TIMER, TURN  
TRANSMITTER ON

SEND DATA PACKET  
USING CKPT\_N  
CKPT\_O=CKPT\_N  
GENERATE NEW CKPT\_N  
START TIMER, SHUT  
TRANSMITTER OFF

WHEN TIMER EXPIRES  
TRANSMIT SYNC\_REQ  
USING TRANSMITTER'S  
CKPT\_O, START TIMER

IF CKPT\_O IN SYNC\_ACK  
MATCHES TRANSMITTER'S  
CKPT\_O  
UPDATE RECEIVER'S  
CKPT\_R  
KILL TIMER, TURN  
TRANSMITTER ON



PASS DATA UP STACK  
CKPT\_O=CKPT\_N  
GENERATE NEW CKPT\_N  
GENERATE NEW CKPT\_R  
FOR TRANSMITTER SIDE  
TRANSMIT SYNC\_ACK  
CONTAINING CKPT\_O

CKPT\_O=CKPT\_N  
GENERATE NEW CKPT\_N  
GENERATE NEW CKPT\_R  
FOR TRANSMITTER SIDE  
TRANSMIT SYNC\_ACK  
CONTAINING CKPT\_O

FIG. 32



PATENT TRIAL AND ADVISORY BOARD  
 JUN 13 2002  
 JCS 896

2153

Please type a plus sign (+) inside this box →

PTO/SB/21 (08-00)  
 Approved for use through 10/31/2002. OMB 0651-0031

U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE  
 Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

<b>TRANSMITTAL FORM</b> <i>(to be used for all correspondence after initial filing)</i>	<b>Application Number</b>	09/504,783
	<b>Filing Date</b>	February 15, 2000
	<b>First Named Inventor</b>	Edmund Colby Munger
	<b>Group Art Unit</b>	2153
	<b>Examiner Name</b>	K. Lim
<b>Total Number of Pages in This Submission</b>	<b>Attorney Docket Number</b>	000479.85672

**RECEIVED**  
 JUN 24 2002  
 Technology Center 2100

ENCLOSURES (check all that apply)		
<input type="checkbox"/> Fee Transmittal Form <input type="checkbox"/> Fee Attached <input checked="" type="checkbox"/> Amendment / Response <input type="checkbox"/> After Final <input type="checkbox"/> Affidavits/declaration(s) <input type="checkbox"/> Extension of Time Request <input type="checkbox"/> Express Abandonment Request <input type="checkbox"/> Information Disclosure Statement <input type="checkbox"/> Certified Copy of Priority Document(s) <input type="checkbox"/> Response to Missing Parts/ Incomplete Application <input type="checkbox"/> Response to Missing Parts under 37 CFR 1.52 or 1.53	<input type="checkbox"/> Assignment Papers (for an Application) <input checked="" type="checkbox"/> Drawing(s) <input type="checkbox"/> Licensing-related Papers <input type="checkbox"/> Petition <input type="checkbox"/> Petition to Convert to a Provisional Application <input type="checkbox"/> Power of Attorney, Revocation Change of Correspondence Address <input type="checkbox"/> Terminal Disclaimer <input type="checkbox"/> Request for Refund <input type="checkbox"/> CD, Number of CD(s) _____	<input type="checkbox"/> After Allowance Communication to Group <input type="checkbox"/> Appeal Communication to Board of Appeals and Interferences <input type="checkbox"/> Appeal Communication to Group (Appeal Notice, Brief, Reply Brief) <input type="checkbox"/> Proprietary Information <input type="checkbox"/> Status Letter <input checked="" type="checkbox"/> Other Enclosure(s) (please identify below): <p style="text-align: center;"><b>Submission of Formal Drawings to Official Draftsman</b></p>
Remarks		

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT	
Firm or Individual name	Bradley C. Wright, Reg. No. 38,061
Signature	<b>Reg. No. 49,024</b>
Date	June 13, 2002

CERTIFICATE OF MAILING			
I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on this date: <input style="width: 150px;" type="text"/>			
Typed or printed name			
Signature		Date	

**Burden Hour Statement:** This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be send to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

**NOTICE OF ALLOWANCE AND FEE(S) DUE**

7590 07/03/2002

Banner & Witcoff, Ltd  
 1001 G Street, NW  
 Washington, DC 20001-4597

EXAMINER	
LIM, KRISNA	
ART UNIT	CLASS-SUBCLASS
2153	709-225000

DATE MAILED: 07/03/2002

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/504,783	02/15/2000	Edmund Colby Munger	00479.85672	8308

TITLE OF INVENTION: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

APPLN. TYPE	SMALL ENTITY	ISSUE FEE	PUBLICATION FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	\$1280	\$0	\$1280	10/03/2002

**THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE 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.**

**THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE REFLECTS A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE APPLIED IN THIS APPLICATION. THE PTOL-85B (OR AN EQUIVALENT) MUST BE RETURNED WITHIN THIS PERIOD EVEN IF NO FEE IS DUE OR THE APPLICATION WILL 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 changed, pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above and notify the United States Patent and Trademark Office of the change in status, or
- B. If the status is the same, pay the TOTAL FEE(S) DUE shown above.

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 now claiming SMALL ENTITY status, check the box below and enclose the PUBLICATION FEE and 1/2 the ISSUE FEE shown above.
  - Applicant claims SMALL ENTITY status. See 37 CFR 1.27.

II. PART B - FEE(S) TRANSMITTAL should be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). Even if the fee(s) have already been paid, Part B - Fee(s) Transmittal should be completed and returned. If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be 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 to Box ISSUE FEE unless advised to the contrary.

**IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.**

**PART B - FEE(S) TRANSMITTAL**

**Complete and send this form, together with applicable fee(s), to: Mail Box ISSUE FEE  
 Commissioner for Patents  
 Washington, D.C. 20231  
 Fax (703)746-4000**

**INSTRUCTIONS:** This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 4 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as 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" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Legibly mark-up with any corrections or use Block 1)  
 7590 07/03/2002

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

**Banner & Witcoff, Ltd**  
 1001 G Street, NW  
 Washington, DC 20001-4597

**Certificate of Mailing or Transmission**  
 I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Box Issue Fee address above, or being facsimile transmitted to the USPTO, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/504,783	02/15/2000	Edmund Colby Munger	00479.85672	8308

TITLE OF INVENTION: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

APPLN. TYPE	SMALL ENTITY	ISSUE FEE	PUBLICATION FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	\$1280	\$0	\$1280	10/03/2002

EXAMINER	ART UNIT	CLASS-SUBCLASS
LIM, KRISNA	2153	709-225000

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. Inclusion of assignee data is only appropriate when an assignment has been previously submitted to the USPTO or is being submitted under separate cover. 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  government

- 4a. The following fee(s) are enclosed:
- Issue Fee
- Publication Fee
- 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 Commissioner is hereby authorized by charge the required fee(s), or credit any overpayment, to Deposit Account Number \_\_\_\_\_ (enclose an extra copy of this form).

Commissioner for Patents 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.

(Authorized Signature) (Date)

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 party in interest as shown by the records of the United States Patent and Trademark Office.

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, Washington, D.C. 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, Washington, DC 20231.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMIT THIS FORM WITH FEE(S)



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

Table with columns: APPLICATION NO., FILING DATE, FIRST NAMED INVENTOR, ATTORNEY DOCKET NO., CONFIRMATION NO., EXAMINER, ART UNIT, PAPER NUMBER. Includes application number 09/504,783, inventor Edmund Colby Munger, and examiner LIM, KRISNA.

DATE MAILED: 07/03/2002

Determination of Patent Term Extension under 35 U.S.C. 154 (b)
(application filed after June 7, 1995 but prior to May 29, 2000)

The patent term extension is 0 days. Any patent to issue from the above identified application will include an indication of the 0 day extension on the front page.

If a continued prosecution application (CPA) was filed in the above-identified application, the filing date that determines patent term extension is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) system. (http://pair.uspto.gov)



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

Table with columns: APPLICATION NO., FILING DATE, FIRST NAMED INVENTOR, ATTORNEY DOCKET NO., CONFIRMATION NO.
Banner & Witcoff, Ltd
1001 G Street, NW
Washington, DC 20001-4597
UNITED STATES
EXAMINER: LIM, KRISNA
ART UNIT: 2153
PAPER NUMBER:
DATE MAILED: 07/03/2002

Notice of Fee Increase on October 1, 2002

If a reply to a "Notice of Allowance and Fee(s) Due" is filed in the Office on or after October 1, 2002, then the amount due may be higher than that set forth in the "Notice of Allowance and Fee(s) Due" since there will be an increase in fees effective on October 1, 2002. See Revision of Patent and Trademark Fees for Fiscal Year 2003; Notice of Proposed Rulemaking, 67 Fed. Reg. 30634, 30636 (May 7, 2002).

If the issue fee paid is the amount shown on the "Notice of Allowance and Fee(s) Due," but not the correct amount in view of the fee increase, a "Notice to Pay Balance of Issue Fee" will be mailed to applicant. In order to avoid processing delays associated with mailing of a "Notice to Pay Balance of Issue Fee," if the response to the Notice of Allowance and Fee(s) due form is to be filed on or after October 1, 2002 (or mailed with a certificate of mailing on or after October 1, 2002), the issue fee paid should be the fee that is required at the time the fee is paid.

Effective October 1, 2002, 37 CFR 1.18 is proposed to be revised to change the patent issue fees as set forth below. As stated above, the final fees may be a different amount, and applicant should check the web site given above when paying the fee.

(a) Issue fee for issuing each original or reissue patent, except a design or plant patent:

By a small entity (Sec. 1.27(a))--\$655.00
By other than a small entity--\$1,310.00

(b) Issue fee for issuing a design patent:

By a small entity (Sec. 1.27(a))--\$235.00
By other than a small entity--\$470.00

(c) Issue fee for issuing a plant patent:

By a small entity (Sec. 1.27(a))--\$315.00
By other than a small entity--\$630.00

Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at (703) 305-8283.

**Notice of Allowability**

Application No.	Applicant(s)	
09/504,783	MUNGER ET AL.	
Examiner	Art Unit	
Krisna Lim	2153	

-- 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 the amendment filed 6/13/02.
- 2.  The allowed claim(s) is/are 28-39 and 67-71.
- 3.  The drawings filed on 13 June 2002 are accepted by the Examiner.
- 4.  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: \_\_\_\_\_.
- 5.  Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).
  - (a)  The translation of the foreign language provisional application has been received.
- 6.  Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

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**

- 7.  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.
- 8.  CORRECTED DRAWINGS 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. \_\_\_\_\_.
  - (b)  including changes required by the proposed drawing correction filed \_\_\_\_\_, which has been approved by the Examiner.
  - (c)  including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No. \_\_\_\_\_.

Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the top margin (not the back) of each sheet. The drawings should be filed as a separate paper with a transmittal letter addressed to the Official Draftsperson.

- 9.  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 Informal Patent Application (PTO-152)
- 3  Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 4  Interview Summary (PTO-413), Paper No. \_\_\_\_\_
- 5  Information Disclosure Statements (PTO-1449), Paper No. 10
- 6  Examiner's Amendment/Comment
- 7  Examiner's Comment Regarding Requirement for Deposit of Biological Material
- 8  Examiner's Statement of Reasons for Allowance
- 9  Other Office Action.

Krisna Lim  
Primary Examiner  
Art Unit: 2153

Art Unit: 2153

1. Applicant's election without traverse of Group II (claims 28-39 and 67-71) in Paper No. 5 (filed 1/28/02) is acknowledged. Claims 82-91 were submitted on 2/22/02; however, Examiner had not received it until 6/13/02. Thus, the previous office action, mailed 3/13/02, did not reflect to the additional claims 82-91.

2. Thus, claims 28-39, 67-71 and 82-91 are pending for examination, and claims 1-27, 40-66 and 72-81 were canceled.

3. Restriction to one of the following inventions is required under 35 U.S.C. § 121:

II. Claims 28-39 and 67-71, drawn to a system for transparently creating a virtual private network (VPN) between a client computer and a target computer, comprising: a ) generating from the client computer a DNS request ..., b) determining whether the DNS ..., c) determining that the DNS ..., classified in Class 709, subclass 249.

IV. Claims 82-91 (similarly to canceled claims 72-81), drawn to a method for establishing an encrypted channel between a client and a secure host, comprising the step of automatically creating the encrypted channel upon intercepting a DNS request

Art Unit: 2153

for a domain name comprising a predetermined domain name extension, classified in Class 713, subclass 201.

4. Inventions II and V are related as subcombinations disclosed as usable together in a single combination. The subcombinations are distinct from each other if they are shown to be separately usable. In the instant case, invention II has separate utility such as a method of registering a node that does not support Mobile IP with a Home Agent that support Mobile IP lacks the step of automatically creating the encrypted channel upon intercepting a DNS request for a domain name comprising a predetermined domain name extension.

5. These inventions are distinct for the reasons given above, and the search required for each Group is different and not co-extensive for examination purpose.

6. For example, the searches for the four inventions would not be co-extensive because these groups would require different searches on PTO's classification class and subclass as following:

1) The Group II search (claims 28-39 and 67-71) would require use of search class 709, subclass 249 ( which would not required for the group IV).



Art Unit: 2153

2) The Group IV search (claims 82-91 (similarly to canceled claims 72-81)) would require use of search class 713, subclass 201 ( which would not required for the group II).

7. During a telephone conversation with Mr. Bradley C. Wright on June 28, 2002, a provisional election was made without traverse to prosecute the invention of Group II, claims 28-39 and 67-71. Affirmation of this election must be made by applicant in replying to this Office action. Claims 82-91 have been withdrawn from further consideration by the examiner, 37 CFR 1.142(b), as being drawn to a non-elected invention.

8. **Examiner's Amendment**

An Examiner's Amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 C.F.R. § 1.312. To ensure consideration of such an amendment, it **MUST** be submitted no later than the payment of the Issue Fee.

9. Authorization for this examiner's amendment was given in a telephone interview with Mr. Bradley C. Wright on June 28, 2002.

10.  **Cancel claims 82-91.**

Art Unit: 2153

11. Claims 28-39 and 67-71 are allowable.

12. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Examiner Krisna Lim whose telephone number is (703) 305-9672. The examiner can normally be reached on Monday-Friday from 9:00 to 5:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Mr. Glen Burgess, can be reached at (703) 305-4772.. The fax phone numbers for the organization where this application or proceeding is assigned is are as following:

(703) 746-7238 [After Final Communication]

or

(703) 746-7239 [Official Communication]

(703) 746-7240 [For Status inquires, draft communication]

and/or

(703) 306-5631, (703) 306-5632 or (703) 306-5633 for [Customer Service Numbers]

Any inquiry of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is (703) 305-3900.

Communications via Internet e-mail regarding this application, other than those under 35 U.S.C. 132 or which otherwise require a signature, may be used by the applicant and should be addressed to [glen.burgess@uspto.gov].

All Internet e-mail communication will be made of record in the application file. PTO employees do not engage in Internet communications where there exists a

Art Unit: 2153

possibility that sensitive information could be identified or exchanged unless the record includes a properly signed express waiver of the confidentiality requirement of 35 U.S.C. 122. This is more clearly set forth in the Interim Internet Usage Policy published in the Office Gazette of the Patent and Trademark on February 25, 1997 at 1195 OG 89.

kl

July 1, 2002



KRISNA LIM  
PRIMARY EXAMINER



USPTO Form 1449 U.S. Department of Commerce Patent and Trademark Office <b>INFORMATION DISCLOSURE</b> <b>CITATION</b> Sheet 1 of 1	Attorney Docket No. 00479.85672	Serial No. 09/504,783
	Applicant(s): Edmund Colby Munger et al.	
	Filing Date: February 15, 2000	Group: 2153

U.S. PATENT DOCUMENTS						
Examiner Initial	Patent No.	Date	Name	Class	Subclass	Filing Date (if appropriate)
<i>K</i>	6,243,749	6/5/01	Sitaraman et al.	<del>X</del>	<del>X</del>	
<i>K</i>	6,119,171	9/12/00	Alkhatib			
<i>K</i>	6,052,788	4/18/00	Wesinger, Jr. et al.			
<i>K</i>	6,006,259	12/21/99	Adelman et al.			
<i>K</i>	5,905,859	5/18/99	Holloway et al.			
<i>K</i>	5,898,830	4/27/99	Wesinger, Jr. et al.			
<i>K</i>	5,892,903	4/6/99	Klaus			
<i>K</i>	5,805,801	9/8/98	Holloway et al.			
<i>K</i>	5,796,942	8/18/98	Esbensen			

FOREIGN PATENT DOCUMENTS							
Examiner Initial	Document No.	Date	Country	Class	Subclass	Translation	
						YES	NO
<i>K</i>	WO 00/70458	11/23/00	PCT	<del>X</del>	<del>X</del>		

OTHER DOCUMENTS (including Author, Title, Date, Pertinent Pages, etc.)	
<i>K</i>	Linux FreeS/WAN Index File, printed from <a href="http://liberty.freeswan.org/freeswan_trees/freeswan-1.3/doc/">http://liberty.freeswan.org/freeswan_trees/freeswan-1.3/doc/</a> on February 21, 2002, 3 Pages
<i>K</i>	J. Gilmore, "Swan: Securing the Internet against Wiretapping", printed from <a href="http://liberty.freeswan.org/freeswan_trees/freeswan-1.3/doc/rationale.html">http://liberty.freeswan.org/freeswan_trees/freeswan-1.3/doc/rationale.html</a> on February 21, 2002, 4 pages
<i>K</i>	Glossary for the Linux FreeS/WAN project, printed from <a href="http://liberty.freeswan.org/freeswan_trees/freeswan-1.3/doc/glossary.html">http://liberty.freeswan.org/freeswan_trees/freeswan-1.3/doc/glossary.html</a> on February 21, 2002, 25 pages
<i>K</i>	Alan O. Frier et al., "The SSL Protocol Version 3.0", November 18, 1996, printed from <a href="http://www.netscape.com/eng/ss13/draft302.txt">http://www.netscape.com/eng/ss13/draft302.txt</a> on February 4, 2002, 56 pages

EXAMINER <i>KRISNA LIM</i>	DATE CONSIDERED <i>6/28/02</i>
----------------------------	--------------------------------

\*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to Applicant.

JMP/BCW/INTL

PATENT COOPERATION TREATY

00479.00  
DOCKETE

From the INTERNATIONAL SEARCHING AUTHORITY

PCT

JUL 08 2002  
Fee due  
8/12

To:  
BANNER & WITCOFF, LTD.  
Attn. Curtin, Joseph W.  
1001 G Street, N.W.  
Eleventh Floor  
Washington, DC 20001-4597 K9  
UNITED STATES OF AMERICA

RECEIVED

JUL 08 2002

BANNER & WITCOFF, LTD.

INVITATION TO PAY ADDITIONAL FEES  
(PCT Article 17(3)(a) and Rule 40.1)

Applicant's or agent's file reference 00479.00027		Date of mailing (day/month/year) 28/06/2002
International application No. PCT/US 01/13261	PAYMENT DUE within 45 <del>XXXX</del> days/days from the above date of mailing	
International filing date (day/month/year) 25/04/2001	Applicant  SCIENCE APPLICATIONS INTERNATIONAL CORPORATION	

1. This International Searching Authority

(i) considers that there are 3 (number of) inventions claimed in the international application covered by the claims indicated ~~XXXX~~ on the extra sheet:

and it considers that the international application does not comply with the requirements of unity of invention (Rules 13.1, 13.2 and 13.3) for the reasons indicated ~~XXXX~~ on the extra sheet:

(ii)  has carried out a partial international search (see Annex)  will establish the international search report on those parts of the international application which relate to the invention first mentioned in claims Nos.:  
1-4, 8-14, 16-19, 23-29, 53-59, 63-70, 74-81, 85-92, 96-102, 106-112

(iii) will establish the international search report on the other parts of the international application only if, and to the extent to which, additional fees are paid

2. The applicant is hereby invited, within the time limit indicated above, to pay the amount indicated below:

$$\frac{\text{EUR } 945,00}{\text{Fee per additional invention}} \times \frac{2}{\text{number of additional inventions}} = \frac{\text{EUR } 1.890,00}{\text{total amount of additional fees}}$$

Or, \_\_\_\_\_ x \_\_\_\_\_ = \_\_\_\_\_

The applicant is informed that, according to Rule 40.2(c), the payment of any additional fee may be made under protest, i.e., a reasoned statement to the effect that the international application complies with the requirement of unity of invention or that the amount of the required additional fee is excessive.

3.  Claim(s) Nos. \_\_\_\_\_ have been found to be unsearchable under Article 17(2)(b) because of defects under Article 17(2)(a) and therefore have not been included with any invention.

Name and mailing address of the International Searching Authority  
European Patent Office, P.B. 5818 Patentlaan 2  
NL-2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
Fax: (+31-70) 340-3016

Authorized officer  
Claude Berthon

1. The present communication is an Annex to the invitation to pay additional fees (Form PCT/ISA/206). It shows the results of the international search established on the parts of the international application which relate to the invention first mentioned in claims Nos.:  
 1-4, 8-14, 16-19, 23-29, 53-59, 63-70, 74-81, 85-92, 96-102, 106-112, 116
2. This communication is not the international search report which will be established according to Article 18 and Rule 43.
3. If the applicant does not pay any additional search fees, the information appearing in this communication will be considered as the result of the international search and will be included as such in the international search report.
4. If the applicant pays additional fees, the international search report will contain both the information appearing in this communication and the results of the international search on other parts of the international application for which such fees will have been paid.

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 838 930 A (DIGITAL EQUIPMENT CORP) 29 April 1998 (1998-04-29)	1, 8, 16, 23, 53, 54, 56-58, 63-65, 67-69, 74-76, 78-80, 85-87, 89-91, 96-98, 100, 101, 106-108, 110, 111, 116
Y	abstract  column 3, line 30 -column 4, line 14 column 16, line 12 -column 17, line 14 column 22, line 56 -column 23, line 20 figures 3, 4, 9, 11-13, 21, 22  ----- -/--	2, 4, 9-14, 17, 18, 24-29, 55, 59, 66, 70, 77, 81, 88, 92, 99, 102, 109, 112

Further documents are listed in the continuation of box C.

Patent family members are listed in 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 but 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

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	GB 2 317 792 A (SECURE COMPUTING CORP) 1 April 1998 (1998-04-01)  abstract page 2, line 3 - line 25 page 3, line 1 - line 10 page 8, line 18 - line 24 page 10, line 20 -page 12, line 2 page 12, line 26 - line 31 page 13, line 16 - line 29 figures 1-4  ---	1,4,8, 14,16, 18,23, 29, 53-58, 63-69, 74-80, 85-91, 96-101, 106-111, 116
Y	EP 0 814 589 A (AT & T CORP) 29 December 1997 (1997-12-29) page 1, line 45 -page 2, line 2 page 5, line 8 - line 14  ---	2,17
X	US 5 588 060 A (AZIZ ASHAR) 24 December 1996 (1996-12-24)  ---	1,3,16, 19
Y	abstract  column 4, line 38 - line 46 column 6, line 50 - line 67 figure 2  ---	59,70, 81,92, 102,112
Y	US 5 689 566 A (NGUYEN MINH TAM C) 18 November 1997 (1997-11-18)  abstract column 9, line 10 -column 10, line 4  ---	4,9,10, 14,18, 24,25, 29,55, 66,77, 88,99, 109
Y	WO 98 27783 A (NORTHERN TELECOM LTD ;HOLMES KIM (US); HUI MARGARET (US); TELLO AN) 25 June 1998 (1998-06-25) abstract figure 3  --- -/--	11,26

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	<p>LAURIE WELLS (LANCASTERB1B@EMAIL.MSN.COM):                      "Subject: Security Icon "                      USENET NEWSGROUP, 'Online!                      19 October 1998 (1998-10-19), XP002200606                      microsoft.public.inetexplorer.ie4.security                      Retrieved from the Internet:                      &lt;URL:http://groups.google.com/&gt;                      'retrieved on 2002-05-30!                      the whole document</p>	<p>12,13,                      27,28</p>
A	<p>STALLINGS W: "CRYPTOGRAPHY AND NETWORK                      SECURITY, PRINCIPLES AND PRACTICE, 2ND                      EDITION"                      CRYPTOGRAPHY AND NETWORK SECURITY, XX, XX,                      8 June 1998 (1998-06-08), pages 399-440,                      XP002167283</p> <p>13.4 Encapsulating security payload                      13.5 Combining security associations</p>	<p>53-59,                      63-70,                      74-81,                      85-92,                      96-102,                      106-112,                      116</p>
L	<p>WILLIAM STALLINGS (WS@SHORE.NET):                      "Subject: new cryptography and network                      security book "                      USENET NEWSGROUP, 'Online!                      8 June 1998 (1998-06-08), XP002200607                      comp.security.misc                      Retrieved from the Internet:                      &lt;URL:http://groups.google.com/&gt;                      'retrieved on 2002-05-30!                      Proof of publication date of XP002167283</p>	



This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. Claims: 1-4 8-14 16-19 23-29 53-59 63-70 74-81 85-92  
96-102 106-112 116

A method and computer readable medium for loading a secure communication software module

2. Claims: 5-7 (as dependent from 1) 20-22 (as dependent from 16) 60-62 (as dependent from 53) 71-73 (as dependent from 64) 82-84 (as dependent from 75) 93-95 (as dependent from 86) 103-105 (as dependent from 97) 113-115 (as dependent from 107)

A method and computer readable medium based on a computer address hopping regime

3. Claims: 15 (as dependent from 1),  
30 (as dependent from 16), 31-52

A method and computer readable medium for sending a query for a secure network address to a secure domain name server

For the following reasoning document EP838930 is taken into account. From this prior art document what is known (see fig. 3) is a system and a method for establishing a secure virtual private network link over a computer network (as defined in claim 1).

With the reference to the prior art document, the first group yields the potential special technical feature of locally storing a secure communication software module when it is not already locally available (as defined in claim 2), hence solving the object problem manually configuring every computer for performing secure communications.

With the reference to the prior art document, the second group yields the potential special technical feature of establishing a connection based on a computer address hopping regime (as defined in claim 5), hence solving the object problem of providing security against eavesdropping on communications over the Internet.

With the reference to the prior art document, the third group yields the potential special technical feature of sending a query for a secure network address to a secure domain name server (as defined in claim 31), hence solving the object problem hiding the true IP address of web sites.

Consequently, neither the objective problems underlying the subjects of the three claimed inventions, nor the solutions as defined by the special technical features described allow for the link of a common inventive concept to be established between said inventions. In

INVITATION TO PAY ADDITIONAL FEES

International application No.

PCT/US 01/13261

conclusion therefore the three groups of claims are not linked by a single general inventive concept. The application hence does not meet the requirements of unity of invention as defined in Rule 13 (1) and (2) of the PCT.

**Patent Family Annex**

Information on patent family members

International Application No

PCT/US 01/13261

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
EP 0838930	A	29-04-1998	US 6101543 A EP 0838930 A2 JP 10178450 A	08-08-2000 29-04-1998 30-06-1998
GB 2317792	A	01-04-1998	US 5983350 A US 5950195 A DE 19741239 A1 DE 19741246 A1 GB 2317539 A ,B	09-11-1999 07-09-1999 07-05-1998 19-03-1998 25-03-1998
EP 0814589	A	29-12-1997	US 6058250 A CA 2204058 A1 EP 0814589 A2	02-05-2000 19-12-1997 29-12-1997
US 5588060	A	24-12-1996	EP 0693836 A1 JP 8008895 A US 5668877 A US 5633933 A US 6091820 A US 6026167 A	24-01-1996 12-01-1996 16-09-1997 27-05-1997 18-07-2000 15-02-2000
US 5689566	A	18-11-1997	US 5638448 A	10-06-1997
WO 9827783	A	25-06-1998	US 6032118 A AU 727878 B2 AU 5131198 A DE 19782193 D2 EP 1008275 A1 GB 2336511 A ,B WO 9827783 A1 SE 9902261 A	29-02-2000 04-01-2001 15-07-1998 25-11-1999 14-06-2000 20-10-1999 25-06-1998 16-06-1999



RECEIVED

MATCH & RETURN #16

AUG 28 2002

Technology Center 2100

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	)	Group Art Unit: 2153
	)	
Edmond Colby Munger et al.	)	Examiner: Krisna Lim
	)	
Serial No. 09/504,783	)	Attorney Docket No. 000479.85672
	)	
Filed: February 15, 2000	)	

For: IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

INFORMATION DISCLOSURE STATEMENT

Assistant Commissioner of Patents  
Washington, D.C. 20231

Sir:

In accordance with Applicants' duty of disclosure, and pursuant to 37 C.F.R. § 1.97(d), the following information is submitted for consideration by the United States Patent and Trademark Office in connection with the above-captioned application. The information is identified on the attached PTO 1449 form.

Applicants do not waive any right to take appropriate action to establish patentability over the listed documents should they be applied as references against the claims of the present application.

The undersigned certifies under 37 C.F.R. § 1.97(e)(1) that each item of information contained in this information disclosure statement was first cited in a communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of this statement. A copy of the foreign search report

is attached.

08/27/2002 GWORDF1 00000002 190733 09504783  
01 FC:126 180.00 CH

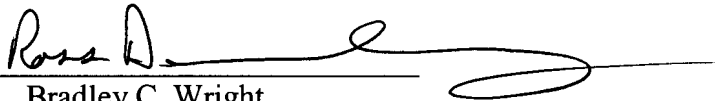
**Information Disclosure Statement**

Serial No. 09/504,783

The Commissioner is authorized to charge the \$180 fee to our Deposit Account No. 19-0733. No additional fees are believed due to ensure consideration of the attached documents by the Examiner. However, if any fees are required or an overpayment of fees made, the Commissioner is hereby authorized to debit or credit our Deposit Account No. 19-0733, as necessary.

Respectfully submitted,

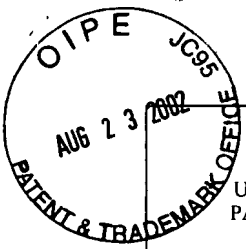
Date: August 23, 2002

By:   
Bradley C. Wright  
Registration No. 38,061

Banner & Witcoff, LTD  
1001 G Street, N.W.  
Washington, D.C. 20001-4597  
(202) 508-9100

**Reg. No. 49,024**

BCW/RAD/mmd



PTO-1449 (Modified) U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE  INFORMATION DISCLOSURE STATEMENT BY APPLICANT	ATTY. DOCKET NO. 000479.85672	SERIAL NUMBER 09/504,783
	APPLICANT Edmond Colby Munger et al.	
	FILING DATE February 15, 2000	GROUP ART UNIT 2153

**U.S. PATENT DOCUMENTS**

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE
<i>K</i>	5,588,060	12/24/96	Aziz	<del>          </del>	<del>          </del>	
<i>K</i>	5,689,566	11/18/9	Nguyen			

**FOREIGN PATENT DOCUMENTS**

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO
<i>K</i>	199 24 575	12/2/99	DE	<del>          </del>	<del>          </del>	
<i>K</i>	0 838 930	4/29/98	EPO			
<i>K</i>	2 317 792	4/1/98	GB			
<i>K</i>	0 814 589	12/29/97	EPO			
<i>K</i>	WO 98/27783	6/25/98	PCT			

**RECEIVED**  
**AUG 28 2002**  
 Technology Center 2100

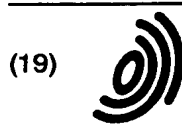
**OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)**

<i>K</i>	Search Report (dated 6/18/02), International Application No. PCT/US01/13260 -
<i>K</i>	Search Report (dated 6/28/02), International Application No. PCT/US01/13261 -
<i>K</i>	Donald E. Eastlake, "Domain Name System Security Extensions", DNS Security Working Group, April 1998, 51 pages
<i>K</i>	D. B. Chapman et al., "Building Internet Firewalls", November 1995, pages 278-297 and pages 351-375
<i>K</i>	P. Srisuresh et al., "DNS extensions to Network Address Translators", July 1998, 27 pages
<i>K</i>	Laurie Wells, "Security Icon", October 19, 1998, 1 page
<i>K</i>	W. Stallings, "Cryptography And Network Security", 2 <sup>nd</sup> Edition, Chapter 13, IP Security, June 8, 1998, pages 399-440
<i>K</i>	W. Stallings, "New Cryptography and Network Security Book", June 8, 1998, 3 pages

EXAMINER <i>KRISNA LIM</i>	DATE CONSIDERED <i>11/8/02</i>
----------------------------	--------------------------------

EXAMINER: Initial citation if reference was considered. Draw line through citation if not in conformance to MPEP 609 and not considered. Include copy of this form with next communication to applicant.

IDS w/1449 form filed: August 23, 2002



Europäisches Patentamt  
European Patent Office  
Office européen des brevets



(11) EP 0 838 930 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:  
29.04.1998 Bulletin 1998/18

(51) Int. Cl.<sup>6</sup>: H04L 29/06

(21) Application number: 97118556.6

(22) Date of filing: 24.10.1997

(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

(72) Inventors:  
• Alden, Kenneth F.  
Boylston, Massachusetts 01505 (US)  
• Lichtenberg, Mitchell P.  
Sunnyvale, CA 94087 (US)  
• Wobber, Edward P.  
Menlo Park, California 94025 (US)

(30) Priority: 25.10.1996 US 738155

(71) Applicant:  
DIGITAL EQUIPMENT CORPORATION  
Maynard, Massachusetts 01754 (US)

(74) Representative: Betten & Resch  
Reichenbachstrasse 19  
80469 München (DE)

(54) Pseudo network adapter for frame capture, encapsulation and encryption

(57) A new pseudo network adapter provides an interface for capturing packets from a local communications protocol stack for transmission on the virtual private network, and includes a Dynamic Host Configuration Protocol (DHCP) server emulator, and an Address Resolution Protocol (ARP) server emulator. The new system indicates to the local communications protocol stack that nodes on a remote private network are reachable through a gateway that is in turn reachable through the pseudo network adapter. A transmit path in the system processes data packets from the local communications protocol stack for transmission through the pseudo network adapter. An encryption engine encrypts the data packets and an encapsulation engine encapsulates the encrypted data packets into tunnel data frames. The network adapter further includes an interface into a transport layer of the local communications protocol stack for capturing received data packets from the remote server node, and a receive path for processing received data packets captured from the transport layer of the local communications protocol stack. The receive path includes a decapsulation engine, and a decryption engine, and passes the decrypted, decapsulated data packets back to the local communications protocol stack for delivery to a user.

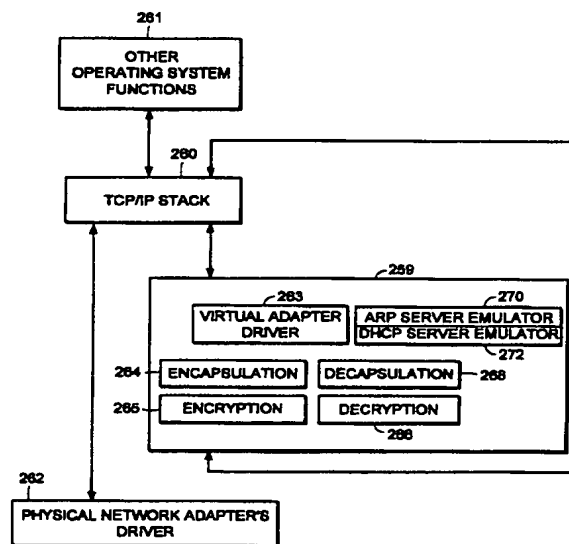


FIG. 15

EP 0 838 930 A2

## Description

### FIELD OF THE INVENTION

The invention relates generally to establishing secure virtual private networks. The invention relates specifically to a pseudo network adapter for capturing, encapsulating and encrypting messages or frames.

### BACKGROUND

In data communications it is often required that secure communications be provided between users of network stations (also referred to as "network nodes") at different physical locations. Secure communications must potentially extend over public networks as well as through secure private networks. Secure private networks are protected by "firewalls", which separate the private network from a public network. Firewalls ordinarily provide some combination of packet filtering, circuit gateway, and application gateway technology, insulating the private network from unwanted communications with the public network.

One approach to providing secure communications is to form a virtual private network. In a virtual private network, secure communications are provided by encapsulating and encrypting messages. Encapsulated messaging in general is referred to as "tunneling". Tunnels using encryption may provide protected communications between users separated by a public network, or among a subset of users of a private network.

Encryption may for example be performed using an encryption algorithm using one or more encryption "keys". When an encryption key is used, the value of the key determines how the data is encrypted and decrypted. When a public-key encryption system is used, a key pair is associated with each communicating entity. The key pair consists of an encryption key and a decryption key. The two keys are formed such that it is unfeasible to generate one key from the other. Each entity makes its encryption key public, while keeping its decryption key secret. When sending a message to node A, for example, the transmitting entity uses the public key of node A to encrypt the message, and then the message can only be decrypted by node A using node A's private key.

In a symmetric key encryption system a single key is used as the basis for both encryption and decryption. An encryption key in a symmetric key encryption system is sometimes referred to as a "shared" key. For example, a pair of communicating nodes A and B could communicate securely as follows: a first shared key is used to encrypt data sent from node A to node B, while a second shared key is to be used to encrypt data sent from node B to node A. In such a system, the two shared keys must be known by both node A and node B. More examples of encryption algorithms and keyed encryption are disclosed in many textbooks, for example

"Applied Cryptography - Protocols, Algorithms, and Source Code in C", by Bruce Schneier, published by John Wiley and Sons, New York, New York, copyright 1994.

Information regarding what encryption key or keys are to be used, and how they are to be used to encrypt data for a given secure communications session is referred to as "key exchange material". Key exchange material may for example determine what keys are used and a time duration for which each key is valid. Key exchange material for a pair of communicating stations can be exchanged in a secure communications session. How key exchange material is made known to the communicating stations for a given secure communications session is referred to as "session key establishment".

A tunnel may be implemented using a virtual or "pseudo" network adapter that appears to the communications protocol stack as a physical device and which provides a virtual private network. A pseudo network adapter must have the capability to receive packets from the communications protocol stack, and to pass received packets back through the protocol stack either to a user or to be transmitted.

A tunnel endpoint is the point at which any encryption/decryption and encapsulation/decapsulation provided by a tunnel is performed. In existing systems, the tunnel end points are pre-determined network layer addresses. The source network layer address in a received message is used to determine the "credentials" of an entity requesting establishment of a tunnel connection. For example, a tunnel server uses the source network layer address to determine whether a requested tunnel connection is authorized. The source network layer address is also used to determine which cryptographic key or keys to use to decrypt received messages.

Existing tunneling technology is typically performed by encapsulating encrypted network layer packets (also referred to as "frames") at the network layer. Such systems provide "network layer within network layer" encapsulation of encrypted messages. Tunnels in existing systems are typically between firewall nodes which have statically allocated IP addresses. In such existing systems, the statically allocated IP address of the firewall is the address of a tunnel end point within the firewall. Existing systems fail to provide a tunnel which can perform authorization based for an entity which must dynamically allocate its network layer address. This is especially problematic for a user wishing to establish a tunnel in a mobile computing environment, and who requests a dynamically allocated IP address from an Internet Service Provider (ISP).

Because existing virtual private networks are based on network layer within network layer encapsulation, they are generally only capable of providing connectionless datagram type services. Because datagram type services do not guarantee delivery of packets, existing



tunnels can only easily employ encryption methods over the data contained within each transmitted packet. Encryption based on the contents of multiple packets is desirable, such as cipher block chaining or stream ciphering over multiple packets. For example, encrypted data would advantageously be formed based not only on the contents of the present packet data being encrypted, but also based on some attribute of the connection or session history between the communicating stations. Examples of encryption algorithms and keyed encryption are disclosed in many textbooks, for example "Applied Cryptography - Protocols, Algorithms, and Source Code in C", by Bruce Schneier, published by John Wiley and Sons, New York, New York, copyright 1994.

Thus there is required a new pseudo network adapter providing a virtual private network having a dynamically determined end point to support a user in a mobile computing environment. The new pseudo network adapter should appear to the communications protocol stack of the node as an interface to an actual physical device. The new pseudo network adapter should support guaranteed, in-order delivery of frames over a tunnel to conveniently support cipher block chaining mode or stream cipher encryption over multiple packets.

### **SUMMARY OF THE INVENTION**

A new pseudo network adapter is disclosed providing a virtual private network. The new system includes an interface for capturing packets from a local communications protocol stack for transmission on the virtual private network. The interface appears to the local communications stack as a network adapter device driver for a network adapter.

The invention, in its broad form, includes a pseudo network adapter as recited in claim 1, providing a virtual network and a method therefor as recited in claim 9.

The system as described hereinafter further includes a Dynamic Host Configuration Protocol (DHCP) server emulator, and an Address Resolution Protocol (ARP) server emulator. The new system indicates to the local communications protocol stack that nodes on a remote private network are reachable through a gateway that is in turn reachable through the pseudo network adapter. The new pseudo network adapter includes a transmit path for processing data packets from the local communications protocol stack for transmission through the pseudo network adapter. The transmit path includes an encryption engine for encrypting the data packets and an encapsulation engine for encapsulating the encrypted data packets into tunnel data frames. The pseudo network adapter passes the tunnel data frames back to the local communications protocol stack for transmission to a physical network adapter on a remote server node.

Preferably, as described hereinafter, the pseudo

network adapter includes a digest value in a digest field in each of the tunnel data frames. A keyed hash function is a hash function which takes data and a shared cryptographic key as inputs, and outputs a digital signature referred to as a digest. The value of the digest field is equal to an output of a keyed hash function applied to data consisting of the data packet encapsulated within the tunnel data frame concatenated with a counter value equal to a total number of tunnel data frames previously transmitted to the remote server node. In another aspect of the system, the pseudo network adapter processes an Ethernet header in each one of the captured data packets, including removing the Ethernet header.

The new pseudo network adapter further includes an interface into a transport layer of the local communications protocol stack for capturing received data packets from the remote server node, and a receive path for processing received data packets captured from the transport layer of the local communications protocol stack. The receive path includes a decapsulation engine, and a decryption engine, and passes the decrypted, decapsulated data packets back to the local communications protocol stack for delivery to a user.

Thus there is disclosed a new pseudo network adapter providing a virtual private network having dynamically determined end points to support users in a mobile computing environment. The new pseudo network adapter provides a system for capturing a fully formed frame prior to transmission. The new pseudo network adapter appears to the communications protocol stack of the station as an interface to an actual physical device. The new pseudo network adapter further includes encryption capabilities to conveniently provide secure communications between tunnel end points using stream mode encryption or cipher block chaining over multiple packets.

### **BRIEF DESCRIPTION OF THE DRAWINGS**

A more detailed understanding of the invention may be had from the following description of a preferred embodiment, given by way of example and to be understood in conjunction with the accompanying drawing in which:

- ◆ Fig. 1 is a block diagram showing the Open Systems Interconnection (OSI) reference model;
- ◆ Fig. 2 is a block diagram showing the TCP/IP internet protocol suite;
- ◆ Fig. 3 is a block diagram showing an exemplary embodiment of a tunnel connection across a public network between two tunnel servers;
- ◆ Fig. 4 is a flow chart showing an exemplary embodiment of steps performed to establish a tunnel con-

nection;

- ◆ Fig. 5 is a flow chart showing an exemplary embodiment of steps performed to perform session key management for a tunnel connection; 5
- ◆ Fig. 6 is a block diagram showing an exemplary embodiment of a relay frame;
- ◆ Fig. 7 is a block diagram showing an exemplary embodiment of a connection request frame; 10
- ◆ Fig. 8 is a block diagram showing an exemplary embodiment of a connection response frame;
- ◆ Fig. 9 is a block diagram showing an exemplary embodiment of a data frame; 15
- ◆ Fig. 10 is a block diagram showing an exemplary embodiment of a close connection frame; 20
- ◆ Fig. 11 is a state diagram showing an exemplary embodiment of a state machine forming a tunnel connection in a network node initiating a tunnel connection; 25
- ◆ Fig. 12 is a state diagram showing an exemplary embodiment of a state machine forming a tunnel connection in a server computer;
- ◆ Fig. 13 is a state diagram showing an exemplary embodiment of a state machine forming a tunnel connection in a relay node;
- ◆ Fig. 14 is a block diagram showing an exemplary embodiment of a tunnel connection between a client computer (tunnel client) and a server computer (tunnel server); 35
- ◆ Fig. 15 is a block diagram showing an exemplary embodiment of a pseudo network adapter; 40
- ◆ Fig. 16 is a block diagram showing an exemplary embodiment of a pseudo network adapter;
- ◆ Fig. 17 is a flow chart showing steps performed by an exemplary embodiment of a pseudo network adapter during packet transmission; 45
- ◆ Fig. 18 is a flow chart showing steps performed by an exemplary embodiment of a pseudo network adapter during packet receipt; 50
- ◆ Fig. 19 is a data flow diagram showing data flow in an exemplary embodiment of a pseudo network adapter during packet transmission; 55
- ◆ Fig. 20 is a data flow diagram showing data flow in

an exemplary embodiment of a pseudo network adapter during packet receipt;

- ◆ Fig. 21 is a diagram showing the movement of encrypted and unencrypted data in an exemplary embodiment of a system including a pseudo network adapter;
- ◆ Fig. 22 is a diagram showing the movement of encrypted and unencrypted data in an exemplary embodiment of a system including a pseudo network adapter; and
- ◆ Fig. 23 is a flow chart showing steps initialization of an exemplary embodiment of a system including a pseudo network adapter.

#### DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Now with reference to Fig. 1 there is described for purposes of explanation, communications based on the Open Systems Interconnection (OSI) reference model. In Fig. 1 there is shown communications 12 between a first protocol stack 10 and a second protocol stack 14. The first protocol stack 10 and second protocol stack 14 are implementations of the seven protocol layers (Application layer, Presentation layer, Session layer, Transport layer, Network layer, Data link layer, and Physical layer) of the OSI reference model. A protocol stack implementation is typically in some combination of software and hardware. Descriptions of the specific services provided by each protocol layer in the OSI reference model are found in many text books, for example "Computer Networks", Second Edition, by Andrew S. Tannenbaum, published by Prentice-Hall, Englewood Cliffs, New Jersey, copyright 1988.

As shown in Fig. 1, data 11 to be transmitted from a sending process 13 to a receiving process 15 is passed down through the protocol stack 10 of the sending process to the physical layer 9 for transmission on the data path 7 to the receiving process 15. As the data 11 is passed down through the protocol stack 10, each protocol layer prepends a header (and possibly also appends a trailer) portion to convey information used by that protocol layer. For example, the data link layer 16 of the sending process wraps the information received from the network layer 17 in a data link header 18 and a data link layer trailer 20 before the message is passed to the physical layer 9 for transmission on the actual transmission path 7.

Fig. 2 shows the TCP/IP protocol stack. Some protocol layers in the TCP/IP protocol stack correspond with layers in the OSI protocol stack shown in Fig. 1. The detailed services and header formats of each layer in the TCP/IP protocol stack are described in many texts, for example "Internetworking with TCP/IP, Vol. 1: Principles, Protocols, and Architecture", Second Edi-

tion, by Douglas E. Comer, published by Prentice-Hall, Englewood Cliffs, New Jersey, copyright 1991. The Transport Control Protocol (TCP) 22 corresponds to the Transport layer in the OSI reference model. The TCP protocol 22 provides a connection-oriented, end to end transport service with guaranteed, in-sequence packet delivery. In this way the TCP protocol 22 provides a reliable, transport layer connection.

The IP protocol 26 corresponds to the Network layer of the OSI reference model. The IP protocol 26 provides no guarantee of packet delivery to the upper layers. The hardware link level and access protocols 32 correspond to the Data link and Physical layers of the OSI reference model.

The Address Resolution Protocol (ARP) 28 is used to map IP layer addresses (referred to as "IP addresses") to addresses used by the hardware link level and access protocols 32 (referred to as "physical addresses" or "MAC addresses"). The ARP protocol layer in each network station typically contains a table of mappings between IP addresses and physical addresses (referred to as the "ARP cache"). When a mapping between an IP address and the corresponding physical address is not known, the ARP protocol 28 issues a broadcast packet (an "ARP request" packet) on the local network. The ARP request indicates an IP address for which a physical address is being requested. The ARP protocols 28 in each station connected to the local network examine the ARP request, and if a station recognizes the IP address indicated by the ARP request, it issues a response (an "ARP response" or "ARP reply" packet) to the requesting station indicating the responder's physical address. The requesting ARP protocol reports the received physical address to the local IP layer which then uses it to send datagrams directly to the responding station. As an alternative to having each station respond only for its own IP address, an ARP server may be used to respond for a set of IP addresses it stores internally, thus potentially eliminating the requirement of a broadcast request. In that case, the ARP request can be sent directly to the ARP server for physical addresses corresponding to any IP address mappings stored within the ARP server.

At system start up, each station on a network must determine an IP address for each of its network interfaces before it can communicate using TCP/IP. For example, a station may need to contact a server to dynamically obtain an IP address for one or more of its network interfaces. The station may use what is referred to as the Dynamic Host Configuration Protocol (DHCP) to issue a request for an IP address to a DHCP server. For example, a DHCP module broadcasts a DHCP request packet at system start up requesting allocation of an IP address for an indicated network interface. Upon receiving the DHCP request packet, the DHCP server allocates an IP address to the requesting station for use with the indicated network interface. The

requesting station then stores the IP address in the response from the server as the IP address to associate with that network interface when communicating using TCP/IP.

Fig. 3 shows an example configuration of network nodes for which the presently disclosed system is applicable. In the example of Fig. 3, the tunnel server A is an initiator of the tunnel connection. As shown in Fig. 3, the term "tunnel relay" node is used to refer to a station which forwards data packets between transport layer connections (for example TCP connections).

For example, in the present system a tunnel relay may be dynamically configured to forward packets between transport layer connection 1 and transport layer connection 2. The tunnel relay replaces the header information of packets received over transport layer connection 1 with header information indicating transport layer connection 2. The tunnel relay can then forward the packet to a firewall, which may be conventionally programmed to pass packets received over transport layer connection 2 into a private network on the other side of the firewall. In the present system, the tunnel relay dynamically forms transport layer connections when a tunnel connection is established. Accordingly the tunnel relay is capable of performing dynamic load balancing or providing redundant service for fault tolerance over one or more tunnel servers at the time the tunnel connection is established.

Fig. 3 shows a Tunnel Server A 46 in a private network N1 48, physically connected with a first Firewall 50. The first Firewall 50 separates the private network N1 48 from a public network 52, for example the Internet. The first Firewall 50 is for example physically connected with a Tunnel Relay B 54, which in turn is virtually connected through the public network 52 with a Tunnel Relay C. The connection between Tunnel Relay B and Tunnel Relay C may for example span multiple intervening forwarding nodes such as routers or gateways through the public network 52.

The Tunnel Relay C is physically connected with a second Firewall 58, which separates the public network 52 from a private network N2 60. The second Firewall 58 is physically connected with a Tunnel Server D 62 on the private network N2 60. During operation of the elements shown in Fig. 3, the Tunnel Server D 62 provides routing of IP packets between the tunnel connection with Tunnel Server A 46 and other stations on the private network N2 60. In this way the Tunnel Server D 62 acts as a router between the tunnel connection and the private network N2 60.

During operation of the elements shown in Fig. 3, the present system establishes a tunnel connection between the private network N1 48 and the private network N2 60. The embodiment of Fig. 3 thus eliminates the need for a dedicated physical cable or line to provide secure communications between the private network 48 and the private network 60. The tunnel connection between Tunnel Server A 46 and Tunnel Server D 62 is

composed of reliable, pair-wise transport layer connections between Tunnel Server A 46 (node "A"), Tunnel Relay B 54 (node "B"), Tunnel Relay C 56 (node "C"), and Tunnel Server D 62 (node "D"). For example, such pair-wise connections may be individual transport layer connections between each node A and node B, node B and node C, and node C and node D. In an alternative embodiment, as will be described below, a tunnel connection may alternatively be formed between a stand-alone PC in a public network and a tunnel server within a private network.

Fig. 4 and Fig. 5 show an example embodiment of steps performed during establishment of the tunnel connection between Tunnel Server A 46 (node "A") and Tunnel Server D 62 (node "D") as shown in Fig. 3. Prior to the steps shown in Fig. 4, node A selects a tunnel path to reach node D. The tunnel path includes the tunnel end points and any intervening tunnel relays. The tunnel path is for example predetermined by a system administrator for node A. Each tunnel relay along the tunnel path is capable of finding a next node in the tunnel path, for example based on a provided next node name (or "next node arc"), using a predetermined naming convention and service, for example the Domain Name System (DNS) of the TCP/IP protocol suite.

During the steps shown in Fig. 4, each of the nodes A, B and C perform the following steps:

- resolve the node name of the next node in the tunnel path, for example as found in a tunnel relay frame;
- establish a reliable transport layer (TCP) connection to the next node in the tunnel path;
- forward the tunnel relay frame down the newly formed reliable transport layer connection to the next node in the tunnel path.

As shown for example in Fig. 4, at step 70 node A establishes a reliable transport layer connection with node B. At step 72 node A identifies the next downstream node to node B by sending node B a tunnel relay frame over the reliable transport layer connection between node A and node B. The tunnel relay frame contains a string buffer describing all the nodes along the tunnel path (see below description of an example tunnel relay frame format). At step 74, responsive to the tunnel relay frame from node A, node B searches the string buffer in the relay frame to determine if the string buffer includes node B's node name. If node B finds its node name in the string buffer, it looks at the next node name in the string buffer to find the node name of the next node in the tunnel path.

Node B establishes a reliable transport layer connection with the next node in the tunnel path, for example node C. Node B further forms an association between the reliable transport layer connection between

Node A and Node B, over which the relay frame was received, and the newly formed reliable transport layer connection between Node B and Node C, and as a result forwards subsequent packets received over the reliable transport layer connection with Node A onto the reliable transport layer connection with Node C, and vice versa. At step 76 node B forwards the tunnel relay frame on the newly formed reliable transport layer connection to node C.

At step 78, responsive to the relay frame forwarded from node B, node C determines that the next node in the tunnel path is the last node in the tunnel path, and accordingly is a tunnel server. Node C may actively determine whether alternative tunnel servers are available to form the tunnel connection. Node C may select one of the alternative available tunnel servers to form the tunnel connection in order to provide load balancing or fault tolerance. As a result node C may form a transport layer connections with one of several available tunnel servers, for example a tunnel server that is relatively underutilized at the time the tunnel connection is established. In the example embodiment, node C establishes a reliable transport layer connection with the next node along the tunnel path, in this case node D.

Node C further forms an association between the reliable transport layer connection between Node B and Node C, over which the relay frame was received, and the newly formed reliable transport layer connection between Node C and Node D, and as a result forwards subsequent packets received over the reliable transport layer connection with Node B to the reliable transport layer connection with Node D, and vice versa. At step 80 node C forwards the relay frame to node D on the newly formed reliable transport layer connection.

Fig. 5 shows an example of tunnel end point authentication and sharing of key exchange material provided by the present system. The present system supports passing authentication data and key exchange material through the reliable transport layer connections previously established on the tunnel path. The following are provided by use of a key exchange/authentication REQUEST frame and a key exchange/authentication RESPONSE frame:

- a) mutual authentication of both endpoints of the tunnel connection;
- b) establishment of shared session encryption keys and key lifetimes for encrypting/authenticating subsequent data sent through the tunnel connection;
- d) agreement on a shared set of cryptographic transforms to be applied to subsequent data; and
- e) exchange of any other connection-specific data between the tunnel endpoints, for example strength and type of cipher to be used, any compression of the data to be used, etc. This data can also be used

by clients of this protocol to qualify the nature of the authenticated connection.

At step 90 a key exchange/authentication request frame is forwarded over the reliable transport layer connections formed along the tunnel path from node A to node D. At step 92, a key exchange/authentication response frame is forwarded from node D back to node A through the reliable transport layer connections. The attributes exchanged using the steps shown in Fig. 5 may be used for the lifetime of the tunnel connection. In an alternative embodiment the steps shown in Fig. 5 are repeated as needed for the tunnel end points to exchange sufficient key exchange material to agree upon a set of session parameters for use during the tunnel connection such as cryptographic keys, key durations, and choice of encryption/decryption algorithms.

Further in the disclosed system, the names used for authentication and access control with regard to node A and node D need not be the network layer address or physical address of the nodes. For example, in an alternative embodiment where the initiating node sending the tunnel relay frame is a stand-alone PC located within a public network, the user's name may be used for authentication and/or access control purposes. This provides a significant improvement over existing systems which base authorization on predetermined IP addresses.

Fig. 6 shows the format of an example embodiment of a tunnel relay frame. The tunnel frame formats shown in Figs. 6, 7, 8 and 9 are encapsulated within the data portion of a transport layer (TCP) frame when transmitted. Alternatively, another equivalent, connection-oriented transport layer protocol having guaranteed, in-sequence frame delivery may be used. The example TCP frame format, including TCP header fields, is conventional and not shown.

The field 100 contains a length of the frame. The field 102 contains a type of the frame, for example a type of RELAY. The field 104 contains a tunnel protocol version number. The field 106 contains an index into a string buffer field 112 at which a name of the originating node is located, for example a DNS host name of the node initially issuing the relay frame (node A in Fig. 3). The fields following the origin index field 106 contain indexes into the string buffer 112 at which names of nodes along the tunnel path are located. For example each index may be the offset of a DNS host name within the string buffer 112. In this way the field 108 contains the index of the name of the first node in the tunnel path, for example node B (Fig. 3). The field 110 contains the index of the name of the second node in the tunnel path, etc. The field 112 contains a string of node names of nodes in the tunnel path.

During operation of the present system, the initiating node, for example node A as shown in Fig. 3, transmits a tunnel relay frame such as the tunnel relay frame shown in Fig. 6. Node A sends the tunnel relay frame to

the first station along the tunnel path, for example node B (Fig. 3), over a previously established reliable transport layer connection. Node B searches the string buffer in the tunnel relay frame to find its node name, for example its DNS host name. Node B finds its node name in the string buffer indexed by path index 0, and then uses the contents of path index 1 110 to determine the location within the string buffer 112 of the node name of the next node along the tunnel path. Node B uses this node name to establish a reliable transport layer connection with the next node along the tunnel path. Node B then forwards the relay frame to the next node. This process continues until the end node of the tunnel route, for example tunnel server D 62 (Fig. 3) is reached.

Fig. 7 shows the format of an example embodiment of a key exchange/key authentication request frame. The field 120 contains a length of the frame. The field 122 contains a type of the frame, for example a type of REQUEST indicating a key exchange/key authentication request frame. The field 124 contains a tunnel protocol version number. The field 126 contains an offset of the name of the entity initiating the tunnel connection, for example the name of a user on the node originally issuing the request frame. This name and key exchange material in the request frame are used by the receiving tunnel end point to authenticate the key exchange/authentication REQUEST. The name of the entity initiating the tunnel connection is also used to authorize any subsequent tunnel connection, based on predetermined security policies of the system. The field 128 contains an offset into the frame of the node name of the destination node, for example the end node of the tunnel shown as node D 62 in Fig. 3.

The field 130 contains an offset into the frame at which key exchange data as is stored, for example within the string buffer field 138. The key exchange data for example includes key exchange material used to determine a shared set of encryption parameters for the life of the tunnel connection such as cryptographic keys and any validity times associated with those keys. The key exchange data, as well as the field 132, further include information regarding any shared set of cryptographic transforms to be used and any other connection-specific parameters, such as strength and type of cipher to be used, type of compression of the data to be used, etc. The field 134 contains flags, for example indicating further information about the frame. The field 136 contains client data used in the tunnel end points to configure the local routing tables so that packets for nodes reachable through the virtual private network are sent through the pseudo network adapters. In an example embodiment, the string buffer 138 is encrypted using a public encryption key of the receiving tunnel end point.

During operation of the present system, one of the end nodes of the tunnel sends a key exchange/authentication REQUEST frame as shown in Fig. 7 to the other end node of the tunnel in order to perform key exchange and authentication as described in step 90 of Fig. 5.

Fig. 8 shows the format of an example embodiment of a key exchange/key authentication response frame, referred to as a connection RESPONSE frame. The field 150 contains a length of the frame. The field 152 contains a type of the frame, for example a type of connection RESPONSE indicating a key exchange/key authentication request frame. The field 154 contains a tunnel protocol version number.

The field 156 contains an offset into the frame at which key exchange data as is stored, for example within the string buffer field 163. The key exchange data for example includes key exchange material to be used for encryption/decryption over the life of the tunnel connection and any validity times associated with that key exchange material. The key exchange data, as well as the field 158, further includes information regarding any shared set of cryptographic transforms to be applied to subsequent data and any other connection-specific parameters, such as strength and type of cipher to be used, any compression of the data to be used, etc. The field 160 contains flags, for example indicating other information about the frame. The client data field 162 contains data used by the pseudo network adapters in the tunnel end points to configure the local routing tables so that packets for nodes in the virtual private network are sent through the pseudo network adapters. The string buffer includes key exchange material. The string buffer is for example encrypted using a public encryption key of the receiving tunnel end point, in this case the initiator of the tunnel connection.

During operation of the present system, one of the end nodes of the tunnel sends a key exchange/authentication RESPONSE frame as shown in Fig. 7 to the other end node of the tunnel in order to perform key exchange and authentication as described in step 92 of Fig. 5.

Fig. 9 shows the format of an example embodiment of an tunnel data frame used to communicate through a tunnel connection. Fig. 9 shows how an IP datagram may be encapsulated within a tunnel frame by the present system for secure communications through a virtual private network. The field 170 contains a length of the frame. The field 172 contains a type of the frame, for example a type of DATA indicating a tunnel data frame. The field 174 contains a tunnel protocol version number.

The fields 176, 178 and 182 contain information regarding the encapsulated datagram. The field 180 contains flags indicating information regarding the frame. The field 184 contains a value indicating the length of the optional padding 189 at the end of the frame. The frame format allows for optional padding in the event that the amount of data in the frame needs to be padded to an even block boundary for the purpose of being encrypted using a block cipher. The field 186 contains a value indicating the length of the digest field 187.

The data frame format includes a digital signature generated by the transmitting tunnel end point referred

to as a "digest". The value of the digest ensures data integrity, for example by detecting invalid frames and replays of previously transmitted valid frames. The digest is the output of a conventional keyed cryptographic hash function applied to both the encapsulated datagram 190 and a monotonically increasing sequence number. The resulting hash output is passed as the value of the digest field 187. The sequence number is not included in the data frame. In the example embodiment, the sequence number is a counter maintained by the transmitter (for example node A in Fig. 3) of all data frames sent to the receiving node (for example node D in Fig. 3) since establishment of the tunnel connection.

In order to determine if the data frame is invalid or a duplicate, the receiving node decrypts the encapsulated datagram 190, and applies the keyed cryptographic hash function (agreed to by the tunnel end nodes during the steps shown in Fig. 5) to both the decrypted encapsulated datagram and the value of a counter indicating the number of data frames received from the transmitter since establishment of the tunnel connection. For example the keyed hash function is applied to the datagram concatenated to the counter value. If the resulting hash output matches the value of the digest field 187, then the encapsulated datagram 190 was received correctly and is not a duplicate. If the hash output does not match the value of the digest field 187, then the integrity check fails, and the tunnel connection is closed. The field 188 contains an encrypted network layer datagram, for example an encrypted IP datagram.

The encapsulated datagram may be encrypted using various encryption techniques. An example embodiment of the present system advantageously encrypts the datagram 190 using either a stream cipher or cipher block chaining encryption over all data transmitted during the life of the tunnel connection. This is enabled by the reliable nature of the transport layer connections within the tunnel connection. The specific type of encryption and any connection specific symmetric encryption keys used is determined using the steps shown in Fig. 5. The fields in the tunnel data frame other than the encapsulated datagram 188 are referred to as the tunnel data frame header fields.

Fig. 10 is a block diagram showing an example embodiment of a "close connection" frame. The field 190 contains the length of the frame. The field 191 contains a frame type, for example having a value equal to CLOSE. Field 192 contains a value equal to the current protocol version number of the tunnel protocol. The field 193 contains a status code indicating the reason the tunnel connection is being closed.

During operation of the present system, when end point of a tunnel connection determines that the tunnel connection should be closed, a close connection frame as shown in Fig. 10 is transmitted to the other end point of the tunnel connection. When a close connection close frame is received, the receiver closes the tunnel

connection and no further data will be transmitted or received through the tunnel connection.

Fig. 11 is a state diagram showing an example embodiment of forming a tunnel connection in a node initiating a tunnel connection. In Fig. 11, Fig. 12, and Fig. 13, states are indicated by ovals and actions or events are indicated by rectangles. For example the tunnel server node A as shown in Fig. 3 may act as a tunnel connection initiator when establishing a tunnel connection with the tunnel server node D. Similarly the client system 247 in Fig. 14 may act as a tunnel connection initiator when establishing a tunnel connection with the tunnel server. The tunnel initiator begins in an idle state 194. Responsive to an input from a user indicating that a tunnel connection should be established, the tunnel initiator transitions from the idle state 194 to a TCP Open state 195. In the TCP Open state 195, the tunnel initiator establishes a reliable transport layer connection with a first node along the tunnel path. For example, the tunnel initiator opens a socket interface associated with a TCP connection to the first node along the tunnel path. In Fig. 3 node A opens a socket interface associated with a TCP connection with node B.

Following establishment of the reliable transport layer connection in the TCP Open state 195, the tunnel initiator enters a Send Relay state 197. In the Send Relay state 197, the tunnel initiator transmits a relay frame at 198 over the reliable transport layer connection. Following transmission of the relay frame, the tunnel initiator enters the connect state 199. If during transmission of the relay frame there is a transmission error, the tunnel initiator enters the Network Error state 215 followed by the Dying state 208. In the Dying state 208, the tunnel initiator disconnects the reliable transport layer connection formed in the TCP Open state 195, for example by disconnecting a TCP connection with Node B. Following the disconnection at 209, the tunnel initiator enters the Dead state 210. The tunnel initiator subsequently transitions back to the Idle state 194 at a point in time predetermined by system security configuration parameters.

In the Connect state 199, the tunnel initiator sends a key exchange/authentication REQUEST frame at 200 to the tunnel server. Following transmission of the key exchange/authentication REQUEST frame 200, the tunnel initiator enters the Response Wait state 201. The tunnel initiator remains in the Response Wait state 201 until it receives a key exchange/authentication RESPONSE frame 202 from the tunnel server. After the key exchange/authentication RESPONSE frame is received at 202, the tunnel initiator enters the Authorized state 203, in which it may send or receive tunnel data frames. Upon receipt of a CLOSE connection frame at 216 in the Authorized state 203, the tunnel initiator transitions to the Dying state 208.

Upon expiration of a session encryption key at 211, the tunnel initiator enters the Reconnect state 212, and sends a CLOSE connection frame at 213 and discon-

nects the TCP connection with the first node along the tunnel path at 214. Subsequently the tunnel initiator enters the TCP Open state 195.

If during the authorized state 203, a local user issues an End Session command at 204, or there is a detection of an authentication or cryptography error in a received data frame at 205, the tunnel initiator enters the Close state 206. During the Close state 206 the tunnel initiator sends a CLOSE connection frame at 207 to the tunnel server. The tunnel initiator then enters the Dying state at 208.

Figure 12 is a state diagram showing the states within an example embodiment of a tunnel server, for example node D in Fig. 3 or tunnel server 253 in Fig. 14. The tunnel server begins in an Accept Wait state 217. In the Accept Wait state 217, the tunnel server receives a request for a reliable transport layer connection, for example a TCP connection request 218 from the last node in the tunnel path prior to the tunnel server, for example Node C in Fig. 3. In response to a TCP connection request 218 the tunnel server accepts the request and establishes a socket interface associated with the resulting TCP connection with Node C.

Upon establishment of the TCP connection with the last node in the tunnel path prior to the tunnel server, the tunnel server enters the Receive Relay state 219. In the Receive Relay state 219, the tunnel server waits to receive a relay frame at 220, at which time the tunnel server enters the Connect Wait state 221. If there is some sort of network error 234 during receipt of the relay frame at 219, the tunnel server enters the Dying state 230. During the Dying state 230 the tunnel server disconnects at 231 the transport layer connection with the last node in the tunnel path prior to the tunnel server. After disconnecting the connection, the tunnel server enters the Dead state 232.

In the Connect Wait state 221, the tunnel server waits for receipt of a key exchange/authentication REQUEST frame at 222. Following receipt of the key exchange/authentication REQUEST frame at 222, the tunnel server determines whether the requested tunnel connection is authorized at step 223. The determination of whether the tunnel connection is authorized is based on a name of the tunnel initiator, and the key exchange material within the key exchange/authentication REQUEST frame.

If the requested tunnel connection is authorized the tunnel server sends a key exchange/authentication RESPONSE frame at 224 back to the tunnel initiator. If the requested tunnel connection is not authorized, the tunnel server enters the Close state 228, in which it sends a close connection frame at 229 to the tunnel client. Following transmission of the CLOSE connection frame at 229, the tunnel server enters the Dying state 230.

If the requested tunnel connection is determined to be authorized at step 223, the tunnel server enters the Authorized state 225. In the Authorized state, the tunnel

server transmits and receives tunnel data frames between itself and the tunnel initiator. If during the Authorized state 225, the tunnel server receives a CLOSE connection frame at 233, the tunnel server transitions to the Dying state 230. If during the authorized state 225, the tunnel server receives an end session command from a user at 226, then the tunnel server transitions to the Close state 228, and transmits a close connection frame at 229 to the tunnel initiator. If the tunnel server in the Authorized state 225 detects an integrity failure in a received packet, the tunnel server transitions to the Close state 228. In the close state 228 the tunnel server sends a CLOSE connection frame at 229 and subsequently enters the Dying state 230.

Fig. 13 is a state diagram showing an example embodiment of a state machine within a tunnel relay node. The tunnel relay node begins in an Accept Wait state 235. When a request is received to form a reliable transport layer connection at 236, a reliable transport layer connection is accepted with the requesting node. For example, a TCP connection is accepted between the relay node and the preceding node in the tunnel path.

The relay node then transitions to the Receive Relay state 237. During the Receive Relay state 237, the relay node receives a relay frame at 238. Following receipt of the relay frame at 238, the relay node determines what forwarding address should be used to forward frames received from the TCP connection established responsive to the TCP connect event 236. If the next node in the tunnel path is a tunnel server, the forwarding address may be selected at 239 so as to choose an underutilized tunnel server from a group of available tunnel servers or to choose an operational server where others are not operational.

Following determination of the forwarding address or addresses in step 239, the relay node enters the Forward Connect state 240. In the Forward Connect state 240, the relay node establishes a reliable transport layer connection with the node or nodes indicated by the forwarding address or addresses determined in step 239.

Following establishment of the new connection at event 241, the tunnel relay enters the Forward state 242. During the Forward state 242, the relay node forwards all frames between the connection established at 236 and those connections established at 241. Upon detection of a network error or receipt of a frame indicating a closure of the tunnel connection at 243, the tunnel relay enters the Dying state 244. Following the Dying state 244, the relay node disconnects any connections established at event 241. The relay node then enters the Dead state 246.

Fig. 14 shows an example embodiment of a virtual private network 249 formed by a pseudo network adapter 248 and a tunnel connection between a tunnel client 247 and a tunnel server 253 across a public network 251. The tunnel server 253 and tunnel client 247 are for example network stations including a CPU or

microprocessor, memory, and various I/O devices. The tunnel server 253 is shown physically connected to a private LAN 256 including a Network Node 1 257 and a Network Node 2 258, through a physical network adapter 254. The tunnel server 253 is further shown physically connected with a firewall 252 which separates the private LAN 256 from the public network 251. The firewall 252 is physically connected with the public network 251. The tunnel server 253 is further shown including a pseudo network adapter 255. The client system 247 is shown including a physical network adapter 250 physically connected to the public network 251.

During operation of the elements shown in Fig. 14, nodes within the virtual private network 249 appear to the tunnel client 247 as if they were physically connected to the client system through the pseudo network adapter 248. Data transmissions between the tunnel client and any nodes that appear to be within the virtual private network are passed through the pseudo network adapter 248. Data transmissions between the tunnel client 247 and the tunnel server 253 are physically accomplished using a tunnel connection between the tunnel client 247 and the tunnel server 253.

Fig. 15 shows elements in an example embodiment of a pseudo network adapter such as the pseudo network adapter 248 in Fig. 14. In an example embodiment the elements shown in Fig. 15 are implemented as software executing on the tunnel client 247 as shown in Fig. 14. In Fig. 15 there is shown a pseudo network adapter 259 including a virtual adapter driver interface 263, an encapsulation engine 264, an encryption engine 265, a decapsulation engine 268, and a decryption engine 266. Further shown in the pseudo network adapter 259 are an ARP server emulator 270 and a Dynamic Host Configuration Protocol (DHCP) server emulator.

The pseudo network adapter 259 is shown interfaced to a TCP/IP protocol stack 260, through the virtual adapter driver interface 260. The TCP/IP protocol stack 260 is shown interfaced to other services in an operating system 261, as well as a physical network adapter's driver 262. The physical network adapter's driver 262 is for example a device driver which controls the operation of a physical network adapter such as physical network adapter 250 as shown in Fig. 14.

During operation of the elements shown in Fig. 15, the pseudo network adapter 259 registers with the network layer in the TCP/IP stack 260 that it is able to reach the IP addresses of nodes within the virtual private network 249 as shown in Fig. 14. For example, the pseudo network adapter on the client system registers that it can reach the pseudo network adapter on the server. Subsequently, a message from the tunnel client addressed to a node reachable through the virtual private network will be passed by the TCP/IP stack to the pseudo network adapter 259. The pseudo network adapter 259 then encrypts the message, and encapsulates the message into a tunnel data frame. The pseudo network adapter 259 then passes the tunnel data frame



back to the TCP/IP protocol stack 260 to be sent through to the physical network adapter in the tunnel server. The tunnel server passes the received data frame to the pseudo network adapter in the server, which de-encapsulates and decrypts the message.

Fig. 16 shows a more detailed example embodiment of a pseudo network adapter 280. The pseudo network adapter 280 includes a virtual network adapter driver interface 288. The transmit path 290 includes an encryption engine 292, and an encapsulation engine 294. The encapsulation engine 294 is interfaced with a TCP/IP transmit interface 312 within a TCP/IP protocol stack, for example a socket interface associated with the first relay node in the tunnel path, or with the remote tunnel end point if the tunnel path includes no relays.

In the example embodiment of Fig. 16, the pseudo network adapter 280 appears to the TCP/IP protocol stack 282 as an Ethernet adapter. Accordingly, ethernet packets 286 for a destination addresses understood by the TCP/IP protocol stack to be reachable through the virtual private network are passed from the TCP/IP protocol stack 282 to the virtual network adapter interface 288 and through the transmit path 290. Similarly, ethernet packets 284 received through the pseudo network adapter 280 are passed from the receive path 296 to the virtual network adapter interface 288 and on to the TCP/IP protocol stack 282.

Further shown in the pseudo network adapter 280 of Fig. 16 is a receive path 296 having a decryption engine 298 interfaced to the virtual network adapter interface 288 and a decapsulation engine 300. The decapsulation engine 300 in turn is interfaced to a TCP/IP receive function 314 in the TCP/IP protocol stack 282, for example a socket interface associated with the first relay in the tunnel path, or with the remote tunnel end point if the tunnel path includes no relays. The pseudo network adapter 280 further includes an ARP server emulator 304 and a DHCP server emulator 306. ARP and DHCP request packets 302 are passed to the ARP server emulator 304 and DHCP server emulator 306 respectively. When a received packet is passed from the receive path 296 to the TCP/IP stack 282, a receive event must be indicated to the TCP/IP stack 282, for example through an interface such the Network Device Interface Specification (NDIS), defined by Microsoft™ Corporation.

Also in Fig. 16 is shown is an operating system 310 coupled with the TCP/IP protocol stack 282. The TCP/IP protocol stack 282 is generally considered to be a component part of the operating system. The operating system 310 in Fig. 16 is accordingly the remaining operating system functions and procedures outside the TCP/IP protocol stack 282. A physical network adapter 308 is further shown operated by the TCP/IP protocol stack 282.

During operation of the elements shown in Fig. 16, a user passes data for transmission to the TCP/IP protocol stack 282, and indicates the IP address of the

node to which the message is to be transmitted, for example through a socket interface to the TCP layer. The TCP/IP protocol stack 282 then determines whether the destination node is reachable through the virtual private network. If the message is for a node that is reachable through the virtual private network, the TCP/IP protocol stack 282 an ethernet packet 286 corresponding to the message to the pseudo network adapter 280. The pseudo network adapter 280 then passes the ethernet packet 286 through the transmit path, in which the ethernet packet is encrypted and encapsulated into a tunnel data frame. The tunnel data frame is passed back into the TCP/IP protocol stack 282 through the TCP/IP transmit function 312 to be transmitted to the tunnel server through the tunnel connection. In an example embodiment, a digest value is calculated for the tunnel data frame before encryption within the transmit path within the pseudo network adapter.

Further during operation of the elements shown in Fig. 16, when the TCP/IP protocol stack 282 receives a packet from the remote endpoint of the TCP/IP tunnel connection, for example the tunnel server, the packet is passed to the pseudo network adapter 280 responsive to a TCP receive event. The pseudo network adapter 280 then decapsulates the packet by removing the tunnel header. The pseudo network adapter further decrypts the decapsulated data and passes it back to the TCP/IP protocol stack 282. The data passed from the pseudo network adapter 280 appears to the TCP/IP protocol stack 282 as an ethernet packet received from an actual physical device, and is the data it contains is passed on to the appropriate user by the TCP/IP protocol stack 282 based on information in the ethernet packet header provided by the pseudo network adapter.

Fig. 17 is a flow chart showing steps performed by an example embodiment of a pseudo network adapter during packet transmission, such as in the transmit path 290 of Fig. 14. The TCP/IP protocol stack determines that the destination node of a packet to be transmitted is reachable through the virtual LAN based on the destination IP address of the packet and a network layer routing table. At step 320 the packet is passed to the pseudo network adapter from the TCP/IP protocol stack. As a result, a send routine in the pseudo adapter is triggered for example in the virtual network adapter interface 288 of Fig. 16.

At step 322 the pseudo network adapter send routine processes the Ethernet header of the packet provided by the TCP/IP stack, and removes it. At step 324, the send routine determines whether the packet is an ARP request packet. If the packet is an ARP request packet for an IP address of a node on the virtual LAN, such as the pseudo network adapter of the tunnel server, then step 324 is followed by step 326. Otherwise, step 324 is followed by step 330.

At step 326, the ARP server emulator in the pseudo network adapter generates an ARP reply packet. For example, if the ARP request were for a physical address

corresponding to the IP address of the pseudo network adapter on the tunnel server, the ARP reply would indicate a predetermined, reserved physical address to be associated with that IP address. At step 328 the pseudo network adapter passes the ARP response to the virtual network adapter interface. The virtual network adapter interface then indicates a received packet to the TCP/IP protocol stack, for example using an NDIS interface. The TCP/IP protocol stack then processes the ARP response as if it had been received over an actual physical network.

At step 330 the send routine determines whether the packet is a DHCP request packet requesting an IP address for the pseudo network adapter. If so, then step 330 is followed by step 332. Otherwise, step 330 is followed by step 334.

At step 334, the DHCP server emulator in the pseudo network adapter generates a DHCP response. The format of DHCP is generally described in the DHCP RFC. At step 328 the pseudo network adapter passes the DHCP response to the virtual network adapter interface, for example indicating an IP address received from the tunnel server in the client data field of the key exchange/authentication RESPONSE frame. The virtual network adapter interface then indicates a received packet to the TCP/IP protocol stack. The TCP/IP protocol stack then processes the DHCP response as if it had been received over an actual physical network.

At step 334 the pseudo network adapter encrypts the message using an encryption engine such that only the receiver is capable of decrypting and reading the message. At step 336 the pseudo network adapter encapsulates the encrypted message into a tunnel data frame. At step 338 the pseudo network adapter transmits the tunnel data frame through the tunnel connection using the TCP/IP protocol stack.

Fig. 18 is a flow chart showing steps performed by an example embodiment of a pseudo network adapter during packet receipt, such as in the receive path 296 of Fig. 14.

At step 350, the pseudo network adapter is notified that a packet has been received over the tunnel connection. At step 352 the pseudo network adapter decapsulates the received message by removing the header fields of the tunnel data frame. At step 354 the pseudo network adapter decrypts the decapsulated datagram from the tunnel data frame. At step 356, in an example embodiment, the pseudo network adapter forms an Ethernet packet from the decapsulated message. At step 358 the pseudo network adapter indicates that an Ethernet packet has been received to the TCP/IP protocol stack through the virtual network adapter interface. This causes the TCP/IP protocol stack to behave as if it had received an Ethernet packet from an actual Ethernet adapter.

Fig. 19 shows the data flow within the transmit path in an example embodiment of a pseudo network adapter. At step 1 370, an application submits data to be

transmitted to the TCP protocol layer 372 within the TCP/IP protocol stack. The application uses a conventional socket interface to the TCP protocol layer 372 to pass the data, and indicates the destination IP address the data is to be transmitted to. The TCP protocol layer 372 then passes the data to the IP protocol layer 374 within the TCP/IP protocol stack. At step 2 376, the TCP/IP protocol stack refers to the routing table 378 to determine which network interface should be used to reach the destination IP address.

Because in the example the destination IP address is of a node reachable through the virtual private network, the IP layer 374 determines from the routing table 378 that the destination IP address is reachable through pseudo network adapter. Accordingly at step 3 380 the TCP/IP protocol stack passes a packet containing the data to the pseudo network adapter 382.

At step 4 384, the pseudo network adapter 382 encrypts the data packets and encapsulates them into tunnel data frames.

The pseudo network adapter 382 then passes the tunnel data frames packets back to the TCP protocol layer 372 within the TCP/IP protocol stack through a conventional socket interface to the tunnel connection with the first node in the tunnel path.

The TCP protocol layer 372 then forms a TCP layer packet for each tunnel data frame, having the tunnel data frame as its data. The TCP frames are passed to the IP layer 374. At step 5 386 the routing table 378 is again searched, and this time the destination IP address is the IP address associated with the physical network adapter on the tunnel server, and accordingly is determined to be reachable over the physical network adapter 390. Accordingly at step 6 388 the device driver 390 for the physical network adapter is called to pass the packets to the physical network adapter. At step 7 392 the physical network adapter transmits the data onto the physical network 394.

Fig. 20 is a data flow diagram showing data flow in an example embodiment of packet receipt involving a pseudo network adapter. At step 1 410 data arrives over the physical network 412 and is received by the physical network adapter and passed to the physical network driver 414. The physical network driver 414 passes the data at step 2 418 through the IP layer 420 and TCP layer 422 to the pseudo network adapter 426 at step 3 424, for example through a conventional socket interface. At step 4 428 the pseudo network adapter 426 decrypts and decapsulates the received data and passes it back to the IP layer of the TCP/IP protocol stack, for example through the TDI (Transport Layer Dependent Interface API) of the TCP/IP stack. The data is then passed through the TCP/IP protocol stack and to the user associated with the destination IP address in the decapsulated datagrams at step 5 430.

Fig. 21 shows data flow in an example embodiment of packet transmission involving a pseudo network adapter. Fig. 21 shows an example embodiment for use

on a Microsoft™ Windows 95™ PC platform. In Fig. 21 a user application 450 passes unencrypted data to an interface into the TCP layer of the TCP/IP protocol, for example the WinSock API 452. The user indicates a destination IP address associated with a node reachable through a virtual private network accessible through the pseudo network adapter.

The TCP layer 454 passes the data to the IP layer 456, which in turn passes the data to the Network Device Interface Specification Media Access Control (NDIS MAC) interface 458. The pseudo network adapter 459 has previously registered with the routing layer (IP) that it is able to reach a gateway address associated with the destination IP address for the user data. Accordingly the IP layer uses the NDIS MAC layer interface to invoke the virtual device driver interface 460 to the pseudo network adapter 459. The pseudo network adapter 459 includes a virtual device driver interface 460, an ARP server emulator 462, and a DHCP server emulator 464.

In the example embodiment of Fig. 19, the pseudo network adapter 459 passes the data to a tunnel application program 466. The tunnel application program 466 encrypts the IP packet received from the IP layer and encapsulates it into a tunnel data frame. The tunnel application then passes the tunnel data frame including the encrypted data to the WinSock interface 452, indicating a destination IP address of the remote tunnel end point. The tunnel data frame is then passed through the TCP layer 454, IP layer 456, NDIS MAC layer interface 458, and physical layer 468, and transmitted on the network 470. Since the resulting packets do not contain a destination IP address which the pseudo network adapter has registered to convey, these packets will not be diverted to the pseudo network adapter.

Fig. 22 is a data flow diagram showing data flow in an example embodiment of packet transmission involving a pseudo network adapter. The embodiment shown in Fig. 22 is for use on a UNIX platform. In Fig. 20 a user application 472 passes unencrypted data to a socket interface to the TCP/IP protocol stack in the UNIX socket layer 474, indicating a destination IP address of a node reachable through the virtual private network.

The UNIX socket layer 474 passes the data through the TCP layer 476 and the IP layer 478. The pseudo network adapter 480 has previously registered with the routing layer (IP) that it is able to reach a gateway associated with the destination IP address for the user data. Accordingly the IP layer 478 invokes the virtual device driver interface 482 to the pseudo network adapter 480. The IP layer 478 passes the data to the pseudo network adapter 480. The pseudo network adapter 480 includes a virtual device driver interface 482, and a DHCP server emulator 484.

In the example embodiment of Fig. 22, the pseudo network adapter 480 passes IP datagrams to be transmitted to a UNIX Daemon 486 associated with the tunnel connection. The UNIX Daemon 486 encrypts the IP

packet(s) received from the IP layer 478 and encapsulates them into tunnel data frames. The UNIX Daemon 486 then passes the tunnel data frames to the UNIX socket layer 474, through a socket associated with the tunnel connection. The tunnel data frames are then processed by the TCP layer 476, IP layer 478, data link layer 488, and physical layer 490 to be transmitted on the network 492. Since the resulting packets are not addressed to an IP address which the pseudo network adapter 480 has registered to convey, the packets will not be diverted to the pseudo network adapter 480.

Fig. 23 is a flow chart showing steps to initialize an example embodiment of a virtual private network. The steps shown in Fig. 23 are performed for example in the tunnel client 247 as shown in Fig. 14. At step 500 a tunnel application program executing in the tunnel client sends a tunnel relay frame to the tunnel server. At step 502 the tunnel application program sends a tunnel key exchange/authentication REQUEST frame to the tunnel server. The tunnel application in the tunnel server ignores the contents of the client data field in the tunnel key exchange/authentication REQUEST frame. The tunnel application in the tunnel server fills in the client data field in the tunnel key exchange/authentication RESPONSE frame with Dynamic Host Configuration Protocol (DHCP) information, for example including the following information in standard DHCP format:

- 1) IP Address for tunnel client Pseudo Network Adapter
- 2) IP Address for tunnel server Pseudo Network Adapter
- 3) Routes to nodes on the private network physically connected to the tunnel server which are to be reachable over the tunnel connection.

At step 504 the tunnel application receives a tunnel key exchange/authentication RESPONSE frame from the tunnel server. The client data field 508 in the tunnel connection response is made available to the pseudo network adapter in the tunnel client. The tunnel application in the tunnel client tells the TCP/IP stack that the pseudo network adapter in the tunnel client is active. The pseudo network adapter in the tunnel client is active and ready to be initialized at step 510.

The tunnel client system is configured such that it must obtain an IP address for the tunnel client pseudo network adapter dynamically. Therefore the TCP/IP stack in the tunnel client broadcasts a DHCP request packet through the pseudo network adapter. Accordingly, at step 512 the pseudo network adapter in the client receives a conventional DHCP request packet from the TCP/IP stack requesting a dynamically allocated IP address to associate with the pseudo network adapter. The pseudo network adapter passes the DHCP request packet to the DHCP server emulator within the pseudo network adapter, which forms a DHCP response based on the client data 508 received from the tunnel applica-

tion. The DHCP response includes the IP address for the client pseudo adapter provided by the tunnel server in the client data. At step 514 the pseudo network adapter passes the DHCP response to the TCP/IP stack.

At step 520, the tunnel application modifies the routing tables within the tunnel client TCP/IP stack to indicate that the routes to the nodes attached to the private network to which the tunnel server is attached all are reachable only through the pseudo network adapter in the tunnel server. The IP address of the pseudo network adapter in the tunnel server provided in the client data is in this way specified as a gateway to the nodes on the private network to which the tunnel server is attached. In this way those remote nodes are viewed by the TCP/IP stack as being reachable via the virtual private network through the client pseudo network adapter.

At step 516 the pseudo network adapter in the tunnel client receives an ARP request for a physical address associated with the IP address of the pseudo network adapter in the tunnel server. The pseudo network adapter passes the ARP request to the ARP server emulator, which forms an ARP reply indicating a reserved physical address to be associated with the IP address of the pseudo network adapter in the tunnel server. At step 518 the pseudo network adapter passes the ARP response to the TCP/IP stack in the tunnel client. In response to the ARP response, the TCP/IP stack determines that packets addressed to any node on the virtual private network must be initially transmitted through the pseudo network adapter.

In an example embodiment the present system reserves two physical addresses to be associated with the pseudo network adapter in the client and the pseudo network adapter in the server respectively. These reserved physical addresses are used in responses to ARP requests passed through the pseudo network adapter for physical addresses corresponding to the IP addresses for the pseudo network adapter in the client and the pseudo network adapter in the server respectively. The reserved physical addresses should have a high likelihood of not being used in any actual network interface.

While the invention has been described with reference to specific example embodiments, the description is not meant to be construed in a limiting sense. Various modifications of the disclosed embodiments, as well as other embodiments of the invention, will be apparent to persons skilled in the art upon reference to this description. Specifically, while various embodiments have been described using the TCP/IP protocol stack, the invention may advantageously be applied where other communications protocols are used. Also, while various flow charts have shown steps performed in an example order, various implementations may use altered orders of step in order to apply the invention. And further, while certain specific software and/or hardware platforms

have been used in the description, the invention may be applied on other platforms with similar advantage. It is therefore contemplated that the appended claims will cover any such modifications or embodiments which fall within the scope of the invention.

## Claims

1. A pseudo network adapter providing a virtual private network, comprising:

an interface for capturing packets from a local communications protocol stack for transmission on said virtual private network, said interface appearing to said local communications protocol stack as a network adapter device driver for a network adapter connected to said virtual private network;

a first server emulator, providing a first reply packet responsive to a first request packet captured by said interface for capturing packets from said local communications protocol stack for transmission on said virtual private network, said first request packet requesting a network layer address for said pseudo network adapter, said first reply indicating a network layer address for said pseudo network adapter; and a second server emulator, providing a second reply packet responsive to an second request packet captured by said interface for capturing packets from said local communications protocol stack for transmission on said virtual private network, said second request packet requesting a physical address corresponding to a network layer address of a second pseudo network adapter, said second pseudo network adapter located on a remote server node, said second reply indicating a predetermined, reserved physical address.

2. The pseudo network adapter of claim 1, further comprising a means for indicating to said local communications protocol stack that said predetermined, reserved physical address is reachable through said pseudo network adapter, wherein said means for indicating modifies a data structure in said local communications protocol stack indicating which nodes or networks are reachable through each network interface of the local system.
3. The pseudo network adapter of claim 1, further comprising a means for indicating to said local communications protocol stack that one or more nodes on a remote private network connected to said remote server node are reachable through a gateway node equal to said second pseudo network adapter on said remote server node.

4. The pseudo network adapter of claim 1, further comprising:
- a transmit path for processing data packets captured by said interface for capturing packets from said local communications protocol stack for transmission on said virtual private network; an encryption engine, within said transmit path, for encrypting said data packets;
  - an encapsulation engine, within said transmit path, for encapsulating said encrypted data packets into tunnel data frames; and
  - a means for passing said tunnel data frames back to said local communications protocol stack for transmission to a physical network adapter on said remote server node.
- 5.
5. The pseudo network adapter of claim 4, wherein said transmit path further includes means for storing a digest value in a digest field in each of said tunnel data frames, said digest value equal to an output of a keyed hash function applied to said data packet encapsulated within said tunnel data frame concatenated with a counter value equal to a total number of tunnel data frames previously transmitted to said remote server node.
- 20
- 25
6. The pseudo network adapter of claim 4, wherein said transmit path further includes means for processing an Ethernet header in each one of said captured data packets, said processing of said Ethernet header including removing said Ethernet header.
- 30
7. The pseudo network adapter of claim 1, further comprising:
- an interface into a transport layer of said local communications protocol stack for capturing received data packets from said remote server node.
- 40
8. The pseudo network adapter of claim 7, further comprising:
- a receive path for processing received data packets captured by said interface into said transport layer of said local communications protocol stack for capturing received data packets from said remote server node;
  - an decapsulation engine, within said receive path, for decapsulating said received data packets by removing a tunnel frame header;
  - an decryption engine, within said receive path, for decrypting said received data packets; and
  - a means for passing said received data packets back to said local communications protocol stack for delivery to a user.
- 45
- 50
- 55
9. A method for providing a pseudo network adapter for a virtual private network, comprising the steps of:
- capturing packets from a local communications protocol stack for transmission on said virtual private network, said capturing through an interface appearing to said local communications stack as a network adapter device driver for a network adapter connected to said virtual private network;
  - issuing a first reply packet responsive to a first request packet captured by said interface for capturing packets from said local communications protocol stack for transmission on said virtual private network, said first request packet requesting a network layer address for said pseudo network adapter, said first reply indicating a network layer address for said pseudo network adapter; and
  - issuing a second reply packet responsive to a second request packet captured by said interface for capturing packets from said local communications protocol stack for transmission on said virtual private network, said second request packet requesting a physical address corresponding to a network layer address of a second pseudo network adapter, said second pseudo network adapter located on a remote server node, said ARP Reply indicating a predetermined, reserved physical address.
10. The method of claim 9, further comprising indicating to said local communications protocol stack that said predetermined, reserved physical address is reachable through said pseudo network adapter, wherein said step of indicating to said local communications protocol stack modifies a data structure in said local communications protocol stack indicating which nodes or networks are reachable through each network interface of the local system.
11. The method of claim 9, further comprising indicating to said local communications protocol stack that one or more nodes on a remote private network connected to said remote server node are reachable through a gateway node equal to said second pseudo network adapter on said remote server node, wherein said step of indicating to said local communications protocol stack that one or more nodes on said remote private network connected to said remote server node are reachable through a gateway node equal to said second pseudo network adapter on said remote server node modifies a network layer routing table in said local communications protocol stack.
12. The method of claim 9, further comprising:

processing data packets captured by said interface for capturing packets from said local communications protocol stack for transmission on said virtual private network in a transmit data path;

5

encrypting said data packets in an encryption engine, within said transmit path;

encapsulating said encrypted data packets into tunnel data frames by an encapsulation engine, within said transmit path; and

10

passing said tunnel data frames back to said local communications protocol stack for transmission to a physical network adapter on said remote server node, wherein said transmit path further includes storing a digest value in a digest field in each of said tunnel data frames, said digest value equal to an output of a keyed hash function applied to said data packet encapsulated within said tunnel data frame concatenated with a counter value equal to a total number of tunnel data frames previously transmitted to said remote server node.

15

20

13. The method of claim 12, wherein said transmit path further includes processing an Ethernet header in each one of said captured data packets, said processing of said Ethernet header including removing said Ethernet header.

25

14. The method of claim 9, further comprising capturing received data packets from said remote server node through an interface into a transport layer of said local communications protocol stack, further comprising:

30

35

processing received data packets captured by said interface into said transport layer of said local communications protocol stack for capturing received data packets from said remote server node in a receive path;

40

decapsulating said received data packets by removing a tunnel frame header in a decapsulation engine, within said receive path;

decrypting said received data packets in a decryption engine within said receive path; and passing said received data frames packets back to said local communications protocol stack for delivery to a user.

45

15. The method of claim 9, wherein said network layer address for said pseudo network adapter and said predetermined, reserved physical address is communicated to said pseudo network adapter from said remote server node as client data in a connection response frame.

50

55

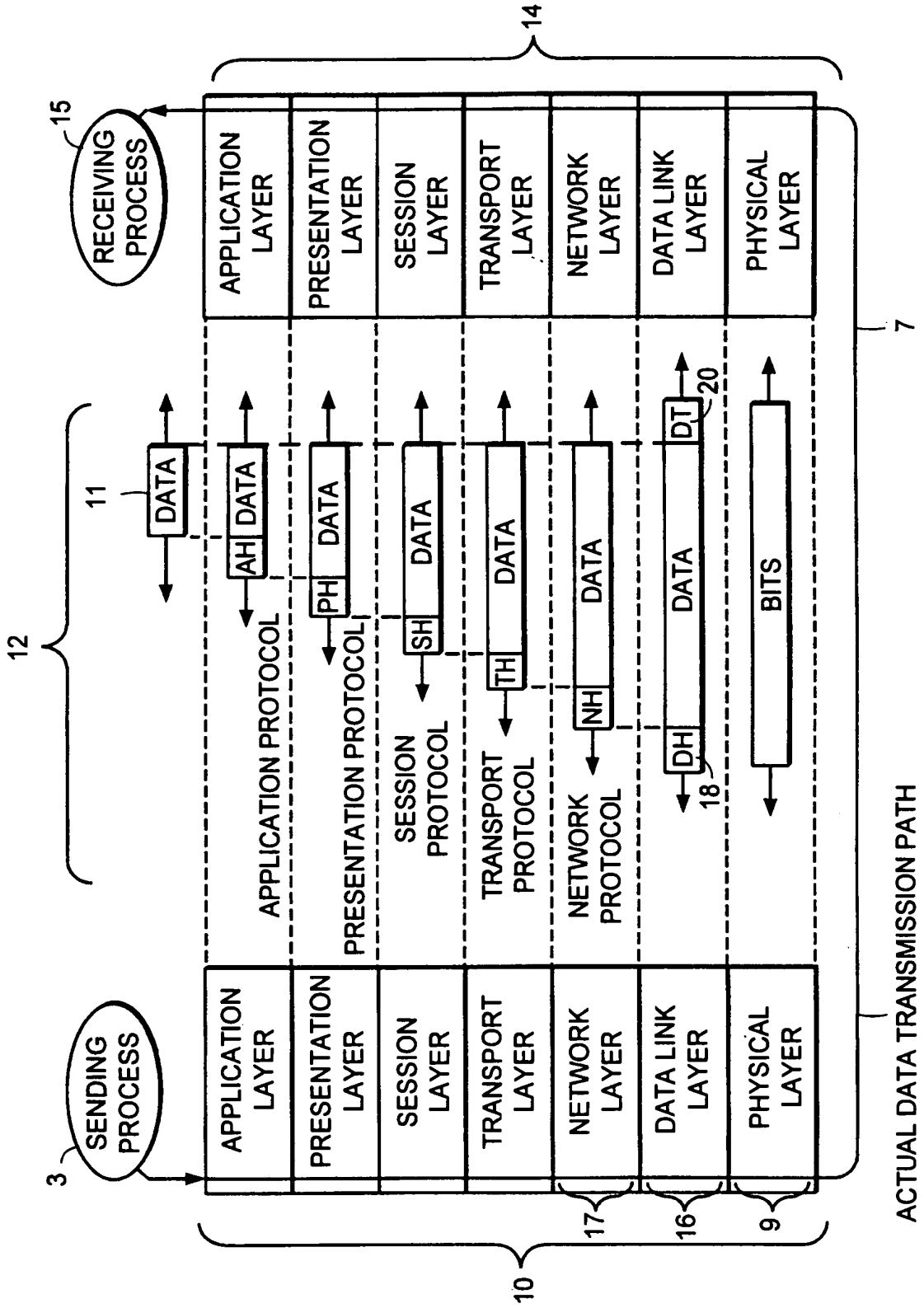


FIG. 1

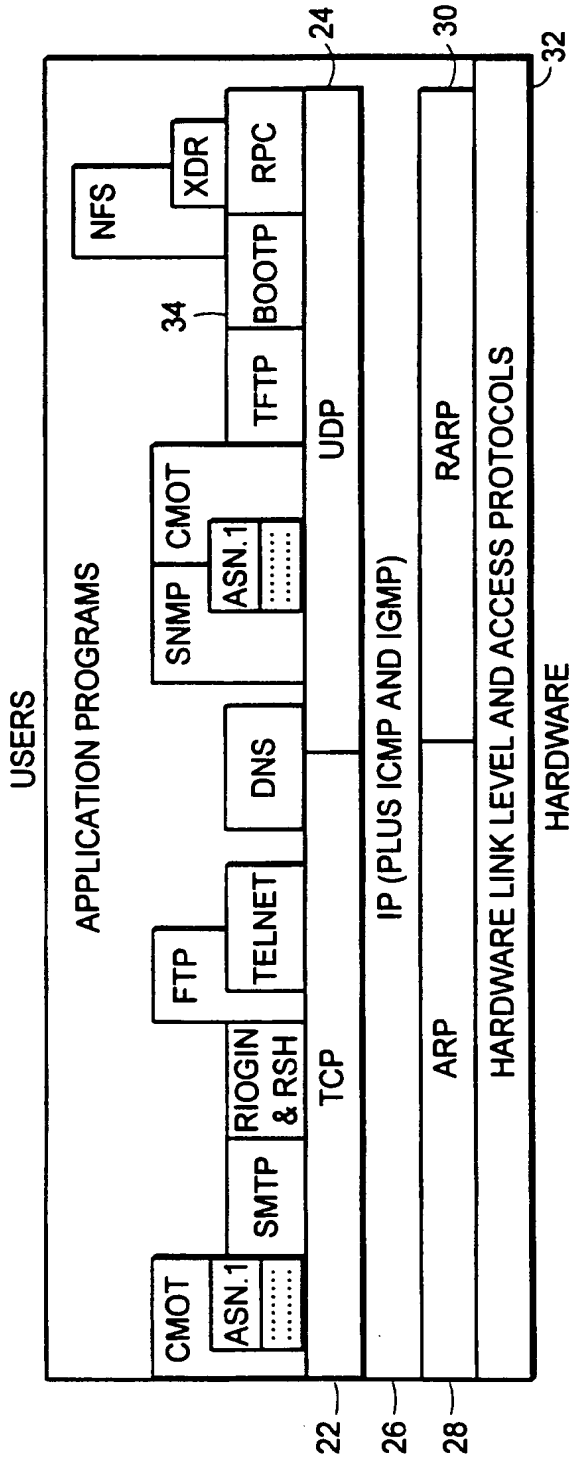


FIG. 2

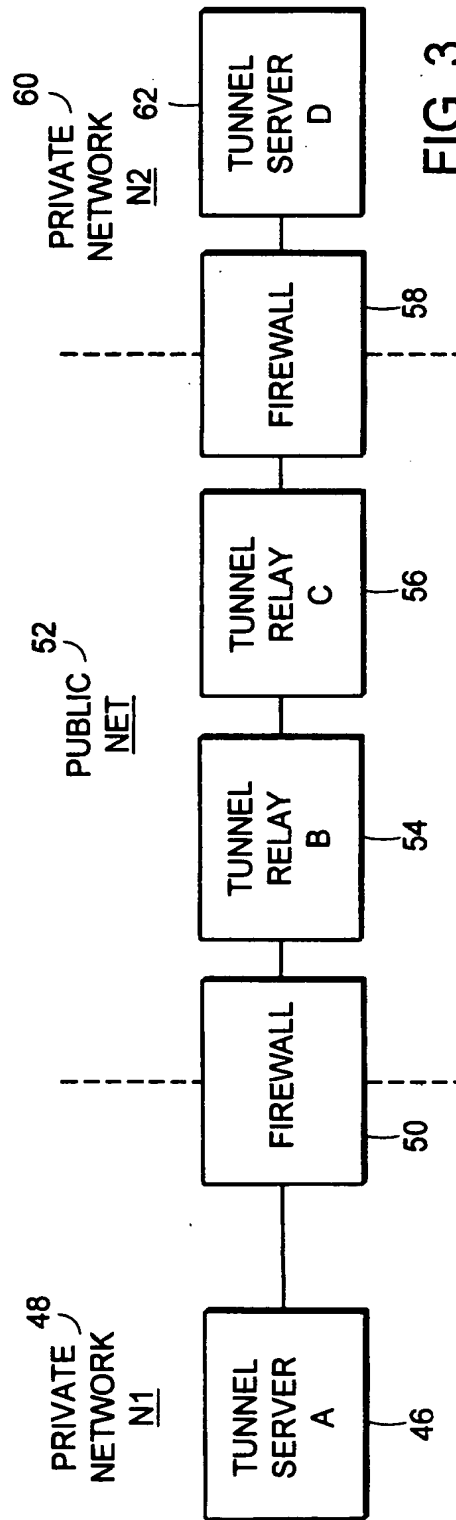


FIG. 3



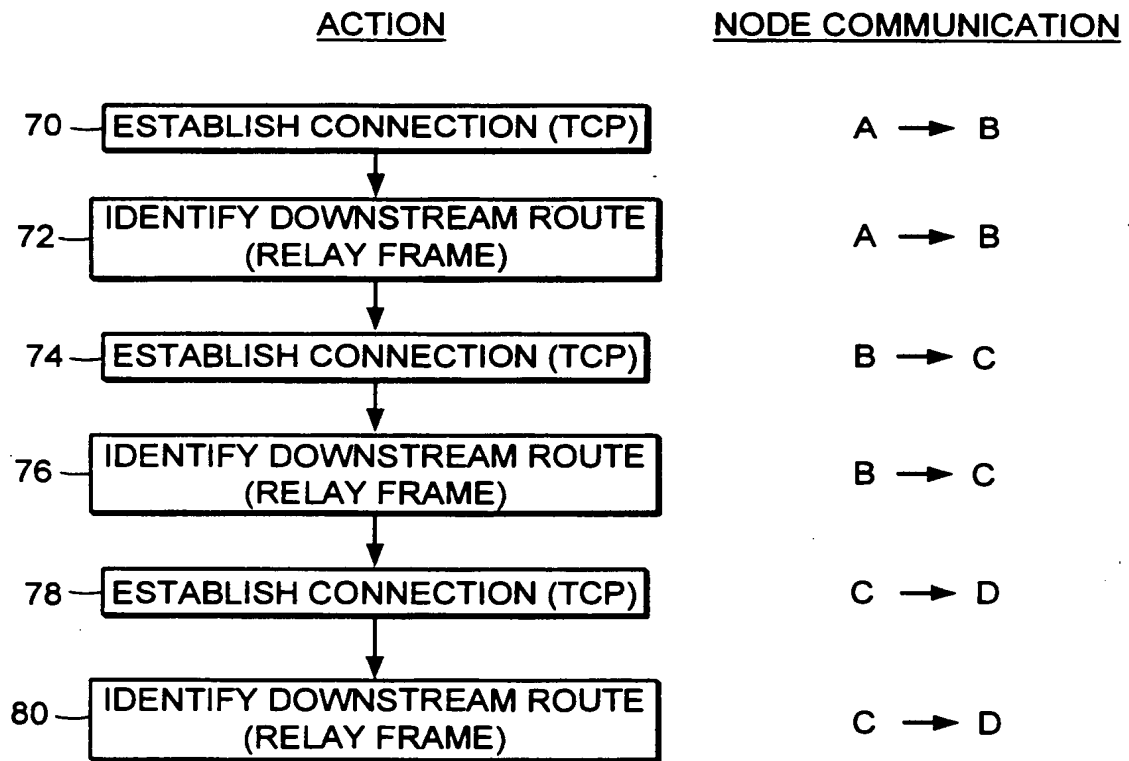


FIG. 4

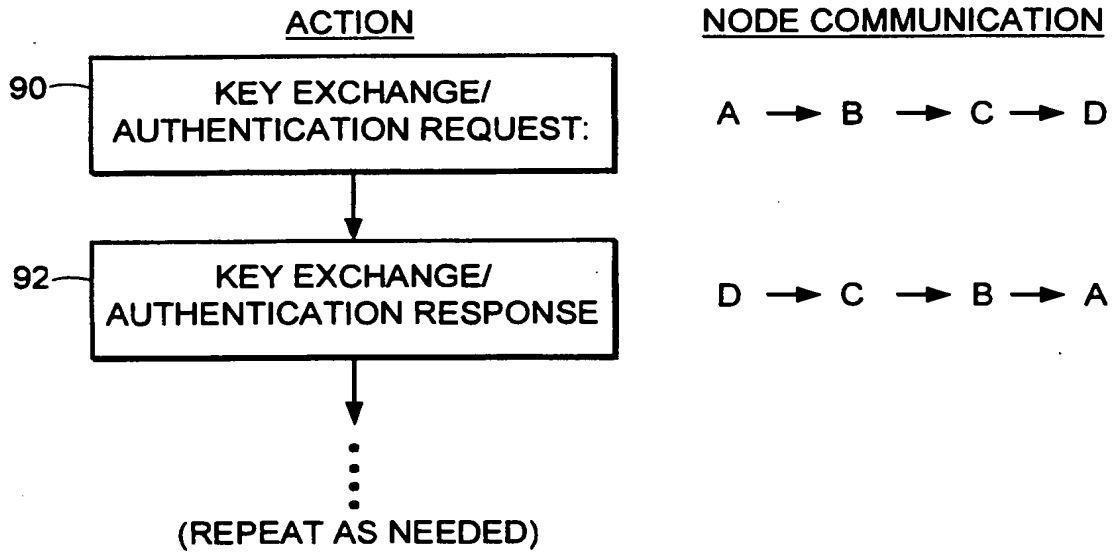


FIG. 5

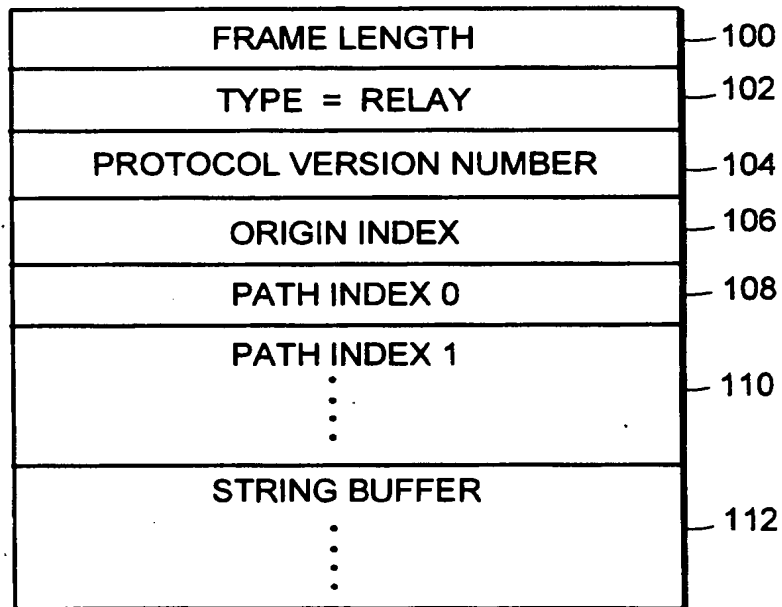


FIG. 6

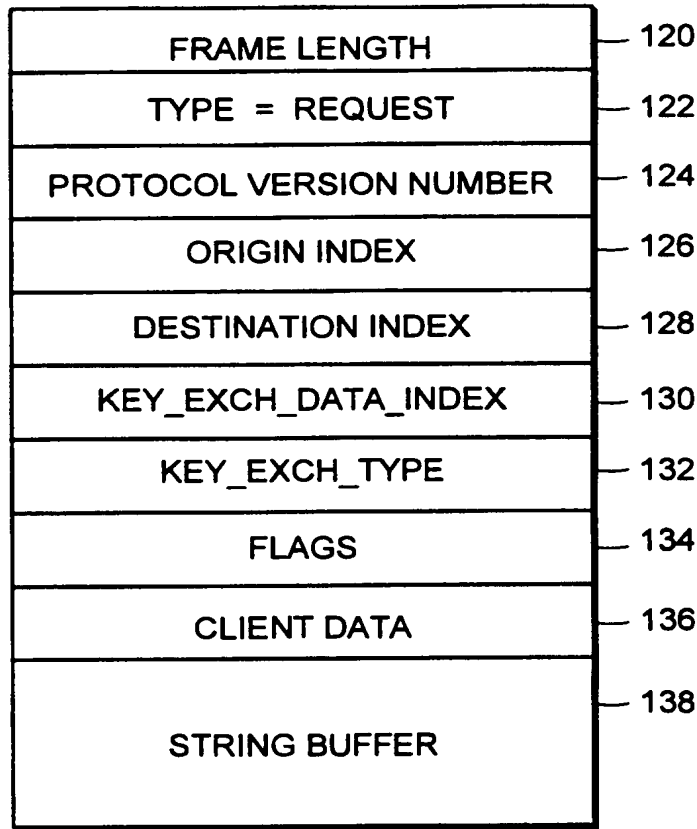


FIG. 7

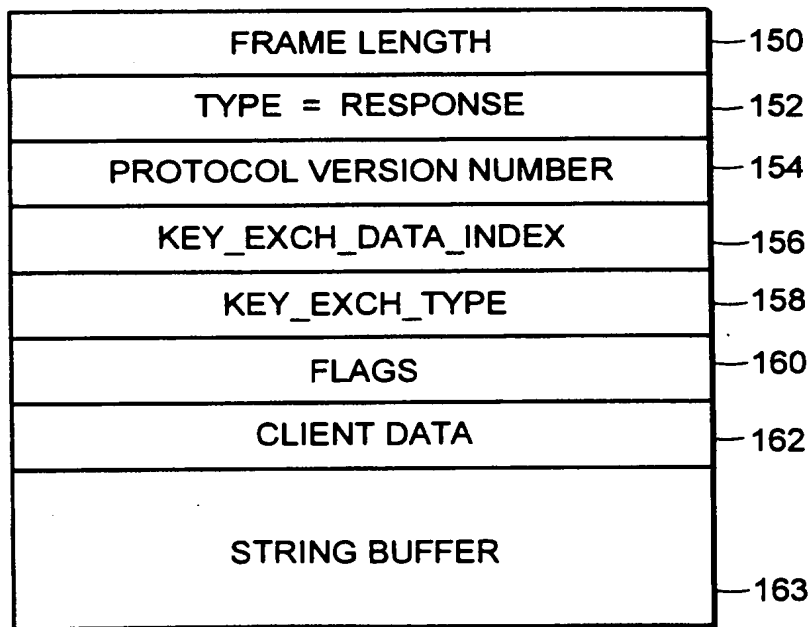


FIG. 8

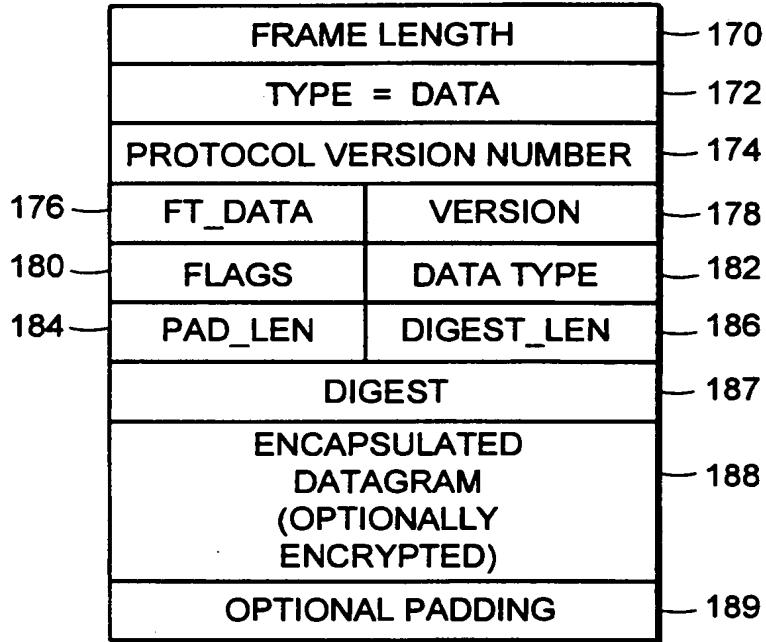


FIG. 9

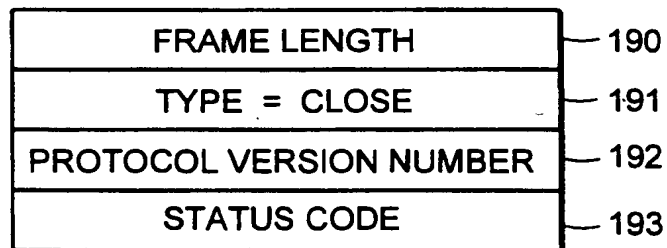


FIG. 10

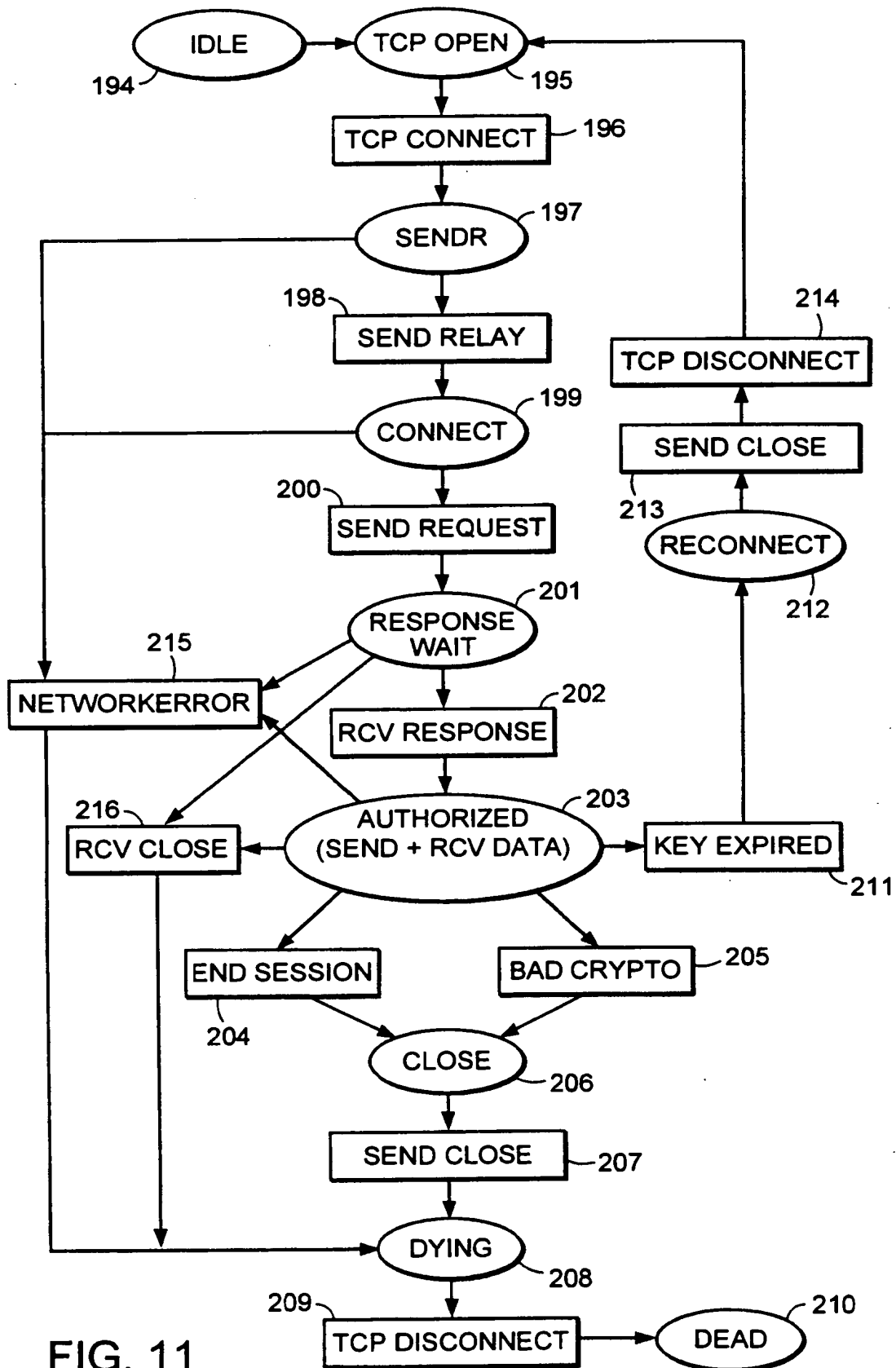


FIG. 11

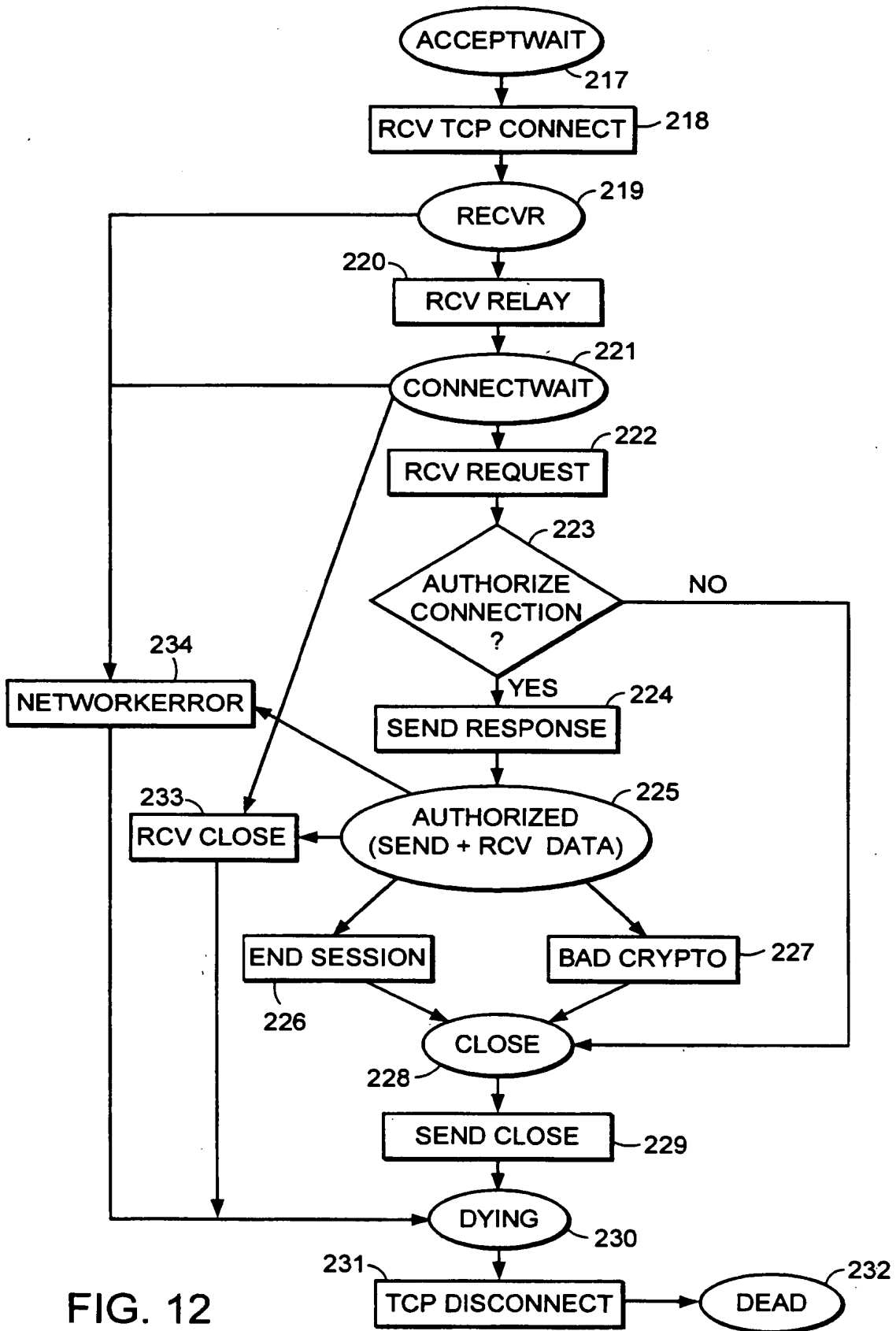


FIG. 12

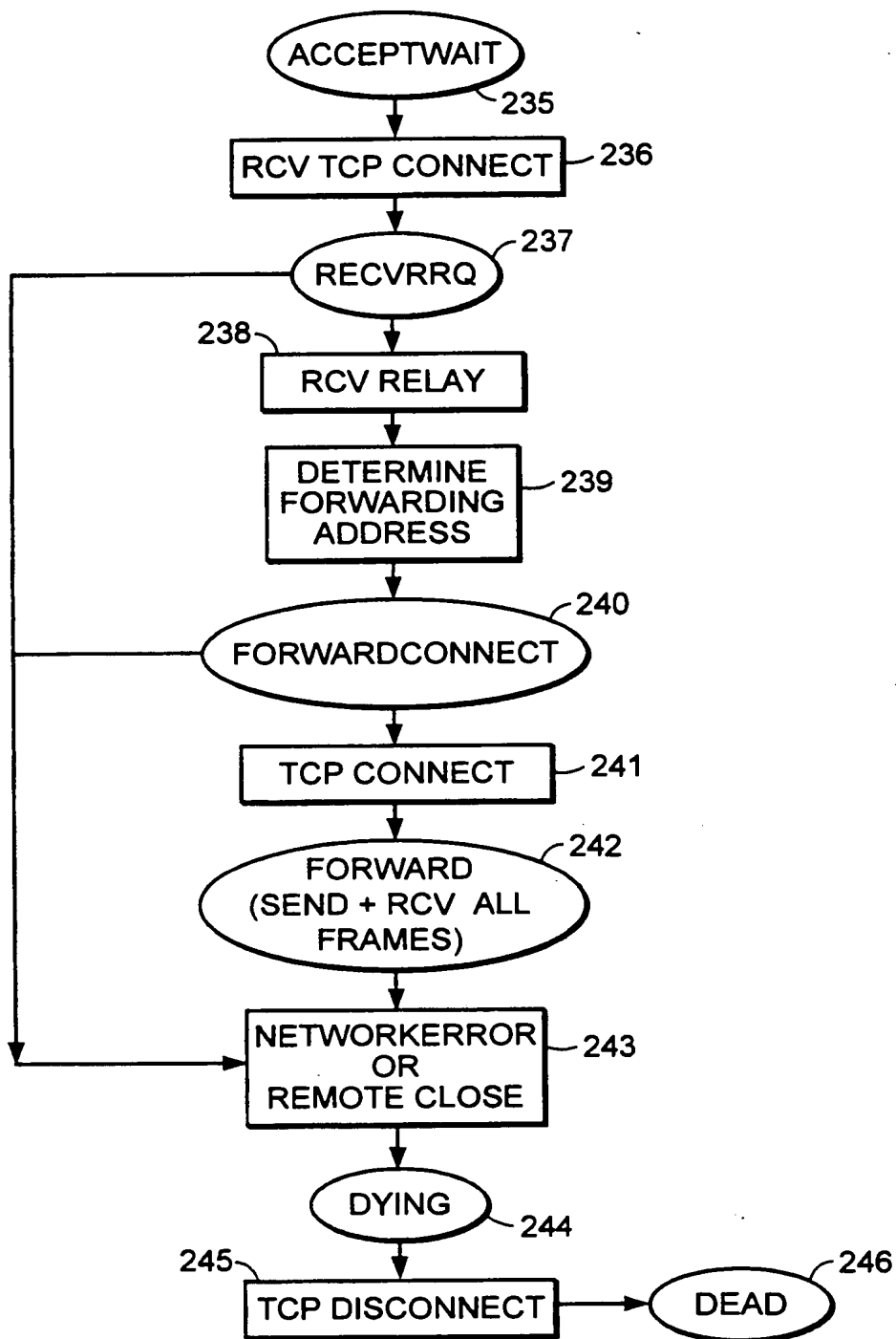


FIG. 13

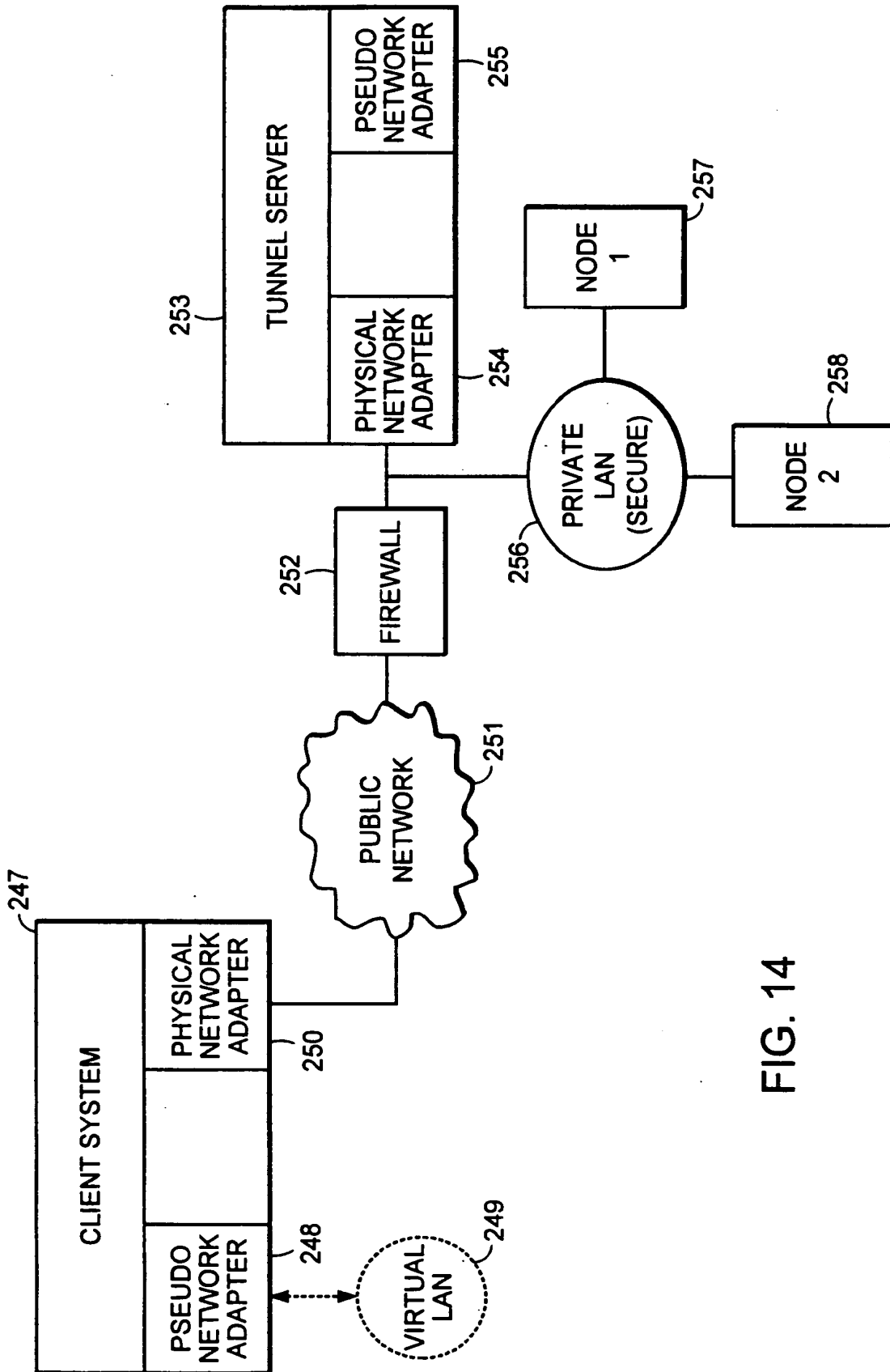


FIG. 14



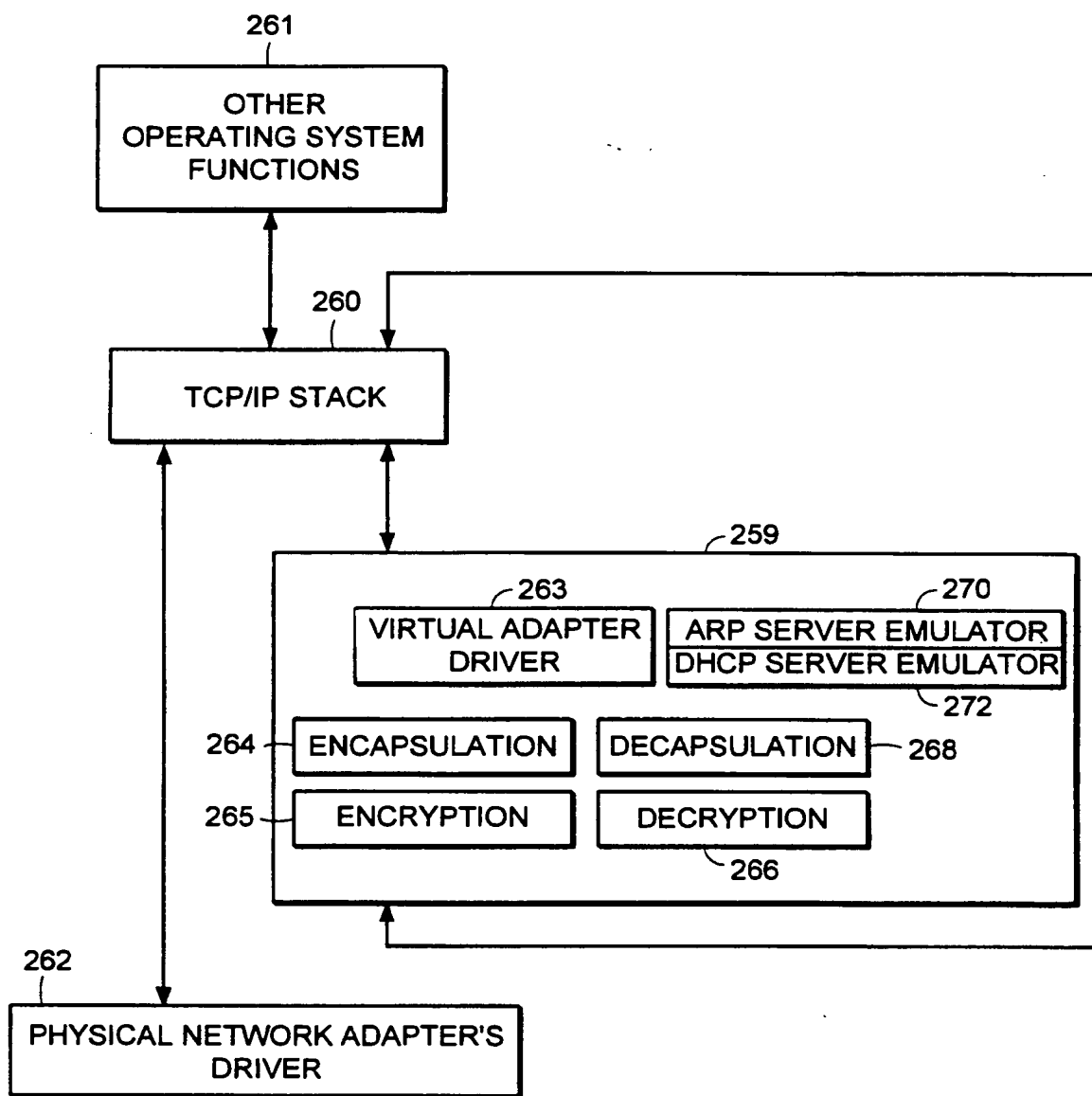


FIG. 15

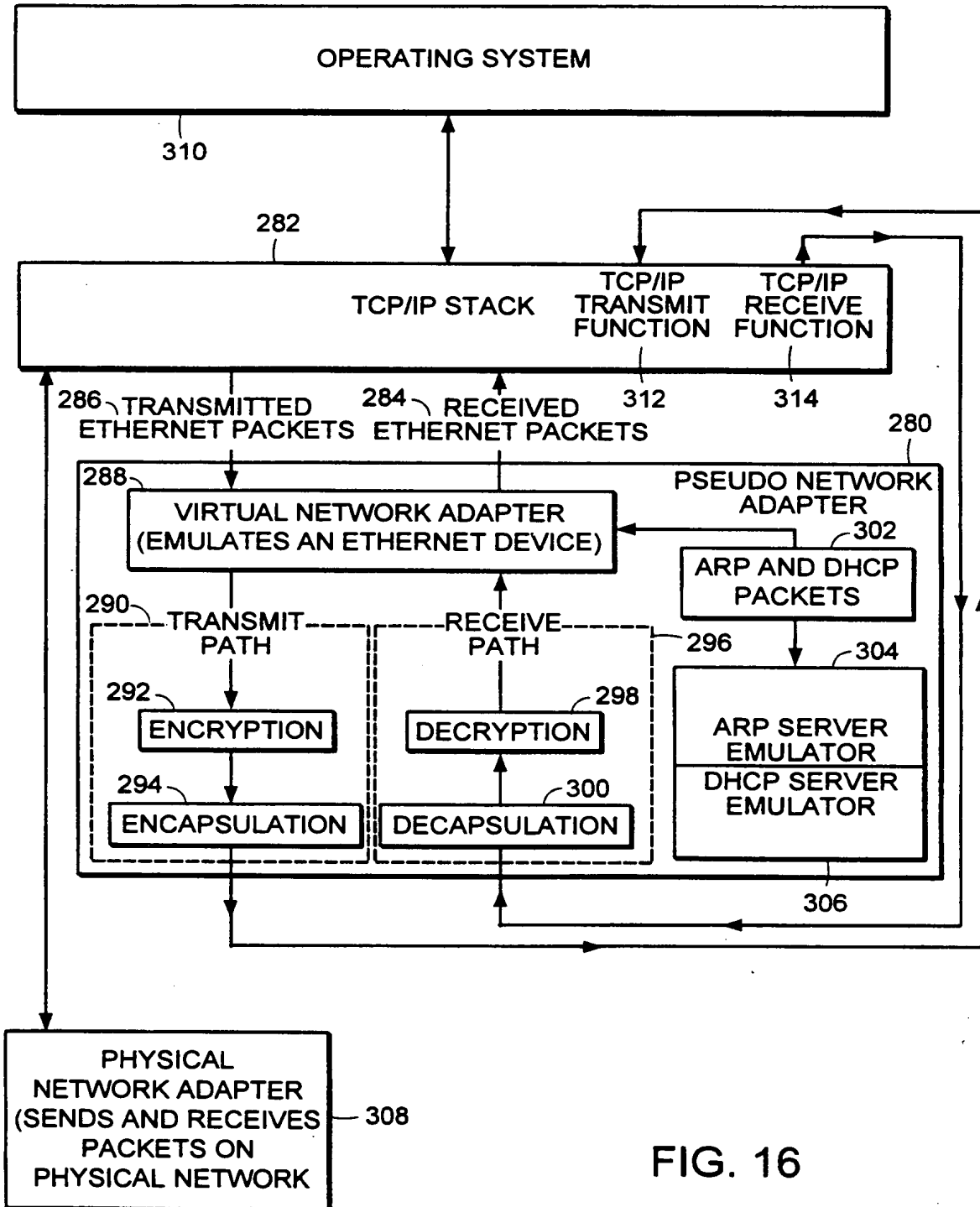


FIG. 16

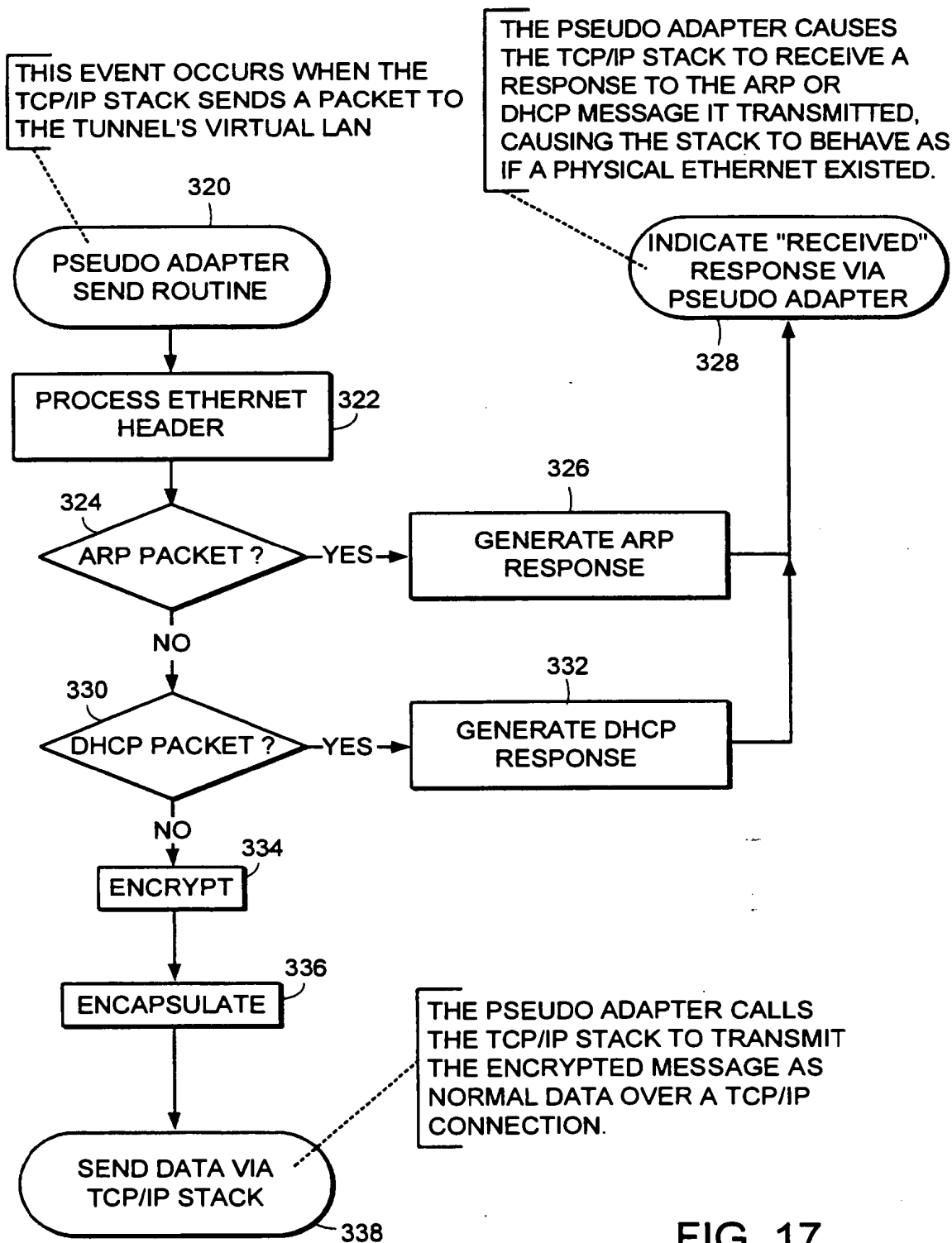


FIG. 17

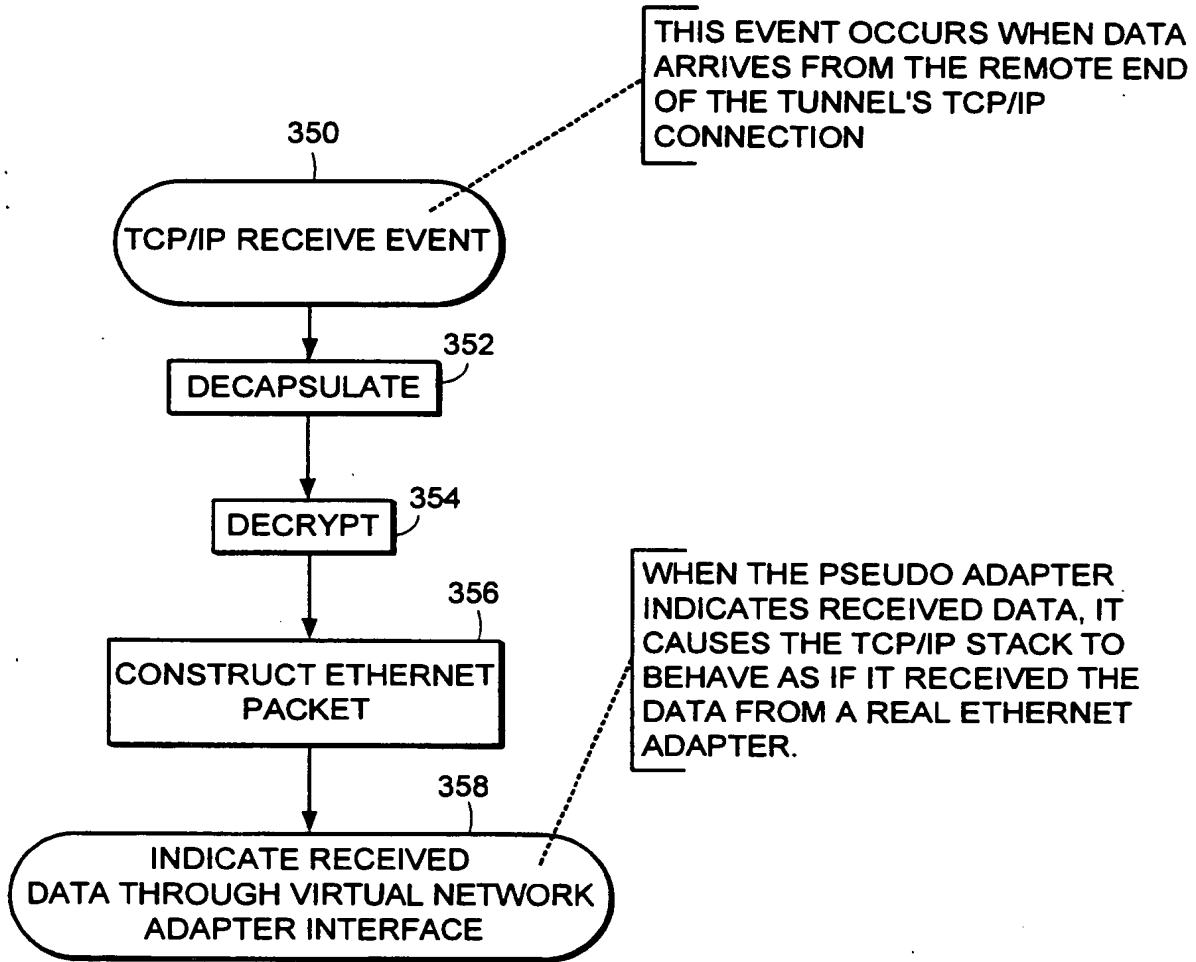


FIG. 18

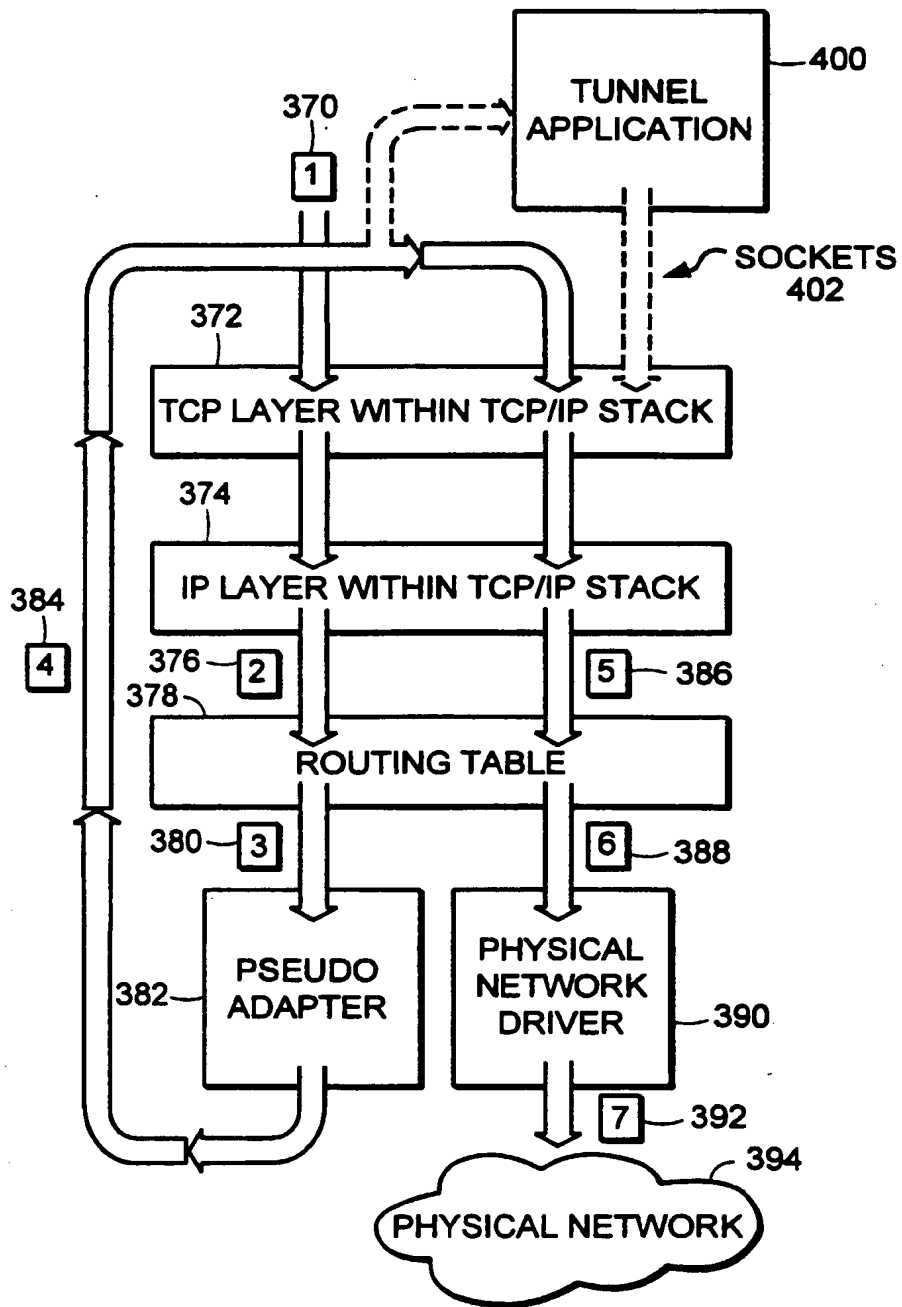


FIG. 19

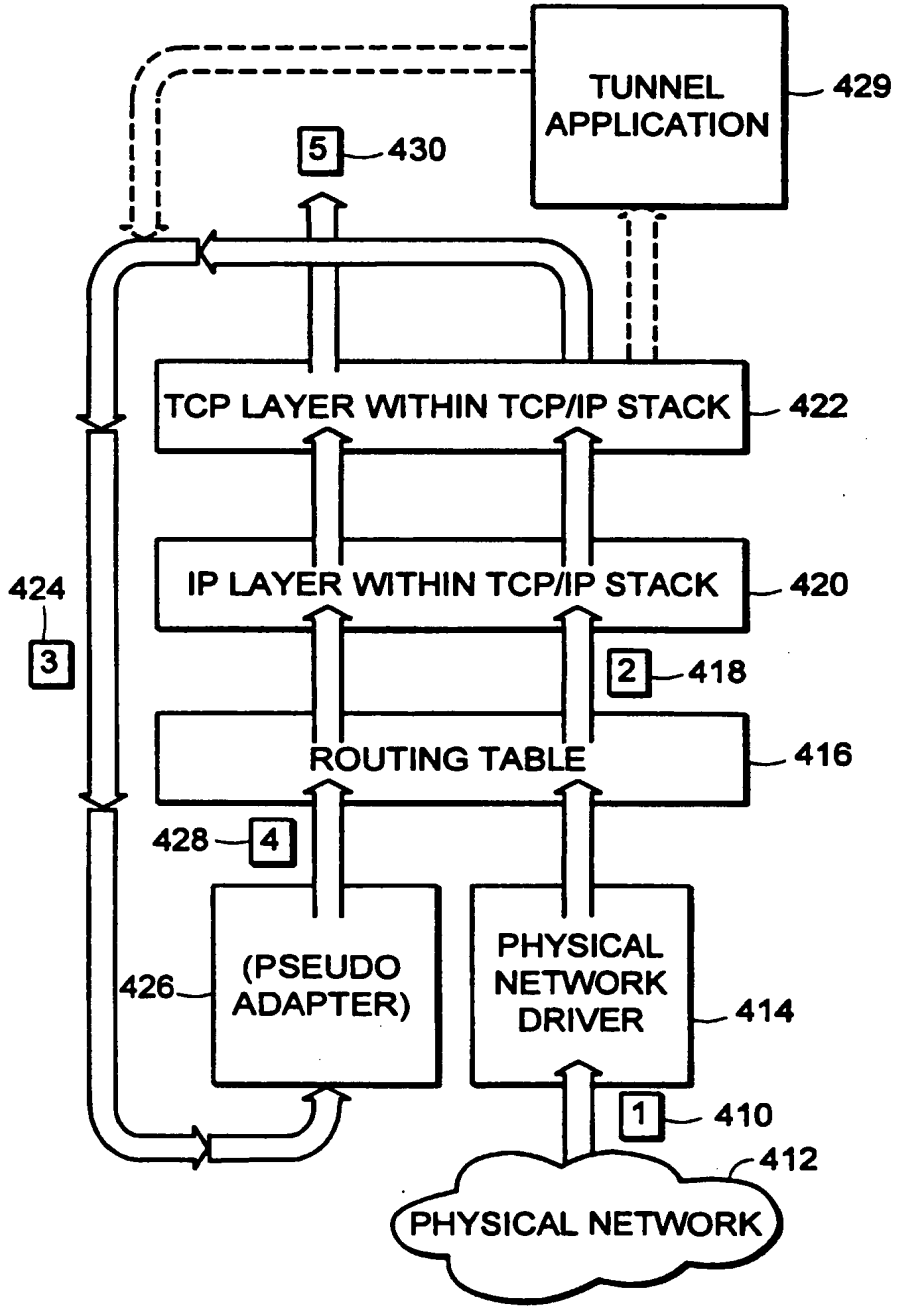


FIG. 20

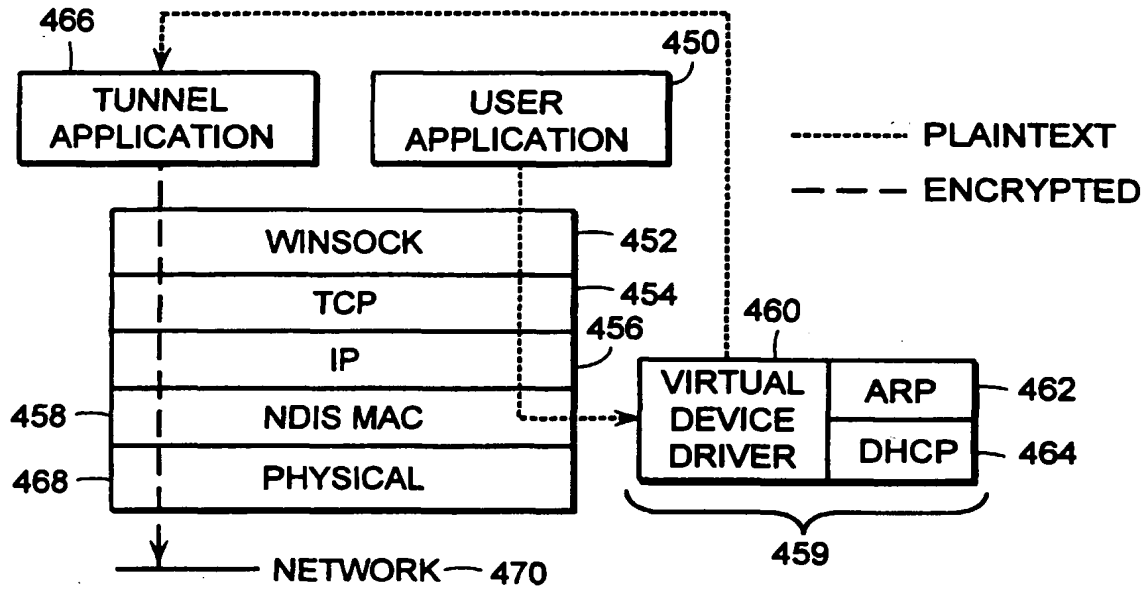


FIG. 21

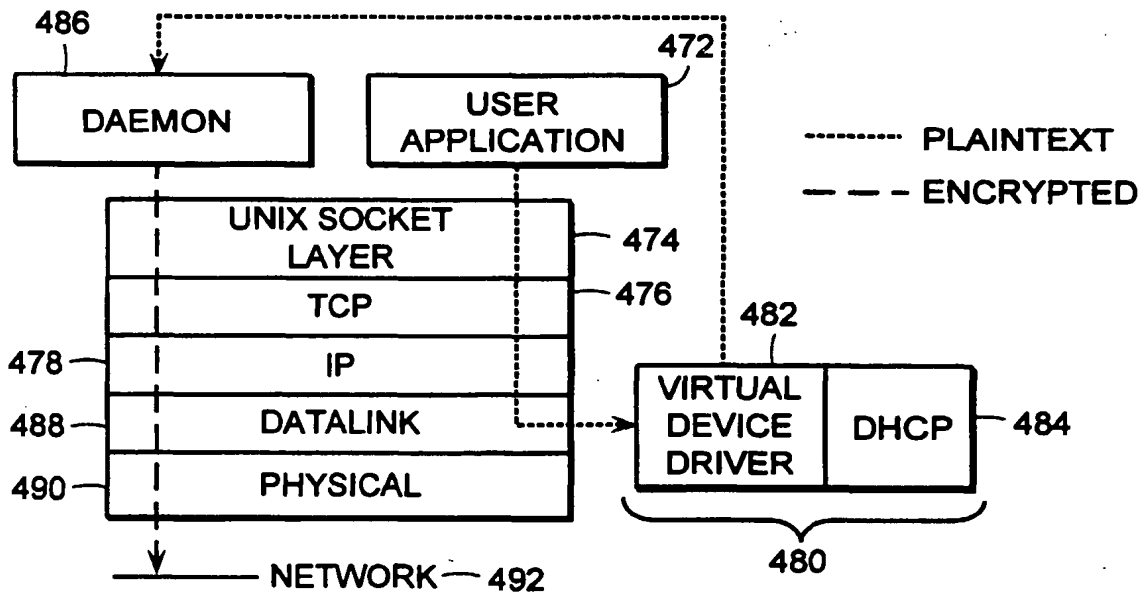


FIG. 22

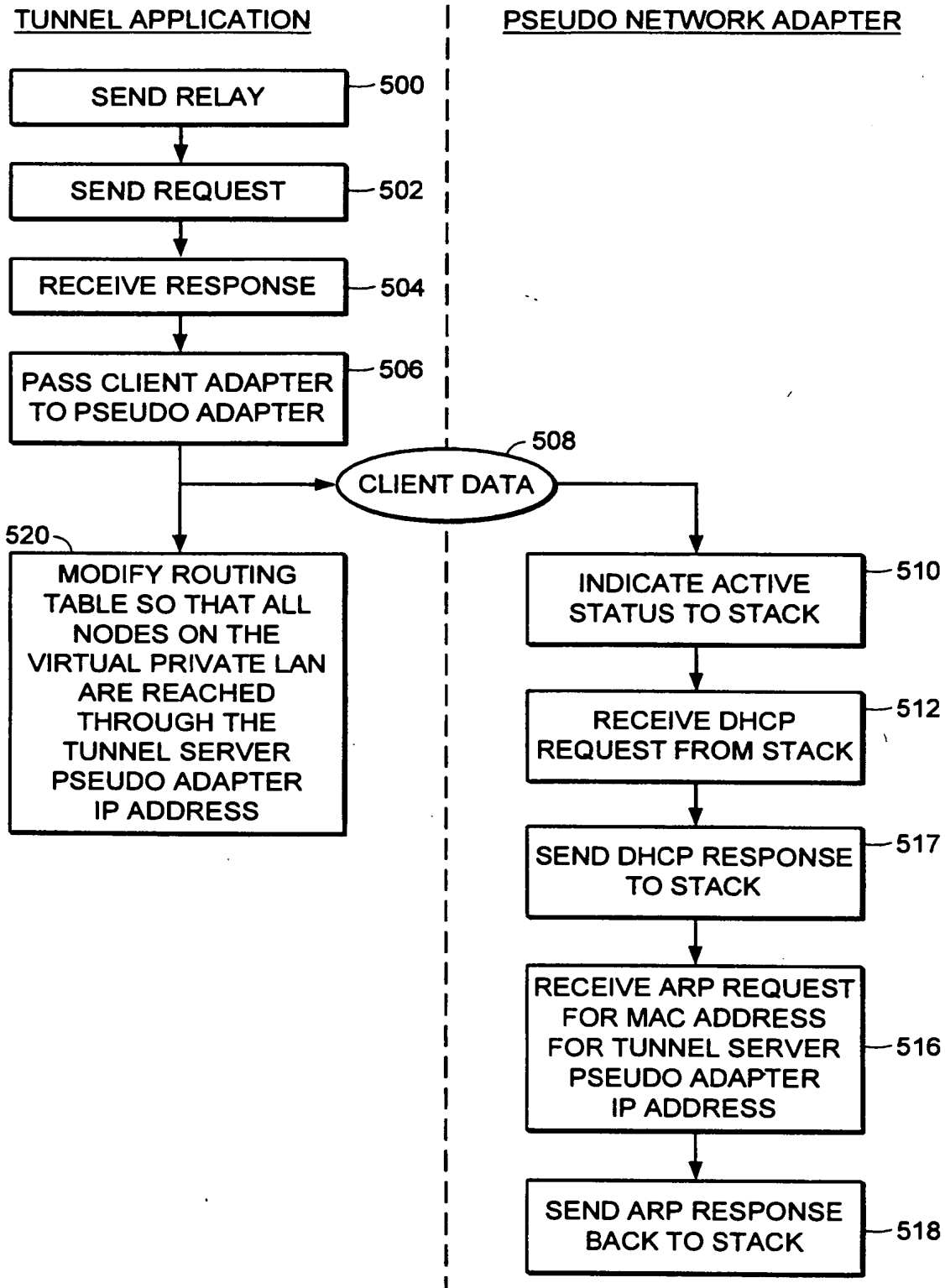


FIG. 23





19 BUNDESREPUBLIK  
DEUTSCHLAND



DEUTSCHES  
PATENT- UND  
MARKENAMT

# Offenlegungsschrift DE 199 24 575 A 1

51 Int. Cl.<sup>6</sup>:  
**H 04 L 29/06**  
H 04 L 12/22  
G 06 F 13/00  
G 06 F 12/14  
// H04L 9/00

21 Aktenzeichen: 199 24 575.4  
22 Anmeldetag: 28. 5. 99  
43 Offenlegungstag: 2. 12. 99

DE 199 24 575 A 1

30 Unionspriorität:  
087823 29. 05. 98 US

71 Anmelder:  
Sun Microsystems, Inc., Palo Alto, Calif., US

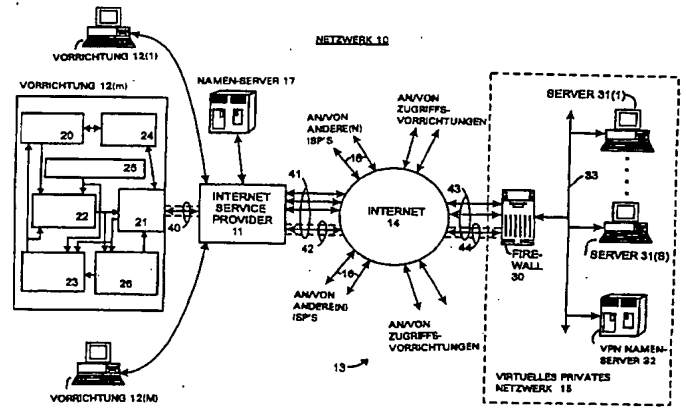
74 Vertreter:  
Samson & Partner, Patentanwälte, 80538 München

72 Erfinder:  
Provino, Joseph E., Cambridge, Mass., US

### Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

54 Kommunikationssystem und -Verfahren

57 Das erfindungsgemäße System umfaßt ein virtuelles privates Netzwerk (15) und eine externe Vorrichtung (12(m)), welche durch ein digitales Netzwerk (14) miteinander verbunden sind. Das virtuelle private Netzwerk (15) weist eine Firewall (30), wenigstens eine interne Vorrichtung (31(s)) und einen Namen-Server (32) auf, welche jeweils eine Netzwerkadresse besitzen. Die interne Vorrichtung (31(s)) besitzt auch eine Sekundäradresse, und der Namen-Server (32) ist derart konfiguriert, daß er eine Zuordnung zwischen der Sekundäradresse und der Netzwerkadresse bereitstellt. In Reaktion auf eine Anfrage von der externen Vorrichtung (12(m)) zum Aufbau einer Verbindung zur Firewall (30) übermittelt die Firewall (30) der externen Vorrichtung (12(m)) die Netzwerkadresse des Namen-Servers (32). In Reaktion auf eine Anfrage von einem Bediener oder ähnlichem, welche die Sekundäradresse der internen Vorrichtung (31(s)) enthält und einen Zugriff an die interne Vorrichtung (31(s)) anfordert, erzeugt die externe Vorrichtung (12(m)) eine Netzwerkadressen-Anfragennachricht zur Übertragung über die Verbindung an die Firewall (30), welche eine Auflösung der Netzwerkadresse, die der Sekundäradresse zugeordnet ist, anfordert. Die Firewall (30) übermittelt die Adressenauflösungsanfrage an den Namen-Server (32), und der Namen-Server (32) übermittelt die Netzwerkadresse, welche der Sekundäradresse zugeordnet ist, an die Firewall (30). Daraufhin stellt die Firewall (30) die Netzwerkadresse in einer ...



DE 199 24 575 A 1

## Beschreibung

Die Erfindung betrifft allgemein das Gebiet der digitalen Kommunikationssysteme und -verfahren, und insbesondere Systeme und Verfahren zum Vereinfachen der Kommunikation zwischen Vorrichtungen, welche mit öffentlichen Netzwerken verbunden sind, z. B. dem Internet, und Vorrichtungen, welche mit privaten Netzwerken verbunden sind.

Digitale Netzwerke wurden entwickelt, um die Übertragung von Information, welche auch Daten und Programme umfaßt, über digitale Computersysteme und andere Digitalvorrichtungen zu ermöglichen. Es wurde eine Vielzahl von Arten von Netzwerken entwickelt und realisiert, einschließlich sog. Fernverbindungsnetze (Wide-Area Networks, nachfolgend "WAN" genannt) und lokale Netzwerke (Local Area Networks, nachfolgend "LAN" genannt), welche eine Information unter Verwendung verschiedener Informationsübertragungsmethoden übermitteln. Im allgemeinen werden LANs innerhalb kleiner geographischer Bereiche realisiert, z. B. innerhalb eines einzelnen Bürogebäudes oder ähnlichem, zum Übertragen von Information innerhalb eines bestimmten Büros, einer Firma oder einer ähnlichen Art von Organisationseinheit. Andererseits werden WANs im allgemeinen auf relativ großen geographischen Bereichen realisiert und können verwendet werden, um Information sowohl zwischen LANs als auch zwischen Vorrichtungen, welche nicht mit LANs verbunden sind, zu übertragen. Derartige WANs umfassen auch öffentliche Netzwerke, z. B. das Internet, welche zur Informationsübertragung zwischen einer Anzahl von Unternehmen verwendet werden können.

Es sind mehrere Probleme im Zusammenhang der Kommunikation über ein Netzwerk aufgetreten, insbesondere in einem großen öffentlichen WAN, wie es z. B. das Internet ist. Im allgemeinen werden Informationen über ein Netzwerk in Nachrichtenpaketen übertragen, welche ausgehend von einer Vorrichtung, als Quelle bzw. Quellenvorrichtung, zu einer anderen Vorrichtung, als Ziel bzw. Zielvorrichtung, über einen oder mehrere Router oder allgemein Schaltungsknoten im Netzwerk übertragen werden. Jedes Nachrichtenpaket enthält eine Zieladresse, welche von den Schaltungsknoten verwendet wird, um das jeweilige Nachrichtenpaket an die geeignete Zielvorrichtung zu leiten. Z. B. im Internet haben solche Adressen die Form von "n"-Bit Zahlen (wobei "n" 32 oder 128 sein kann), wobei solche Zahlenkolonnen für einen Benutzer schwierig sind zu merken und einzugeben, wenn die oder der Benutzer die Übertragung eines Nachrichtenpakets veranlassen möchte. Um einen Benutzer von der Notwendigkeit zu befreien, sich solche spezifische Zahlen-Internetadressen zu merken und einzugeben, stellt das Internet einen zweiten Adressierungsmechanismus bereit, der durch Benutzer der jeweiligen Vorrichtungen einfacher handzuhaben ist. Bei diesem Adressierungsmechanismus werden Internet-Domains, wie etwa LANs, Internet-Service-Provider (nachfolgend "ISP" genannt) und ähnliche, welche im Internet verbunden sind, durch für einen Benutzer relativ einfach les- und merkbare Namen identifiziert, die nachfolgend als "Klartextnamen" bezeichnet werden. Um den Einsatz von solchen Klartextnamen umzusetzen, werden Namen-Server, auch als DNS-Server für "Domain Name Server" bezeichnet, bereitgestellt, um die Klartextnamen in die geeigneten Internetadressen umzuwandeln. Wenn ein Bediener einer Vorrichtung, der die Übertragung eines Nachrichtenpakets an eine andere Vorrichtung wünscht, den Klartextnamen der anderen Vorrichtung eingibt, nimmt die Vorrichtung zuerst Kontakt mit einem Namen-Server auf. Im allgemeinen kann der Namen-Server ein Teil des ISP selbst sein oder er kann eine spezielle Vorrichtung sein, welche durch den ISP über das Internet zugäng-

lich ist; in jedem Fall wird der ISP den Namen-Server identifizieren, welcher für die Vorrichtung zu verwenden ist, wenn sich die Vorrichtung beim ISP einloggt, d. h. anmeldet. Falls der Namen-Server, nachdem die Vorrichtung einen Kontakt hergestellt hat, eine Zahlen-Internetadresse für den Klartext-Domainnamen besitzt oder erhalten kann, übermittelt der Namen-Server die Zahlen-Internetadresse, welche dem Klartext-Domainnamen entspricht, zu der Vorrichtung des Bedieners. Die Vorrichtung kann sodann die Zahlen-Internetadresse, welche von dem Namen-Server zurückgesendet wurde, in das Nachrichtenpaket einfügen und das Nachrichtenpaket an den ISP für die Übertragung über das Internet auf konventioneller Weise liefern. Die Internet-Schaltungsknoten verwenden die Zahlen-Internetadresse, um das Nachrichtenpaket an die gewünschte Zielvorrichtung zu übermitteln.

Andere Probleme treten insbesondere in Verbindung mit der Übertragung von Information über ein öffentliches WAN, z. B. das Internet, auf. Ein Problem besteht darin, sicherzustellen, daß die über das WAN übertragene Information, welche die Quellenvorrichtung und die Zielvorrichtung vertraulich behalten möchten, auch tatsächlich vertraulich bleibt gegenüber möglichen Lauschern, welche die Information abfangen können. Um die Vertraulichkeit zu wahren, wurden verschiedene Formen von Verschlüsselung entwickelt und werden verwendet, um die Information vor der Übertragung durch die Quellenvorrichtung zu verschlüsseln und die Information nach deren Empfang durch die Zielvorrichtung zu entschlüsseln. Falls gewünscht wird, daß beispielsweise die gesamte Information, welche zwischen einer bestimmten Quellenvorrichtung und einer bestimmten Zielvorrichtung übertragen wird, vertraulich bleiben soll, können die Vorrichtungen einen sog. "Sicherheitstunnel" zwischen den Vorrichtungen einrichten, der im wesentlichen sicherstellt, daß die gesamte Information, welche von der Quellenvorrichtung an die Zielvorrichtung übertragen wird, vor der Übertragung verschlüsselt wird (mit Ausnahme von bestimmten Protokollinformationen, wie Adresseninformation, welche den Fluß von Netzpaketen über das Netzwerk zwischen der Quellen- und Zielvorrichtung steuert), und daß die verschlüsselte Information vor der Verwendung durch die Zielvorrichtung entschlüsselt wird. Die Quellen- und Zielvorrichtungen können jeweils für sich eine Verschlüsselung bzw. Entschlüsselung durchführen, oder die Verschlüsselung und Entschlüsselung kann durch andere Vorrichtungen durchgeführt werden, bevor die Nachrichtenpakete über das Internet übertragen werden.

Ein weiteres Problem, welches insbesondere im Zusammenhang mit Unternehmen, Regierungsämtern und privaten Organisationen auftritt, deren private Netzwerke, welche LANs, WANs oder etwaige Kombinationen derselben sein können, mit öffentlichen WANs, z. B. dem Internet, verbunden sind, besteht darin, sicherzustellen, daß deren private Netzwerke sicher sind gegenüber anderen Netzwerken, zu welchen z. B. die Unternehmen keinen Zugriff haben möchten, oder einen Zugriff durch andere zu regulieren und zu kontrollieren, zu welchen z. B. die jeweiligen Organisationen einen begrenzten Zugriff haben möchten. Um dies umzusetzen, verbinden die Organisationen in der Regel ihre privaten Netzwerke mit öffentlichen WANs über eine begrenzte Anzahl von Gateways, welche manchmal als "Firewalls" bezeichnet werden, durch welche der gesamte Netzwerkverkehr zwischen dem internen und dem öffentlichen Netzwerk läuft. In der Regel sind Netzwerkadressen von Domains und Vorrichtungen in dem privaten Netzwerk "hinter" der Firewall den Namen-Servern bekannt, welche in den privaten Netzwerken vorgesehen sind; sie sind aber nicht zugänglich für Namen-Server oder andere Vorrichtungen au-

Berhalb der privaten Netzwerke, was die Kommunikation zwischen einer Vorrichtung außerhalb des privaten Netzwerkes und einer Vorrichtung innerhalb des privaten Netzwerkes schwierig macht.

Ein Ziel der vorliegenden Erfindung ist es, hier Abhilfe zu schaffen.

Dieses Ziel erreicht die Erfindung durch die Gegenstände der Ansprüche 1, 7 und 13. Bevorzugte Ausführungsbeispiele der Erfindung sind in den jeweils abhängigen Ansprüchen beschrieben.

Danach schafft die Erfindung ein neuartiges und verbessertes System und ein Verfahren zum Vereinfachen von Kommunikation zwischen Vorrichtungen, welche mit öffentlichen Netzwerken, z. B. dem Internet, verbunden sind, und Vorrichtungen, welche mit privaten Netzwerken verbunden sind, wobei die Auflösung von Sekundäradressen, wie etwa Text- bzw. Klartextnamen im Internet, in die zugehörigen Netzwerkadressen durch Namen-Server oder ähnliche Vorrichtungen, die mit den privaten Netzwerken verbunden sind, ermöglicht wird.

Hierfür stellt die Erfindung ein System zur Verfügung mit einem virtuellen Privaten Netzwerk und einer externen Vorrichtung, welche durch ein digitales Netzwerk miteinander verbunden sind, sowie ein Kommunikationsverfahren und ein Computerprogrammprodukt zum gemeinsamen Verwenden mit einem derartiges System. Das virtuelle private Netzwerk weist eine Firewall bzw. ein Firewall-System, wenigstens eine interne Vorrichtung und einen Namen-Server auf, welche jeweils eine Netzwerkadresse besitzen. Die interne Vorrichtung besitzt ferner eine Sekundäradresse, und der Namen-Server ist derart konfiguriert, daß er eine Zuordnung zwischen der Sekundäradresse und der Netzwerkadresse bereitstellt. In Reaktion auf eine Anfrage von der externen Vorrichtung zum Aufbau einer Verbindung zur Firewall übermittelt die Firewall der externen Vorrichtung die Netzwerkadresse des Namen-Servers. In Reaktion auf eine Anfrage von einem Bediener oder ähnlichem, welche die Sekundäradresse der internen Vorrichtung enthält und einen Zugriff an die interne Vorrichtung anfordert, erzeugt die externe Vorrichtung eine Netzwerkadressen-Anfragennachricht zur Übertragung über die Verbindung an die Firewall, welche eine Auflösung der Netzwerkadresse, die der Sekundäradresse zugeordnet ist, anfordert. Die Firewall übermittelt die Adressenauflösungsanfrage an den Namen-Server und der Namen-Server übermittelt die Netzwerkadresse, welche der Sekundäradresse zugeordnet ist, an die Firewall. Daraufhin stellt die Firewall die Netzwerkadresse in einer Netzwerkadressenantwortnachricht zur Übertragung über die Verbindung an die externe Vorrichtung bereit. Die externe Vorrichtung kann sodann die auf diese Weise bereitgestellte Netzwerkadresse in nachfolgenden an die interne Vorrichtung gerichtete Kommunikationen mit der Firewall verwenden.

Weitere Vorteile und Ausgestaltungen der Erfindung ergeben sich aus der nachfolgenden detaillierten Beschreibung eines bevorzugten Ausführungsbeispiels. In der Beschreibung wird auf die beigefügte schematische Zeichnung Bezug genommen. Darin zeigt:

Fig. 1 ein funktionelles Blockdiagramm eines erfindungsgemäßen Netzwerkes.

Fig. 1 zeigt ein funktionelles Blockdiagramm eines Netzwerkes 10, welches gemäß der vorliegenden Erfindung aufgebaut ist. Das Netzwerk 10 gemäß Fig. 1 umfaßt einen Internet-Service-Provider (nachfolgend "ISP") 11, welcher die Übertragung von Nachrichtenpaketen zwischen einer oder mehreren Vorrichtungen 12(1) bis 12(M) (nachfolgend allgemeinen mit dem Bezugszeichen 12(m) identifiziert), welche mit dem ISP 11 verbunden sind, und anderen Vorrich-

tungen, welche allgemein durch ein Bezugszeichen 13 gekennzeichnet sind, über das Internet 14 ermöglicht, wobei die Übertragung von Information in Nachrichtenpaketen zwischen den Vorrichtungen 12(m) und 13 realisiert wird.

Der ISP 11 verbindet das Internet 14 über eine oder mehrere logische Verbindungen oder Gateways oder ähnlichem (im vorliegenden allgemein als "Verbindungen" bezeichnet), welche allgemein durch das Bezugszeichen 41 gekennzeichnet sind. Der ISP 11 kann ein öffentlicher ISP sein, welcher in diesem Falle die Verbindung mit Vorrichtungen 12(m) herstellt, welche durch Bediener betrieben werden können, die der allgemeinen Öffentlichkeit angehören, so daß diese Bediener Zugang zu dem Internet erlangen. Alternativ dazu kann der ISP 11 ein privater ISP sein. In diesem Falle werden die damit verbundenen Vorrichtungen 12(m) im allgemeinen beispielsweise durch Angestellte eines bestimmten Unternehmens oder einer Regierungseinrichtung, Mitgliedern von einer privaten Organisation oder ähnlichen betrieben, um diesen Angestellten oder Mitglieder einen Zugang in das Internet bereit zu stellen.

In an sich konventioneller Weise weist das Internet ein Netz von Schaltungsknoten auf (welche nicht separat dargestellt sind), welche die ISPs 11 und die Vorrichtungen 13 miteinander verbinden, um dazwischen die Übertragung von Nachrichtenpaketen zu ermöglichen. Die Nachrichtenpakete, welche über das Internet 14 übertragen werden, stimmen mit denjenigen überein, welche durch das sog. Internetprotokoll (IP) definiert werden, und umfassen einen Kopfabschnitt, einen Datenabschnitt und können einen Fehlererfassungs- und/oder Korrekturabschnitt aufweisen. Der Kopfabschnitt enthält Information, welche verwendet wird, um das Nachrichtenpaket über das Internet 14 zu übertragen, beispielsweise eine Zieladresse, welche die Vorrichtung identifiziert, welche das Nachrichtenpaket als Zielvorrichtung empfangen soll, und eine Quellenadresse, welche diejenige Vorrichtung identifiziert, welche das Nachrichtenpaket erzeugt hat. In jedem Nachrichtenpaket haben die Ziel- und Quellenadresse jeweils die Form einer Zahl, welche eindeutig die jeweilige Ziel- bzw. Quellenvorrichtung identifiziert. Die Schaltungsknoten im Internet 14 verwenden wenigstens die Zieladresse eines jeweiligen Nachrichtenpaketes, um das jeweilige Nachrichtenpaket an die Zielvorrichtung zu übermitteln, wenn die Zielvorrichtung an das Internet angeschlossen ist, oder an einen ISP 11 oder andere Vorrichtungen, welche an das Internet 14 angeschlossen sind, welche sodann das Nachrichtenpaket an das geeignete Ziel senden werden. Der Datenabschnitt eines jeden Nachrichtenpakets enthält die in dem Nachrichtenpaket übertragenen Daten; und der Fehlererfassungs- und/oder Korrekturabschnitt enthält Fehlererfassungs- und/oder Korrekturinformationen, welche verwendet werden können, um zu verifizieren, daß das Nachrichtenpaket in korrekter Weise von der Quelle zu der Zielvorrichtung übertragen wurde (im Fall der Fehlererfassungsinformation), und um ausgewählte Arten von Fehlern zu korrigieren, falls das Nachrichtenpaket nicht korrekt übertragen wurde (im Falle der Fehlerkorrekturinformation).

Die Vorrichtungen 12(m), welche mit dem ISP 11 verbunden sind, können jede beliebige Anzahl von Arten von Vorrichtungen umfassen, welche über das Internet 14 mit anderen Vorrichtungen 13 kommunizieren, umfassend z. B. Personalcomputer, Computer-Workstations und ähnliches. Jede Vorrichtung 12(m) kommuniziert mit dem ISP 11, um Nachrichtenpakete für die Übertragung über das Internet 14 an diesen zu übertragen, oder um Nachrichtenpakete, welche durch den ISP 11 über das Internet empfangen werden, von diesem zu empfangen. Dabei kann jedes geeignete Protokoll verwendet werden, z. B. das bekannte Point-to-Point Proto-

koll (allgemein mit "PPP" abgekürzt), falls die Vorrichtung 12(m) über eine Point-to-Point Verbindung mit dem ISP 11 verbunden ist, oder irgendein konventionelles "Multi-Drop" Protokoll, falls die Vorrichtung 12(m) mit dem ISP 11 über ein "Multi-Drop"-Netzwerk, z. B. das Ethernet, verbunden ist, oder ähnliches. Die Vorrichtungen 12(m) sind im allgemeinen entsprechend der üblichen Computerarchitektur mit gespeicherten Programmen aufgebaut, welche z. B. eine Systemeinheit, eine Bildschirmanzeigeinheit und Bedieneingabeeinrichtungen, wie etwa eine Tastatur oder eine Maus, umfaßt. Eine Systemeinheit weist im allgemeinen eine oder mehrere Prozessor-, Speicher-, Massenspeichereinrichtungen, z. B. Festplatten- und/oder Bandspeicherelemente, oder andere Elemente (nicht separat gezeigt) auf, wie etwa Netzwerk- und/oder Telefonschnittstelleneinrichtungen, um die jeweilige Vorrichtung an den ISP 11 anzukoppeln. Die Prozessor- bzw. Verarbeitungseinrichtungen verarbeiten Programme, einschließlich Anwendungsprogramme, unter der Steuerung eines Betriebssystems, um verarbeitete Daten zu erzeugen. Die Bildschirmeinheit ermöglicht es der Vorrichtung, die verarbeiteten Daten und einen Verarbeitungsstatus der Daten dem Benutzer anzuzeigen, und die Bedieneingabeeinrichtung ermöglicht es dem Bediener, Daten einzugeben und die Verarbeitung zu steuern.

Diese Elemente der Vorrichtung 12(m) arbeiten in Verbindung mit einer geeigneten Programmierung so zusammen, um eine Vorrichtung 12(m) mit einer Anzahl von funktionellen Elementen bereit zustellen, beispielsweise eine Bedienerchnittstelle 20, eine Netzwerkschnittstelle 21, einen Nachrichtenpaketgenerator 22, einen Nachrichtenpaketempfänger und -prozessor 23, eine ISP Einloggsteuerung bzw. Anmeldungssteuerung 24, einen Internetparameterspeicher 25 und im Zusammenhang mit der vorliegenden Erfindung einen Sicherheits-Nachrichtenpaketprozessor 26. Die Bedienerchnittstelle 20 ermöglicht, daß die Vorrichtung 12(m) Eingabeinformationen von der/den Bedieneingabeeinrichtung(en) der Vorrichtung 12(m) empfängt und die Ausgabeinformationen dem Bediener auf der/den Bildschirmeinrichtung(en) der Vorrichtung 12(m) angezeigt werden. Die Netzwerkschnittstelle 21 ermöglicht eine Verbindung der Vorrichtung 12(m) mit dem ISP 11 unter Verwendung des geeigneten PPP oder Netzwerkprotokolls, um Nachrichtenpakete an den ISP 11 zu übertragen und von diesem Nachrichtenpakete zu empfangen. Die Netzwerkschnittstelle 21 kann eine Verbindung mit dem ISP 11 über das öffentliche Telefonnetz vorsehen, um einen Wählverbindungsnetzwerkbetrieb (sog. Dial-Up Betrieb) der Vorrichtung 12(m) über das öffentliche Telefonnetz zu ermöglichen. Alternativ oder zusätzlich dazu kann die Netzwerkschnittstelle 21 eine Verbindung durch den ISP 11 über beispielsweise ein konventionelles LAN ermöglichen, wie etwa das Ethernet. In Reaktion auf eine durch die Bedienerchnittstelle 20 gelieferte Eingabe und/oder in Reaktion auf Anfragen aus Programmen (nicht gezeigt), welche durch die Vorrichtung 12(m) verarbeitet werden, kommuniziert die ISP Einloggsteuerung 24 über die Netzwerkschnittstelle 21, um die Initialisierung (sog. "Log-On") einer Kommunikationssitzung zwischen der Vorrichtung 12(m) und dem ISP 11 zu ermöglichen. Während dieser Kommunikationssitzung kann die Vorrichtung 12(m) Information in der Form von Nachrichtenpaketen an andere Vorrichtungen über das Internet 14 sowie an andere Vorrichtungen 12(m') (wobei  $m' \neq m$ ), welche mit der ISP 11 oder mit anderen ISPs verbunden sind, übertragen. Während eines Log-On-Betriebs empfängt die ISP Einloggsteuerung 24 die Internetprotokollparameter (IP-Parameter), welche im Zusammenhang mit einer Nachrichtenpaketerzeugung während der Kommunikationssitzung verwendet werden.

Während einer Kommunikationssitzung erzeugt der Nachrichtenpaketgenerator 22 Nachrichtenpakete zur Übertragung durch die Netzwerkschnittstelle 21 in Reaktion auf eine Eingabe, welche durch den Bediener über die Bedienerchnittstelle 20 geliefert wird und/oder in Reaktion auf Anfragen aus Programmen (nicht separat gezeigt), welche durch die Vorrichtung 12(m) verarbeitet werden. Die Netzwerkschnittstelle 21 empfängt auch Nachrichtenpakete aus dem ISP 11 und liefert diese an den Nachrichtenpaketempfänger und -prozessor 23 zur Verarbeitung und Bereitstellung an die Bedienerchnittstelle 20 und/oder anderen Programmen (nicht gezeigt), welche durch die Vorrichtung 12(m) verarbeitet werden. Falls die empfangenen Nachrichtenpakete eine Information enthalten, z. B. Web-Seiten oder ähnliches, welche dem Bediener angezeigt werden soll, kann die Information der Bedienerchnittstelle 20 geliefert werden, damit die Information auf der Bildschirmeinheit der Vorrichtung angezeigt wird. Zusätzlich oder alternativ dazu kann die Information an andere Programme (nicht gezeigt) zur Verarbeitung geliefert werden, welche durch die Vorrichtung 12(m) verarbeitet werden.

Im allgemeinen können die Elemente, wie die Bedienerchnittstelle 20, der Nachrichtenpaketgenerator 22, der Nachrichtenpaketempfänger und -prozessor 23, die ISP Einloggsteuerung 24 und der Internetparameterspeicher 25 Elemente eines konventionellen Internet-Browsers enthalten, wie die von Mosaic, Netscape Navigator und Microsoft Internet Explorer.

Wie es oben erwähnt wurde, weist die Vorrichtung 12(m) im Zusammenhang mit der vorliegenden Erfindung einen Sicherheits-Nachrichtenpaketprozessor 26 auf. Der Sicherheits-Nachrichtenpaketprozessor 26 ermöglicht den Aufbau und Verwendung eines "Sicherheits隧nNELS" zwischen der Vorrichtung 12(m) und anderen Vorrichtungen 12(m') (wobei  $m' \neq m$ ) oder 13, wie es welches weiter unten beschrieben wird. Im allgemeinen wird in einem solchen Sicherheitstunnel Information in wenigstens dem Datenabschnitt der zwischen der Vorrichtung 12(m) und einer spezifischen anderen Vorrichtung 12(m') (wobei  $m' \neq m$ ) oder 13 übertragenen Nachrichtenpakete geheimgehalten, beispielsweise durch Verschlüsselung des Datenabschnittes vor der Übertragung durch die Quellenvorrichtung. Die Information in anderen Abschnitten eines derartigen Nachrichtenpakets kann ebenfalls geheimgehalten werden, mit Ausnahme der Information, welche benötigt wird, um die Übertragung des jeweiligen Nachrichtenpakets zwischen den Vorrichtungen zu ermöglichen, also z. B. wenigstens die Zielinformation, damit die Schaltungsknoten des Internets und die ISPs die Vorrichtung identifizieren können, welche das Nachrichtenpaket empfangen soll.

Zusätzlich zu dem ISP 11 kann eine Vielzahl von anderen ISPs die Verbindung zum Internet herstellen, wie es durch die Pfeile 16 angedeutet ist, um eine Kommunikation zwischen Vorrichtungen, welche an diesen anderen ISPs angeschlossen sind, mit anderen Vorrichtungen über das Internet zu ermöglichen, welche die Vorrichtungen 12(n), welche an dem ISP 11 angeschlossen sind, umfassen können.

Die Vorrichtungen 13, auf welche die Vorrichtungen 12(m) zugreifen und mit welchen diese kommunizieren, können auch von jeder beliebigen Anzahl von Arten von Vorrichtungen sein, einschließlich Personalcomputer, Computer-Workstations und ähnliches, oder auch Minicomputer und Großrechner, Großspeichersysteme, Rechenserver, lokale Netzwerke (LANs) und Fernverbindungsnetzwerke (WANs), welche derartige Vorrichtungen und zahlreiche andere Arten von Vorrichtungen enthalten, die direkt oder indirekt mit den Netzwerken verbunden werden können. Nach der vorliegenden Erfindung umfaßt wenigstens eine der Vor-

richtungen wenigstens ein privates Netzwerk, welches als virtuelles privates Netzwerk 15 gekennzeichnet ist und z. B. die Form eines LAN oder eines WAN haben kann. Das virtuelle private Netzwerk 15 kann jede der Vorrichtungen 12(m') (wobei  $m' \neq m$ ) aufweisen (wobei die Verbindung zu dem Internet 14 über einen ISP erfolgt) oder der Vorrichtungen 13 (wobei die Verbindung zu dem Internet 14 unmittelbar erfolgt). Bei dem vorliegend beschriebenen Ausführungsbeispiel wird angenommen, daß das virtuelle Netzwerk 15 eine Vorrichtung 13 aufweist. Das virtuelle private Netzwerk 15 umfaßt selbst mehrere Vorrichtungen, welche hier als eine Firewall bzw. ein Firewall-System 30, mehrere Server 31(1) bis 31(S) (im nachfolgenden allgemein mit dem Bezugszeichen 31(s) angegeben) und ein Namen-Server 32 gekennzeichnet sind, wobei allesamt durch eine Übertragungsverbindung 33 miteinander verbunden sind. Die Firewall 30 und die Server 31(s) können ähnlich sein wie jede der verschiedenen Arten von Vorrichtungen 12(m) und 13, die hier beschrieben sind, und können daher beispielsweise umfassen Personalcomputer, Computer-Workstations und ähnliches, aber auch Minicomputer und Großrechner, Großspeichersysteme, Rechnerserver, lokale Netzwerke (LANs) und Fernverbindungsnetzwerke (WANs), welche derartige Vorrichtungen und zahlreiche andere Arten von Vorrichtungen umfassen, welche direkt oder indirekt mit den Netzwerken verbunden werden können.

Wie oben ausgeführt wurde, kommunizieren diese Vorrichtungen einschließlich der Vorrichtungen 12(m) und der Vorrichtungen 13 durch Übertragung von Nachrichtenpaketen über das Internet. Die Vorrichtungen 12(m) und 13 können Information in einem Peer-to-Peer bzw. gleichrangigem Modus, in einem Client-Server Modus oder nach beiden dieser Modi übertragen. Im allgemeinen überträgt eine Vorrichtung in einer Peer-to-Peer Nachrichtenpaketübertragung Information in einem oder mehreren Nachrichtenpaketen an die andere Vorrichtung. Andererseits kann eine Vorrichtung, welche in einem Client-Server Modus als Client fungiert, ein Nachrichtenpaket an eine andere Vorrichtung übertragen, welche als Server fungiert, um beispielsweise einen Dienst durch die andere Vorrichtung auszulösen. Mehrere Arten derartiger Dienste sind dem Fachmann bekannt, beispielsweise das Wiedergewinnen bzw. Auslesen von Information aus der anderen Vorrichtung, damit diese aktiviert wird, um Verarbeitungsoperationen und dergleichen durchzuführen. Falls der Server dazu dient, dem Client vor allem Informationen zu liefern, kann dieser allgemein als ein Serverserver bezeichnet werden. Falls der Server andererseits Verarbeitungsoperationen auf Anfrage des Client ausführen soll, kann dieser allgemein als ein Rechnerserver bezeichnet werden. Andere Arten von Servern zum Ausführen von anderen Arten von Diensten und Operationen auf Anfrage von Clients sind dem Fachmann ebenfalls bekannt.

Wenn in einer Client-Server Anordnung eine Vorrichtung 12(m) einen Dienst durch beispielsweise eine Vorrichtung 13 ausgeführt haben möchte, erzeugt die Vorrichtung 12(m) eines oder mehrere Anfragenachrichtenpakete zur Übertragung an die Vorrichtung 13, welche den benötigten Dienst anfordern. Das Anfragenachrichtenpaket enthält die Internetadresse der Vorrichtung 13, welche als die Zielvorrichtung das Nachrichtenpaket empfängt und den Dienst ausführt. Die Vorrichtung 12(m) überträgt das/die Anfragenachrichtenpaket(e) an den ISP 11. Der ISP 11 überträgt daraufhin das Nachrichtenpaket über das Internet an die Vorrichtung 13.

Falls die Vorrichtung 13 die Form eines WAN oder LAN hat, empfängt das WAN oder LAN das/die Nachrichtenpaket(e) und leitet dieses/diese zu einer dort angeschlossenen Vorrichtung weiter, welche den angeforderten Dienst aus-

führen soll.

In jedem Fall wird die Vorrichtung 13, welche den angeforderten Dienst ausführen soll, nach Empfang des/der Anfragenachrichtenpaket(e) die Anfrage bearbeiten. Falls die Vorrichtung 12(m), welche das/die Anfragenachrichtenpaket(e) erzeugt hat, oder deren Bediener die notwendigen Befugnisse hat, um den Dienst von der Vorrichtung 13 anzufordern, und falls der angeforderte Dienst die Einleitung einer Informationsübertragung aus der Vorrichtung 13 als ein Speicherserver an die Vorrichtung 12(m) als ein Client umfaßt, erzeugt die Vorrichtung 13 eines oder mehrere Antwortnachrichtenpakete, welche die angeforderten Information enthalten, und überträgt das/die Paket(e) über das Internet 14 an den ISP 11. Daraufhin überträgt der ISP 11 das/die Nachrichtenpaket(e) an die Vorrichtung 12(m). Falls andererseits der angeforderte Dienst die Einleitung eines Verarbeitungsvorganges durch die Vorrichtung 13 als ein Rechnerserver beinhaltet, wird die Vorrichtung 13 den/die angeforderten Rechendienst(e) ausführen. Falls die Vorrichtung 13 verarbeitete Daten, welche während den Rechenvorgängen erzeugt wurden, an die Vorrichtung 12(m) als Client zurücksenden soll, erzeugt die Vorrichtung 13 zusätzlich eines oder mehrere Antwortnachrichtenpakete, welche die verarbeiteten Daten enthalten und überträgt das/die Paket(e) über das Internet 14 an den ISP 11. Der ISP 11 überträgt daraufhin das/die Nachrichtenpaket(e) an die Vorrichtung 12(m). Entsprechende Operationen können durch die Vorrichtungen 12(m) und 13, dem ISP 11 und dem Internet 14 in Verbindung mit anderen Arten von Diensten ausgeführt werden, welche durch die Server-Vorrichtungen 13 bereitgestellt werden können.

Wie oben angemerkt wurde, enthält jedes Nachrichtenpaket, welches durch die Vorrichtungen 12(m) und 13 zur Übertragung über das Internet 14 erzeugt wird, eine Zieladresse, welche von den Schaltungsknoten verwendet wird, um das jeweilige Nachrichtenpaket an die geeignete Zielvorrichtung zu leiten. Adressen im Internet haben die Form von "n"-Bit Zahlen (wobei "n" beim gegenwärtigen Standard 32 oder 128 sein kann). Um insbesondere einen Bediener einer Vorrichtung 12(m) von der Notwendigkeit zu befreien, sich spezifische Zahlenkolonnen bzw. Zahlen-Internetadressen zu merken und diese der Vorrichtung 12(m) einzugeben, um die Erzeugung eines Nachrichtenpakets zur Übertragung über das Internet einzuleiten, stellt das Internet einen zweiten Adressierungsmechanismus zur Verfügung, welcher einfacher durch menschliche Bediener der jeweiligen Vorrichtungen handhabbar ist. Bei diesem Adressierungsmechanismus werden Internet-Domains, wie etwa LANs, Internet-Service-Provider (ISPs) und ähnliche, welche in bzw. mit dem Internet verbunden sind, durch relativ einfach les- und merkbare Namen, sog. Klartextnamen, identifiziert. Dabei soll sich hier die Bezeichnung "Klartextname" auf jede Art von Namenstext beziehen, z. B. auch auf Abkürzungen, generische Bezeichnungen, Phantasiebegriffe, etc. Um das System der Klartext-Domainnamen umzusetzen, ist der ISP 11 mit einem Namen-Server 17 (der auch als ein DNS Server (Domain Name Server) bezeichnet werden kann) verbunden, welcher die Klartext-Domainnamen auflösen bzw. in eine gültige Internetadresse umwandeln kann, um die geeignete Internetadresse für das in dem jeweiligen Klartextnamen angegebene Ziel bereitzustellen. Im allgemeinen kann der Namen-Server ein Teil des ISP 11 oder damit direkt verbunden sein, wie es in Fig. 1 gezeigt ist, oder er kann eine bestimmte Vorrichtung sein, welche durch den ISP über das Internet zugänglich ist. Jedenfalls wenn sich die Vorrichtung 12(m) bei dem ISP 11 während einer Kommunikationssitzung einloggt, wird der ISP 11, wie oben hingewiesen wurde, verschiedene Internet-Proto-

kollparameter (IP-Parameter) zuordnen, welche die Vorrichtung 12(m) während der Kommunikationssitzung verwendet, und welche in dem Internetparameterspeicher 25 gespeichert sind. Diese IP-Parameter enthalten Informationen, wie

- (a) eine Internetadresse für die Vorrichtung 12(m), welche die Vorrichtung 12(m) während der Kommunikationssitzung identifiziert; und
- (b) die Identifizierung eines Namen-Servers 17, welchen die Vorrichtung 12(m) während der Kommunikationssitzung verwendet.

Wenn die Vorrichtung 12(m) Nachrichtenpakete zur Übertragung erzeugt, fügt sie ihre Internetadresse (oberer Punkt (a)) als die Quellenadresse ein. Die Vorrichtung(en) 13, welche die jeweiligen Nachrichtenpakete empfängt/empfangen, kann/können die Quellenadresse aus den Nachrichtenpaketen, welche von der Vorrichtung 12(m) empfangen werden, in Nachrichtenpaketen verwenden, welche die Vorrichtung(en) 13 zur Übertragung an die Vorrichtung 12(m) erzeugt/erzeugen, so daß das Internet in der Lage ist, die durch die jeweilige Vorrichtung 13 erzeugten Nachrichtenpakete an die Vorrichtung 12(m) zu leiten. Falls die Vorrichtung 12(m) auf den Namen-Server 17 über das Internet 14 zugreift, hat die durch den ISP 11 bereitgestellte Identifizierung des Namen-Servers 17 (siehe oben unter (b)) die Form einer Zahlen-Internetadresse, welche es der Vorrichtung 12(m) ermöglicht, für den Namen-Server 17 Nachrichten zu erzeugen, welche eine Auflösung der Klartext-Internetadressen in Zahlen-Internetadressen anfordern. Der ISP 11 kann der Vorrichtung 12(m) auch andere IP-Parameter zuordnen, wenn diese sich beim ISP 11 einloggt, beispielsweise die Identifizierung einer Verbindung zu dem Internet 14, welche für Nachrichten zu verwenden ist, die durch die Vorrichtung 12(m) übersandt werden, insbesondere falls der ISP 11 Mehrfach-Gateways aufweist. In der Regel speichert die Vorrichtung 12(m) die Internetparameter im Internetparameterspeicher 25 für die Verwendung während der Kommunikationssitzung.

Wenn ein Bediener die Vorrichtung 12(m) veranlassen möchte, daß sie ein Nachrichtenpaket an eine Vorrichtung 13 überträgt gibt der oder die Bediener(in) die Internetadresse der Vorrichtung 13 an die Vorrichtung 12(m) über die Bedienerchnittstelle 20 ein, sowie eine Information oder die Identifizierung der in der Vorrichtung 12(m) aufbewahrten Information, welche in der Nachricht übertragen werden sollen. Die Bedienerchnittstelle 20 aktiviert daraufhin den Paketgenerator 22 zur Freigabe der benötigten Pakete zur Übertragung durch den ISP 11 über das Internet 14. Falls

- (i) der Bediener die Zahlen-Internetadresse bereitgestellt hat, oder
- (ii) der Bediener die Klartext-Internetadresse bereitgestellt hat, aber der Paketgenerator 22 bereits die Zahlen-Internetadresse besitzt, welche der durch den Bediener eingegebenen Klartext-Internetadresse entspricht,

kann der Paketgenerator 22 unmittelbar nach Aktivierung durch die Bedienerchnittstelle 20 die Pakete erzeugen und diese an die Netzwerkschnittstelle 21 zur Übertragung an den ISP 11 liefern.

Falls aber der Bediener die Klartext-Internetadresse der Vorrichtung 13, an welche die Pakete zu übertragen sind, eingegeben hat, und falls der Paketgenerator 22 die entsprechende Zahlen-Internetadresse davon nicht bereits besitzt,

ermöglicht es der Paketgenerator 22, daß die Netzwerkadresse von dem Namen-Server 17, der in dem IP-Parameterspeicher 25 identifiziert ist, erhalten wird.

Bei diesem Vorgang wird der Paketgenerator 22 anfänglich den Namen-Server 17 kontaktieren, um zu versuchen, die geeignete Zahlen-Internetadresse von dem Namen-Server 17 zu erhalten. Bei diesem Vorgang wird die Vorrichtung 12(m) geeignete Nachrichtenpakete zur Übertragung an den Namen-Server 17 unter Verwendung der Zahlen-Internetadresse des Namen-Servers 17 erzeugen, welche durch den ISP 11 bereitgestellt wird, wenn sich die Vorrichtung 12(m) zu Beginn der Kommunikationssitzung einloggt. Jedenfalls wenn der Namen-Server 17 die Zahlen-Internetadresse für den Klartextnamen besitzt oder erhalten kann, wird der Namen-Server 17 die Zahlen-Internetadresse an die Vorrichtung 12(m) übermitteln. Die Zahlen-Internetadresse wird durch den Paketgenerator 22 über die Netzwerkschnittstelle 21 und den Paketempfänger und -prozessor 23 empfangen. Nachdem der Paketgenerator 22 die Zahlen-Internetadresse empfangen hat, kann er die notwendigen Nachrichtenpakete zur Übertragung an die Vorrichtung 13 durch die Netzwerkschnittstelle 21 und den ISP 11 erzeugen.

Wie oben ausgeführt wurde, ist in Fig. 1 eine der Vorrichtungen 13, welche an das Internet 14 angeschlossen sind, ein virtuelles privates Netzwerk 15, wobei das virtuelle private Netzwerk 15 eine Firewall bzw. ein Firewall-System 30, mehrere als Server 31(s) gekennzeichnete Vorrichtungen und einen Namen-Server 32 aufweist, die durch eine Übertragungsverbindung 33 miteinander verbunden sind. Die Server 31(s), die Firewall 30 und der Namen-Server 32 können als z. B. in einem LAN oder WAN verbundene Vorrichtungen untereinander Information in Form von Nachrichtenpaketen austauschen. Da die Firewall 30 mit dem Internet 14 verbunden ist und darüber Nachrichtenpakete empfangen kann, hat sie auch eine Internetadresse. Zusätzlich haben wenigstens die Server 31(s), welche über das Internet zugänglich sind, auch jeweilige Internetadressen. Dabei dient der Namen-Server 32 der Umwandlung von Klartext-Internetadressen für die Server 31(s) innerhalb des virtuellen privaten Netzwerkes 15 in die jeweiligen Zahlen-Internetadressen.

Im allgemeinen wird das virtuelle private Netzwerke 15 von einem Unternehmen, einem Regierungsamt, einer Organisation oder ähnlichem gehalten, welche möchten, daß die Server 31(s) Zugriff auf andere Vorrichtungen außerhalb des virtuellen privaten Netzwerkes 15 haben und an diese Information über das Internet 14 übertragen können, aber welche ebenfalls möchten, daß der Zugriff an die Server 31(s) durch Vorrichtungen 12(m) und andere externe Vorrichtungen über das Internet 14 in einer kontrollierten Weise begrenzt ist. Die Firewall 30 dient dazu, den Zugriff durch Vorrichtungen außerhalb des virtuellen privaten Netzwerkes 15 auf Server 31(s) innerhalb des virtuellen privaten Netzwerkes 15 zu kontrollieren. Bei diesem Vorgang stellt die Firewall 30 auch die Verbindung zum Internet 14 her und empfängt Nachrichtenpakete darüber zur Übertragung an einen Server 31(s). Falls das Nachrichtenpaket angibt, daß die Quelle des Nachrichtenpaketes einen Zugriff auf einen bestimmten Server 31(s) anfordert, und falls die Quelle für den Zugriff an den Server 31(s) autorisiert ist, sendet die Firewall 30 das Nachrichtenpaket über die Übertragungsverbindung 33 an den Server 31(s). Falls andererseits die Quelle nicht autorisiert ist, auf den Server 31(s) zuzugreifen, wird die Firewall 30 das Nachrichtenpaket nicht an den Server 31(s) übersenden, und kann anstelle ein Antwortnachrichtenpaket an die Quellenvorrichtung übermitteln, welches angibt, daß die Quelle nicht für den Zugriff an den Server 31(s) autorisiert ist. Die Firewall kann ähnlich aufgebaut sein wie die ande-



ren Vorrichtungen 31(s) in dem virtuellen privaten Netzwerk 15, wobei zusätzlich eine oder mehrere Verbindungen mit dem Internet vorhanden sind, welche allgemein durch das Bezugszeichen 43 gekennzeichnet sind.

Kommunikationen zwischen Vorrichtungen außerhalb des virtuellen privaten Netzwerkes 15, z. B. der Vorrichtung 12(m), und einer Vorrichtung, z. B. einem Server 31(s), innerhalb des virtuellen privaten Netzwerkes 15 kann über einen Sicherheitstunnel zwischen der Firewall 30 und der externen Vorrichtung, wie es oben beschrieben ist, erreicht werden, damit die ausgetauschten Information geheim bleiben, während diese über das Internet 14 und durch den ISP 11 übertragen werden. Ein Sicherheitstunnel zwischen der Vorrichtung 12(m) und dem virtuellen privaten Netzwerk 15 ist in Fig. 1 durch logische Verbindungen dargestellt, welche durch die Bezugszeichen 40, 42 und 44 gekennzeichnet sind; es versteht sich, daß die logische Verbindung 42 eine der logischen Verbindungen 41 zwischen dem ISP 11 und dem Internet 14 und die logische Verbindung 44 eine der logischen Verbindungen 43 zwischen dem Internet 14 und der Firewall 30 umfaßt.

Der Aufbau eines Sicherheitstunnels kann durch eine Vorrichtung 12(m), die extern zu dem virtuellen privaten Netzwerk 15 ist, ausgelöst werden. Bei diesem Vorgang erzeugt die Vorrichtung 12(m) in Reaktion auf eine Aufforderung durch deren Bediener ein Nachrichtenpaket zur Übertragung durch den ISP 11 und das Internet 14 an die Firewall 30, welches den Aufbau eines Sicherheitstunnels zwischen der Vorrichtung 12(m) und der Firewall 30 anfordert. Das Nachrichtenpaket kann an eine bestimmte Zahlen-Internetadresse gerichtet sein, welche der Firewall 30 zugeordnet ist und welche für Sicherheitstunnelaufbauanfragen reserviert ist, und welche ferner der Vorrichtung 12(m) bekannt ist und durch den Namen-Server 17 bereitgestellt wird. Falls die Vorrichtung 12(m) autorisiert ist, auf einen Server 31(s) in dem virtuellen privaten Netzwerk 15 zuzugreifen, nehmen die Vorrichtung 12(m) als Client und die Firewall 30 einen Dialog auf, welcher den Austausch von einem oder mehreren Nachrichtenpaketen über das Internet 14 umfaßt. Während des Dialogs kann die Firewall 30 der Vorrichtung 12(m) die Identifizierung eines Entschlüsselungsalgorithmus und einen zugehörigen Entschlüsselungsschlüssel bereitstellen, welche die Vorrichtung 12(m) beim Entschlüsseln der verschlüsselten Abschnitte der Nachrichtenpakete zu verwenden hat, welche das virtuelle private Netzwerk an die Vorrichtung 12(m) überträgt. Zusätzlich dazu kann die Firewall 30 der Vorrichtung 12(m) auch die Identifizierung eines Verschlüsselungsalgorithmus und einen zugehörigen Verschlüsselungsschlüssel bereitstellen, welche die Vorrichtung 12(m) beim Verschlüsseln der Abschnitte der Nachrichtenpakete zu verwenden hat, welche die Vorrichtung 12(m) an das virtuelle private Netzwerk 15 überträgt und welche verschlüsselt werden sollen. Alternativ dazu kann die Vorrichtung 12(m) die Identifizierung des Verschlüsselungsalgorithmus und des Verschlüsselungsschlüssels, welche die Vorrichtung 12(m) verwenden wird, an die Firewall 30 während des Dialogs liefern. Die Vorrichtung 12(m) kann in ihrem IP-Parameterspeicher 25 Informationen betreffend den Sicherheitstunnel speichern, einschließlich der Information in Verbindung mit der Identifizierung der Firewall 30 und der Identifizierungen der Verschlüsselungs- und Entschlüsselungsalgorithmen und dazugehöriger Schlüssel für Nachrichtenpakete, welche durch den Sicherheitstunnel übertragen werden.

Sodann können die Vorrichtung 12(m) und die Firewall 30 Nachrichtenpakete über den Sicherheitstunnel übertragen. Beim Erzeugen von Nachrichtenpaketen zur Übertragung über den Sicherheitstunnel verwendet die Vorrichtung

12(m) den Sicherheits-Paketprozessor 26, um die Abschnitte der Nachrichtenpakete zu verschlüsseln, welche vor der Übertragung durch die Netzwerkschnittstelle 21 an den ISP 11 zur Übertragung über das Internet 14 an die Firewall 30 verschlüsselt werden sollen, und um die verschlüsselten Abschnitte der Nachrichtenpakete zu entschlüsseln, welche durch die Vorrichtung 12(m) empfangen werden und welche verschlüsselt sind. Insbesondere nachdem der Paketgenerator 22 ein Nachrichtenpaket zur Übertragung an die Firewall 30 über den Sicherheitstunnel erzeugt hat, liefert er das Nachrichtenpaket an den Sicherheits-Paketprozessor 26. Der Sicherheits-Paketprozessor 26 verschlüsselt daraufhin die Abschnitte des Nachrichtenpakets, welche verschlüsselt werden sollen, unter Verwendung des Verschlüsselungsalgorithmus und des Verschlüsselungsschlüssels. Nachdem die Firewall 30 ein Nachrichtenpaket von der Vorrichtung 12(m) über den Sicherheitstunnel empfangen hat, wird sie dieses entschlüsseln und, falls der beabsichtigte Empfänger des Nachrichtenpakets eine andere Vorrichtung, z. B. ein Server 31(s), in dem virtuellen privaten Netzwerk 15 ist, wird die Firewall 30 das Nachrichtenpaket an diese andere Vorrichtung über die Übertragungsverbindung 33 übertragen.

Wenn ein Nachrichtenpaket von einer Vorrichtung, z. B. einem Server 31(s), in dem virtuellen privaten Netzwerk 15 an die Vorrichtung 12(m) über den Sicherheitstunnel übertragen werden soll, empfängt die Firewall 30 ein solches Nachrichtenpaket über die Übertragungsverbindung 33 und verschlüsselt das Nachrichtenpaket zur Übertragung über das Internet 14 an den ISP 11. Der ISP 11 sendet daraufhin das Nachrichtenpaket an die Vorrichtung 12(m), insbesondere an deren Netzwerkschnittstelle 21. Die Netzwerkschnittstelle 21 liefert das Nachrichtenpaket an den Sicherheits-Paketprozessor 26, welcher die verschlüsselten Abschnitte des Nachrichtenpakets unter Verwendung des Entschlüsselungsalgorithmus und -schlüssels entschlüsselt.

Ein Problem tritt auf im Zusammenhang mit Zugriffen durch eine Vorrichtung, z. B. einer Vorrichtung 12(m), welche extern zum virtuellen privaten Netzwerk 15 ist, und einer Vorrichtung, z. B. einem Server 31(s), welche extern zu der Firewall ist, nämlich dann, wenn dem Namen-Server 17 keine Zahlen-Internetadressen für die Server 31(s) und andere Vorrichtungen bereitgestellt sind, die sich innerhalb des virtuellen privaten Netzwerkes 15 befinden – mit Ausnahme der Zahlen-Internetadressen, welche der Firewall 30 zugeordnet sind. Folglich wird die Vorrichtung 12(m) nach Eingabe der Klartext-Internetadresse durch den Bediener nicht in der Lage sein, die Zahlen-Internetadresse des Servers 31(s) zu erhalten, wenn er auf den Namen-Server 17 zugreift.

Wenn die Vorrichtung 12(m) und die Firewall 39 zusammenarbeiten, um einen dazwischenliegenden Sicherheitstunnel aufzubauen, liefert die Firewall 30 zur Behebung des obigen Problems an die Vorrichtung 12(m) zusätzlich zu möglichen Identifikationen der Verschlüsselungs- und Entschlüsselungsalgorithmen und -schlüsseln, welche im Zusammenhang mit der Übertragung der Nachrichtenpakete über den Sicherheitstunnel zu verwenden sind, an die Vorrichtung 12(m) auch die Identifizierung eines Namen-Servers, z. B. eines Namen-Servers 32, innerhalb des virtuellen privaten Netzwerkes 15, auf welchen die Vorrichtung 12(m) zugreifen kann, um die geeigneten Zahlen-Internetadressen für die Klartext-Internetadressen zu erhalten, welche durch den Bediener einer Vorrichtung 12(m) eingegeben werden. Die Identifizierung des Namen-Servers 32 wird ebenfalls in dem IP-Parameterspeicher 25 gespeichert, zusammen mit der Identifizierung des Namen-Servers 17, welche durch den ISP 11 bereitgestellt wurde, sobald die Vorrichtung 12(m)

beim ISP 11 zu Beginn einer Kommunikationssitzung eingeloggt wurde. Wenn daher die Vorrichtung 12(m) ein Nachrichtenpaket an eine Vorrichtung, z. B. einen Server 31(s), in dem virtuellen privaten Netzwerk 15 unter Verwendung einer Klartext-Internetadresse übertragen möchte, welche z. B. durch einen Bediener bereitgestellt bzw. eingegeben wurde, greift die Vorrichtung 12(m) zu Beginn auf den Namen-Server 17 zu, wie es oben beschrieben wurde, um zu versuchen, die zu der Klartext-Internetadresse zugehörige Zahlen-Internetadresse zu erhalten. Da der Namen-Server 17 außerhalb des virtuellen privaten Netzwerkes 15 ist und die durch die Vorrichtung 12(m) angeforderten Information nicht besitzt, sendet er ein entsprechend lautendes Antwortnachrichtenpaket. Die Vorrichtung 12(m) wird sodann ein Anfragenachrichtenpaket zur Übertragung an den Namen-Server 32 durch die Firewall 30 und über den Sicherheitstunnel erzeugen. Falls der Namen-Server 32 eine Zahlen-Internetadresse besitzt, welche zu der Klartext-Internetadresse in dem Anfragenachrichtenpaket gehört, welches durch die Vorrichtung 12(m) geliefert wird, stellt er die Zahlen-Internetadresse in einer Weise bereit, welche im allgemeinen derjenigen ähnlich ist, welche oben im Zusammenhang mit dem Namen-Server 17 beschrieben wurde mit der Ausnahme, daß die Zahlen-Internetadresse durch den Namen-Server 32 in einem an die Firewall 30 gerichteten Nachrichtenpaket geliefert wird, und die Firewall 30 sodann das Nachrichtenpaket über den Sicherheitstunnel an die Vorrichtung 12(m) übermittelt. Es versteht sich, daß sich in dem Nachrichtenpaket, welches durch die Firewall 30 übertragen wird, die Zahlen-Internetadresse in dem Nachrichtenpaket im Datenabschnitt des Nachrichtenpakets befindet, welches über den Sicherheitstunnel übertragen wird und entsprechend verschlüsselt sein wird. Das Nachrichtenpaket wird durch die Vorrichtung 12(m) in einer ähnlichen Weise verarbeitet, wie sie oben im Zusammenhang mit anderen Nachrichtenpaketen beschrieben wurde, welche durch die Vorrichtung 12(m) über den Sicherheitstunnel empfangen werden. Das heißt, daß das Nachrichtenpaket durch den Sicherheits-Paketprozessor 26 vor dem Übermitteln an den Paketempfänger und -prozessor 23 zur Verarbeitung entschlüsselt wird. Die Zahlen-Internetadresse für den Server 31(s) kann in einem Cache in einer Zugriffskontrollliste (ACL) in dem IP-Parameterspeicher 25 gespeichert werden, zusammen mit der Zuordnungsinformation bezüglich der zugehörigen Klartext-Internetadresse, einer Angabe, daß der Server 31(s), der dieser Klartext-Internetadresse zugeordnet ist, über die Firewall 30 des virtuellen privaten Netzwerkes 15 zugänglich ist, und die Identifizierungen der Verschlüsselungs- und Entschlüsselungsalgorithmen und -schlüssel, welche für eine Verschlüsselung und Entschlüsselung der geeigneten Abschnitte der Nachrichtenpakete zu verwenden sind, welche an den Server 31(s) übertragen und von diesem erhalten werden.

Es versteht sich, daß in Reaktion auf ein Nachrichtenpaket von der Vorrichtung 12(m), welches beim Namen-Server 32 die Bereitstellung einer Zahlen-Internetadresse für eine durch die Vorrichtung 12(m) angegebene Klartext-Internetadresse anfordert, falls der Namen-Server 32 keine Zuordnungsinformation zwischen der Klartext-Internetadresse und einer Zahlen-Internetadresse besitzt, der Namen-Server 32 ein Antwortnachrichtenpaket, das entsprechend lautet, übertragen kann. Falls die Vorrichtung 12(m) eine Identifizierung von anderen Namen-Servern besitzt, welche z. B. mit anderen virtuellen privaten Netzwerken (nicht gezeigt) verbunden sein können und zu welchen die Vorrichtung 12(m) Zugriff hat, dann kann die Vorrichtung 12(m) versuchen, auf die anderen Namen-Server in einer ähnlichen Weise, wie es oben beschrieben ist, zuzugreifen. Falls die

Vorrichtung 12(m) nicht in der Lage ist, eine Zahlen-Internetadresse, welche der Klartext-Internetadresse zugeordnet ist, von irgendeinem der Namen-Server zu erhalten, zu welchem sie Zugriff hat und welche im allgemeinen im IP-Parameterspeicher 25 der Vorrichtung 12(m) identifiziert sind, wird sie allgemein nicht in der Lage sein, auf eine Vorrichtung mit der vorgegebenen Klartext-Internetadresse zuzugreifen und wird den Bediener oder ein Programm, welche den Zugriff angefordert haben, dementsprechend unterrichten.

Mit diesem Hintergrund werden nun Operationen, welche durch die Vorrichtung 12(m) und das virtuelle private Netzwerk 15 in Verbindung mit der vorliegenden Erfindung durchgeführt werden, im Detail beschrieben. Im allgemeinen laufen die Operationen in zwei Phasen ab. In einer ersten Phase arbeiten die Vorrichtung 12(m) und das virtuelle private Netzwerk 15 zusammen, um einen Sicherheitstunnel durch das Internet 14 aufzubauen. In dieser ersten Phase liefert das virtuelle private Netzwerk 15, insbesondere die Firewall 30, die Identifizierung eines Namen-Servers 32, und es kann auch die den Verschlüsselungs- und Entschlüsselungsalgorithmus und -schlüssel betreffende Information bereitstellen, wie es oben beschrieben wurde. In der zweiten Phase, nachdem der Sicherheitstunnel eingerichtet wurde, kann die Vorrichtung 12(m) die während der ersten Phase gelieferten Information im Zusammenhang mit der Erzeugung und Übertragung von Nachrichtenpaketen an einen oder mehrere Server 31(s) in dem virtuellen privaten Netzwerk 15 und bei dem notwendigen Umwandlungsvorgang der Klartext-Internetadressen zu Zahlen-Internetadressen aus dem Namen-Server 32, welcher durch die Firewall 30 während der ersten Phase identifiziert wurde, verwenden.

Folglich erzeugt die Vorrichtung 12(m) in der ersten (Sicherheitstunnelaufbau)phase zu Beginn ein Nachrichtenpaket zur Übertragung an die Firewall 30, welches einen Aufbau eines Sicherheitstunnels anfordert. Das Nachrichtenpaket enthält eine Zahlen-Internetadresse für die Firewall, (welche durch den Bediener der Vorrichtung oder ein Programm bereitgestellt werden kann, welches durch die Vorrichtung 12(m) verarbeitet wird, oder durch den Namen-Server 17 bereitgestellt werden kann, nachdem eine Klartext-Internetadresse durch den Bediener oder ein Programm bereitgestellt wurde), und welche insbesondere dazu dient, die Firewall 30 zu veranlassen, mit der Vorrichtung 12(m) einen Sicherheitstunnel aufzubauen. Falls die Firewall 30 die Anfrage bezüglich des Sicherheitstunnelaufbaus akzeptiert und falls die Firewall 30 die Verschlüsselungs- und Entschlüsselungsalgorithmen und -schlüssel bereitstellt, so wie es oben angegeben wurde, erzeugt die Firewall 30 ein Antwortnachrichtenpaket zur Übertragung an die Vorrichtung 12(m), welches die Verschlüsselungs- und Entschlüsselungsalgorithmen und -schlüssel identifiziert. Wie oben beschrieben, wird dieses Antwortnachrichtenpaket nicht verschlüsselt. Wenn die Vorrichtung 12(m) die Antwort empfängt, werden die Identifizierungen der Verschlüsselungs- und Entschlüsselungsalgorithmen und -schlüssel in dem IP-Parameterspeicher 25 gespeichert.

Zu einem späteren Zeitpunkt in der ersten Phase erzeugt die Firewall 30 auch ein Nachrichtenpaket zur Übertragung an die Vorrichtung 12(m), welches die Zahlen-Internetadresse des Namen-Servers 32 enthält. Bei diesem Nachrichtenpaket wird der Abschnitt des Nachrichtenpakets, welcher die Zahlen-Internetadresse des Namen-Servers 32 enthält, unter Verwendung eines Verschlüsselungsalgorithmus und Verschlüsselungsschlüssels verschlüsselt, und dies kann unter Verwendung des Entschlüsselungsalgorithmus und -schlüssels, die durch das zuvor beschriebene Antwortnachrichtenpaket geliefert wurden, wieder entschlüsselt



werden. Diese Nachricht hat im allgemeinen die folgende Struktur:

```
"<IIA(FW),IIA(DEV_12(m))><SEC_TUN>
<ENCR<<IIA(FW),IIA(DEV_12(m))><(DNS_ADRS:IIA(-
NS_2)>>>"
```

wobei

- (i) "IIA(FW)" die Quellenadresse darstellt, d. h. eine Zahlen-Internetadresse der Firewall 30,
- (ii) "IIA(DEV\_12(m))" die Zieladresse darstellt, d. h. die Zahlen-Internetadresse der Vorrichtung 12 (m),
- (iii) "DNS\_ADRS:IIA(NS)" angibt, daß "IIA(NS\_32)" die Zahlen-Internetadresse des Namen-Servers 32 darstellt, für dessen Benutzung die Vorrichtung 12(m) autorisiert ist, und
- (iv) "ENCR<...>" bedeutet, daß die Information, zwischen den Klammern "<" und ">" verschlüsselt ist.

Der Anfangsabschnitt der Nachricht "IIA(FW),IIA(DEV\_12(m))>" bildet wenigstens einen Teil des Kopfabschnitts der Nachricht, und "<ENCR<<IIA(FW),IIA(DEV\_12(m))><IIA(NS)>>>" stellt wenigstens einen Teil des Datenabschnitts der Nachricht dar. "<SEC\_TUN>" stellt einen Hinweis in dem Kopfabschnitt dar, welcher angibt, daß die Nachricht über den Sicherheitstunnel übertragen wird, wodurch auch angezeigt wird, daß der Datenabschnitt der Nachricht verschlüsselte Information enthält.

Nachdem die Vorrichtung 12(m) die Nachricht von der Firewall 30 empfängt, wie es oben beschrieben wurde, und weil das Nachrichtenpaket den <SEC\_TUN> Hinweis enthält, überträgt deren Netzwerkschnittstelle 21 den verschlüsselten

Abschnitt "<ENCR<<IIA(FW),IIA(DEV\_12(m))><(DNS\_ADRS:IIA(NS\_32)>>>" an den Sicherheits-Paketprozessor 26 zur Verarbeitung. Der Sicherheits-Paketprozessor 26 entschlüsselt den verschlüsselten Abschnitt, bestimmt weiter, daß der Abschnitt "IIA(NS\_32)" die Zahlen-Internetadresse des Namen-Servers darstellt, insbesondere des Namen-Servers 32, für dessen Benutzung die Vorrichtung 12(m) autorisiert ist, und speichert diese Adresse in dem IP-Parameterspeicher 25 zusammen mit einer Angabe, daß die dorthin gerichteten Nachrichtenpakete zu der Firewall 30 zu übertragen sind, und daß die Daten in den Nachrichtenpaketen unter Verwendung des Verschlüsselungsalgorithmus und -schlüssels, die davor durch die Firewall 30 übermittelt wurden, zu verschlüsseln sind. Es versteht sich, daß aufgrund der Tatsache, daß die Zahlen-Internetadresse des Namen-Servers 32 von der Firewall an die Vorrichtung 12(m) in verschlüsselter Form übertragen wird, diese vertraulich bleibt, selbst wenn das Paket durch einen Dritten abgefangen wird.

In Abhängigkeit des speziellen Protokolls, welches für den Aufbau des Sicherheitstunnels verwendet wird, können die Firewall 30 und die Vorrichtung 12(m) auch Nachrichtenpakete austauschen, welche andere Information enthalten als die oben beschriebenen.

Wie oben erwähnt wurde, kann die Vorrichtung 12(m) in der zweiten Phase nach der Einrichtung des Sicherheitstunnels die Information, welche während der ersten Phase bereitgestellt wurde, im Zusammenhang mit dem Erzeugen und Übertragen von Nachrichtenpaketen zu einem oder mehreren der Server 31(s) in dem virtuellen privaten Netzwerk 15 nutzen. Falls bei diesen Operationen der Bediener einer Vorrichtung 12(m) oder ein Programm, welches durch eine Vorrichtung 12(m) verarbeitet wird, möchte, daß die Vorrichtung 12(m) ein Nachrichtenpaket an einen Server

31(s) in dem virtuellen privaten Netzwerk 15 überträgt, und falls der Bediener durch die Bedienerchnittstelle 20 oder das Programm eine Klartext-Internetadresse bereitstellt, wird zunächst die Vorrichtung 12(m), insbesondere der Paketgenerator 22, bestimmen, ob der IP-Parameterspeicher 25 dort in einem Cache eine Zahlen-Internetadresse gespeichert hat, welche zu der Klartext-Internetadresse gehört. Falls dies nicht der Fall ist, erzeugt der Paketgenerator 22 ein Anfragenachrichtenpaket zur Übertragung an den Namen-Server 17, um von diesem die zu der Klartext-Internetadresse gehörige Zahlen-Internetadresse anzufordern. Falls der Namen-Server 17 eine zu der Klartext-Internetadresse gehörige Zahlen-Internetadresse besitzt, wird dieser die Zahlen-Internetadresse an die Vorrichtung 12(m) liefern. Es versteht sich, daß dies nur erfolgen kann, wenn die Klartext-Internetadresse im Anfragenachrichtenpaket sowohl einer Vorrichtung 13 außerhalb des virtuellen privaten Netzwerkes 15 als auch einem Server 32(s) in dem virtuellen privaten Netzwerk 15 zugeordnet wurde. Danach kann die Vorrichtung 12(m) die Zahlen-Internetadresse verwenden, um Nachrichtenpakete zur Übertragung über das Internet zu erzeugen, wie es oben beschrieben wurde.

Falls andererseits angenommen wird, daß der Namen-Server 17 keine der Klartext-Internetadresse zugeordnete Zahlen-Internetadresse besitzt, wird der Namen-Server 17 ein entsprechend lautendes Antwortnachrichtenpaket an die Vorrichtung 12(m) übermitteln. Sodann erzeugt der Paketgenerator 22 der Vorrichtung 12(m) ein Anfragenachrichtenpaket zur Übertragung an den nächsten Namen-Server, der in ihrem IP-Parameterspeicher 25 identifiziert ist, um von diesem Namen-Server die der Klartext-Internetadresse zugeordnete Zahlen-Internetadresse anzufordern. Falls dieser nächste Namen-Server der Namen-Server 32 ist, liefert der Paketgenerator 22 das Nachrichtenpaket an den Sicherheits-Paketprozessor 26 zur weiteren Verarbeitung. Der Sicherheits-Paketprozessor 26 erzeugt daraufhin ein Anfragenachrichtenpaket zur Übertragung über den Sicherheitstunnel an die Firewall 30. Diese Nachricht hat im allgemeinen folgende Struktur:

```
"<IIA(DEV_12(m)),IIA(FW)><SEC_TUN>
<ENCR<<IIA(DEV_12(m)),IIA(NS_32)>><IIA_REQ>>>"
```

wobei

- (i) "IIA(DEV\_12(m))" die Quellenadresse darstellt, d. h. die Zahlen-Internetadresse der Vorrichtung 12(m),
- (ii) "IIA(FW)" die Zieladresse darstellt, d. h. die Zahlen-Internetadresse der Firewall 30,
- (iii) "IIA(NS\_32)" die Adresse des Namen-Servers 32 darstellt,
- (iv) "<<IIA(DEV\_12(m)),IIA(NS\_32)>><IIA\_REQ>>" das Anfragenachrichtenpaket darstellt, welches durch den Paketgenerator 22 erzeugt wird, wobei "<IIA(DEV\_12(m)),IIA(NS\_32)>" den Kopfabschnitt des Anfragenachrichtenpakets und "<IIA\_REQ>" den Datenabschnitt des Anfragenachrichtenpakets darstellt,
- (v) "ENCR<...>" angibt, daß die Information zwischen den Klammern "<" und ">" verschlüsselt ist, und
- (vi) "<SEC\_TUN>" einen Hinweis in dem Kopfabschnitt des Nachrichtenpakets darstellt, welches durch den Sicherheitspaketgenerator 26 erzeugt wird und angibt, daß die Nachricht über den Sicherheitstunnel übertragen wird, wobei hierdurch angegeben wird, daß der Datenabschnitt der Nachricht verschlüsselte Information enthält.

Wenn die Firewall 30 das durch den Sicherheitspaketgenerator 26 erzeugte Anfragennachrichtenpaket empfängt, wird diese den verschlüsselten Abschnitt des Nachrichtenpakets entschlüsseln, um "`<<IIA(DEV_12(m)),IIA(NS_32)>><IIA_REQ>>`" zu erhalten. Dies stellt das Anfragennachrichtenpaket dar, welches durch den Paketgenerator 22 erzeugt wird. Nachdem das Anfragennachrichtenpaket erhalten wurde, überträgt die Firewall 30 dieses über die Übertragungsverbindung 33 an den Namen-Server 32. In Abhängigkeit von dem Protokoll zur Übertragung von Nachrichtenpaketen über die Übertragungsverbindung 33 kann es bei diesem Prozeß für die Firewall 30 notwendig sein, das Anfragennachrichtenpaket zu modifizieren, damit es dem Protokoll der Übertragungsverbindung 33 entspricht.

Nachdem der Namen-Server 32 das Anfragennachrichtenpaket erhalten hat, wird dieser das Anfragennachrichtenpaket verarbeiten, um zu bestimmen, ob er eine der Klartext-Internetadresse, welche in dem Anfragennachrichtenpaket gesendet wird, zugeordnete Zahlen-Internetadresse besitzt. Falls der Namen-Server feststellt, daß er eine solche Zahlen-Internetadresse aufweist, wird dieser ein Antwortnachrichtenpaket zur Übertragung an die Firewall erzeugen, welches die Zahlen-Internetadresse enthält. Im allgemeinen hat das Antwortnachrichtenpaket die folgende Struktur:

```
"<<IIA(NS_32),IIA(DEV_12(m))>><IIA_RESP>>"
```

wobei

- (i) "IIA(NS\_32)" die Quellenadresse darstellt, d. h. die Zahlen-Internetadresse des Namen-Servers 32,
- (ii) "IIA(DEV\_12(m))" die Zieladresse darstellt, d. h. die Zahlen-Internetadresse der Vorrichtung 12(m), und
- (iii) "IIA\_RESP" die Zahlen-Internetadresse darstellt, welche der Klartext-Internetadresse zugeordnet ist.

Nachdem die Firewall 30 das Antwortnachrichtenpaket empfangen hat, und weil die Kommunikation mit der Vorrichtung 12(m) über den dazwischenliegenden Sicherheitstunnel stattfindet, verschlüsselt die Firewall 30 das von dem Namen-Server 32 empfangene Antwortnachrichtenpaket und erzeugt ein Nachrichtenpaket zur Übertragung an die Vorrichtung 12(m), welches das verschlüsselte Antwortnachrichtenpaket enthält. Im allgemeinen hat das durch die Firewall 30 erzeugte Nachrichtenpaket die folgende Struktur:

```
"<IIA(FW),IIA(DEV12(m))><SEC_TUN)>  
<ENCR<<IIA(NS_32),IIA(DEV_12(m))>><IIA_RESP>>  
>"
```

wobei

- (i) "IIA(FW)" die Quellenadresse darstellt, d. h. die Zahlen-Internetadresse der Firewall 30,
- (ii) "IIA(DEV\_12(m))" die Zieladresse darstellt, d. h. die Zahlen-Internetadresse der Vorrichtung 12(m),
- (iii) "SEC\_TUN" einen Hinweis in dem Kopfabschnitt des Nachrichtenpakets darstellt, welches durch den Sicherheitspaketgenerator 26 erzeugt wird, und angibt, daß die Nachricht über den Sicherheitstunnel übertragen wird, und wobei auch angegeben wird, daß der Datenabschnitt der Nachricht verschlüsselte Information enthält,
- (iv) "ENCR<...>" angibt, daß die Information zwischen den Klammern "<" und ">" (was dem von dem Namen-Server 32 empfangenen Antwortnachrichten-

paket entspricht) verschlüsselt ist.

Zusätzlich kann es je nach dem Protokoll zur Übertragung von Nachrichtenpaketen über die Übertragungsverbindung 33 für die Firewall 30 notwendig sein, das Nachrichtenpaket zu bearbeiten und/oder zu modifizieren, damit dieses dem Protokoll des Internets 14 entspricht.

Wenn die Vorrichtung 12(m) das Nachrichtenpaket von der Firewall 30 empfängt, wird das Nachrichtenpaket an den Sicherheits-Paketprozessor 26 geliefert. Der Sicherheitspaketprozessor 26 entschlüsselt daraufhin den verschlüsselten Abschnitt des Nachrichtenpakets, um die der Klartext-Internetadresse zugeordnete Zahlen-Internetadresse zu erhalten und lädt diese Information in den IP-Parameterspeicher 25. Danach kann die Vorrichtung diese Zahlen-Internetadresse beim Erzeugen von Nachrichtenpaketen zur Übertragung an den Server 31(s) verwenden, welcher zu der Klartext-Internetadresse gehört.

Es versteht sich, daß, falls der Namen-Server 32 keine Zahlen-Internetadresse besitzt, welche der durch die Vorrichtung 12(m) in dem Anfragennachrichtenpaket gelieferte Klartext-Internetadresse zugeordnet ist, dies der Namen-Server 32 in dem durch ihn erzeugten Antwortnachrichtenpaket entsprechend anzeigen. Die Firewall 30 erzeugt dann in Reaktion auf das durch den Namen-Server 32 gelieferte Antwortnachrichtenpaket auch ein Nachrichtenpaket zur Übertragung an die Vorrichtung 12(m), welches einen verschlüsselten Abschnitt enthält, der das Antwortnachrichtenpaket umfaßt, das durch den Namen-Server 32 erzeugt wurde. Nachdem die Vorrichtung 12(m) das Nachrichtenpaket empfangen hat, wird der verschlüsselte Abschnitt durch den Sicherheitspaketprozessor 26 entschlüsselt, welcher daraufhin den Paketgenerator 22 darüber informiert, daß der Namen-Server 32 keine der Klartext-Internetadresse zugeordnete Zahlen-Internetadresse besitzt. Falls der IP-Parameterspeicher 25 die Identifizierung eines anderen Namen-Servers enthält, erzeugt sodann der Paketgenerator 22 der Vorrichtung 12(m) ein Anfragennachrichtenpaket zur Übertragung an den nächsten Namen-Server, der in deren IP-Parameterspeicher 25 identifiziert ist, um von diesem Namen-Server die Zahlen-Internetadresse anzufordern, welche der Klartext-Internetadresse zugeordnet ist. Falls andererseits der IP-Parameterspeicher 25 keine Identifizierung eines anderen Namen-Servers enthält, kann der Paketgenerator 22 die Bedienerschnittstelle 20 oder ein Programm darüber informieren, daß er nicht in der Lage ist, ein Nachrichtenpaket zur Übertragung an eine Vorrichtung zu erzeugen, welche der Klartext-Internetadresse zugeordnet ist, welche durch die Bedienerschnittstelle 20 oder ein Programm eingegeben bzw. bereitgestellt wurde.

Die Erfindung liefert eine Anzahl von Vorteilen. Insbesondere schafft die Erfindung ein System zum Vereinfachen der Kommunikation zwischen Vorrichtungen, welche mit einem öffentlichen Netzwerk verbunden sind, z. B. mit dem Internet 14, und Vorrichtungen, welche mit privaten Netzwerken verbunden sind, z. B. mit dem virtuellen privaten Netzwerk 15, indem die Umwandlung von Klartextadressen in Netzwerkadressen durch einen Namen-Server, der bevorzugt über einen Sicherheitstunnel mit den privaten Netzwerken verbunden ist, ermöglicht wird.

Es versteht sich, daß eine Vielzahl von Modifikationen an der im Zusammenhang mit Fig. 1 beschriebenen Anordnung durchgeführt werden können. Obwohl das Netzwerk 10 so beschrieben wurde, daß die Identifizierung der Verschlüsselungs- und Entschlüsselungsalgorithmen und -schlüssel durch die Vorrichtung 12(m) und die Firewall 30 während des Dialogs, währenddessen der Sicherheitstunnel eingerichtet wird, ausgetauscht wird, versteht es sich, daß bei-

spielsweise Information durch die Vorrichtung 12(m) und die Firewall 30 getrennt von dem Aufbau eines solchen Sicherheitstunnels bereitgestellt werden können.

Obwohl die Erfindung im Zusammenhang mit dem Internet beschrieben wurde, versteht es sich ferner, daß die Erfindung in Verbindung mit jedem, insbesondere globalen, Netzwerk verwendet werden kann. Obwohl die Erfindung im Zusammenhang mit einem Netzwerk beschrieben wurde, welches ein System von Klartext-Netzwerkadressen bereitstellt, versteht es sich ferner, daß die Erfindung nicht darauf beschränkt ist sondern in Verbindung mit jedem Netzwerk verwendet werden kann, welches irgendeine Form einer – den systemeigenen Netzwerkadressen übergeordnete – Sekundär-Netzwerkadresseneinrichtung oder vergleichbare nicht-formeller Netzwerkadresseneinrichtung vorsieht.

Es versteht sich ferner, daß ein erfindungsgemäßes System als ganzes oder in Teilen aus speziell hierfür geeigneter Hardware oder einem allgemein geeigneten Computersystem oder jeder Kombination davon aufgebaut werden kann, wobei jeder Abschnitt davon durch ein geeignetes Programm gesteuert werden kann. Jedes Programm kann als ganzes oder in Teilen einen Teil des Systems umfassen oder auf dem System in einer konventionellen Weise gespeichert sein, oder es kann als ganzes oder in Teilen in das System über ein Netzwerk oder andere Mechanismen zur Übertragung von Information in einer konventionellen Weise bereitgestellt werden. Zusätzlich versteht es sich, daß das System betrieben und/oder auf andere Art und Weise mittels Information gesteuert werden kann, welche durch einen Bediener mittels Bedieneringabeelementen (nicht gezeigt) bereitgestellt wird, welche direkt an das System angeschlossen sein können oder welche die Information über ein Netzwerk oder andere Mechanismen zur Übertragung von Information in einer konventionellen Weise übertragen können.

Die vorstehende Beschreibung hat sich auf ein spezifisches Ausführungsbeispiel der Erfindung bezogen. Es versteht sich jedoch, daß verschiedene Variationen und Modifikationen der Erfindung gemacht werden können, bei welchen einige oder alle der Vorteile der Erfindung erreicht werden. Diese und andere Variationen und Modifikationen fallen in den Schutzbereich der vorliegenden Erfindung, der durch die nachfolgenden Ansprüche bestimmt ist.

#### Patentansprüche

1. System umfassend ein virtuelles privates Netzwerk (15) und eine externe Vorrichtung (12 (m)), welche über ein digitales Netzwerk (14) kommunizieren, wobei:  
das virtuelle private Netzwerk (15) eine Firewall (30), wenigstens eine interne Vorrichtung (31(s)) und einen Namen-Server (32) aufweist, welche jeweils eine Netzwerkadresse besitzen, wobei die interne Vorrichtung (31(s)) auch eine Sekundäradresse besitzt und der Namen-Server (32) derart konfiguriert ist, daß er eine Zuordnung zwischen der Sekundäradresse und der Netzwerkadresse bereitstellt,  
die Firewall (30) derart konfiguriert ist, daß sie der externen Vorrichtung (12(m)) in Reaktion auf deren Anfrage zum Aufbau einer Verbindung zur Firewall (30) die Netzwerkadresse des Namen-Servers (32) liefert, und  
die externe Vorrichtung (12(m)) derart konfiguriert ist, daß sie in Reaktion auf eine Anfrage zum Zugriff auf die interne Vorrichtung (31(s)), welche die Sekundäradresse der internen Vorrichtung (31(s)) enthält, eine Netzwerkadressen-Anfragennachricht zur Übertragung über die Verbindung an die Firewall (30) erzeugt, wel-

che eine Auflösung der der Sekundäradresse zugeordneten Netzwerkadresse erfordert, wobei die Firewall (30) derart konfiguriert ist, daß sie die Adressenauflosungsanfrage an den Namen-Server (32) übermittelt, der Namen-Server (32) derart konfiguriert ist, daß er die der Sekundäradresse zugeordnete Netzwerkadresse bereitstellt, und die Firewall (30) daraufhin die Netzwerkadresse in einer Netzwerkadressen-Antwortnachricht zur Übertragung über die Verbindung an die externe Vorrichtung (12(m)) bereitstellt.

2. System nach Anspruch 1, bei welchem die externe Vorrichtung (12(m)) derart konfiguriert ist, daß sie die in der Netzwerkadressen-Antwortnachricht bereitgestellte Netzwerkadresse beim Erzeugen von wenigstens einer Nachricht zur Übertragung an die interne Vorrichtung (31(s)) verwendet.

3. System nach Anspruch 1 oder 2, bei welchem die externe Vorrichtung (12(m)) derart konfiguriert ist, daß sie mit dem Netzwerk (14) durch einen Netzwerk-Service-Provider (11) verbunden wird.

4. System nach Anspruch 3, bei welchem die externe Vorrichtung (12(m)) derart konfiguriert ist, daß sie eine Kommunikationssitzung mit dem Netzwerk-Service-Provider (11) aufbaut, wobei der Netzwerk-Service-Provider (11) der externen Vorrichtung (12(m)) die Identifizierung eines weiteren Namen-Servers übermittelt, wobei der weitere Namen-Server derart konfiguriert ist, daß er eine Zuordnung zwischen einer Sekundäradresse und einer Netzwerkadresse für wenigstens eine Vorrichtung bereitstellt.

5. System nach einem der vorstehenden Ansprüche, bei welchem die externe Vorrichtung (12(m)) derart konfiguriert ist, daß sie eine Liste von Namen-Servern erhält, welche der externen Vorrichtung (12(m)) identifiziert wurden, und die externe Vorrichtung (12(m)) die Namen-Server in der Liste nacheinander in Reaktion auf eine Anfrage zum Zugriff auf eine andere Vorrichtung abfragt, wobei die Anfrage eine Sekundäradresse der anderen Vorrichtung enthält, solange bis die externe Vorrichtung (12(m)) eine Netzwerkadresse empfängt, wobei die externe Vorrichtung (12(m)) in jedem Abfragevorgang eine Netzwerkadressen-Anfragennachricht zur Übertragung über das Netzwerk (14) erzeugt, welche durch einen der Namen-Server in der Liste zu beantworten ist, und von diesem eine Netzwerkadressen-Antwortnachricht empfängt.

6. System nach einem der vorstehenden Ansprüche, bei welchem die Verbindung zwischen der externen Vorrichtung (12(m)) und der Firewall (30) ein Sicherheitstunnel ist, in welchem wenigstens ein der zwischen der externen Vorrichtung (12(m)) und der Firewall (30) übertragenen Nachrichten verschlüsselt ist.

7. Verfahren zum Betreiben eines Systems umfassend ein virtuelles privates Netzwerk (15) und eine externe Vorrichtung (12(m)), welche durch ein digitales Netzwerk (14) miteinander verbunden sind, wobei das virtuelle private Netzwerk (15) eine Firewall (30), wenigstens eine interne Vorrichtung (31(s)) und einen Namen-Server (32) aufweist, welche jeweils eine Netzwerkadresse besitzen, wobei die interne Vorrichtung (31(s)) auch eine Sekundäradresse besitzt, und der Namen-Server (32) derart konfiguriert ist, daß er eine Zuordnung zwischen der Sekundäradresse und der Netzwerkadresse bereitstellt, wobei:

A. in Reaktion auf eine Anfrage der externen Vorrichtung (12(m)) zum Aufbau einer Verbindung zur Firewall (30) die Firewall (30) der externen Vorrichtung (12(m)) die Netzwerkadresse des

Namen-Servers (32) übermittelt; und

B. (i) in Reaktion auf eine Anfrage zum Zugriff auf die interne Vorrichtung (31(s)), welche die Sekundäradresse der internen Vorrichtung (31(s)) enthält, die externe Vorrichtung (12(m)) eine Netzwerkadressen-Anfragenachricht zur Übertragung über die Verbindung an die Firewall (30) erzeugt, welche eine Auflösung der Netzwerkadresse, welche der Sekundäradresse zugeordnet ist, anfordert,

(ii) die Firewall (30) die Adressenauflösungsanfrage an den Namen-Server (32) übermittelt, (iii) der Namen-Server (32) die der Sekundäradresse zugeordnete Netzwerkadresse bereitstellt, und

(iv) die Firewall (30) die Netzwerkadresse in einer Netzwerkadressen-Antwortnachricht zur Übertragung über die Verbindung an die externe Vorrichtung (12(m)) bereitstellt.

8. Verfahren nach Anspruch 7, bei welchem die externe Vorrichtung (12(m)) ferner die in der Netzwerkadressen-Antwortnachricht bereitgestellte Netzwerkadresse beim Erzeugen von wenigstens einer Nachricht zur Übertragung an die interne Vorrichtung (31(s)) verwendet.

9. Verfahren nach Anspruch 7 oder 8, bei welchem die externe Vorrichtung (12(m)) mit dem Netzwerk (14) durch einen Netzwerk-Service-Provider (11) verbunden werden kann.

10. Verfahren nach Anspruch 9, bei welchem die externe Vorrichtung (12(m)) eine Kommunikationssitzung mit dem Netzwerk-Service-Provider (11) aufbaut, wobei der Netzwerk-Service-Provider (11) der externen Vorrichtung (12(m)) die Identifizierung eines weiteren Namen-Servers übermittelt, wobei der weitere Namen-Server eine Zuordnung zwischen einer Sekundäradresse und einer Netzwerkadresse für wenigstens eine Vorrichtung bereitstellt.

11. Verfahren nach einem der Ansprüche 7 bis 10, bei welchem die externe Vorrichtung (12(m)) eine Liste von Namen-Servern erhält, welche der externen Vorrichtung (12(m)) identifiziert wurden, und die externe Vorrichtung (12(m)) die Namen-Server in der Liste nacheinander in Reaktion auf eine Anfrage zum Zugriff auf eine andere Vorrichtung abfragt, wobei die Anfrage eine Sekundäradresse der anderen Vorrichtung enthält, solange bis die externe Vorrichtung (12(m)) eine Netzwerkadresse empfängt, wobei die externe Vorrichtung (12(m)) in jedem Abfragevorgang eine Netzwerkadressen-Anfragenachricht zur Übertragung über das Netzwerk (14) erzeugt, welche durch einen der Namen-Server in der Liste zu beantworten ist, und von diesem eine Netzwerkadressen-Antwortnachricht empfängt.

12. Verfahren nach einem der Ansprüche 7 bis 11, bei welchem die Verbindung zwischen der externen Vorrichtung (12(m)) und der Firewall (30) ein Sicherheitstunnel ist, in welchem wenigstens ein Abschnitt der zwischen der externen Vorrichtung (12(m)) und der Firewall (30) übertragenen Nachrichten verschlüsselt ist.

13. Computerprogramm-Produkt zur gemeinsamen Verwendung mit einem virtuellen privaten Netzwerk (15) und einer externen Vorrichtung (12(m)), welche durch ein digitales Netzwerk (14) miteinander verbunden sind, wobei das virtuelle private Netzwerk eine Firewall (30), wenigstens eine interne Vorrichtung (31(s)) und einen Namen-Server (32) aufweist, welche jeweils eine Netzwerkadresse besitzen, wobei die interne Vorrichtung (31(s)) auch eine Sekundäradresse

besitzt, und der Namen-Server (32) derart konfiguriert ist, daß er eine Zuordnung zwischen der Sekundäradresse und der Netzwerkadresse bereitstellt, wobei das Computerprogrammprodukt ein maschinenlesbares Medium mit folgenden Codes aufweist:

A. ein Namen-Server-Identifizierungscodemodul, welches veranlaßt, daß die Firewall (30) der externen Vorrichtung (12(m)) in Reaktion auf deren Anfrage zum Aufbau einer Verbindung zur Firewall (30) die Netzwerkadresse des Namen-Servers (32) übermittelt,

B. ein Codemodul zur Erzeugung einer Netzwerkadressen-Anfragenachricht, welches veranlaßt, daß die externe Vorrichtung (12(m)) in Reaktion auf eine Anfrage zum Zugriff auf die interne Vorrichtung (31(s)), welche die Sekundäradresse der internen Vorrichtung (31(s)) enthält, eine Netzwerkadressen-Anfragenachricht zur Übertragung über die Verbindung an die Firewall (30) erzeugt, welche die Auflösung der der Sekundäradresse zugeordneten Netzwerkadresse anfordert,

C. ein Modul zur Übermittlung einer Adressenauflösungsanfrage, welches veranlaßt, daß die Firewall (30) die Adressenauflösungsanfrage an den Namen-Server (32) übermittelt,

D. ein Namen-Server-Steuerungsmodul, welches veranlaßt, daß der Namen-Server (32) die der Sekundäradresse zugeordnete Netzwerkadresse bereitstellt, und

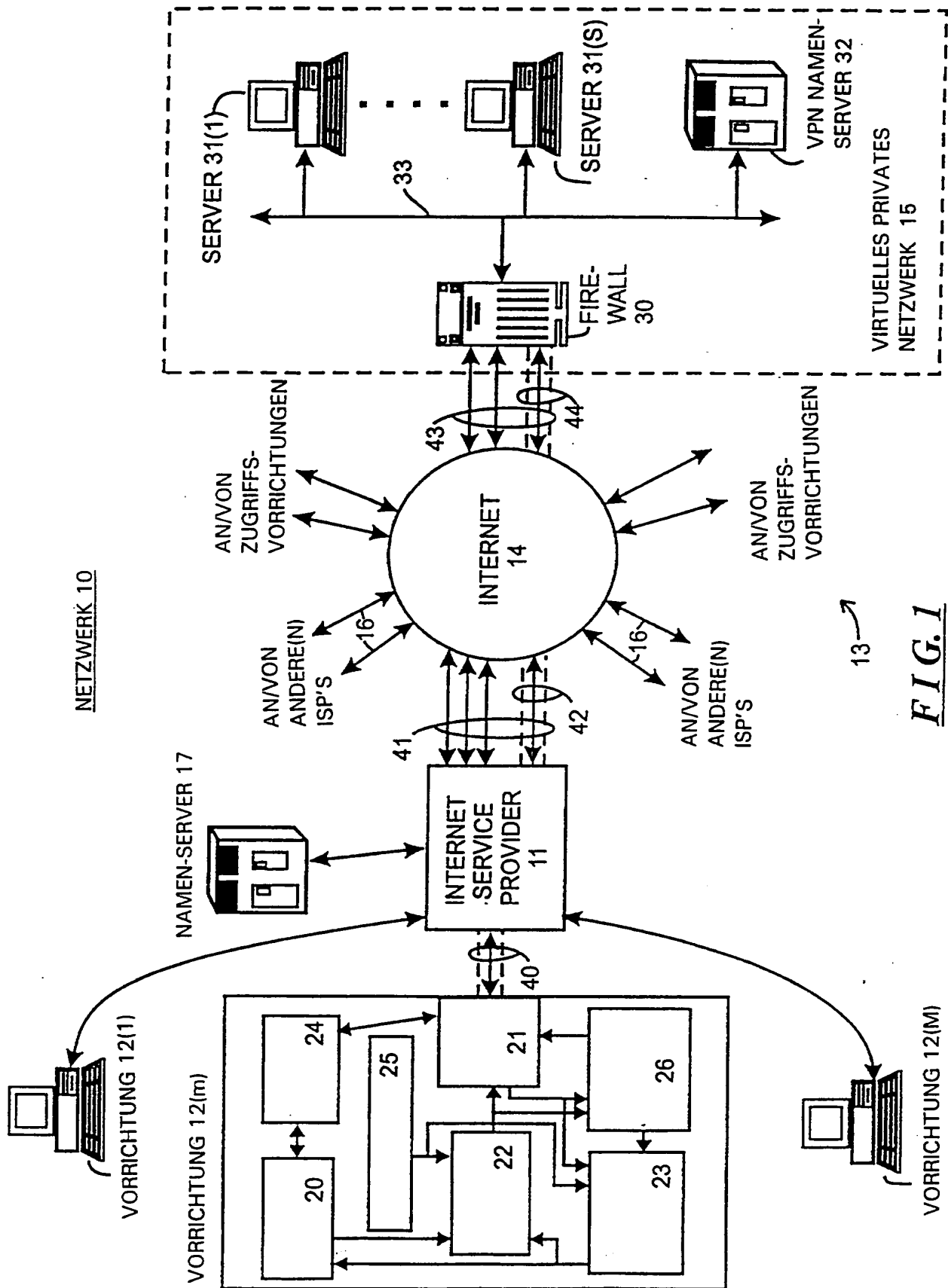
E. ein Modul zur Übermittlung einer Netzwerkadressen-Antwortnachricht, welches veranlaßt, daß die Firewall (30) die Netzwerkadresse in einer Netzwerkadressen-Antwortnachricht zur Übertragung über die Verbindung an die externe Vorrichtung (12(m)) bereitstellt.

14. Computerprogramm-Produkt nach Anspruch 13, welches ferner ein Netzwerkadressenverwendungsmodul aufweist, welches veranlaßt, daß die externe Vorrichtung (12(m)) die in der Netzwerkadressen-Antwortnachricht übermittelte Netzwerkadresse beim Erzeugen von wenigstens einer Nachricht zur Übertragung an die interne Vorrichtung (31(s)) verwendet.

15. Computerprogramm-Produkt nach Anspruch 13 oder 14, welches ferner ein Netzwerk-Service-Provider-Steuerungsmodul aufweist, welches veranlaßt, daß die externe Vorrichtung (12(m)) mit dem Netzwerk (14) durch einen Netzwerk-Service-Provider (11) verbunden wird.

16. Computerprogramm-Produkt nach Anspruch 15, bei welchem das Netzwerk-Service-Provider-Steuerungsmodul ein Kommunikationssitzungsaufbaumodul umfaßt, welches veranlaßt, daß die externe Vorrichtung (12(m)) mit dem Netzwerk-Service-Provider (11) eine Kommunikationssitzung aufbaut und von diesem eine Identifizierung von einem weiteren Namen-Server empfängt.

17. Computerprogramm-Produkt nach einem der Ansprüche 13 bis 16, welches ferner ein Namen-Server-Abfragesteuerungsmodul aufweist, welches veranlaßt, daß die externe Vorrichtung (12(m)) eine Liste von Namen-Servern erhält, welche der externen Vorrichtung (12(m)) identifiziert wurden, und die Namen-Server in der Liste nacheinander in Reaktion auf eine Anfrage zum Zugriff auf eine andere Vorrichtung abfragt, wobei die Anfrage eine Sekundäradresse der anderen Vorrichtung enthält, solange bis die externe Vorrichtung (12(m)) eine Netzwerkadresse empfängt, und wobei die externe Vorrichtung (12(m)) in jedem Abfragevor-



13 → **FIG. 1**

gang eine Netzwerkadressen-Anfragesnachricht zur Übertragung über das Netzwerk (14) erzeugt, welche durch einen der Namen-Server in der Liste zu beantworten ist, und von diesem eine Netzwerkadressen-Antwortnachricht empfängt.

5

18. Computerprogramm-Produkt nach einem der Ansprüche 13 bis 17, bei welchem die Verbindung zwischen der externen Vorrichtung (12(m)) und der Firewall (30) ein Sicherheitstunnel ist, in welchem wenigstens ein Abschnitt der zwischen der externen Vorrichtung (12(m)) und der Firewall (30) übertragenen Nachrichten verschlüsselt ist.

10

Hierzu 1 Seite(n) Zeichnungen

15

20

25

30

35

40

45

50

55

60

65

(12) UK Patent Application (19) GB (11) 2 317 792 (13) A

(43) Date of A Publication 01.04.1998

(21) Application No 9719816.2

(22) Date of Filing 17.09.1997

(30) Priority Data

(31) 08715343 (32) 18.09.1996 (33) US  
08715668 18.09.1996

(71) Applicant(s)

Secure Computing Corporation

(Incorporated in USA - Delaware)

2675 Long Lake Road, Roseville,  
Minnesota 55113-2536, United States of America

(72) Inventor(s)

Spence Minear  
Edward B Stockwell  
Troy De Jongh

(74) Agent and/or Address for Service

Beresford & Co  
2-5 Warwick Court, High Holborn, LONDON,  
WC1R 5DJ, United Kingdom

(51) INT CL<sup>6</sup>  
H04L 9/00

(52) UK CL (Edition P )  
H4P PPEB  
U1S S2124 S2209

(56) Documents Cited

WO 97/26735 A1 WO 97/26734 A1 WO 97/26731 A1  
WO 97/23972 A1 WO 97/13340 A1

(58) Field of Search

UK CL (Edition P ) H4P PDCSA PDCSC PPEB  
INT CL<sup>6</sup> H04L 9/00 9/32 29/06 29/08  
Online: WPI, INSPEC

(54) Virtual Private Network for encrypted firewall

(57) A system (10) for regulating the flow of messages through a firewall (18) having a network protocol stack, wherein the network protocol stack includes an Internet Protocol (IP) layer where if the message is not encrypted, it passes the unencrypted message up the network protocol stack to an application level proxy (50), and if the message is encrypted, it decrypts the message and passes the decrypted message up the network protocol stack to the application level proxy. The step of decrypting the message includes the step of executing a process at the IP layer to decrypt the message.

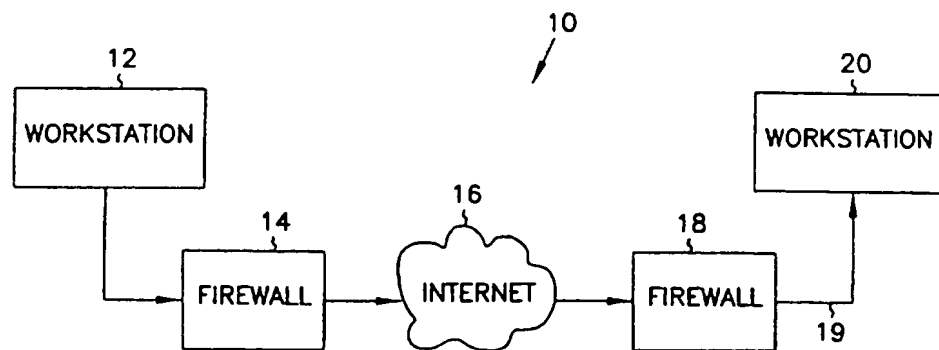


FIG. 1

At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

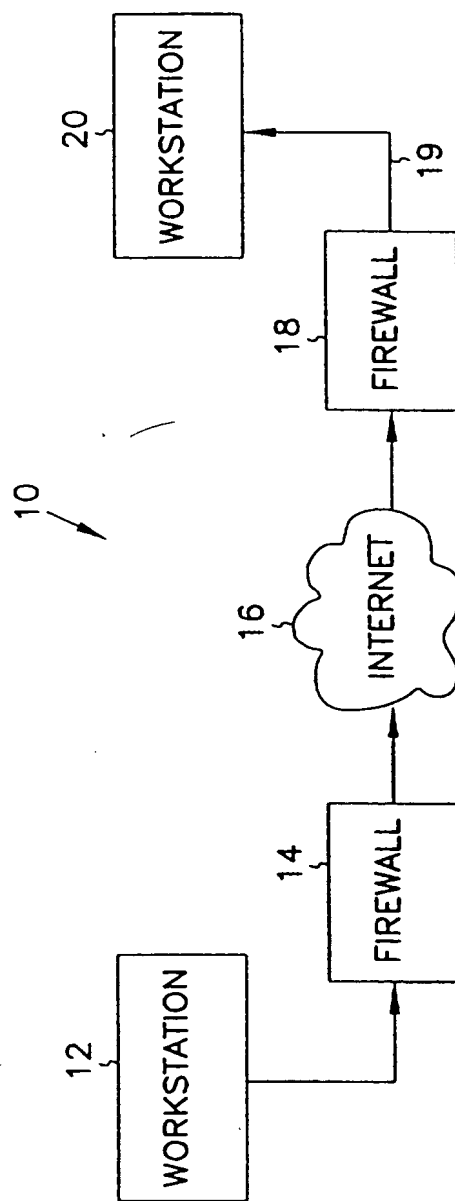


FIG. 1



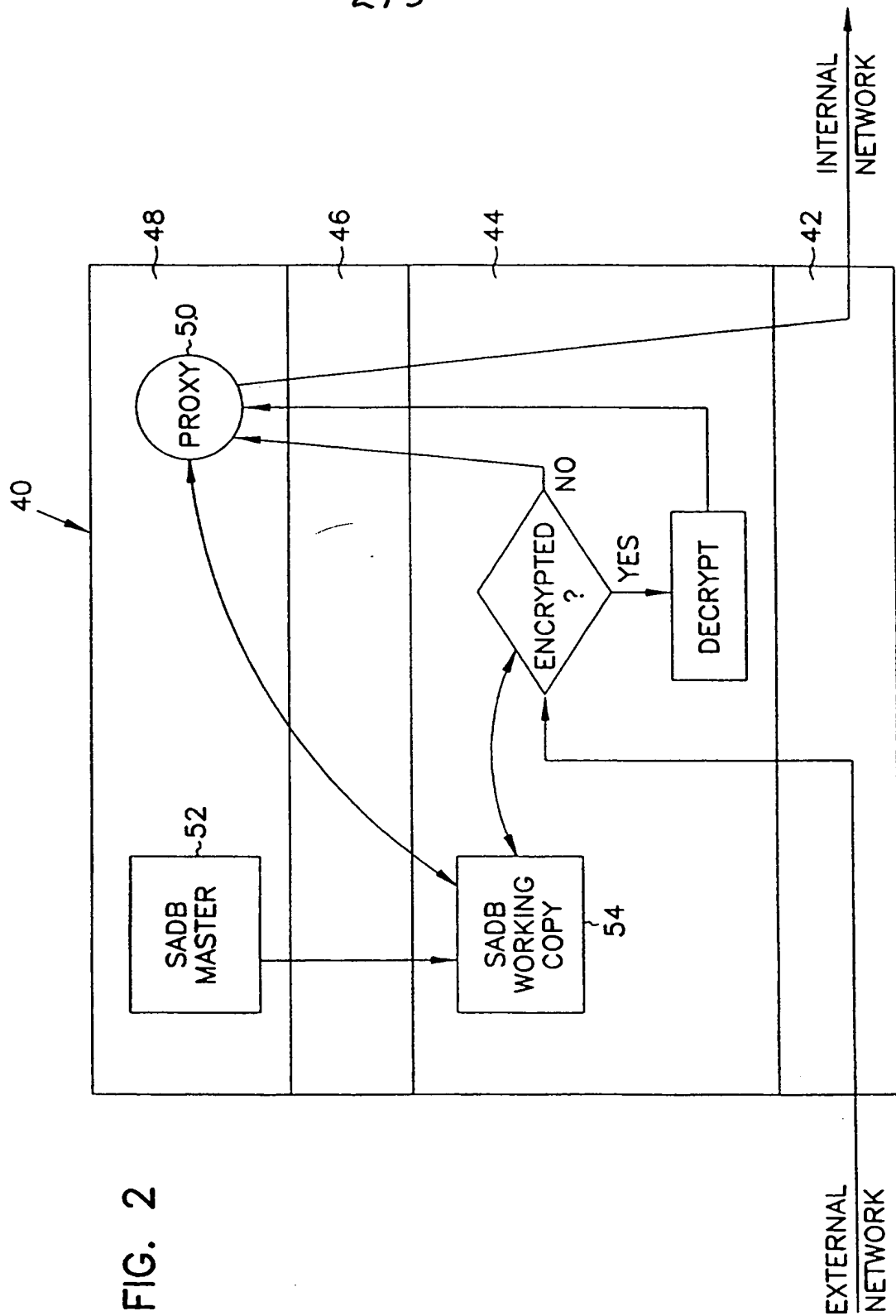
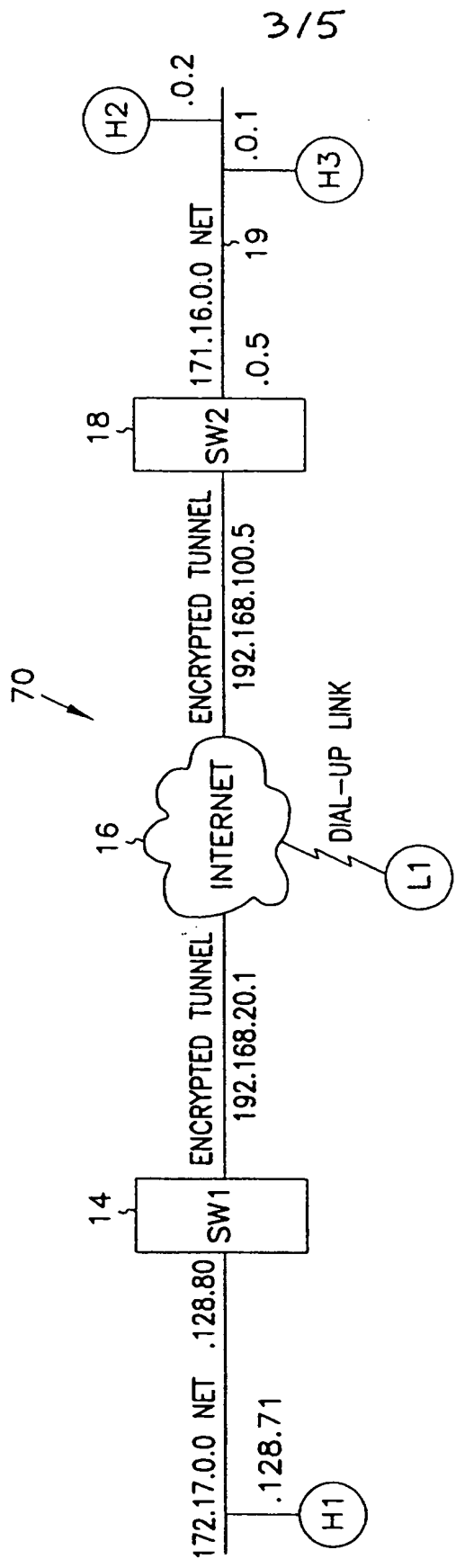


FIG. 2



3/5

FIG. 3

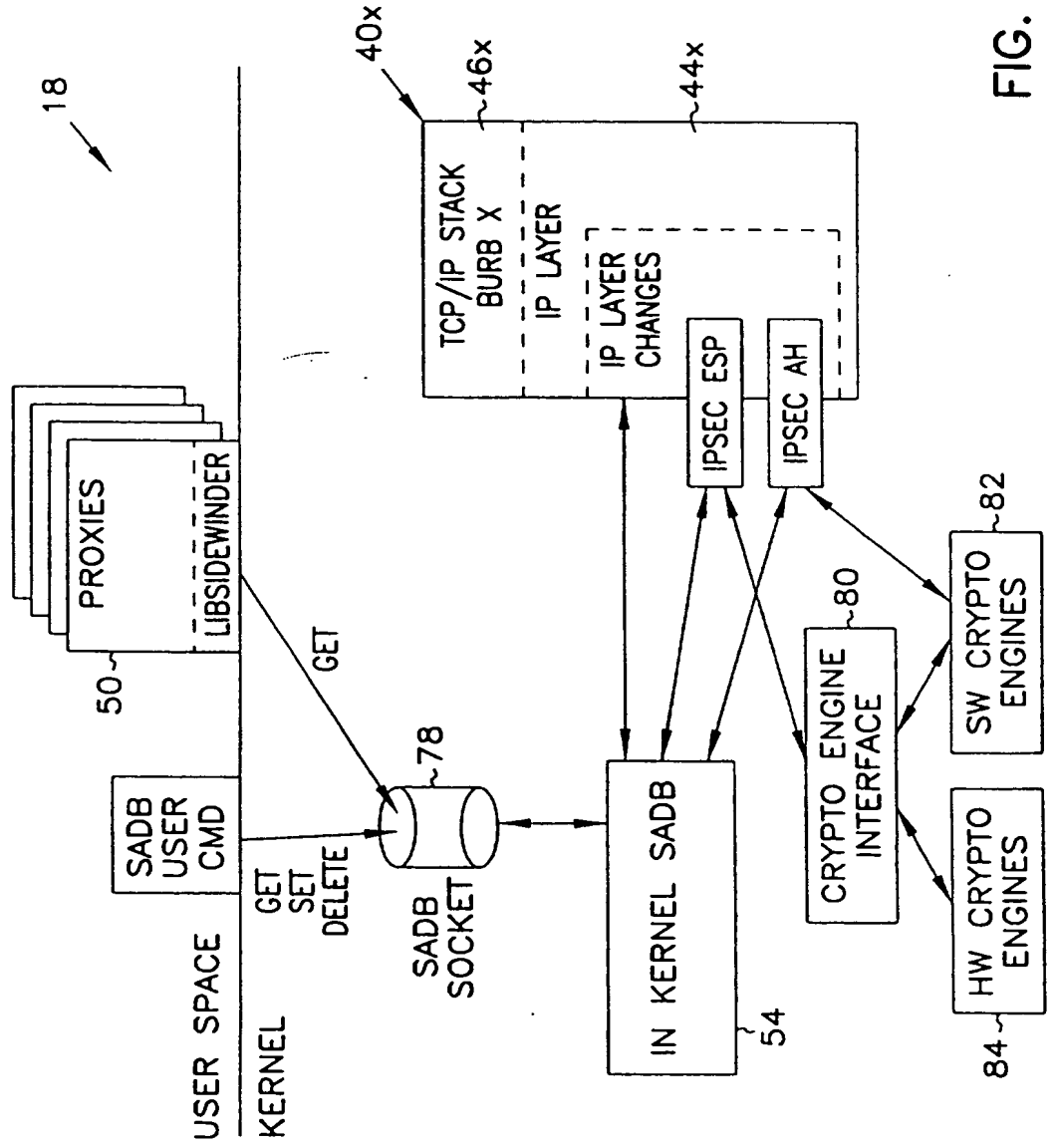


FIG. 4

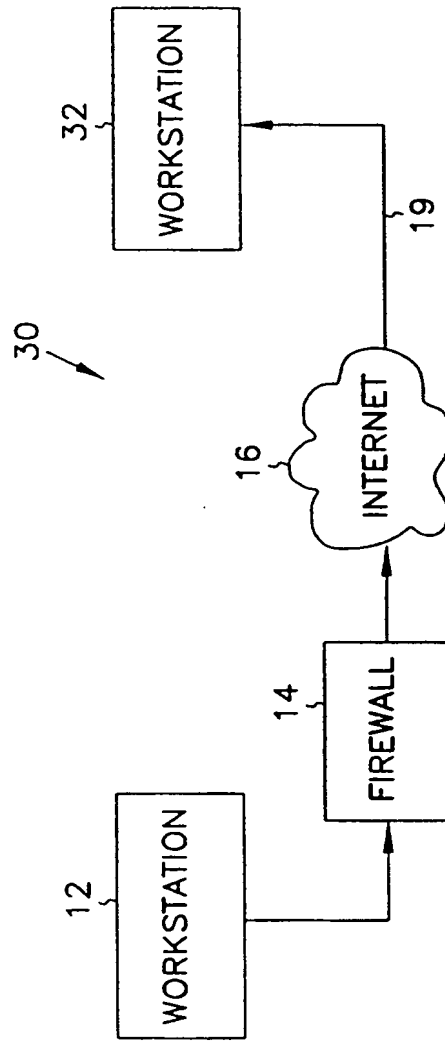


FIG. 5

## VIRTUAL PRIVATE NETWORK ON APPLICATION GATEWAY

5 Background of the InventionField of the Invention

The present invention pertains generally to network communications, and in particular to a system and method for securely transferring information between firewalls over an unprotected network.

10 Background Information

Firewalls have become an increasingly important part of network design. Firewalls provide protection of valuable resources on a private network while allowing communication and access with systems located on an unprotected network such as the Internet. In addition, they operate to block attacks on a private network arriving from the unprotected network by providing a single connection with limited services. A well designed firewall limits the security problems of an Internet connection to a single firewall computer system. This allows an organization to focus their network security efforts on the definition of the security policy enforced by the firewall. An example of a firewall is given in 20 "SYSTEM AND METHOD FOR PROVIDING SECURE INTERNETWORK SERVICES" by Boebert et al. (PCT Published Application No. WO 96/13113, published on May 2, 1996), the description of which is hereby incorporated by reference. Another description of a firewall is provided by Dan Thomsen in "Type Enforcement: the new security model", *Proceedings: Multimedia: Full-* 25 *Service Impact on Business, Education, and the Home*, SPIE Vol. 2617, p. 143, August 1996. Yet another such system is described in "SYSTEM AND METHOD FOR ACHIEVING NETWORK SEPARATION" by Gooderum et al. (PCT Published Application No. WO 97/29413, published on August 14, 1997), the description of which is hereby incorporated by reference. All the above 30 systems are examples of application level gateways. Application level gateways use proxies or other such mechanisms operating at the application layer to process traffic through the firewall. As such, they can review not only the

message traffic but also message content. In addition, they provide authentication and identification services, access control and auditing.

Data to be transferred on unprotected networks like the Internet is susceptible to electronic eavesdropping and accidental (or deliberate) corruption.

5 Although a firewall can protect data within a private network from attacks launched from the unprotected network, even that data is vulnerable to both eavesdropping and corruption when transferred from the private network to an external machine. To address this danger, the Internet Engineering Task Force (IETF) developed a standard for protecting data transferred between firewalls  
10 over an unprotected network. The Internet Protocol Security (IPSEC) standard calls for encrypting data before it leaves the first firewall, and then decrypting the data when it is received by the second firewall. The decrypted data is then delivered to its destination, usually a user workstation connected to the second firewall. For this reason IPSEC encryption is sometimes called *firewall-to-*  
15 *firewall encryption* (FFE) and the connection between a workstation connected to the first firewall and a client or server connected to the second firewall is termed a *virtual private network*, or VPN.

The two main components of IPSEC security are data encryption and sender authentication. Data encryption increases the cost and time required for  
20 the eavesdropping party to read the transmitted data. Sender authentication ensures that the destination system can verify whether or not the encrypted data was actually sent from the workstation that it was supposed to be sent from. The IPSEC standard defines an encapsulated payload (ESP) as the mechanism used to transfer encrypted data. The standard defines an authentication header (AH)  
25 as the mechanism for establishing the sending workstation's identity.

Through the proper use of encryption, the problems of eavesdropping and corruption can be avoided; in effect, a protected connection is established from the internal network connected to one firewall through to an internal network connected to the second firewall. In addition, IPSEC can be used to provide a  
30 protected connection to an external computing system such as a portable personal computer.

IPSEC encryption and decryption work within the IP layer of the network protocol stack. This means that all communication between two IP addresses will be protected because all interfirewall communication must go through the IP layer. Such an approach is preferable over encryption and decryption at higher levels in the network protocol stack since when encryption is performed at layers higher than the IP layer more work is required to ensure that all supported communication is properly protected. In addition, since IPSEC encryption is handled below the Transport layer, IPSEC can encrypt data sent by any application. IPSEC therefore becomes a transparent add-on to such protocols as TCP and UDP.

Since, however, IPSEC decryption occurs at the IP layer, it can be difficult to port IPSEC to an application level gateway while still maintaining control at the proxy over authentication, message content, access control and auditing. Although the IPSEC specification in RFC 1825 suggests the use of a mandatory access control mechanism in a multi-level secure (MLS) network to compare a security level associated with the message with the security level of the receiving process, such an approach provides only limited utility in an application level gateway environment. In fact, implementations on application level gateways to date have simply relied on the fact that the message was IPSEC-encrypted as assurance that the message is legitimate and have simply decoded and forwarded the message to its destination. This creates, however, a potential chink in the firewall by assuming that the encrypted communication has access to all services.

What is needed is a method of handling IPSEC messages within an application level gateway which overcomes the above deficiencies. The method should allow control over access by an IPSEC connection to individual services within the internal network.

#### Summary of the Invention

The present invention is a system and method for regulating the flow of messages through a firewall having a network protocol stack, wherein the network protocol stack includes an Internet Protocol (IP) layer, the method

comprising the steps of determining, at the IP layer, if a message is encrypted, if the message is not encrypted, passing the unencrypted message up the network protocol stack to an application level proxy, and if the message is encrypted, decrypting the message and passing the decrypted message up the network  
5 protocol stack to the application level proxy, wherein the step of decrypting the message includes the step of executing a procedure at the IP layer to decrypt the message.

According to another aspect of the present invention, a system and method is described for authenticating the sender of a message within a  
10 computer system having a network protocol stack, wherein the network protocol stack includes an Internet Protocol (IP) layer, the method comprising the steps of determining, at the IP layer, if the message is encrypted, if the message is encrypted, decrypting the message, wherein the step of decrypting the message includes the step of executing a procedure at the IP layer to decrypt the message,  
15 passing the decrypted message up the network protocol stack to an application level proxy, determining an authentication protocol appropriate for the message, and executing the authentication protocol to authenticate the sender of the message.

#### Brief Description of the Drawings

20 In the following detailed description of example embodiments of the invention, reference is made to the accompanying drawings which form a part hereof, and which is shown by way of illustration only, specific embodiments in which the invention may be practiced. It is to be understood that other embodiments may be utilized and structural changes may be made without  
25 departing from the scope of the present invention.

In the drawings, where like numerals refer to like components throughout the several views:

Figure 1 is a functional block diagram of an application level gateway-  
implemented firewall-to-firewall encryption scheme according to the present  
30 invention;



Figure 2 is a block diagram showing access control checking of both encrypted and unencrypted messages in network protocol stack according to the present invention;

Figure 3 is a block diagram of a representative application level gateway-  
5 implemented firewall-to-firewall encryption scheme;

Figure 4 is a block diagram of one embodiment of a network-separated protocol stack implementing IPSEC according to the present invention; and

Figure 5 is a functional block diagram of a firewall-to-workstation encryption scheme according to the present invention.

10

### Description of the Preferred Embodiments

In the following detailed description of the preferred embodiment, references made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration specific preferred embodiments in which  
15 the invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention, and it is to be understood that other embodiments may be utilized and that structural, logical, physical, architectural, and electrical changes may be made without departing from the spirit and scope of the present invention. The following detailed  
20 description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined only by the appended claims and their equivalents.

A system 10 which can be used for firewall-to-firewall encryption (FFE) is shown in Figure 1. In Figure 1, system 10 includes a workstation 12 communicating through a firewall 14 to an unprotected network 16 such as the  
25 Internet. System 10 also includes a workstation 20 communicating through a firewall 18 to unprotected network 16. In one embodiment, firewall 18 is an application level gateway.

As noted above, IPSEC encryption and decryption work within the IP layer of the network protocol stack. This means that all communications  
30 between two IP addresses will be protected because all interfirewall communication must pass through the IP layer. IPSEC takes the standard

Internet packet and converts it into a carrier packet. The carrier packet is designed to do two things: to conceal the contents of the original packet (encryption) and to provide a mechanism by which the receiving firewall can verify the source of the packet (authentication). In one embodiment of the present invention, each IPSEC carrier packet includes both an authentication header used to authenticate the sending machine and an encapsulated payload containing encrypted data. The authentication header and the encapsulated payload features of IPSEC can, however, be used independently. As required in RFC 1825, DES-CBC is provided for use in encrypting the encapsulated payload while the authentication header uses keyed MD5.

To use IPSEC, you must create a *security association* (SA) for each destination IP address. In one embodiment, each SA contains the following information:

- Security Parameters Index (SPI) - The index used to find a SA on receipt of an IPSEC datagram.
- Destination IP address - The address used to find the SA and trigger use of IPSEC processing on output.
- The peer SPI - The SPI value to put on a IPSEC datagram on output.
- The peer IP address - The destination IP address to be put into the packet header if IPSEC Tunnel mode is used.
- The Encryption Security Payload (ESP) algorithm to be used.
- The ESP key to used for decryption of input datagrams.
- The ESP key to used for encryption of output datagrams.
- The authentication (AH) algorithm to be used.
- The AH key to be used for validation of input packets.
- The AH key to be used for generation of the authentication data for output datagrams.

The combination of a given Security Parameter Index and Destination IP address uniquely identifies a particular "Security Association." In one

embodiment, the sending firewall uses the sending userid and Destination Address to select an appropriate Security Association (and hence SPI value). The receiving firewall uses the combination of SPI value and Source address to obtain the appropriate Security Association.

5           A security association is normally one-way. An authenticated communications session between two firewalls will normally have two Security Parameter Indexes in use (one in each direction). The combination of a particular Security Parameter Index and a particular Destination Address uniquely identifies the Security Association.

10           More information on the specifics of an IPSEC FFE implementation can be obtained from the standards developed by the IPSEC work group and documented in *Security Architecture for IP* (RFC 1825) and in RFC's 1826-1829.

          When a datagram is received from unprotected network 16 or is to be  
15 transmitted to a destination across unprotected network 16, the firewall must be able to determine the algorithms, keys, etc. that must be used to process the datagram correctly. In one embodiment, this information is obtained via a security association lookup. In one such embodiment, the lookup routine is passed several arguments: the source IP address if the datagram is being received  
20 from network 16 or the destination IP address if the datagram is to be transmitted across network 16, the SPI, and a flag that is used to indicate whether the lookup is being done to receive or transmit a datagram.

          When an IPSEC datagram is received by firewall 18 from unprotected network 16, the SPI and source IP address are determined by looking in the  
25 datagram. In one embodiment a Security Association Database (SADB) stored within firewall 18 is searched for the entry with a matching SPI. In one such embodiment, security associations can be set up based on network address as well as a more granular host address. This allows the network administrator to create a security association between two firewalls with only a couple of lines in  
30 a configuration file on each machine. For such embodiments, the entry in the Security Association Database that has both the matching SPI and the longest

address match is selected as the SA entry. In another such embodiment, each SA has a prefix length value associated with the address. An address match on a SA entry means that the addresses match for the number of bits specified by the prefix length value.

5           There are two exceptions to this search process. First, when an SA entry is set marked as being dynamic it implies that the user of this SA may not have a fixed IP address. In this case the match is fully determined by the SPI value. Thus it is necessary that the SPI values for such SA entries be unique in the SADB. The second exception is for SA entries marked as tunnel mode entries.  
10   In this case it is normally the case that the sending entity will hide its source address so that all that is visible on the public wire is the destination address. In this case, like in the case where the SA entries are for dynamic IP addresses, the search is done exclusively on the basis of the SPI.

          When transmitting a datagram across unprotected network 16 the SADB  
15 is searched using only the destination address as an input. In this case the entry which has the longest address match is selected and returned to the calling routine.

          In one embodiment, if firewall 18 receives datagrams which are identified as either an IP\_PROTO\_IPSEC\_ESP or IP\_PROTO\_IPSEC\_AH  
20 protocol datagram, there must be a corresponding SA in the SADB or else firewall 18 will drop the packet and an audit message will be generated. Such an occurrence might indicate a possible attack or it might simply be a symptom of an erroneous key entry in the Security Association Database.

          In a system such as system 10, application level gateway firewall 18 acts  
25 as a buffer between unprotected network 16 and workstations such as workstation 20. Messages coming from unprotected network 16 are reviewed and a determination is made as to whether execution of an authentication and identification protocol is warranted. In contrast to previous systems, system 10 also performs this same determination on IPSEC-encrypted messages. If  
30 desired, the same authentication and identification can be made on messages to be transferred from workstation 20 to unprotected network 16. Figure 2

illustrates one way of authenticating both encrypted and unencrypted messages in a system such as system 10.

In the system of Figure 2 a network protocol stack 40 includes a physical layer 42, an Internet protocol (IP) layer 44, a Transport layer 46 and an application layer 48. Such a protocol stack exists, for instance on application level gateway firewall 18 of Figure 1. An application executing in application layer 48 can communicate to an application executing on another system by preparing a message and transmitting it through one of the existing transport services executing on transport layer 46. Transport layer 46 in turn uses a process executing in IP layer 44 to continue the transfer. Physical layer 42 provides the software needed to transfer data through the communication hardware (e.g., a network interface card or a modem). As noted above, IPSEC executes within IP layer 44. Encryption and authentication is transparent to the host as long as the network administrator has the Security Association Database correctly configured and a key management mechanism is in place on the firewall.

In application level gateway firewall 18, a proxy 50 operating within application layer 48 processes messages transferred between internal and external networks. All network-to-network traffic must pass through one of the proxies within application layer 48 before being the transfer across networks is allowed. A message arriving from external network 16 is examined at IP layer 44 and an SADB is queried to determine if the source address and SPI are associated with an SA. In the embodiment shown in Figure 2, an SADB Master copy 52 is maintained in persistent memory at application layer 48 while a copy 54 of SADB is maintained in volatile memory within the kernel. If the message is supposed to be encrypted, the message is decrypted based on the algorithm and key associated with the particular SA and the message is transferred up through transport layer 46 to proxy 50. Proxy 50 examines the source and destination addresses and the type of service desired and decides whether authentication of the sender is warranted. If so, proxy 50 initiates an authentication protocol. The protocol may be as simple as requesting a user

name and password or it may include a challenge/response authentication process. Proxy 50 also looks to see whether the message coming in was encrypted or not and may factor that into whether a particular type of authentication is needed. In Telnet, for instance, user name/password authentication may be sufficient for an FFE link while the security policy may dictate that a more stringent challenge/response protocol is needed for unencrypted links. In that case, proxy 50 will be a Telnet proxy and it will base its authentication protocol on whether the link was encrypted or not.

Since IPSEC executes within IP layer 44 there is no need for host firewalls to update their applications. Users that already have IPSEC available on their own host machine will, however, have to request that the firewall administrator set up SA's in the SADB for their traffic.

In the embodiment shown in Figure 2, a working copy of the Security Association Database consisting of all currently active SA's is kept resident in memory for ready access by IP layer processing as datagrams are received and transmitted. In addition, a working master copy of the SADB is maintained in a file in nonvolatile memory. During system startup and initialization processing the content of all of the required SA's in master SADB is added to the working copy stored in kernel memory.

In one embodiment, firewall 18 maintains different levels of security on internal and external network interfaces. It is desirable for a firewall to have different levels of security on both the internal and external interfaces. In one embodiment, firewall 18 supports three different levels, numbered 0 through 2. These levels provide a simple policy mechanism that controls permission for both in-bound and out-bound packets.

Level 0 - do not allow any in-bound or out-bound traffic unless there is a security association between the source and destination.

- Level 1 - Allow both in-bound and out-bound non-IPSEC traffic but force the use of IPSEC if a SA exists for the address. (To support this firewall 18 must look for a SA for each in-bound datagram.)

- Level 2 - allow NULL security associations to exist. NULL associations 5 are just like normal security associations, except no encryption or authentication transform is performed on in-bound or out-bound packets that correspond to this NULL association. With Level 2 enabled, the machine will still receive unprotected traffic, but it will not transmit unless Level 1 is enabled.

The default protection level established when the Security Association 10 Database (SADB) is initialized at boot time is 1 for in-bound traffic and 2 for out-bound traffic.

An Access Control List, or ACL, is a list of rules that regulate the flow of Internet connections through a firewall. These rules control how a firewall's servers and proxies will react to connection attempts. When a server or proxy 15 receives an incoming connection, it performs an ACL check on that connection.

An ACL check compares a set of parameters associated with the connection against a list of ACL rules. The rules determine whether the connection is allowed or denied. A rule can also have one or more side effects. A side effect causes the proxy to change its behavior in some fashion. For 20 example, a common side effect is to redirect the destination IP address to an alternate machine. In addition to IP connection attempts, ACL checks can also be made on the console logins and on logins made from serial ports. Finally, ACL checks can also be made on behalf of IP access devices, such as a Cisco box, through the use of the industry standard TACACS+ protocol.

25 In one embodiment, the ACL is managed by an acld daemon running in the kernel of firewalls 10 and 30. The acld daemon receives two types of requests, one to query the ACL and one to administer it. In one such embodiment, the ACL is stored in a relational database such as the Oracle database for fast access. By using such a database, query execution is 30 asynchronous and many queries can be executing concurrently. In addition, these types of databases are designed to manipulate long lists of rules quickly

and efficiently. These qualities ensure that a given query cannot hang up the process that issued the query for any appreciable time (> 1-2 seconds).

In one such embodiment, the database can hold up to 100,000 users and up to 10,000 hosts but can be scaled up to the capacity of the underlying  
 5 database engine. The results of an ACL check is cached, allowing repeated checks to be turned around very quickly.

Applications on firewalls 10 and 30 can query `acd` to determine if a given connection attempt should be allowed to succeed. In one embodiment, the types of applications (i.e. "agents") that can make ACL queries can be divided  
 10 into four classes:

- 1) Proxies. These allow connections to pass through firewall 10 or 30 in order to provide access to a remote service. They include `tnauthp` (authenticated telnet proxy), `ptp` (FTP proxy), `httpp` (HTTP proxy), and `tcpgps` (TCP generic service proxy).
- 15 2) Servers. These provide a service on the firewall itself. They include `ftpd` and `httpd`.
- 3) Login agents. Login agent is a program on the firewall that can create a Unix shell. It is not considered a server because it cannot receive IP connections. One example is `/usr/bin/login` when used to create a dialup  
 20 session or a console session on firewall 10 or 30. Another example is the command `srole`.
- 4) Network Access Servers (NAS). NAS is a remote IP access device, typically a dialup box manufactured by such companies as Cisco or Bridge. The NAS usually provides dialup telnet service and may also  
 25 provide SLIP or PPP service.

Proxies, servers, login agents, and NASes make queries to `acd` to determine if a given connection attempt should be allowed to succeed. All of the agents except NAS make their queries directly. NAS, because it is remote, must communicate via an auxiliary daemon that typically uses an industry standard  
 30 protocol such as RADIUS or TACACS+. The auxiliary daemon (e.g., `tacradd`) in turn forwards the query to local `acd`.



As a side effect of the query, `acd` tells the agent if authentication is needed. If no authentication is needed, the connection proceeds immediately. Otherwise `acd` provides (as another side effect) a list of allowed authentication methods that the user can choose from. The agent can present a menu of choices  
 5 or simply pick the first authentication method by default. Typical authentication methods include plain password, SNK DSS, SDI SecurID, LOCKout DES, and LOCKout FORTEZZA. In one embodiment, the list of allowed authentication methods varies depending on the host name, user name, time of day, or any combination thereof.

10 In the case of a Level 0 policy, it would be safe to assume that all incoming traffic is encrypted or authenticated. In the case of Levels 1 through 2, a determination must be made whether or not a security association exists for a given peer. Otherwise an application may believe that in-bound traffic has been authenticated when it really has not. (That is why it is necessary to look for an  
 15 SA on input of each non-IPSEC datagram.)

In one embodiment, a flag which accompanies the message as it is sent from IP layer 44 to proxy 50 indicates whether the incoming message was or was not encrypted. In another embodiment, proxy 50 accesses Security Association Database 54 (the table in the kernel can be queried via an SADB routing socket  
 20 (PF-SADB)) to determine whether or not a security association exists for a given peer.. The SADB socket is much like a routing socket found in the stock BSD 4.4 kernel (protocol family PF-ROUTE) except that PF-SADB sockets are used to maintain the Security Association Database (SADB) instead of the routing table. Because the private keys used for encryption, decryption, and keyed  
 25 authentication are stored in this table, access must be strictly prohibited and allowed to only administrators and key management daemons. Care must be taken when allowing user-level daemons access to `/dev/mem` or `/dev/kmem` as well, since the keys are stored in kernel memory and could be exposed with some creative hacking.

30 In one embodiment, a command-line tool called `sadb` is used to support the generation and maintenance of in-kernel version 54 of SADB. The primary

interface between this tool and the SADB is the PF-SADB socket. The kernel provides socket processing to receive client requests to add, update, or change entries in in-kernel SADB 54. As noted above, the default protection level established when the Security Association Database (SADB) is initialized at boot time is 1 for in-bound traffic and 2 for out-bound traffic. This may be changed by the use of the `sadb` command.

The existing `sadb` command was derived from the NIST implementation of IPSEC. As noted above, this tool is much like `route` in that it uses a special socket to pass data structures in and out of the kernel. There are three commands recognized by the `sadb` command: `get`, `set`, `delete`. The following simple shell script supports adding and removing a single SA entry to SADB 54. It shows one embodiment of a parameter order for adding a SA to the SADB.

```

# ! /bin/sh
15 if [ $# -ne 1 ]
    then
        echo "usage: $0 <on>|<off>" >&2
        exit 1
    fi
20 ONOFF=$1

    addsa ()
    {
        IPADDRESS=$2
25 PEERADDRESS=0.0.0.0
        PREFIXLEN=0                # Num of bits, 0 => full 32
        bit match
        LOCALADDRESS=0.0.0.0
        REALADDRESS=0.0.0.0
30 PORT=0
        PROTOCOL=0
        UID=0
        DESALG=1                    # I = DES-CBC
        IVLEN=4                    # bytes
35 DESKEY=0b0b0b0b0b0b0b0b
        DESKEYLEN=8                # bytes
        AHALG=1                    # 1 = MD5
        AHKEY=30313233343536373031323334353637
        AHKEYLEN=16                # bytes
40 LOCAL_SPI=$1

```

```

PEER_SPI=$1
TUNNEL_MODE=0
AHRESULTLEN=4
COMBINED_MODE=1           # On output, 1 = ESP, then
5 AH; 0 = AH, then ESP
DYNAMIC_FLAG=0

if [ "$SONOFF" = "on"
then
10     ./sadb add dst $IPADDRESS $PREFIXLEN $LOCAL_SPI
      $UID $PEERADDRESS $PEER_SPI $TUNNEL_MODE $LOCALADDRESS
      $REALADDRESS $PROTOCOL $PORT $DESALG $IVLEN $DESKEYLEN
      $DESKEY $DESKEYLEN $DESKEY $AHALG $AHKEYLEN $AHKEY
15 $AHKEYLEN $AHKEY $AHRESULTLEN $COMBINED_MODE
      $DYNAMIC_FLAG
      else
          ./sadb delete dst $IPADDRESS $LOCAL-SPI
      fi
20 }

# Get down to work:
addsa 500 172.17.128.115           # number6.sctc.com

```

The current status of in-kernel SADB 54 can be obtained with the `sadb` command. The `get` option allows dumping the entire SADB or a single entry. In one embodiment, the complete dump approach uses `/dev/kmem` to find the information. The information may be presented as follows:

```

30 # sadb get dst
Local-SPI Address-Family Destination-Addr
Preflx_length UID
Peer-Address Peer-SPI Transport-Type
Local-Address Real-Address
35 Protocol Port
ESP_Alg_ID ESP_IVEC_Length
ESP_Enc_Key_length ESP_Enc_ESP_Key
ESP_Dec_Key_length ESP_Dec_ESP_Key
AH_Alg_ID AH_Data_Length
40 AH_Gen_Key_Length AH_Gen_Key
AH_Check_Key_Length AH_Check_Key
Combined_Mode Dynamic_Flag

```

```

-----
500 INET: number6.sctc.com 0 0
      0.0.0.0 500 Transport(0) 0
5      0.0.0.0 0.0.0.0
      None None
      DES/CBC-RFC1829(1) 4
          8 0b0b0b0b0b0b0b0b
          8 0b0b0b0b0b0b0b0b
10     MDS-RFC1828(1) 4
          16 30313233343536373031323334353637
          16 30313233343536373031323334353637
      ESP+AH(1) 0
501 INET: spokes.sctc.com 0 0
15     0.0.0.0 501 Transport(0) 0
      0.0.0.0.0.0.0.0.0
      None None
      DES/CBC-RFC1829(1) 4
          8 0b0b0b0b0b0b0b0b
20     8 0b0b0b0b0b0b0b0b
      MDS-RFC1828(1) 4
          16 30313233343536373031323334353637
          16 30313233343536373031323334353637
      ESP+AH(1) 0
25
End of list.

```

When a new entry is added to in-kernel SADB 54, the add process first checks to see that no existing entry will match the values provided in the new entry. If no match is found then the entry is added to the end of the existing SADB list.

To illustrate the use and administration of an FFE, we'll go through an example using FFE 70 in Figure 3. Firewalls 14 and 18 are both application level gateway firewalls implemented according to the present invention. Workstations H2 and H3 both want to communicate with H1. For the administrator of firewalls 14 and 18, this is easy to accomplish. The administrator sets up a line something like this (we'll only show the IP address part and SPI parts of the SA, since they're the trickiest values to configure. Also, assume that we are using tunnel mode):

```

40 # Hypothetical SW1 Config File

```

```

#
# Fields are laid out in the following manner:
# srcaddrornet= localSPI= peeraddr= peerSPI=
# realsrcaddr= localaddr= key=
5
# The following entry sets up a tunnel between hosts
# behind SW1
# and hosts behind SW2.
src=172.16.0.0 localSPI=666 peer=192.168.100.5
10 peerSPI=777 \
    realsrcaddr=192.168.100.5 localaddr=0.0.0.0
    key=0xdeadbeeffadebabe

# Hypothetical SW2 Config File
15 #
# Fields are laid out in the following manner:
# srcaddrornet= localSPI= peeraddr= peerSPI=
# realsrcaddr= localaddr= key=

20 # The following entry sets up a tunnel between hosts
# behind SW1 and
# hosts behind SW2.
src=172.17.0.0 localSPI=777 peer=192.168.20.1
peerSPI=666 \
25 realsrcaddr=192.168.20.1 localaddr=0.0.0.0 \
    key=0xdeadbeeffadebabe

```

With this setup, all traffic is encrypted using one key, no matter who is talking to whom. For example, traffic from H2 to H1 as well as traffic from H3 to H1 will be encrypted with one key. Although this setup is small and simple, it may not be enough.

What happens if H2 cannot trust H3? In this case, the administrator can set up security associations at the host level. In this case, we have to rely on the SPI field of the SA, since the receiving firewall cannot tell from the datagram header which host behind the sending firewall sent the packet. Since the SPI is stored in IPSEC datagrams, we can do a lookup to obtain its value. Below are the sample configuration files for both firewalls again, but this time, each host combination communicates with a different key. Moreover, H2 excludes H3 from communications with H1, and H3 excludes H2 in the same way.

40

```

# Hypothetical SW1 Config File
#
# Fields are laid out in the following manner:
# srcaddr= localSPI= peeraddr= peerSPI=
5 realsrcaddr= localaddr= key=

# The following entry sets up a secure link between H2
and H1
src=172.16.0.2 localSPI=666 peer=192.168.100.5
10 peerSPI=777 \
    realsrcaddr=192.168.100.5
localaddr=178.17.128.71 \
    key=0x0a0a0a0a0a0a0a0a

15 # The following entry sets up a secure link between H3
and H1
src=172.16.0.1 localSPI=555 peer=192.168.100.5
peerSPI=888 \
    realsrcaddr=192.168.100.5
20 localaddr=178.17.128.71 \
    key=0x0b0b0b0b0b0b0b0b

# Hypothetical SW2 Config File
#
# Fields are laid out in the following manner:
# srcaddr= localSPI= peeraddr= peerSPI=
25 realsrcaddr= localaddr= key=

# The following entry sets up a secure link between H2
and H1
30 src=172.17.128.71 localSPI=777 peer=192.168.20.1
peerSPI=666 \
    realsrcaddr=192.168.20.1 localaddr=172.16.0.2 \
    key=0x0a0a0a0a0a0a0a0a

35 # The following entry sets up a secure link between H3
and H1
src=172.17.128.71 localSPI=888 peer=192.168.20.1
peerSPI=555 \
40 realsrcaddr=192.168.20.1 localaddr=172.16.0.1 \
    key=0x0b0b0b0b0b0b0b0b

```

Figure 4 is a block diagram showing in more detail one embodiment of an IPSEC-enabled application level gateway firewall 18. Application level gateway firewall 18 provides access control checking of both encrypted and

unencrypted messages in a more secure environment due to its network-separated architecture. Network separation divides a system into a set of independent regions or burbs, with a domain and a protocol stack assigned to each burb. Each protocol stack 40x has its own independent set of data structures, including routing information and protocol information. A given socket will be bound to a single protocol stack at creation time and no data can pass between protocol stacks 40 without going through proxy space. A proxy 50 therefore acts as the go-between for transfers between domains. Because of this, a malicious attacker who gains control of one of the regions is prevented from being able to compromise processes executing in other regions. Network separation and its application to an application level gateway is described in "SYSTEM AND METHOD FOR ACHIEVING NETWORK SEPARATION", U.S. Application No. 08/599,232, filed February 9, 1996 by Gooderum et al.

In the system shown in Figure 4, the in-bound and out-bound datagram processing of a security association continues to follow the conventions defined by the network separation model. Thus all datagrams received on or sent to a given burb remain in that burb once decrypted. In one such embodiment SADB socket 78 has been defined to have the type 'sadb'. Each proxy 50 that requires access to SADB socket 78 to execute its query as to whether the received message was encrypted must have create permission to the sadb type.

The following is list of specific requirements that a system such as is shown in Figure 4 must provide. Many of the requirements were discussed in the information provided earlier in this document.

1. Firewall applications may query the IPSEC subsystem to determine if traffic with a given address is guaranteed to be encrypted.
2. Receipt of an unencrypted datagram from an address that has a SA results in the datagram being dropped and an audit message being generated.
3. On receipt of encrypted protocol datagrams the SADB searches will be done using the SPI as the primary key. The source address will be a secondary key. The SA returned by the search will be the SA which matches the SPI exactly and has the longest match with the address.

4. A search of the SADB for a SPI that finds an entry that is marked as SA for a dynamic IP will not consider the address in the search process.
5. A search of the SADB for a SPI that finds an entry that is marked as a SA for a tunnel mode connection will to consider the address if it is (0.0.0.0) i.e INADDR.
6. On receipt of a non-IPSEC datagram the SADB will be searched for an entry that matches the src address. If a SA is found the datagram will be dropped and an audit message sent.
7. SADB searches on output will be done using the DST address as key. If more than one SA entry in the SADB has that address the first one with the maximum address match will be returned.
8. The SADB must be structured so that searches are fast regardless if the search is done by SPI or by address.
9. The SADB must provide support for connections to a site with a fixed SPI but changing IP address. SA entries for such connections will be referred to as Dynamic Address Sites, or just Dynamic entries.
10. When a dynamic entry is found by a SPI search, the current datagram's SRC address, which is required to ensure that the return datagrams are properly encrypted, will be recorded in the SA only after the AH checking has passed successfully. (This is because if the address is recorded before AH passes then an attacker can cause return packets of an outgoing connection to be transmitted in the clear.)
11. A failure of an AH check on a dynamic entry results in an audit message.
12. In an embodiment where the firewall requires that all connections use both AH and ESP, on receipt the order should be AH first ESP second.
13. The processing structure on both input and output should try to minimize the number of SADB required lookups.

Returning to Figure 4, in one embodiment firewall 18 includes a crypto engine interface 80 used to encrypt an IPSEC payload. Crypto engine interface 80 may be connected to a software encryption engine 82 or to a hardware



encryption engine 84. Engines 82 and 84 perform the actual encryption function using, for example, DES-CBC. In addition, software encryption engine 82 may include the keyed MD5 algorithm used for AH.

In one embodiment, crypto engine interface 80 is a utility which provides  
5 a consistent interface between the software and hardware encryption engines. As shown in Figure 4, in one such embodiment interface 80 only supports the use of the use of hardware cryptographic engine 84 for IPSEC ESP processing. The significant design issue that interface 80 must deal with is that use of a hardware encryption engine requires that the processing be down in disjoint steps  
10 operating in different interrupt contexts as engine 84 completes the various processing steps.

The required information is stored in a request structure that is bound to the IP datagram being processed. The request is of type `crypto_request_t`. This structure is quite large and definitely does not contain a minimum state set.

15 In addition to the definition of the request data structure, this software implementing interface 80 provides two functions which isolate the decision of which cryptographic engine to use. The `crypt_des_encrypt` function is for use by the IP output processing to encrypt a datagram. The `crypt_des_decrypt` function is for use by the IP input processing to  
20 decrypt a datagram. If hardware encryption engine 84 is present and other hardware usage criteria are met the request is enqueued on a hardware processing queue and a return code indicating that the cryptographic processing is in progress is returned. If software engine 82 is used, the return code indicates that the cryptographic processing is complete. In the former case, the continuation of  
25 the IP processing is delayed until after hardware encryption is done. Otherwise it is completed as immediately in the same processing stream.

There are two software cryptographic engines 82 provided in the IPSEC software. One provides the MD5 algorithm used by the IPSEC AH processing, and the other provides the DES algorithm used by the IPSEC ESP processing.  
30 This software can be obtained from the US Government IPSEC implementation.

In one embodiment hardware cryptographic engine 84 is provided by a Cylink SafeNode processing board. The interface to this hardware card is provided by the Cylink device driver. A significant aspect of the Cylink card that plays a major part in the design of the IPSEC Cylink driver is that the card functions much like a low level subroutine interface and requires software support to initiate each processing step. Thus to encrypt or decrypt an individual datagram there are a minimum of two steps, one to set the DES initialization vector and one to do the encryption. Since the IP processing can not suspend itself and wait while the hardware completes and then be rescheduled by the hardware interrupt handler, in one embodiment a finite state machine is used to tie sequences of hardware processing elements together. In one such embodiment the interrupt handler looks at the current state, executes a defined after state function, transitions to the state and then executes that state's start function.

One function, `cyl_enqueue_request`, is used to initiate either an encrypt or a decrypt action. This function is designed to be called by cryptographic engine interface 80. All of the information required to initiate the processing as well as the function to be performed after the encryption operation is completed is provided in the request structure. This function will enqueue the request on the hardware request queue and start the hardware processing if necessary.

A system 30 which can be used for firewall-to-workstation encryption is shown in Figure 5. In Figure 5, system 30 includes a workstation 12 communicating through a firewall 14 to an unprotected network 16 such as the Internet. System 30 also includes a workstation 32 communicating directly with firewall 14 through unprotected network 16. Firewall 14 is an application level gateway incorporating IPSEC handling as described above. (It should be noted that IPSEC security cannot be used to authenticate the personal identity of the sender for a firewall to firewall transfer. When IPSEC is used, however, on a single user machine such as a portable personal computer, IPSEC usage should

be protected with a personal identification number (PIN). In these cases IPSEC can be used to help with user identification to the firewall.)

According to the IPSEC RFC's, you can use either tunnel or transport mode with this embodiment based on your security needs. In certain situations,  
5 the communications must be sent in tunnel mode to hide unregistered addresses.

Although specific embodiments have been illustrated and described herein, it will be appreciated by those of ordinary skill in the art that any arrangement which is calculated to achieve the same purpose may be substituted for the specific embodiment shown. This application is intended to cover any  
10 adaptations or variations of the present invention. Therefore, it is intended that this invention be limited only by the claims and the equivalents thereof.

What is claimed is:

1. A method of regulating the flow of messages through a firewall having a network protocol stack, wherein the network protocol stack includes an Internet Protocol (IP) layer, the method comprising the steps of:
  - 5 determining, at the IP layer, if a message is encrypted;
  - if the message is not encrypted, passing the unencrypted message up the network protocol stack to an application level proxy; and
  - 10 if the message is encrypted, decrypting the message and passing the decrypted message up the network protocol stack to the application level proxy, wherein the step of decrypting the message includes the step of executing a procedure at the IP layer to decrypt the message.
  
2. A method of authenticating the sender of a message within a computer system having a network protocol stack, wherein the network protocol stack includes an Internet Protocol (IP) layer, the method comprising the steps of:
  - 15 determining, at the IP layer, if the message is encrypted;
  - if the message is encrypted, decrypting the message, wherein the step of decrypting the message includes the step of executing a process at the IP layer to
  - 20 decrypt the message;
  - passing the decrypted message up the network protocol stack to an application level proxy;
  - determining an authentication protocol appropriate for the message; and
  - 25 executing the authentication protocol to authenticate the sender of the message.
  
3. The method according to claim 2 wherein the step of determining an authentication protocol appropriate for the message includes the steps of:
  - 30 determining a source IP address associated with the message; and
  - determining the authentication protocol associated with the source IP address.

4. The method according to claim 2 wherein the message includes security parameters index and wherein the step of determining an authentication protocol appropriate for the message includes the steps of:

5 determining the authentication protocol associated with a dynamic IP address, wherein the step of determining the authentication protocol includes the step of looking up a security association based on the security parameters index; determining a current address associated with the dynamic source IP address; and binding the current address to the security parameters index.

10

5. A firewall, comprising:

a first communications interface;

a second communications interface;

a network protocol stack connected to the first and the second

15 communications interfaces, wherein the network protocol stack includes an Internet Protocol (IP) layer and a transport layer;

a decryption procedure, operating at the IP layer, wherein the decryption procedure decrypts encrypted messages received at one of said first and second communications interfaces and outputs decrypted messages; and

20 a proxy, connected to the transport layer of said network protocol stack, wherein the proxy receives decrypted messages from the decryption procedure and executes an authentication protocol based on the content of the decrypted message.

25 6. A firewall, comprising:

a first communications interface;

a second communications interface;

a first network protocol stack connected to the first communications

interface, wherein the first network protocol stack includes an Internet Protocol

30 (IP) layer and a transport layer;

a second network protocol stack connected to the second communications interface, wherein the second network protocol stack includes an Internet Protocol (IP) layer and a transport layer;

5 a decryption procedure, operating at the IP layer of the first network protocol stack, the decryption procedure receiving encrypted messages received by said first communications interface and outputting decrypted messages; and  
 a proxy, connected to the transport layers of said first and second network protocol stacks, the proxy receiving decrypted messages from the decryption procedure and executing an authentication protocol based on the content of the  
 10 decrypted message.

7. The firewall according to claim 6 wherein the firewall further includes:  
 a third communications interface; and

15 a third network protocol stack connected to the third communications interface and to the proxy, wherein the third network protocol stack includes an Internet Protocol (IP) layer and a transport layer and wherein the second and third network protocol stacks are restricted to first and second burbs, respectively.

20 8. A method of establishing a virtual private network between a first and a second network, wherein each network includes an application level gateway firewall which uses a proxy operating at the application layer to process traffic through the firewall, wherein each firewall includes a network protocol stack and wherein each network protocol stack includes an Internet Protocol (IP) layer, the  
 25 method comprising the steps of:

transferring a connection request from the first network to the second network;

determining, at the IP layer of the network protocol stack of the second network's firewall, if the connection request is encrypted;

if the connection request is encrypted, decrypting the request, wherein the step of decrypting the request includes the step of executing a procedure at the IP layer of the second network's firewall to decrypt the message;

- 5       passing the connection request up the network protocol stack to an application level proxy;
- determining an authentication protocol appropriate for the connection request;
- executing the authentication protocol to authenticate the connection request; and
- 10       if the connection request is authentic, establishing an active connection between the first and second networks.

9.       The method according to claim 8 wherein the step of executing the authentication protocol includes the step of executing program code within the  
15       firewall of the second network to mimic a challenge/response protocol executing on a server internal to the second network.

10.      The method according to claim 8 wherein the step of executing the authentication protocol includes the step of executing program code to execute  
20       the authentication protocol in line to the session.

11.      The method according to claim 8 wherein the step of determining an authentication protocol includes the step of determining if the connection request  
25       arrived encrypted and selecting the authentication protocol based on whether the connection request was encrypted or not encrypted.



Application No: GB 9719816.2
Claims searched: 1-11

Examiner: B.J.SPEAR
Date of search: 21 January 1998

Patents Act 1977
Search Report under Section 17

Databases searched:

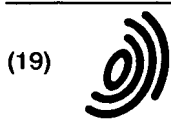
UK Patent Office collections, including GB, EP, WO & US patent specifications, in:
UK Cl (Ed.P): H4P (PPEB,PDCSA,PDCSC)
Int Cl (Ed.6): H04L 9/00, 9/32, 29/06, 29/08
Other: Online:WPI, INSPEC

Documents considered to be relevant:

Table with 3 columns: Category, Identity of document and relevant passage, Relevant to claims. Rows include XP WO97/26734A1, XP WO97/26731A1, XP WO97/26735A1, XP WO97/23972A1, XP WO97/13340A1.

X Document indicating lack of novelty or inventive step
Y Document indicating lack of inventive step if combined with one or more other documents of same category.
& Member of the same patent family
A Document indicating technological background and/or state of the art.
P Document published on or after the declared priority date but before the filing date of this invention.
E Patent document published on or after, but with priority date earlier than, the filing date of this application.





Europäisches Patentamt  
 European Patent Office  
 Office européen des brevets



(11) EP 0 814 589 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication: 29.12.1997 Bulletin 1997/52 (51) Int. Cl.<sup>6</sup>: H04L 29/06

(21) Application number: 97109792.8

(22) Date of filing: 16.06.1997

(84) Designated Contracting States:  
 AT BE CH DE DK ES FR GB GR IE IT LI LU MC NL  
 PT SE

(30) Priority: 19.06.1996 US 667524

(71) Applicant: AT&T Corp.  
 New York, NY 10013-2412 (US)

(72) Inventors:  
 • Harwood, Jonathan P.  
 Morganville, N.J. 07751 (US)

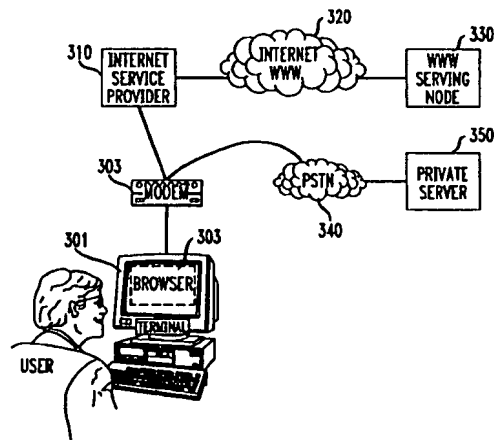
• Kimmeth, Thomas  
 Gladstone, N.J. 07977 (US)  
 • Nusbaum, Kurt  
 Downers Grove, Illinois 60515 (US)

(74) Representative:  
 KUHNEN, WACKER & PARTNER  
 Alois-Steinecker-Strasse 22  
 85354 Freising (DE)

(54) System and method for automated network reconfiguration

(57) A method is disclosed for providing an enhanced level of security for sensitive or proprietary information associated with information transactions in a public network, such as the Internet. In carrying out that method, an on-line information transaction is bifurcated between a generalized information access portion of such a transaction and an exchange of sensitive user information. With such a bifurcation, the generalized information access portion of the transaction, which generally would constitute the more substantial (in terms of network resources) portion of the transaction, would be handled via a non-secure network, usually a public network such as the Internet. The portion of the transaction involving sensitive user information, on the other hand, would be handled by a separate secure connection, such as a private network, or intranetwork. An important characteristic of this bifurcation arrangement is the provision of a means for automated reconfiguration of a user terminal as between accessing the generalized information via the non-secure network and access to the secure communications network for the exchange of sensitive user information. Such an automated reconfiguration will be carried out without the necessity for any action on the part of the user, and indeed will be largely invisible to the user.

FIG. 3



EP 0 814 589 A2

**Description****FIELD OF THE INVENTION**

5 This invention is related to the field of data communications, and more particularly to a method and means for establishing an automatic reconfiguration of a user terminal among alternative tasks.

**BACKGROUND OF THE INVENTION**

10 With the increasing popularity of personal computers over the last several years has come a striking growth in transaction-oriented computer-to-computer communications (as opposed to bulk-data transfers among such computers). For convenience herein such transaction-oriented computer-to-computer communications will be described by the shorthand term "information transaction". That growth in the use of computers for such information transactions has unquestionably been fueled by the existence of an international infrastructure for implementing such data communications, known as the Internet. And, driven by the burgeoning demand for such information transaction services, the Internet has itself experienced explosive growth in the amount of traffic handled.

At least partly in response to that demand, a new level of accessibility to various information sources has recently been introduced to the Internet, known as the World Wide Web ("WWW"). The WWW allows a user to access a universe of information which combines text, audio, graphics and animation within a hypermedia document. Links are contained within a WWW document which allow simple and rapid access to related documents. Using a system known as the HyperText Markup Language ("HTML"), pages of information in the WWW contain pointers to other pages, those pointers typically being a key word (commonly known as a hyperlink word). When a user selects one of those key words, a hyperlink is created to another information layer (which may be in the same, or a different information server), where typically additional detail related to that key word will be found.

25 In order to facilitate implementation of the WWW on the Internet, new software tools have been developed for user terminals, usually known as Web Browsers, which provide a user with a graphical user interface means for accessing information on the Web, and navigating among information layers therein. A commonly used such Web Browser is that provided by Netscape.

The substantial growth in the use of computer networks, and particularly the WWW, for such information transactions, has predictably led to significant commercialization of this communications medium. For example, with the WWW, a user is not only able to access numerous information sources, some public and some commercial, but is also able to access "catalogs" of merchandise, where individual items from such a catalog can be identified and ordered, and is able to carry out a number of banking and other financial transactions. As will be obvious, such commercial transactions will typically involve sensitive and proprietary information, such as credit card numbers and financial information of a user. Thus, with the growth of commercial activity in the Internet, has also come a heightened concern with security.

It is well known that there are persons with a high level of skill in the computer arts, commonly known as "hackers", who have both the ability and the will to intercept communications via the Internet. Such persons are thereby able to gain unauthorized access to various sensitive user information, potentially compromising or misappropriating such information.

40 The vulnerability of such sensitive user information to misuse when so transmitted via the Internet is a phenomena which has only recently received wide public attention. Unless such security concerns can be quickly addressed and alleviated, the commercial development of this new communications medium may be slowed or even stalled altogether.

**SUMMARY OF THE INVENTION**

45 Accordingly, it is an object of the invention to provide an acceptable level of security for sensitive or proprietary information associated with information transactions in a public network, such as the Internet. That object is realized through an arrangement whereby an on-line information transaction is bifurcated between a generalized information access portion of such a transaction and an exchange of sensitive user information. With such a bifurcation, the generalized information access portion of the transaction, which generally would constitute the more substantial (in terms of network resources) portion of the transaction would be handled via a non-secure network, usually a public network such as the Internet. The portion of the transaction involving sensitive user information, on the other hand, would be handled by a separate secure connection, such as a private network, or intranetwork. An important characteristic of this bifurcation arrangement is the provision of a means for automated reconfiguration of a user terminal as between accessing the generalized information via the non-secure network and access to the secure communications network for the exchange of sensitive user information. Such an automated reconfiguration will be carried out without the necessity for any action on the part of the user, and indeed will be largely invisible to the user. In a further embodiment of the invention, a transfer of data is provided from a public to a private network, wherein data selected by a user from a public net-

work site may be arranged and displayed at a user terminal and, subject to further user selection/confirmation activity, thereafter transferred to a private network.

#### BRIEF DESCRIPTION OF THE DRAWINGS

5

Figure 1 depicts an illustrative case of information transactions carried out via a public network such as the Internet.

Figure 2 shows the architecture of a browser as would typically be applied for accessing a hypermedia web page.

Figure 3 illustrates the primary elements of the reconfigurable dual-path method of the invention.

Figure 4 depicts in flow chart form the basic jump capability of the methodology of the invention.

10 Figures 5A & 5B (generally designated collectively herein as "Figure 5") depict in flow chart form the "shopping cart" capability of the methodology of the invention.

Figure 6A & 6B (generally designated collectively herein as "Figure 6") depict in flow chart form the stored configuration capability of the methodology of the invention.

15 Figure 7A & 7B (generally designated collectively herein as "Figure 7") depict in flow chart form the off-line form capability of the methodology of the invention.

#### DETAILED DESCRIPTION

20 For clarity of explanation, the illustrative embodiment of the present invention is presented as comprising individual functional blocks. The functions these blocks represent may be provided through the use of either shared or dedicated hardware, including, but not limited to, hardware capable of executing software.

Figure 1 depicts an illustrative case of information transactions carried out via the Internet. As seen in the figure, an exemplary user obtains access to the Internet by first connecting, via a Terminal 110 having an associated Browser 111, to an Internet Service Provider 112 selected by the user. That connection between the user and the Internet Service Provider will typically be made via the Public Switched Telephone Network (PSTN) from a modem associated with the user's Terminal to a network node in the Internet maintained by the selected Internet Service Provider.

25 Once the user has obtained access to the selected Internet Service Provider, an address is provided for connection to another user or other termination site and such a connection is made via the Internet to that destination location. As can be seen from the figure, communication via the Internet may be either user-to-user, as from Terminal 110 to Terminal 130, or from a user to a node representing an information source accessed via the Internet, such as Public Site 120.

30 It will of course be understood that the Internet provides service to a large number of users and includes a large number of such Public Sites, but the illustration provides the essential idea of the communication paths established for such Internet communication. It will also be understood that a number of service classifications are supported by the Internet, with the World Wide Web service, which represents a preferred embodiment for the public network aspect of the method of the invention, being one of the currently most heavily trafficked of such services.

35 The Web Browser, such as depicted at 111, can be seen as a software application operating in conjunction with a user terminal (such as Terminal 110) which provides an interface between such a user terminal and the particular functionality of the WWW information site. The architecture of such a browser is generally described in terms of three main components, as illustrated in Figure 2. At the top level is the Browser 201, which enables the acquisition of information pages from a WWW server (beginning, in all cases, with the "home page" for that server), for display at a display device associated with the terminal. The Browser also provides the necessary interface for the terminal with the HTML functionality used by the server to provide access to other linked information layers.

40 The second level of the browser architecture is the TCP/IP Stack 202, which handles the communications protocols used for connecting the terminal to the WWW server. The bottom level of this architecture is the Dialer 203, which typically handles the function of providing dialing and setup digits to a modem, as illustrated at 204, such a modem generally being a part of the terminal. Normally, upon receiving dialing and other setup information from the dialer, the modem would cause a connection to be made via the PSTN to the Internet Service Provider selected for that terminal.

45 After a connection is established in this manner to the Internet Service Provider, an address would be provided for the WWW information node sought to be contacted, a connection to that node made through the Internet, and the home page for that node caused to be displayed at the terminal's display device. A user would then select a key word in that home page, typically by clicking on the word with a mouse or similar device, and, upon transmission of that selection signal to the WWW server, a hyperlink would be created to the linked information layer and the open page of that layer would be caused to be displayed at the user terminal.

50 As explained above, serious questions have been raised in respect to the security of communications via the public Internet. (Note, that the discussion herein is focused on the Internet, and particularly the WWW functionality of the Internet, as a preferred embodiment of such public data communication networks generally, but the methodology of the invention will be applicable to any such network.) To address this problem, the methodology of the invention begins with a bifurcation of the information transaction between a user and the selected information transaction provider into a por-

tion related to sensitive or proprietary user information, and other information comprising that transaction. With such a bifurcation, it becomes possible to provide substantial security for that proprietary information by use of an alternative communications path for that separated portion of the transaction via a private network, or intranetwork -- *i.e.*, a connection between a user's terminal and a secure serving node on that private network. It is anticipated that a coordination means will be established in respect to the management of information among the public and private network elements of the bifurcated information transaction.

In its basic form, this methodology may be carried out by the user terminal initiating a call via the Internet to a selected WWW node, and upon establishing connection to that node, proceeding with the desired information transaction up to the point where an exchange of sensitive or proprietary information were required. At that point the user terminal would be instructed by the WWW server to terminate that connection (*i.e.*, hangup) and to place a new call to an identified private network server for the necessary exchange of sensitive information.

However, in order to accomplish such a dual-path transaction, it is necessary that the browser at the user terminal be reconfigured to provide the dialing, authorization (*i.e.*, login and password), and other needed information for accessing the alternative private network, in order to implement the proprietary portion of the transaction. It will also usually be the case that, upon completion of that private-network transaction, the original dialer, stack and browser configurations will need to be restored, in order for the terminal to retain its normal Internet access functionality. Such a reconfiguration and subsequent restoral of the necessary parameters in the browser, stack and dialer is likely to be well beyond the capabilities of the average user.

Accordingly, as a further embodiment of the inventive methodology, an automated browser reconfiguration means is provided which interoperates with the browser. This browser reconfiguration means is described in detail hereafter and will be referred to as the "Bridging Software".

Figure 3 provides an illustration of the primary elements of the reconfigurable dual-path method of the invention. As seen in the figure, a first path comparable to the Internet link shown in Figure 1, between User Terminal 301 and WWW Serving Node 330 (via Browser 302, Modem 303, Internet Service Provider 310, and Internet 320) is provided. However, an alternative path is now provided from the output of Modem 303 to Private Server 350. That path is illustrated as being via the PSTN, which is generally regarded as being highly secure, but an alternative dedicated or other more-secure path between the User Terminal 301 and the Private Server 350 could as well be provided. In keeping with the discussion above, Browser 302 shown in Figure 3 would also include the Bridging Software installed as a helper application for implementing the automatic reconfiguration of the Browser.

In the operation of this system, a user would normally make an initial connection to an Internet application, such as the application represented by WWW Serving Node 330, which, *e.g.*, might be a shopping application, a financial transaction, or the provision of an enrollment form for off-line preparation. After conducting all, or some portion of an information transaction short of an exchange of sensitive or proprietary information, including a capture by the user's terminal of needed information from the public site, a user provides a signal indicative of an end to that portion of that transaction. During the course of the public portion of the information transaction, specially configured files are sent from the WWW serving node to the Bridging Software associated with Browser 302. Such files contain instructions for the Bridging Software to store information-like products -- *e.g.*, for selected items from a catalog, forms for enrollment, or non-secure portions of a financial transaction, and reconfiguration information for dialing and logging into the private portion of the transaction. The Bridging Software then hangs up the Internet connection, edits the user terminal's browser, stack and dialer files to reconfigure the terminal to connect to the private server. Prior to automatic redialing of the new private site for the user, the Bridging Software may be instructed by the application operating at WWW Server Node 330 to display items chosen for purchase, or to display a form for the end-user to complete off-line before dialing the private application. Upon connecting to the private application and completing the transaction as to the user sensitive information in a private environment, the Bridging Software then restores the end-user software to the dialing and authorization parameters required to dial to the public Internet.

A particularly advantageous application of the automated reconfiguration and information transfer methodology of the Bridging Software is that it adds value to certain WWW servers which do not possess the Common Gateway Interface ("CGI") capability -- *i.e.*, a provision of specialized functions on the server beyond just displaying HTML files, and are accordingly unable to accomplish any transactional processing in respect to items selected by a user. In effect, such a non-CGI server, on its own, can only serve as a "billboard" for the items represented in its database.

However, with the collection and redelivery process of the Bridging Software, a data capture and processing mechanism can be implemented for servers operating in a non-CGI environment -- such servers being incapable of more than the simple delivery of static data packets corresponding to available items. The data set enabled by the Bridging Software is a mechanism for augmenting such limited server capabilities by defining a flexible mechanism for the receipt, display, and delivery of arbitrary data from one site to another.

In such a scenario, the Bridging Software receives a "shopping cart" item list from the host as a data-set defined with a static MIME data packet associated with the Bridging Software. This information comprising the data-set may be updated, displayed to the user in a "read-only" fashion, or presented to the user for order selection.

During the process of interacting with the WWW server, a user may trigger HTML links resulting in additional MIME packets for the Bridging Software being delivered to the client. These packets allow items to be added and/or removed from the specified data set or presented to the user for local confirmation. The user will interact with a pop-up screen provided by the Bridging Software which presents the items available with product information, such as part number, description, unit cost, etc. The user identifies those items which are to be placed into the "shopping cart" and the quantity of items desired. Upon completion of the form, the Bridging Software stores the order in a format suitable for subsequent delivery to the private server site.

An additional feature provided by the methodology of the Bridging Software is an automated mechanism for providing compatibility with user terminals not previously having the Bridging Software included with the terminal's browser. To that end, the Bridging Software located at an accessed public network site initially checks to see if the browser counterpart for that software is loaded at the calling user terminal. If yes, the heretofore described processes of the Bridging Software go forward. If not however, a request is sent through the public host to download the Bridging Software to the calling terminal. After such a download, a helper application loads the Bridging Software to the terminal's browser.

## I. Illustrative Embodiments

A variety of browser reconfiguration applications are supported by the automated browser reconfiguration means of the invention. Four essentially diverse capabilities of this invention, which support such applications, are described hereafter as illustrative embodiments of the invention.

### A. Basic Jump Capabilities

In this configuration, which is illustrated in flow chart form in Figure 4, an end-user is connected to a chosen WWW serving node (where a desired information product is made available) via a modem and an Internet browser associated with the user's terminal (Step 401 of Figure 4). After conducting an information transaction with the selected WWW serving node for some interval (determined in relation to the specific application accessed), the user clicks on a hyper-text link, or picture, to begin an automated process which will cause that public session to be terminated and a new connection established to an alternate private data network (Step 402).

In response to that user action, a data message containing parameter reconfiguration instructions is passed from the WWW server application to the Bridging Software at the user's terminal (Step 403). Upon receiving such instructions, the Bridging Software edits the user's on-line communications software parameters, reconfiguring that software to dial the alternate data network (Step 404). This reconfiguration is fully automatic and transparent to the user, and includes parameters such as modem dial number, login, password, and TCP/IP addresses. At that point, the Bridging Software causes the modem to disconnect the current data network connection, shutting down the browser, and to then dial the alternate private data network (Step 405).

With the establishment of a connection to the private server on the alternate data network, the user interacts with the alternate data network application as appropriate (Step 406), and after an interval completes his activity with the alternate data network and provides an indication of such completion (Step 407). A data message containing parameter reconfiguration instructions is then passed from the alternate data network application to the Bridging Software (Step 408).

At that point, the Bridging Software again edits the user's on-line communications software parameters, reconfiguring them to dial the original public data network, or another preselected network (Step 409). As with the first reconfiguration, this configuration is automatic and includes parameters such as modem dial number, login, password, and TCP/IP addresses. The Bridging Software automatically causes the current private data network to be disconnected by the modem (Step 410), and if appropriate, causes the original public data network to be redialed (Step 411). When such a reconnection to the public data network is established, the end-user would then continue his application in the public data network.

### B. "Shopping Cart" Capability

With this configuration, illustrated in flow chart form in Figure 5, a user begins by establishing a connection to a WWW application (assuming for the moment that the application is non-CGI enabled) at a serving node for that application, using the Internet browser and modem associated with the user's terminal (Step 501 of Figure 5). Upon finding an item in that application to be saved, or remembered for later consideration, or purchase, the user clicks on a hyper-text link, or picture, representing that item (Step 502). That application then sends a data message to the Bridging Software containing information about the items selected (Step 503) and such information is stored by the Bridging Software.

ware in the "shopping cart" file in the user's terminal (Step 504). Such selection download and storage steps (*i.e.*, steps 502, 503 & 504) are repeated for as many items as the user chooses to select. At any point after the Bridging Software has received the first set of item selection information, the user can instruct the Bridging Software to cause those selected items about which such information has been received to be displayed locally (at the user's terminal), where the user may review or edit (including deletion if desired) the collection of items theretofore selected. The application may also control display characteristics such as color and font for such locally displayed items. Note that in the case of a CGI-enabled application, the application itself will keep track of the items selected by the user and only download the totality of the selected items at the end of the selection process, and accordingly, the described local display option will not be applicable to such a CGI-enabled application.

At the point of completion of his "shopping", the user clicks on a hyper-text link or picture to "check out" (Step 505), which will begin a process of causing a jump to an alternate data network for the completion of sensitive portions of the transaction. To that end, a data message containing parameter reconfiguration instructions is passed from the WWW application to the Bridging Software (Step 506). It is to be noted that, as a security measure, information such as the new dial number, IP address, home page, configuration data (*e.g.*, login, password, DNS address) may be passed over the public network in encrypted form.

Upon receiving such reconfiguration instructions, the Bridging Software edits the user's on-line communications software parameters, reconfiguring that software to dial the alternate data network (Step 507). This reconfiguration is fully automatic and transparent to the user, and includes parameters such as modem dial number, login, password, and TCP/IP addresses. At that point, the Bridging Software causes the modem to disconnect the current data network connection, shutting down the browser, and to then dial the alternate data network (Step 508).

The Bridging Software passes the stored "shopping cart" data captured from the WWW application to the alternate network application (Step 509), where that data may be displayed for the user, permitting the user to confirm and/or modify the data (Step 510). The user interacts with the alternate data network application as appropriate, and after an interval completes his activity with the alternate data network (Step 511) and thus, by providing an appropriate completion signal to the application, completing the private portion of the information transaction (Step 512). A data message containing parameter reconfiguration instructions is then passed from the alternate data network application to the Bridging Software (Step 513).

The Bridging Software, at this point, again edits the user's on-line communications software parameters, reconfiguring them to dial the original (or another pre-defined) data network (Step 514). As with the first reconfiguration, this configuration is automatic and includes parameters such as modem dial number, login, password, and TCP/IP addresses. The Bridging Software automatically causes the current private data network to be disconnected by the modem (Step 515), and if appropriate, causes the original public data network to be redialed (Step 516). When such a reconnection is established to the point in the public data network where the user had left off to handle the secured aspects of his information transaction, the user would then continue his application in the public data network.

### C. Stored Configuration Capabilities

For this configuration, depicted in flow chart form in Figure 6, an end-user is connected to a chosen WWW serving node (where a desired information product is made available) via a modem and an Internet browser associated with the user's terminal (Step 601 of Figure 6). The user selects a hypertext link or picture associated with the WWW application by clicking on such link or picture (Step 602). A data message containing parameter reconfiguration instructions and an application icon (related to the selected hypertext link or picture) is passed from the WWW application to the Bridging Software (Step 603).

The Bridging Software creates an icon for display at the user's terminal, and saves a Bridging Software configuration file that is associated with that icon (Step 604). Such Bridging Software actions are automatic and multiple selections may be captured in this manner. At this point the user may continue the on-line session, or, if all desired selections have been made, a signal is provided from the user that the session should be discontinued (Step 605). The Bridging Software then automatically disconnects the current data network connection (Step 606).

After disconnecting from the WWW application, and following an interval determined by the user, a new application is selected by the user by clicking on the appropriate new icon displayed at the user's terminal (Step 607). The Bridging Software receives the reconfiguration instructions from the file associated with the selected icon (Step 608).

The Bridging Software edits the user's on-line communications software parameters, reconfiguring that software to dial the alternate data network (Step 609). The Bridging Software then automatically starts the user's Internet browser software and causes the alternate network application to be dialed by the modem associated with that terminal (Step 610). Upon establishing a connection to the alternate network, the user interacts with that application and completes the transaction to the user's satisfaction (Step 611). After a signal is sent to the alternate network indicating such completion of the user's activity (Step 612), a data message containing parameter reconfiguration instructions is passed from the alternate data network application to the Bridging Software (Step 613). That Software then causes the user's

terminal configuration parameters to be reset (Step 614) and the alternate data network to be automatically disconnected (Step 615).

#### D. Off-Line Form Capability

5

In this configuration, depicted in flow chart form in Figure 7, an end-user is connected to a chosen WWW serving node (where a desired information product is made available) via a modem and an Internet browser associated with the user's terminal (Step 701 of Figure 7). The user selects a hypertext link or picture associated with an off-line form application -- an exemplary such form being an HTML-based form -- by clicking on such link or picture (Step 702). A data message containing parameter reconfiguration instructions for the Bridging Software, the selected off-line-form application, and an optional icon (related to the selected hypertext link or picture) is passed from the WWW application to the Bridging Software (Step 703). Note that the selected off-line form may be for either single or multiple use.

10

In the case of a delayed or multiple use of the selected form, the Bridging Software may create an icon for display at the user's terminal, and will save a Bridging Software configuration file that is associated with that icon (Step 704). The form in question is also saved on the user's terminal. Such Bridging Software actions are automatic. At this point the user may continue the on-line session, or, if all desired selections have been made, a signal is provided from the user that the session should be discontinued (Step 705). The Bridging Software then automatically disconnects the current data network connection (Step 706).

15

After disconnecting from the WWW application, two cases are to be considered as to the further processing of the selected form: (1) an immediate single use of the form and (2) either a delayed or multiple use of the form. In the first case, the Bridging Software edits the user's on-line communications software parameters, reconfiguring that software to dial the alternate data network. The Bridging Software then automatically starts the user's Internet browser software which is caused to display the off-line form. The user then completes the off-line form and chooses a "Submit Form" button displayed at his terminal.

20

In the second case, the Bridging Software will have created an icon for display at the user's terminal and saved a Bridging Software configuration file associated with that icon. Following an interval determined by the user, the off-line-form application is started by the user by clicking on the new form icon displayed at the user's terminal (Step 707). The Bridging Software receives the reconfiguration instructions from the file associated with the selected icon (Step 708).

25

The Bridging Software edits the user's on-line communications software parameters, reconfiguring that software to dial the alternate data network (Step 709). The Bridging Software then automatically starts the user's Internet browser software which is caused to display the off-line form (Step 710). The user then completes the off-line form and chooses a "Submit Form" button displayed at his terminal (Step 711).

30

In either the first or second case, following activation of the "Submit Form" button, the alternate network application is then caused to be dialed by the Bridging Software. Upon establishing a connection to the alternate network, the form data is passed to the alternate network (Step 712). The user then interacts with that application and completes the application (Step 713). After a signal is sent to the alternate network indicating such completion of the user's activity (Step 714), a data message containing parameter reconfiguration instructions is passed from the alternate data network application to the Bridging Software (Step 715). That Software then causes the user's terminal configuration parameters to be reset (Step 716) and the alternate data network to be automatically disconnected (Step 717).

35

40

#### CONCLUSION

A system and method has been described for the automatic switching of an information transaction between two or more alternate networks. This functionality, which incorporates a reconfiguration means designated herein as the Bridging Software, supports the movement of application specific data from one on-line environment to another. Among potential applications of this process for passing data between different environments are: selected items for purchase ("shopping cart"), captured data from forms, and other server captured data such as web pages visited.

45

The Bridging Software reconfiguration means is intended to work with various Web Browser software implementations, including the Netscape Personal Edition (NPE) Software for Windows 3.1 and 3.11, and which represents a working embodiment for the invention. The Bridging Software installs itself as a helper application within the browser application and utilizes a special MIME type configuration file to pass reconfiguration and "shopping cart" information from the server to the client software.

50

When an application requires a user to re-connect to a private application, a reconfiguration file is passed to the Bridging Software helper application via a CGI script or simple hyper-text link. The helper application disconnects the current data connection, reconfigures the dial parameters (dial #, login password, DNS address, and home page) and initiates the dial program so the end-user can access the private application.

55

When the end-user connects to the private application, the Bridging Software reconfiguration means provides the new "private server" application with data collected from the "public server", and the application resumes in a private,

secure environment.

The Bridging Software allows both short term and long term storage of dial configurations. Configurations passed to the Bridging Software can be designated as single use configurations and discarded after the application has terminated, or saved and displayed to the end-user as a dial choice by the Bridging Software.

5 Although the present embodiment of the invention has been described in detail, it should be understood that various changes, alterations and substitutions can be made therein without departing from the spirit and scope of the invention as defined by the appended claims. In particular, it is noted that, while the invention has been primarily described in terms of a preferred embodiment based on an automatic reconfiguration between a public and a private data network, any the methodology of the invention will be equally applicable to any set of alternate networks.

10

### Claims

1. A method for managing a transaction via a communications path between a terminal device and a serving node in a data network, said method comprising the steps of:

15

establishing an initial communications path via a first connection between said terminal device and a serving node in a first data network;  
receiving information from said serving node in said first data network for effecting a reconfiguration of said communications path for said transaction from said first connection in said first data network to a second connection in a second data network; and  
20 automatically connecting said terminal device to a serving node in said second data network via said second connection.

20

2. A method for managing a transaction via a communications path between a terminal device and a serving node in a data network, said method comprising the steps of:

25

establishing an initial communications path via a first connection between said terminal device and a serving node in a first data network;  
selecting at least one information item from a data base of said information items provided at said serving node in said first data network;  
30 causing said selected information items to be downloaded to said terminal device via said first connection;  
receiving information from said serving node in said first data network for effecting a reconfiguration of said communications path for said transaction from said first connection in said first data network to a second connection in a second data network; and  
35 automatically connecting said terminal device to a serving node in said second data network via said second connection.

30

35

3. A method for managing a transaction via a communications path between a terminal device and a serving node in a data network, said method comprising the steps of:

40

establishing an initial communications path via a first connection between said terminal device and a serving node in a first data network;  
identifying at least one data network application from a data base of said data network applications provided at said serving node in said first data network;  
45 receiving information from said serving node in said first data network for reconfiguring said terminal device for implementation of a communication path via an alternate connection between said terminal device and at least one of said identified data network applications in a second data network; and  
in response to a selection signal from a user, automatically connecting said terminal device to a selected one of said identified data network applications via said alternate connection.

50

4. A method for managing a transaction via a communications path between a terminal device and a serving node in a data network, said method comprising the steps of:

55

establishing an initial communications path via a first connection between said terminal device and a serving node in a first data network;  
selecting an off-line form application from a data base provided at said serving node in said first data network;  
receiving information from said serving node in said first data network for reconfiguring said terminal device for implementation of a communication path via a second connection between said terminal device and said



selected off-line form application in a second data network; and  
in response to, a selection signal from a user, automatically connecting said terminal device to said selected off-line form application.

- 5 5. The method for managing a transaction of Claim 1 or 2 including the further step of recognizing a signal to reconfigure said communications path from said first connection to said second connection.
6. The method for managing a transaction of Claim 3 wherein said selected data network application is operated at a serving node in said second data network.
- 10 7. The method for managing a transaction of Claim 4 wherein said selected off-line form application is operated at a serving node in said second data network.
8. The method for managing a transaction of one of the Claims 1, 2, 6 or 7 wherein said serving nodes in said first and said second data networks are manifested in a common node.
- 15 9. The method for managing a transaction of Claim 1 or 2 wherein said step of receiving information includes the further step of effecting said reconfiguration of said communications path.
- 20 10. The method for managing a transaction of Claim 1 or 2 wherein said step of automatically connecting includes the step of automatically disconnecting said first connection prior to implementation of said second connection.
11. The method for managing a transaction of Claim 1 or 2 including the further steps of:  
25 automatically disconnecting said second connection in response to a user signal; and  
reconfiguring said terminal device to enable, in response to user instruction, an implementation of a connection via an identified data network.
- 30 12. The method for managing a transaction of Claim 11 wherein said step of automatically reconfiguring said terminal device includes the step of effecting said implementation of said connection via said identified data network.
13. The method for managing a transaction of Claim 2 wherein said step of causing said selected information items to be downloaded includes the further step of causing said selected information items to be displayed at said terminal device.
- 35 14. The method for managing a transaction of Claim 13 wherein said displayed selected items can be edited by a user at said terminal device.
15. The method for managing a transaction of Claim 13 wherein display characteristics for said displayed selected items can be controlled at said terminal device.
- 40 16. The method for managing a transaction of Claim 2 wherein said step of automatically connecting includes the step of uploading said selected information items from said terminal device to said service provider via said second connection.
- 45 17. The method for managing a transaction of Claim 3 including the further steps of:  
automatically disconnecting said alternate connection in response to a user signal; and  
50 reconfiguring said terminal device to enable implementation of a pre-selected connection between said terminal device and an identified data network.
18. The method for managing a transaction of Claim 17 wherein said step of automatically reconfiguring said terminal device includes the further step of effecting said implementation of said pre-selected connection.
- 55 19. The method for managing a transaction of Claim 4 including the further step of downloading from said serving node in said first data network to said terminal device of an off-line form related to said off-line form application.
20. The method for managing a transaction of Claim 4 including the further step of uploading said downloaded off-line

form from said terminal device to said selected off-line form application, after processing by a user.

21. The method for managing a transaction of Claim 4 including the further steps of:

5 automatically disconnecting said connection to said selected off-line form application in response to a user signal; and  
reconfiguring said terminal device to enable implementation of a pre-selected connection between said terminal device and an identified data network.

10 22. The method for managing a transaction of Claim 21 wherein said step of automatically reconfiguring said terminal device includes the further step of effecting said implementation of said pre-selected connection.

23. A method for managing connections between a terminal device and at least one information source/processor wherein at least two of said connections are implemented via separate communications networks, comprising the steps of:

15 recognizing a signal for connection to an information source/processor via a communications network other than a communications network for which a predetermined connection is configured;  
causing said terminal device to implement a connection to said information source/processor via said other communications network; and  
20 upon termination of said information source/processor connection via said other communications network, automatically reconfiguring a connection criteria in said terminal device to enable said terminal device to implement, in response to user instruction, a connection via an alternative one of said communications networks.

25 24. The method for managing connections of Claim 23 wherein said recognizing step occurs at a point when said terminal device is connected to a given source/processor.

25. The method for managing connections of Claim 23 wherein information items may be selected by a user at said terminal device from said given source/processor, and including the further step of causing said selected information items to be downloaded from said source/processor to said terminal device.

30 26. The method for managing connections of Claim 25 wherein said step of effecting connection includes the further step of uploading said selected information items from said terminal device to said other information source/processor.

35 27. The method for managing connections of Claim 26 wherein said selected information items are processed by said user at said terminal device prior to uploading to said other information source/processor.

40 28. The method for managing connections of Claim 24 including the further step of causing said given source/processor to download to said terminal device configuration data for enabling said step of effecting connection to said other information source/processor.

45 29. The method for managing connections of Claim 24 including the further step of causing said other source/processor to download to said terminal device configuration data for enabling said step of automatically restoring a prior connection criteria in said terminal device.

30. A method for enhancing security of certain data in an on-line information transaction comprising the steps of:

50 bifurcating said information transaction into a first portion comprising said certain data and a remaining portion, wherein said remaining portion is carried out via a public on-line communications connection between a terminal device and a public information server;  
causing said first portion to be carried out via a secure private on-line communications connection between said terminal device and a private information server; and  
55 automatically reconfiguring network access means in said terminal device to switch between said public connection and said private connection.

FIG. 1

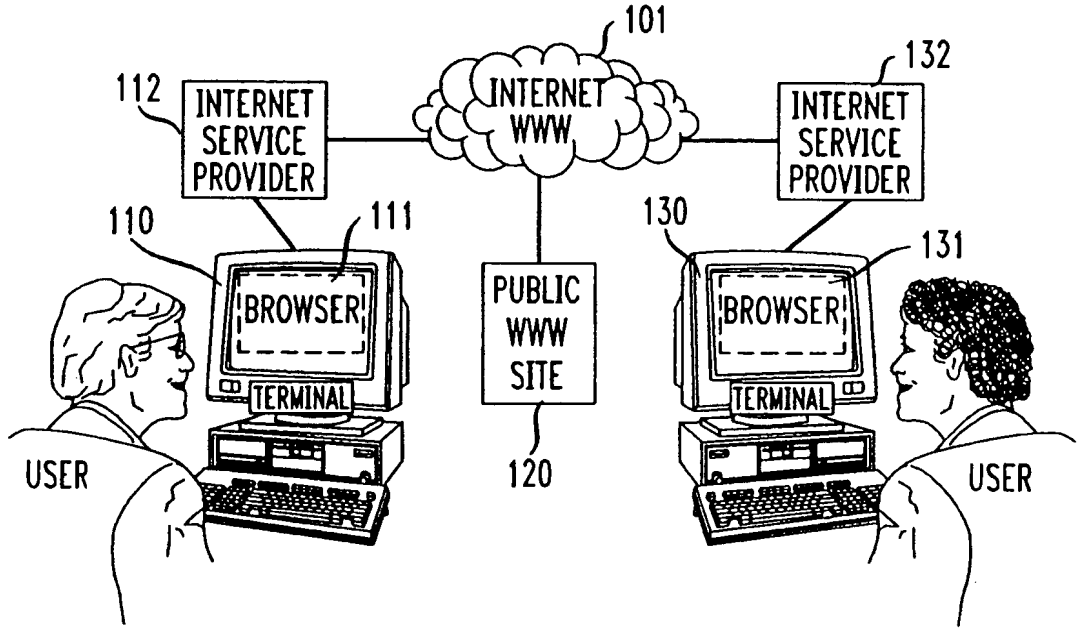


FIG. 2

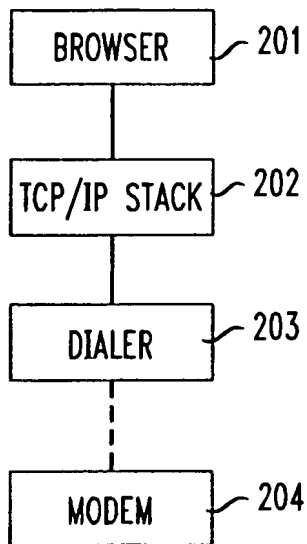


FIG. 3

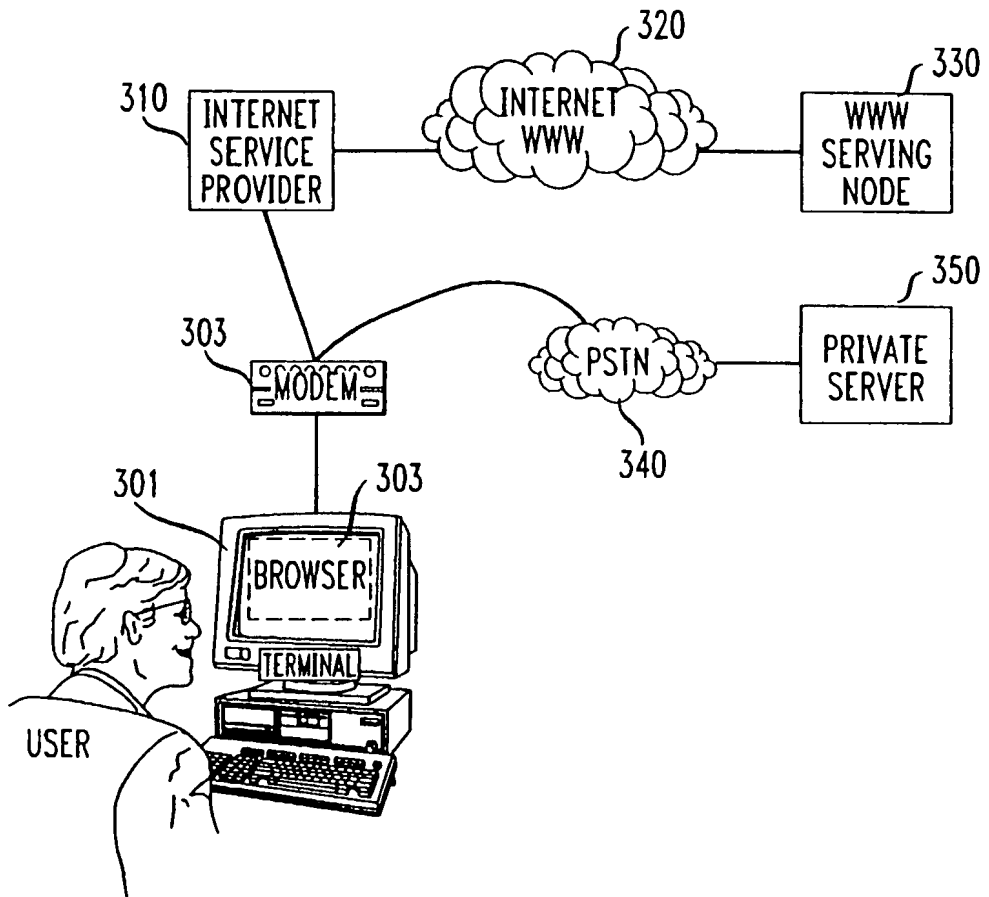


FIG. 4

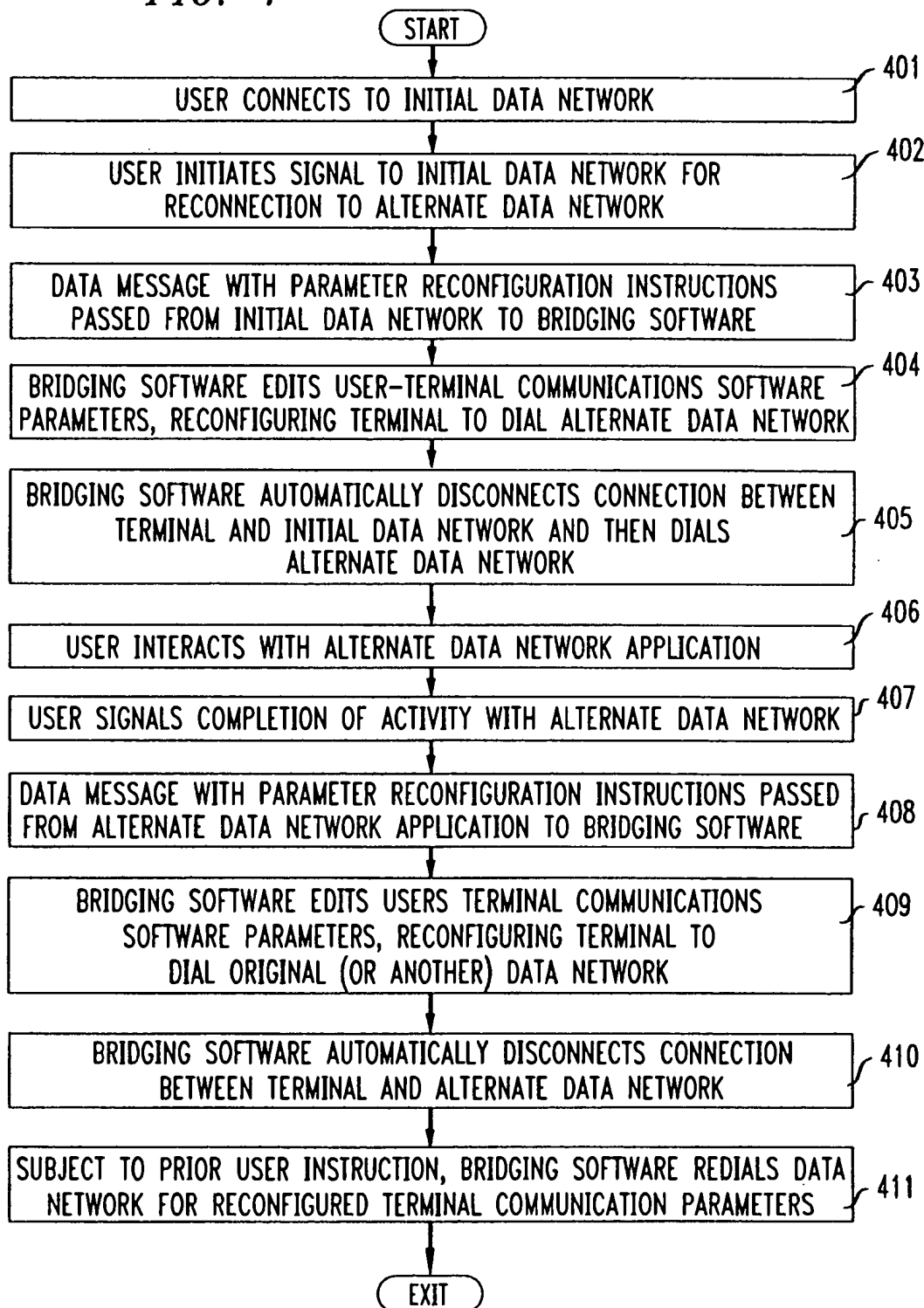


FIG. 5A

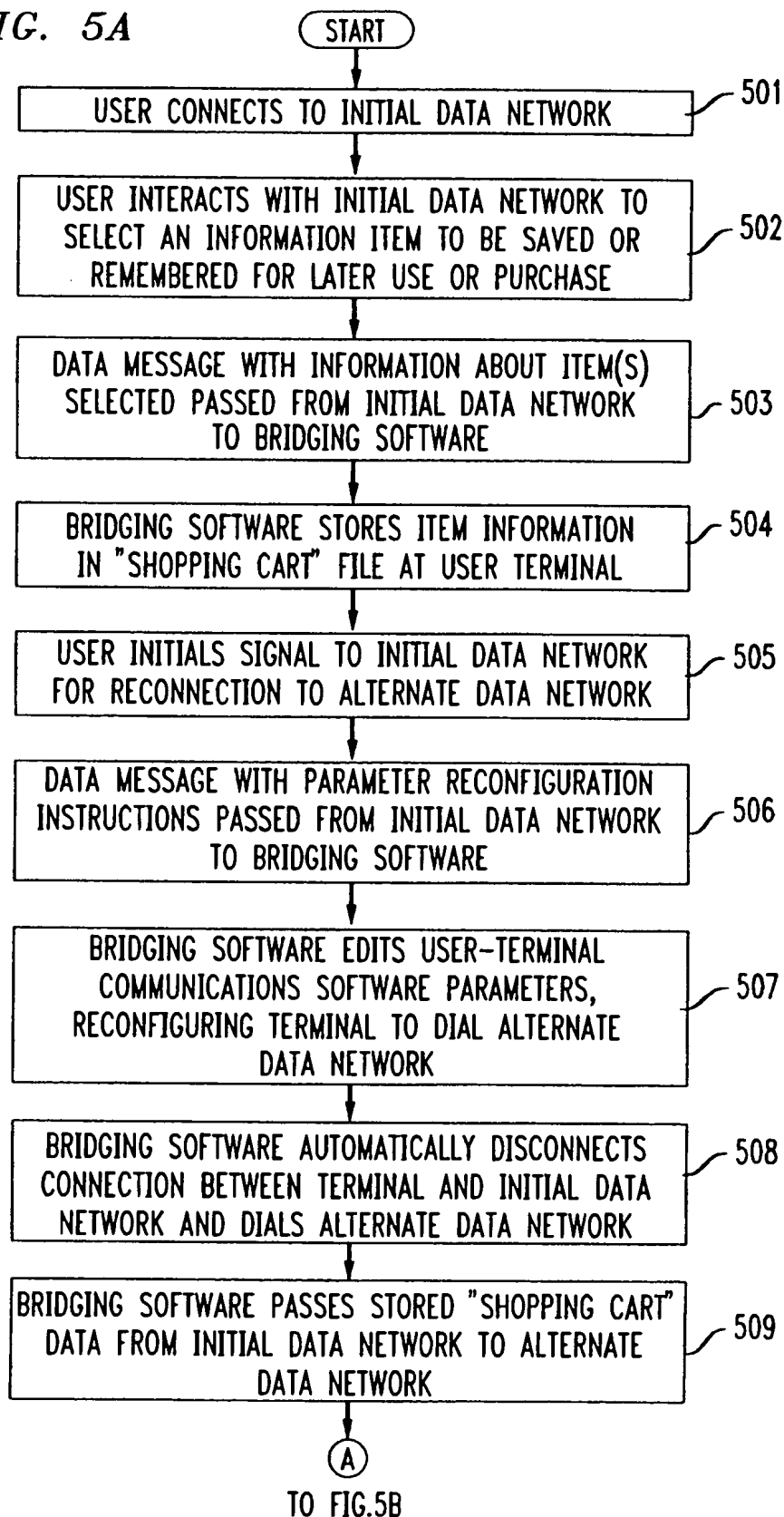


FIG. 5B

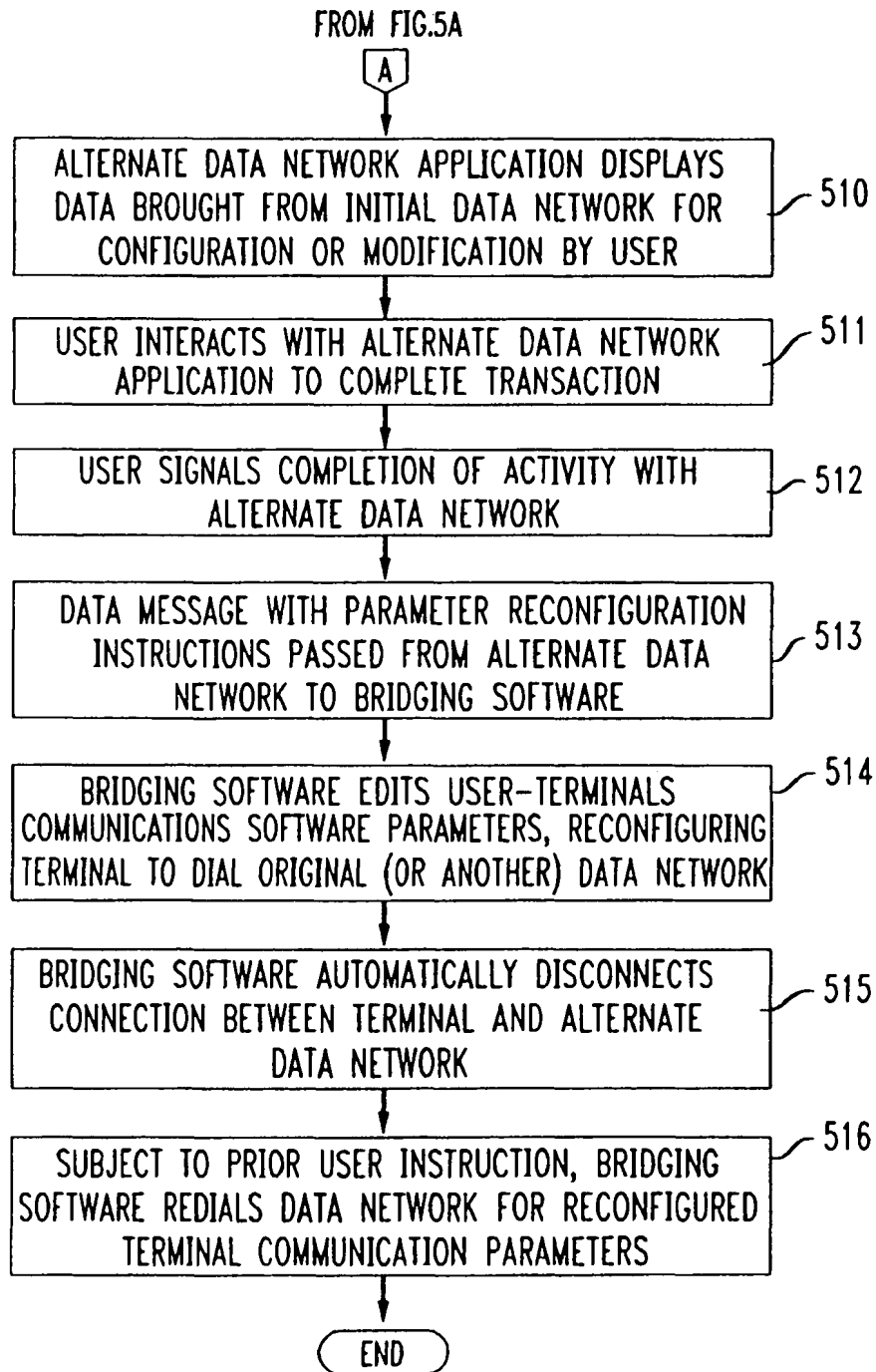


FIG. 6A

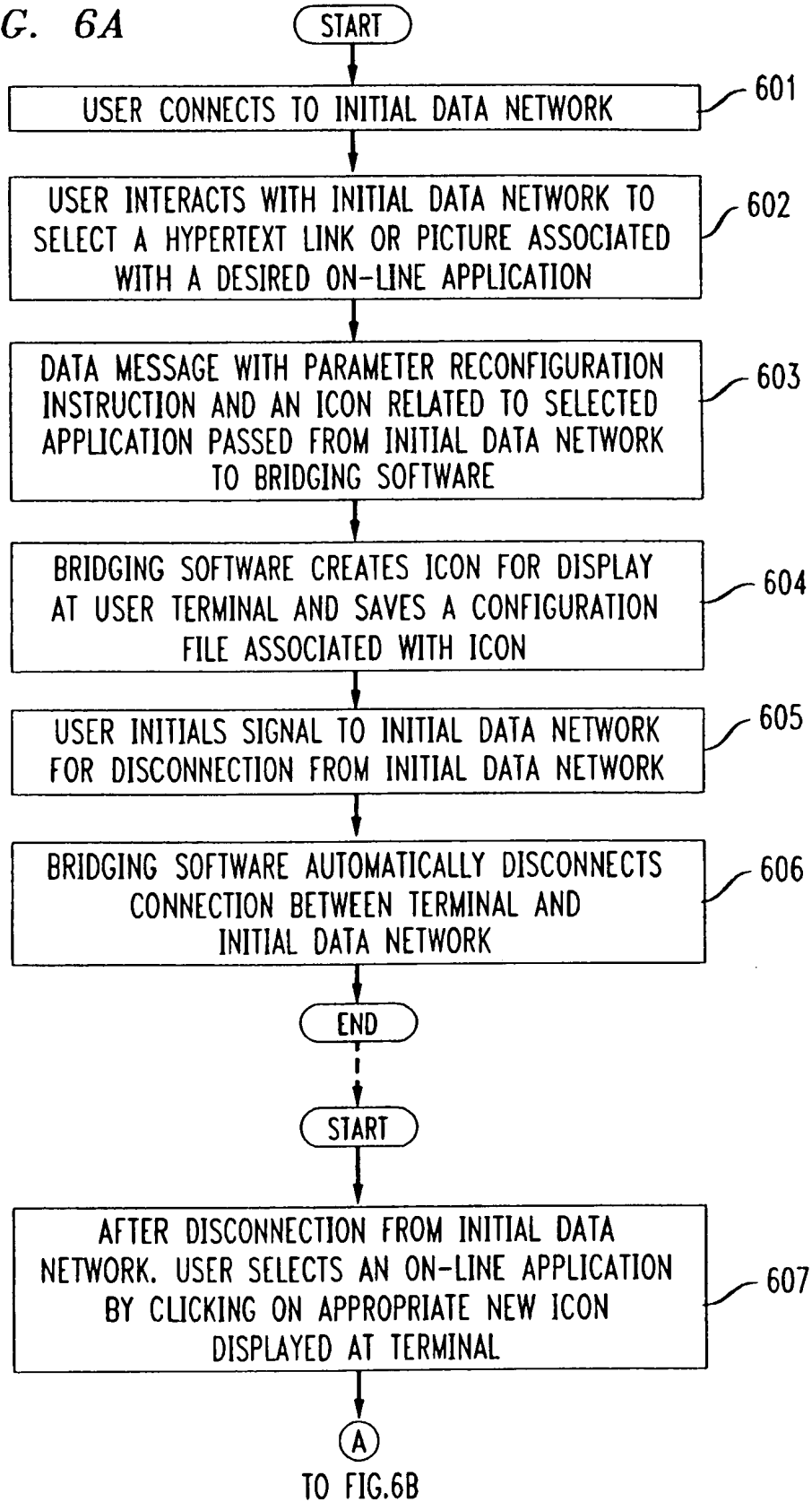




FIG. 6B

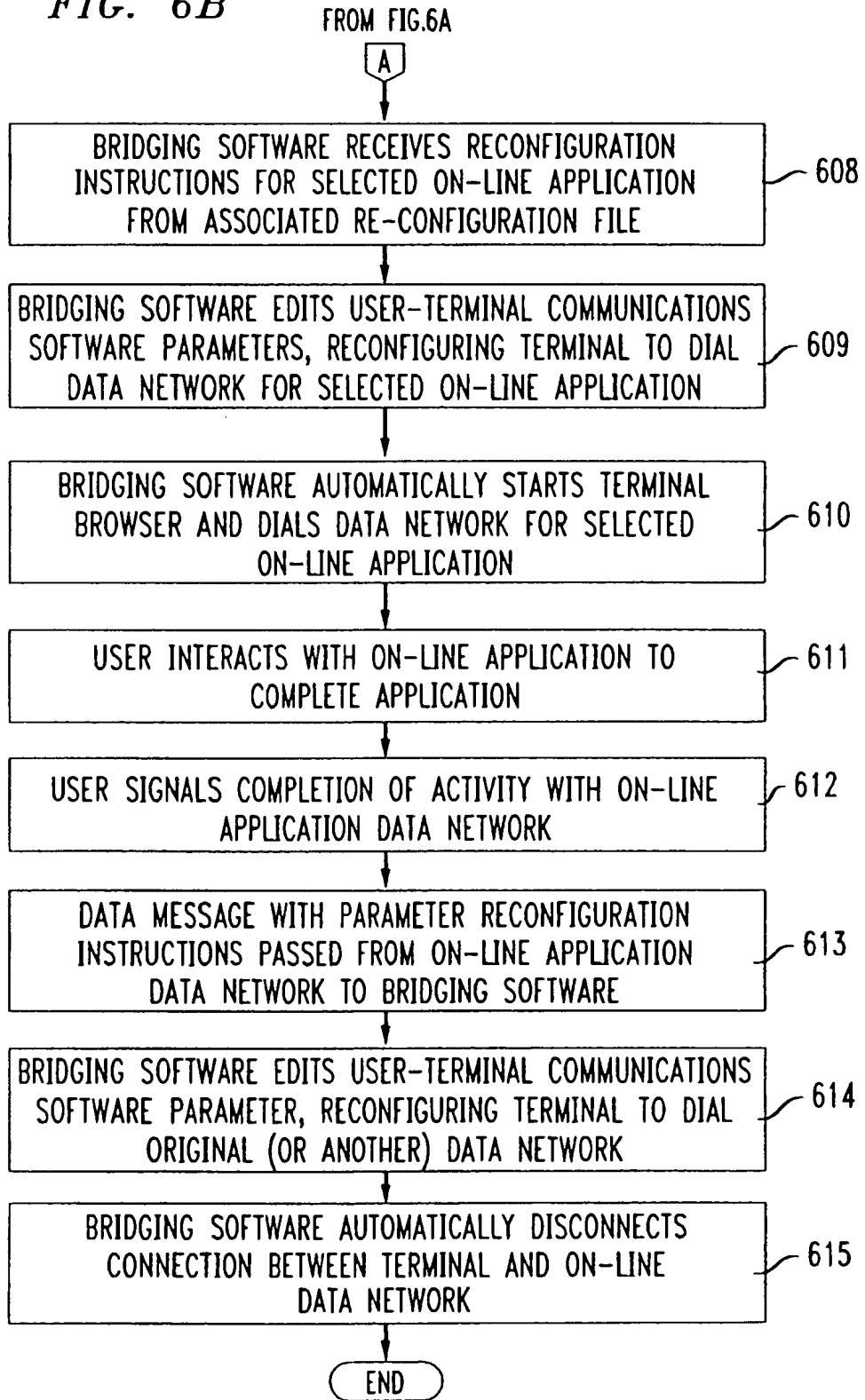


FIG. 7A

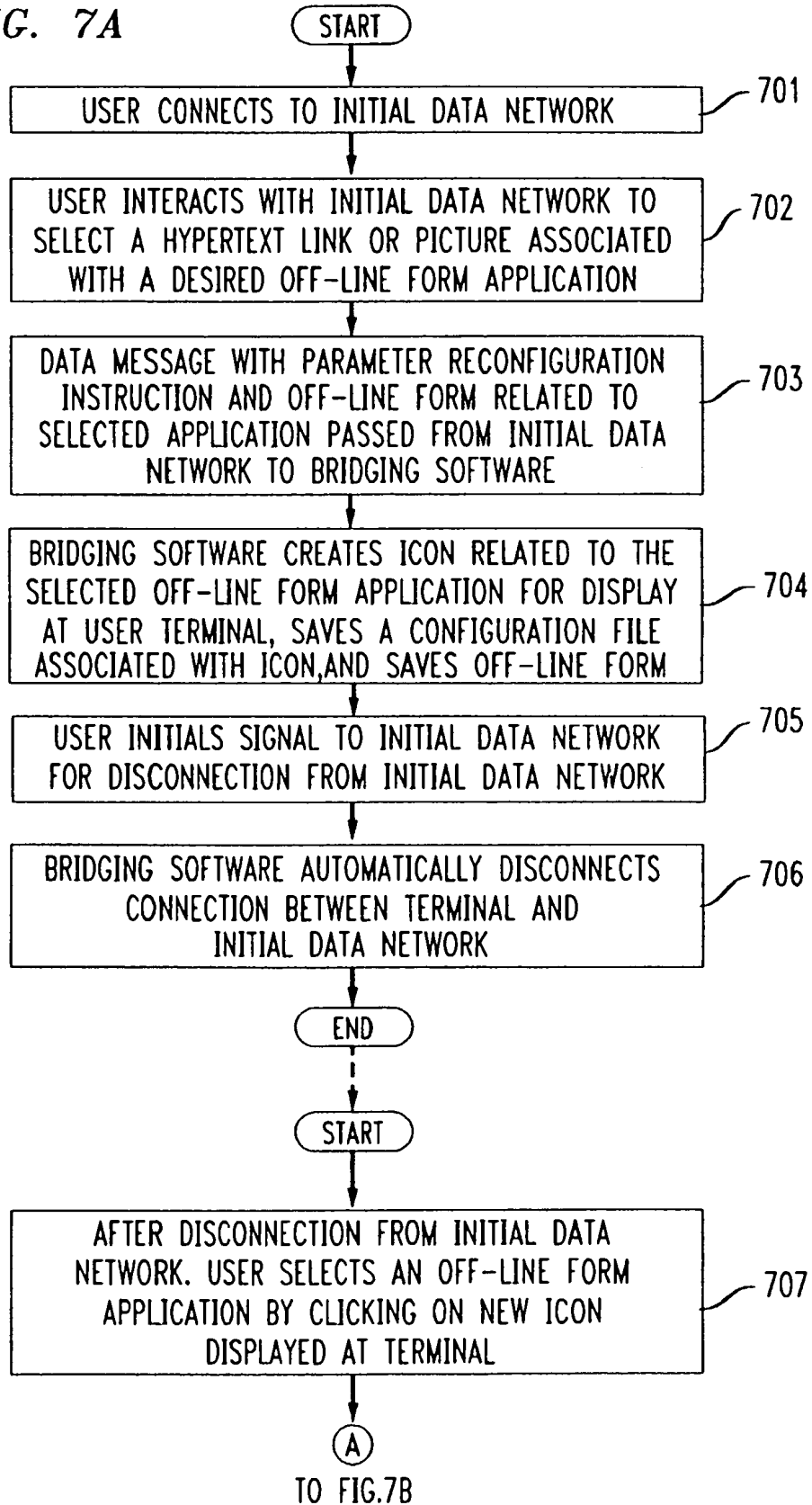
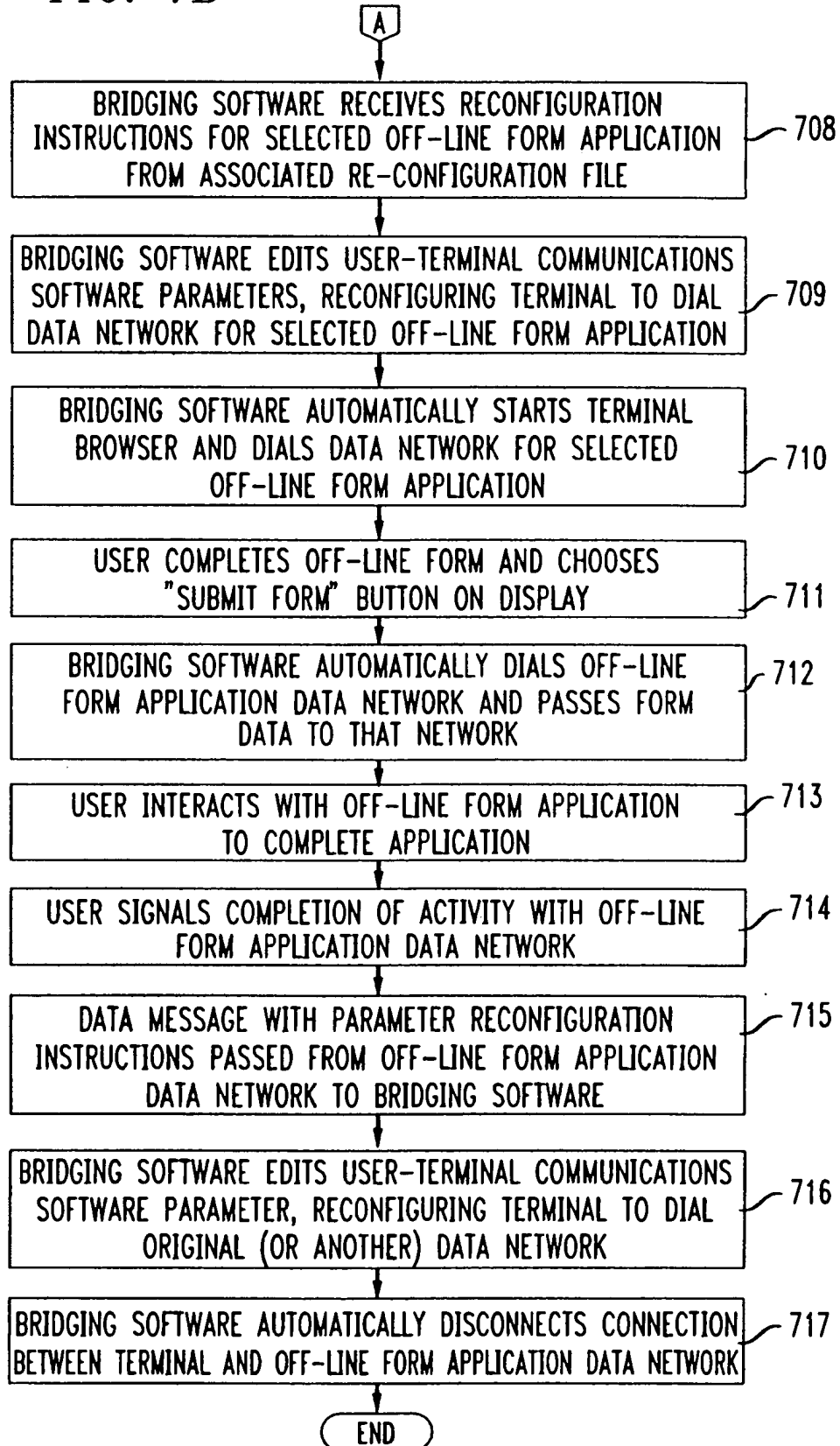


FIG. 7B

FROM FIG.7A





Europäisches Patentamt  
European Patent Office  
Office européen des brevets



(11) EP 0 814 589 A3

(12) EUROPEAN PATENT APPLICATION

(88) Date of publication A3:  
30.08.2000 Bulletin 2000/35

(51) Int. Cl.<sup>7</sup>: H04L 29/06, G06F 17/60,  
G07F 19/00

(43) Date of publication A2:  
29.12.1997 Bulletin 1997/52

(21) Application number: 97109792.8

(22) Date of filing: 16.06.1997

(84) Designated Contracting States:  
AT BE CH DE DK ES FR GB GR IE IT LI LU MC NL  
PT SE

• Kimmeth, Thomas  
Gladstone, N.J. 07977 (US)  
• Nusbaum, Kurt  
Downers Grove, Illinois 60515 (US)

(30) Priority: 19.06.1996 US 667524

(71) Applicant: AT&T Corp.  
New York, NY 10013-2412 (US)

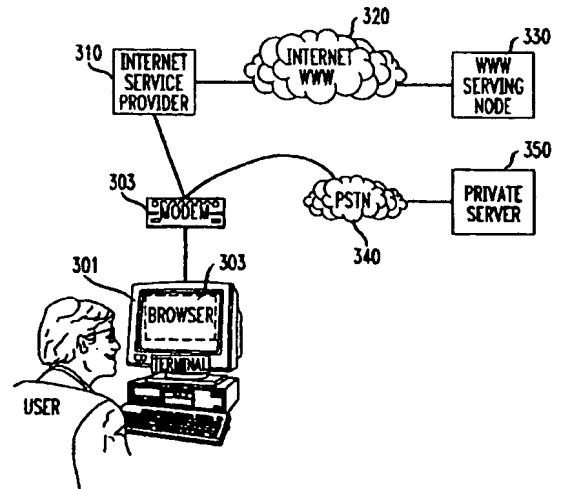
(74) Representative: Kuhnen & Wacker  
Patentanwaltsgesellschaft mbH,  
Alois-Steinecker-Strasse 22  
85354 Freising (DE)

(72) Inventors:  
• Harwood, Jonathan P.  
Morganville, N.J. 07751 (US)

(54) System and method for automated network reconfiguration

(57) A method is disclosed for providing an enhanced level of security for sensitive or proprietary information associated with information transactions in a public network, such as the Internet (101). In carrying out that method, an on-line information transaction is bifurcated between a generalized information access portion of such a transaction and an exchange of sensitive user information. With such a bifurcation, the generalized information access portion of the transaction, which generally would constitute the more substantial (in terms of network resources) portion of the transaction, would be handled via a non-secure network, usually a public network such as the Internet (320). The portion of the transaction involving sensitive user information, on the other hand, would be handled by a separate secure connection, such as a private network, or intranetwork (340). An important characteristic of this bifurcation arrangement is the provision of a means for automated reconfiguration of a user terminal as between accessing the generalized information via the non-secure network and access to the secure communications network for the exchange of sensitive user information. Such an automated reconfiguration will be carried out without the necessity for any action on the part of the user, and indeed will be largely invisible to the user.

FIG. 3



EP 0 814 589 A3



European Patent Office

EUROPEAN SEARCH REPORT

Application Number  
EP 97 10 9792

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.8)
A	EP 0 590 861 A (AMERICAN TELEPHONE & TELEGRAPH) 6 April 1994 (1994-04-06) * column 1, line 9 - line 24 * * column 3, line 15 - line 17 * * column 3, line 43 - column 4, line 7 *	1-30	H04L29/06 G06F17/60 G07F19/00
A	BAGSHAW E: "NET PROFITS" APRIL 1995 PC PRO, pages 176,178-182, XP002059701 ISSN: 1355-4603 * left-hand column, line 181, last paragraph *	1-30	
A	BELLARE M ET AL: "IKP - A FAMILY OF SECURE ELECTRONIC PAYMENT PROTOCOLS" PROCEEDINGS OF THE FIRST USENIX WORKSHOP ON ELECTRONIC COMMERCE, JULY 11-12, 1995, pages 89-106, XP000579445 * page 95, right-hand column *	1-30	
A	WO 96 00485 A (TELEFON AB LM ERICSSON) 4 January 1996 (1996-01-04) * page 6, line 1 - line 3; figure 1 * * page 10, line 13 - page 11, line 29 *	1-30	TECHNICAL FIELDS SEARCHED (Int.Cl.8)  H04L G06F G07F
The present search report has been drawn up for all claims			
Place of search	Date of completion of the search	Examiner	
THE HAGUE	5 July 2000	Vercauteren, S	
CATEGORY OF CITED DOCUMENTS		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 A : technological background O : non-written disclosure P : intermediate document & : member of the same patent family, corresponding document	
X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category			

EPO FORM 1503 (03.92) (PatCOI)

**ANNEX TO THE EUROPEAN SEARCH REPORT  
ON EUROPEAN PATENT APPLICATION NO.**

EP 97 10 9792

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.

05-07-2000

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0590861 A	06-04-1994	CA 2100134 A	30-03-1994
		JP 7129671 A	19-05-1995
		MX 9305830 A	30-06-1994
		US 5485510 A	16-01-1996
WO 9600485 A	04-01-1996	US 5668876 A	16-09-1997
		AU 692881 B	18-06-1998
		AU 2688795 A	19-01-1996
		CA 2193819 A	04-01-1996
		EP 0766902 A	09-04-1997
		FI 965161 A	13-02-1997
		JP 10502195 T	24-02-1998

EPD FORM P0459

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

FIG. 1 (Prior Art)

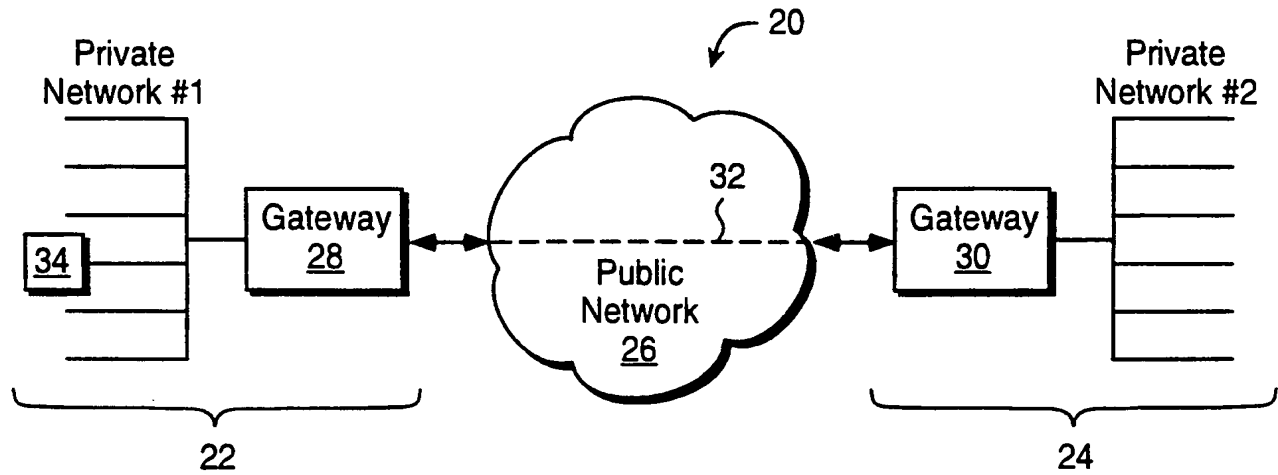


FIG. 2

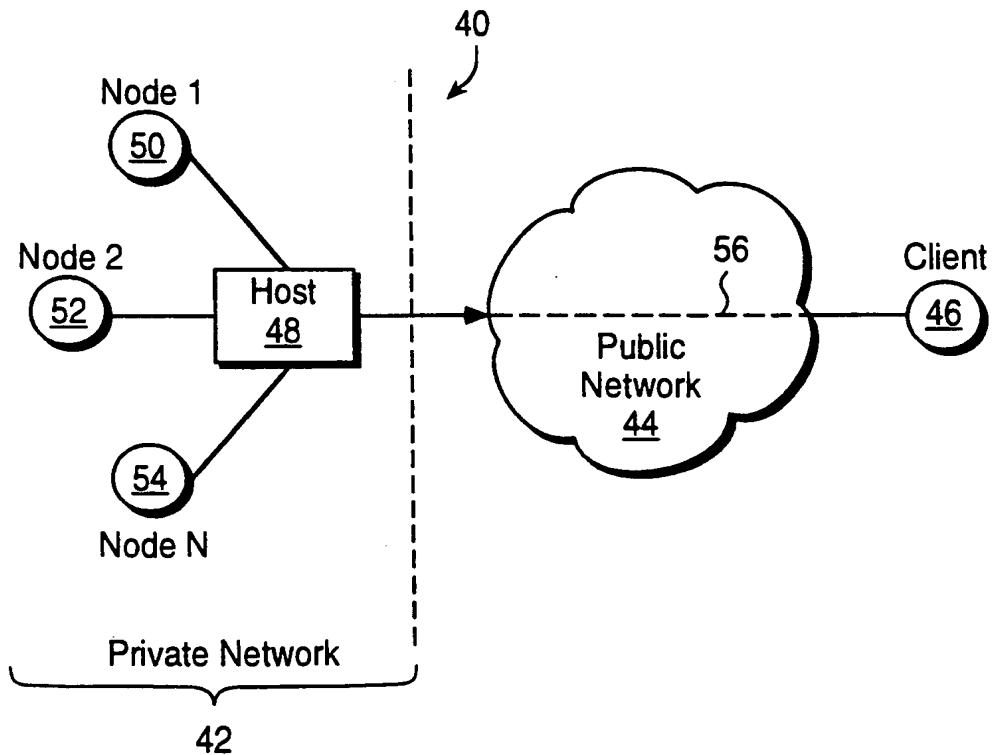


FIG. 3

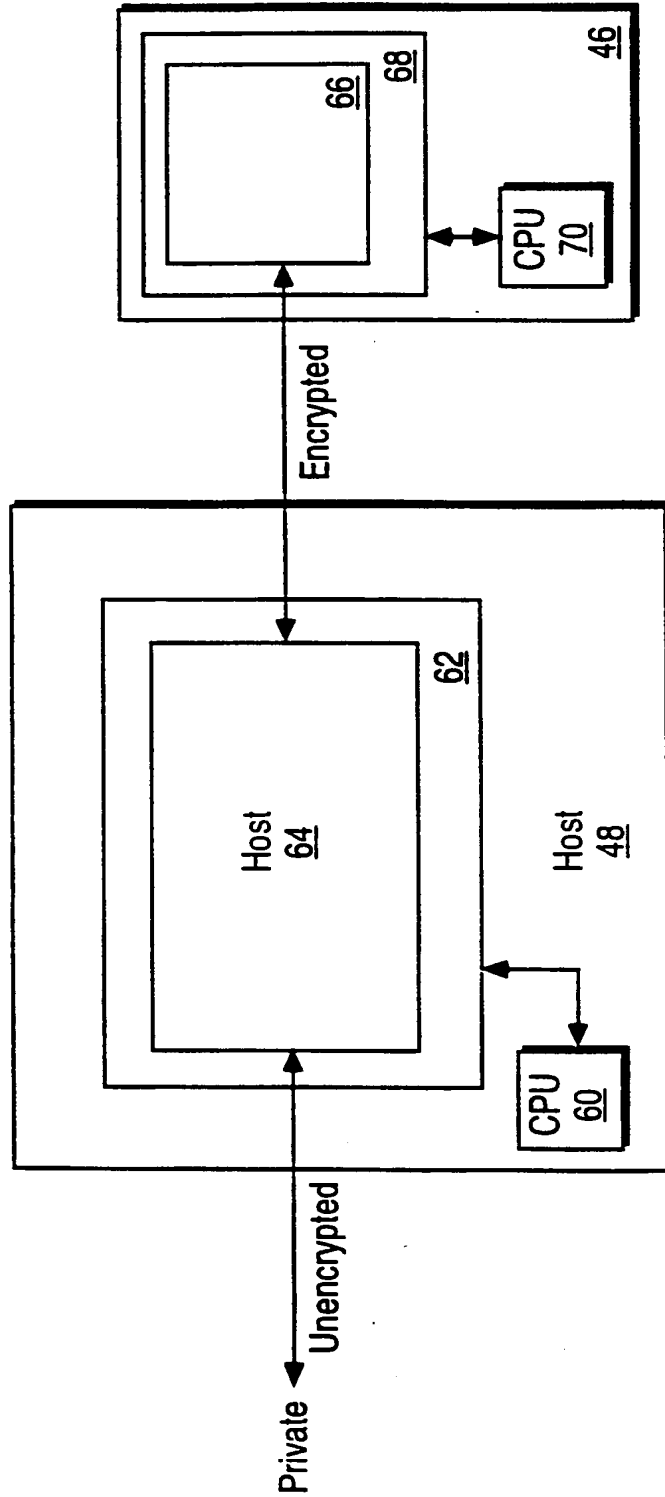
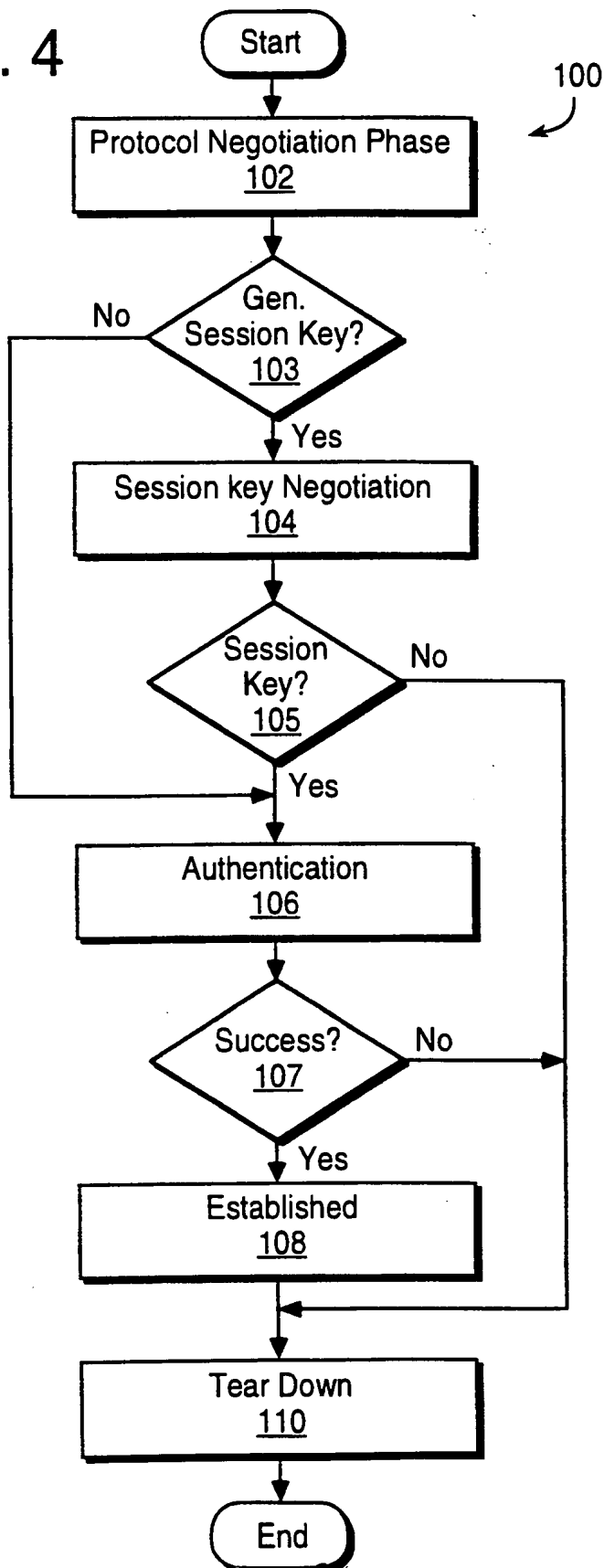




FIG. 4



INTERNATIONAL SEARCH REPORT

International application No.  
PCT/US99/01523

**A. CLASSIFICATION OF SUBJECT MATTER**  
IPC(6) : G06F 13/00; H04L 9/30  
US CL : 709/245 . 229 ; 380/30  
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. : 709/245 . 229 . 228 . 226; 380/30 . 23

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, Internet  
Search terms : private and public network, protocols, authentication, session key, encrypt, decrypt, encapsulation.

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,416,842 A (AZIZ) 16 MAY 1995, Abstract, Figs. 2 and 5 and 6, col. 4 line 65 - col. 5 line 48, col. 6 lines 3-35 and lines 40-51, col. 7 lines 16-35, col. 7 line 63 - col. 8 line 2	1-14
Y	US 5,550,984 A (GELB) 27 AUGUST 1996, Abstract, Fig. 1, col. 5 line 45 - col. 6 line 51, col. 7 line 58 - col. 8 line 19	1-14
Y	US 5,548,646 A (AZIZ ET. AL.) 20 AUGUST 1996, Abstract, Figs. 5 and 6, col. 9 lines 1-50, col. 10 line 32 - col. 11 line 67	1-14
Y,P	US 5,835,726 A (SHWED ET. AL.) 10 November 1998, Abstract, Fig. 21, col. 20 line 41 - col. 21 line 7, col. 22 line 19 - col. 23 line 9.	1-14

Further documents are listed in the continuation of Box C.  See patent family annex.

* Special categories of cited documents.	*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
*A* document defining the general state of the art which is not considered to be of particular relevance	*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
*E* earlier document published on or after the international filing date	*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
*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)	*Z* document member of the same patent family
*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	

Date of the actual completion of the international search: 21 MAY 1999  
Date of mailing of the international search report: 02 JUN 1999

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: AHMAD MATAR  
Telephone No. (703) 305-4731

PATENT COOPERATION TREAT

PCT

INTERNATIONAL SEARCH REPORT

(PCT Article 18 and Rules 43 and 44)

Applicant's or agent's file reference <b>00479.00029</b>	<b>FOR FURTHER ACTION</b> see Notification of Transmittal of International Search Report (Form PCT/ISA/220) as well as, where applicable, item 5 below.	
International application No. <b>PCT/US 01/ 04340</b>	International filing date (day/month/year) <b>12/02/2001</b>	(Earliest) Priority Date (day/month/year) <b>15/02/2000</b>

Applicant

**SCIENCE APPLICATIONS INTERNATIONAL CORPORATION**

This International Search Report has been prepared by this International Searching Authority and is transmitted to the applicant according to Article 18. A copy is being transmitted to the International Bureau.

This International Search Report consists of a total of 5 sheets.

It is also accompanied by a copy of each prior art document cited in this report.

1. **Basis of the report**

a. With regard to the **language**, the international search was carried out on the basis of the international application in the language in which it was filed, unless otherwise indicated under this item.

the international search was carried out on the basis of a translation of the international application furnished to this Authority (Rule 23.1(b)).

b. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, the international search was carried out on the basis of the sequence listing :

contained in the international application in written form.

filed together with the international application in computer readable form.

furnished subsequently to this Authority in written form.

furnished subsequently to this Authority in computer readable form.

the statement that the subsequently furnished written sequence listing does not go beyond the disclosure in the international application as filed has been furnished.

the statement that the information recorded in computer readable form is identical to the written sequence listing has been furnished

2.  **Certain claims were found unsearchable** (See Box I).

3.  **Unity of invention is lacking** (see Box II).

4. With regard to the **title**,

the text is approved as submitted by the applicant.

the text has been established by this Authority to read as follows:

**AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY**

5. With regard to the **abstract**,

the text is approved as submitted by the applicant.

the text has been established, according to Rule 38.2(b), by this Authority as it appears in Box III. The applicant may, within one month from the date of mailing of this international search report, submit comments to this Authority.

6. The figure of the **drawings** to be published with the abstract is Figure No.

as suggested by the applicant.

because the applicant failed to suggest a figure.

because this figure better characterizes the invention.

3a

None of the figures.

INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 01/04340

A. CLASSIFICATION OF SUBJECT MATTER  
 IPC 7 H04L12/56 H04L29/06 H04L12/46

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)  
 IPC 7 H04L

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 practical, search terms used)

EPO-Internal, WPI Data, PAJ, INSPEC, COMPENDEX

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of documents with such designations as appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 858 189 A (HITACHI LTD) 12 August 1998 (1998-08-12) column 6, line 35 -column 10, line 13 --- -/--	1-27

Further documents are listed in the continuation of box C.  Patent family members are listed in annex.

° Special categories of cited documents :

*A* document defining the general state of the art which is not considered to be of particular relevance	*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
*E* earlier document but published on or after the international filing date	*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
*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)	*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.
*O* document referring to an oral disclosure, use, exhibition or other means	*&* document member of the same patent family
*P* document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search  6 August 2002	Date of mailing of the international search report  20. 08. 2002
--	--

Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl. Fax: (+31-70) 340-3016	Authorized officer  Ströbeck, A.
--	--

INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 01/04340

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>MURTHY ET AL: "Congestion-oriented shortest multipath routing"                      PROCEEDINGS OF IEEE INFOCOM 1996.                      CONFERENCE ON COMPUTER COMMUNICATIONS.                      FIFTEENTH ANNUAL JOINT CONFERENCE OF THE                      IEEE COMPUTER AND COMMUNICATIONS                      SOCIETIES. NETWORKING THE NEXT GENERATION.                      SAN FRANCISCO, MAR. 24 - 28, 1996,                      PROCEEDINGS OF INFOCOM, L,                      vol. 2 CONF. 15,                      24 March 1996 (1996-03-24), pages                      1028-1036, XP010158171                      ISBN: 0-8186-7293-5                      abstract                      page 1028, left-hand column, line 38                      -right-hand column, line 29</p>	1-27
E	<p>WO 01 50688 A (TELEFONAKTIEBOLAGET LM                      ERICSSON (PUBL)) 12 July 2001 (2001-07-12)                      page 11, line 18 -page 13, line 21</p>	28,29,34
A	<p>WO 98 59470 A (TELEFONAKTIEBOLAGET LM                      ERICSSON (PUBL))                      30 December 1998 (1998-12-30)                      page 4, line 5 -page 5, line 2</p>	28-39
X	<p>WO 99 48303 A (CISCO TECHNOLOGY, INC.)                      23 September 1999 (1999-09-23)                      page 1, line 8 -page 2, line 5                      page 5, line 33 -page 6, line 15                      page 7, line 21 - line 33</p>	40,50
A		41-49, 51-59
A	<p>JONES JIM ET AL: "Distributed Denial of                      Service Attacks: Defenses"                      INTERNET ARTICLE, 'Online! 2000,                      XP002208785                      Retrieved from the Internet:                      &lt;URL:www.bai.org/pdf/DDOS-defense.pdf &gt;                      'retrieved on 2002-08-05!                      paragraph '0005!</p>	60-66
X	<p>WO 99 38081 A (ASCEND COMMUNICATIONS INC)                      29 July 1999 (1999-07-29)                      page 9, line 13 -page 10, line 17                      page 11, line 10 -page 12, line 2</p>	67
A		68-71

INTERNATIONAL SEARCH REPORT

International application No.  
PCT/US 01/04340

Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This International Search Report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

- 1.  Claims Nos.:  
because they relate to subject matter not required to be searched by this Authority, namely:
  
- 2.  Claims Nos.:  
because they relate to parts of the International Application that do not comply with the prescribed requirements to such an extent that no meaningful International Search can be carried out, specifically:
  
- 3.  Claims Nos.:  
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

- 1.  As all required additional search fees were timely paid by the applicant, this International Search Report covers all searchable claims.
  
- 2.  As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.
  
- 3.  As only some of the required additional search fees were timely paid by the applicant, this International Search Report covers only those claims for which fees were paid, specifically claims Nos.:
  
- 4.  No required additional search fees were timely paid by the applicant. Consequently, this International Search Report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

- The additional search fees were accompanied by the applicant's protest.
- No protest accompanied the payment of additional search fees.

**FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210**

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. Claims: 1-27

A system and a method to balance the load between communication paths with varying transmission quality.

2. Claims: 28-39

A system and a method to prevent someone from learning requested IP addresses by intercepting DNS requests.

3. Claims: 40-59

A method to prevent a denial-of-service attack from an unauthenticated user flooding dummy data packets on to a low bandwidth link.

4. Claims: 60-66

A method to prevent an authenticated user residing within a secure system from flooding it with dummy data packets.

5. Claims: 67-71

A method to allocate memory in a central computer communicating with a potentially large number of client computers.

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 01/04340

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
EP 0858189	A	12-08-1998	JP 10224400 A	21-08-1998
			EP 0858189 A2	12-08-1998
			US 6112248 A	29-08-2000
WO 0150688	A	12-07-2001	SE 517217 C2	07-05-2002
			AU 2564501 A	16-07-2001
			WO 0150688 A1	12-07-2001
			SE 9904841 A	30-06-2001
			US 2001006523 A1	05-07-2001
WO 9859470	A	30-12-1998	AU 8052398 A	04-01-1999
			SE 9702385 A	24-12-1998
			WO 9859470 A2	30-12-1998
WO 9948303	A	23-09-1999	AU 3098299 A	11-10-1999
			WO 9948303 A2	23-09-1999
WO 9938081	A	29-07-1999	US 6055575 A	25-04-2000
			AU 2562599 A	09-08-1999
			CA 2318267 A1	29-07-1999
			EP 1064602 A1	03-01-2001
			WO 9938081 A1	29-07-1999



#17



# MATCH & RETURN

## PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	)	Group Art Unit: 2153
	)	
Edmond Colby Munger et al.	)	Examiner: Krisna Lim
	)	
Serial No. 09/504,783	)	Attorney Docket No. 000479.85672
	)	
Filed: February 15, 2000	)	

For: IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

### INFORMATION DISCLOSURE STATEMENT

RECEIVED

Assistant Commissioner of Patents  
Washington, D.C. 20231

SEP 10 2002  
Technology Center 2100

Sir:

In accordance with Applicants' duty of disclosure, and pursuant to 37 C.F.R. § 1.97(d), the following information is submitted for consideration by the United States Patent and Trademark Office in connection with the above-captioned application. The information is identified on the attached PTO 1449 form.

Applicants do not waive any right to take appropriate action to establish patentability over the listed documents should they be applied as references against the claims of the present application.

The undersigned certifies under 37 C.F.R. § 1.97(e)(1) that each item of information contained in this information disclosure statement was first cited in a communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of this statement. A copy of the foreign search report is attached.

09/13/2002 HMKZ11 00000084 190733 09504783

01 FC:126 180.00 CH

**Information Disclosure Statement**

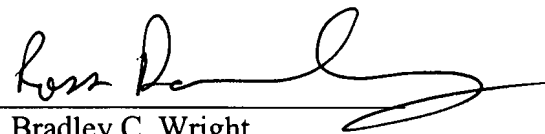
Serial No. 09/504,783

The Commissioner is authorized to charge the \$180 fee to our Deposit Account No. 19-0733. No additional fees are believed due to ensure consideration of the attached documents by the Examiner. However, if any fees are required or an overpayment of fees made, the Commissioner is hereby authorized to debit or credit our Deposit Account No. 19-0733, as necessary.

Respectfully submitted,

Date: September 12, 2002

By: \_\_\_\_\_



Bradley C. Wright  
Registration No. 38,061

Banner & Witcoff, LTD  
1001 G Street, N.W.  
Washington, D.C. 20001-4597  
(202) 508-9100

**Reg. No. 49,024**

BCW/RAD/mmd

PTO-1449 (Modified)  U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE  INFORMATION DISCLOSURE STATEMENT BY APPLICANT	ATTY. DOCKET NO. 000479.85672	SERIAL NUMBER 09/504,783
	APPLICANT Edmond Colby Munger et al.	
	FILING DATE February 15, 2000	GROUP ART UNIT 2153



U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO
<i>K</i>	0 858 189	8/12/98	EPO			
<i>K</i>	WO 01 50688	7/12/01	PCT			
<i>K</i>	WO 98 59470	12/30/98	PCT			
<i>K</i>	WO 99 48303	9/23/99	PCT			
<i>K</i>	WO 99 38081	7/29/99	PCT			

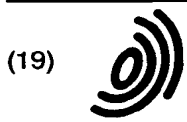
**RECEIVED**  
**SEP 19 2002**  
 Technology Center 2100

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

<i>K</i>	Search Report (dated 8/20/02), International Application No. PCT/US01/04340
<i>K</i>	Shree Murthy et al., "Congestion-Oriented Shortest Multipath Routing", Proceedings of IEEE INFOCOM, 1996, pages 1028-1036
<i>K</i>	Jim Jones et al., "Distributed Denial of Service Attacks: Defenses", Global Integrity Corporation, 2000, pages 1-14

EXAMINER <i>KRISNA LIM</i>	DATE CONSIDERED <i>11/7/02</i>
EXAMINER: Initial citation if reference was considered. Draw line through citation if not in conformance to MPEP 609 and not considered. Include copy of this form with next communication to applicant.	

IDS w/1449 form filed: September 12, 2002



Europäisches Patentamt  
 European Patent Office  
 Office européen des brevets



(11) EP 0 858 189 A2

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication: 12.08.1998 Bulletin 1998/33  
 (51) Int. Cl.<sup>6</sup>: H04L 12/46  
 (21) Application number: 98101286.7  
 (22) Date of filing: 26.01.1998

(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 MK RO SI  
 (30) Priority: 05.02.1997 JP 22402/97  
 (71) Applicant: Hitachi, Ltd.  
 Chiyoda-ku, Tokyo 101-0062 (JP)  
 (72) Inventors:  
 • Maciel, Frederico Buchholz  
 Kokubunji-shi, Tokyo (JP)

• Kitai, Katsuyoshi  
 Tokyo (JP)  
 • Higuchi, Tatsuo  
 Hillsboro, OR 97124 (US)  
 • Yoshizawa, Satoshi  
 Musashino-shi, Tokyo (JP)  
 • Murahashi, Hideki  
 Hachioji-shi, Tokyo (JP)  
 (74) Representative:  
 Beetz & Partner  
 Patentanwälte  
 Steinsdorfstrasse 10  
 80538 München (DE)

(54) **Networking method**

(57) This invention provides dynamic balance of the traffic among data processing devices interconnecting networks and thereby improve the networking performance. For network traffic flowing between a first network and a second network, the traffic is distributed among

the data processing devices that act as routers according to the traffic amount. An algorithm for balancing the traffic is used to select appropriate data processing devices as routers.

TARGET	ROUTER			NEXT HOP	PRIORITY
	ID	PHYSICAL	NETWORK		
...	...	...	...	...	...
NETWORK 1	3a	31a-P	30a-N	31a	1
	3b	31b-P	30b-N	31b	1
	3c	—	30c-N	42b	2
...	...	...	...	...	...

**FIG. 6**

EP 0 858 189 A2

## Description

Background of the Invention

The present invention relates to the interconnection among data networks and more specifically to traffic load balancing, for instance, among multiple routers or among multiple interfaces in a parallel processing system (devices).

Computer networks are usually connected to other computer networks and the connection between the computer networks forms internets. Connection between two given networks is implemented by one or more data processing devices (see D.E. Comer, "Internetworking with TCP/IP," volume 1, Prentice Hall, 1991).

These data processing devices include, but are not limited to, routers, gateways and switches, and will be generally and interchangeably referred to as routers or gateways in this specification.

An example shown in Figure 1 represents a plurality of data processing devices (3a, 3b, 3c, 3d, 3e, 4a, 4b) and three networks (1, 2, 5). The data processing devices 3a, 3b act as routers between network 1 and network 2. The data processing device 3c acts as a router between networks 2 and 5. The data processing device 4b acts as a router between networks 1 and 5. The data processing devices are connected to the networks by network interfaces (30a, 30b, 30c, 30d, 30e, 31a, 31b, 31c, 40b, 41a and 41b). These network interfaces (30a, 30b, 30c, 30d, 30e, 31a, 31b, 31c, 42b, 41a and 41b) are identified by physical addresses and also by network addresses.

On various kinds of networks, the ARP protocol (Address Resolution Protocol) (see "Internet Engineering Task Force RFC 826") is used to correlate physical and network addresses. Physical and network addresses hereafter are given by a notation consisting of the interface number of Figure 1, a dash, and a suffix "P" or "N" indicating that the address is a physical address or a network address.

When a data processing device in one network communicates with a data processing device in another network, the communication between the two data processing devices is done by using one or more routers (data processing devices) between one network and the other network, to transfer a communication message (packet) from one network to the other network. Generally, the correlation between a network (target network) to which a packet (communication) is transferred to and a router that transfers the packet to the target network is shown in a routing table of the data processing devices (see "Internetworking with TCP/IP" cited above). This method for indicating the route is, hereafter, referred to as an "explicit routing table setup". In Figure 1, data processing devices 3a, 3b, 3c, 3d and 3e have routing table 32a, 32b, 32c, 32d and 32e, respectively. Figures 2a - 2e show an example of routing tables in the explicit routing setup for routes which trans-

fer packets from the data processing devices (3a, 3b, 3c, 3d, 3e) connected to the network 2 of Figure 1 to the network 1. Routing tables (32a, 32b, 32c, 32d, 32e) each have an entry (321a, 321b, 321c, 321d, 321e) representing a target network, an entry (322a, 322b, 322c, 322d, 322e) representing a next hop address of the target network, and a flag (323a, 323b, 323c, 323d, 323e). The flag may have the values "interface" or "gateway". When the value of the flag is "interface", the next hop address means the address of the network interface which is directly connected to the target network, in case the data processing device in question is directly connected to the target network. When the value of the flag is "gateway", the next hop address means the address of a router which transfers packets to the target network. This value of the flag is used in case a data processing device in question is not connected to the target network.

In the routing tables of the data processing devices 3a, 3b, the network addresses (31a-N, 31b-N) of their respective interfaces to the network 1 are shown as the next hops (322a, 322b), and the value of the flag is "interface" (323a, 323b). The routing tables of the data processing devices 3c, 3d, 3e give 30a-N as the next hop (322c, 322d, 322e) and the value of the flag is "gateway" (323c, 323d, 323e), thus showing that the data processing device 3a is a router to network 1.

Other methods can be used to interconnect two or more networks. Two of these methods (Proxy ARP, OSPF protocol) are explained below.

Proxy ARP (where "ARP" is the Address Resolution Protocol) is a method for making routers transparent in communication between two or more networks (see RFC 1027), by making one or more routers in the networks act as proxies. On communications from one network to another network, the routers reply ARP requests on the former network querying a network addresses in the later network, then receive communications on the former network addressed to the later network and route them to the later network. Thus, these routers transparently bridge two or more networks (refer to "Internetworking with TCP/IP" cited above). To this end, the correlation between the physical addresses and the network addresses needs to be set up.

Figure 3 shows an example of such a setup (proxy ARP setup), in which the correlation between network addresses 711 and physical addresses 712 is set as special entries on the ARP cache 71. In this example, the router 3a acts as a proxy for communication flowing from the network 1 to the network 2. A "public" flag 713 indicates that the entry should be used to answer ARP queries. In this example, any of ARP queries to the network addresses 30c-N, 30d-N and 30e-N in the network 2 will be answered by 31a-P. Figure 3 shows an example of this setup accomplished in the ARP cache of the data processing device 3a. This setup can be implemented in the ARP cache of any of the data processing

devices 3a, 3b, 4a and 4b connected to the network 1.

The proxy ARP setup as implemented above and the explicit routing table setup are both performed by the administrator of each data processing device. This means that these setups are static, i.e., once performed they remain the same and can only be changed by manual intervention by the administrator. Hence, when there is a malfunction or when a new router is installed, these setups must be changed manually.

A method of changing the routing table dynamically according to changes in the network is provided by the OSPF protocol (see RFC 1245, 1246 and 1247). In this case, the routers exchange routing information and change their routing tables according to this information.

The basic algorithm of the OSPF protocol is shown in Figure 4. The data processing device broadcasts a message including the networks it can reach and the distances to these networks determined by the number of hops (step 822), and also receives such messages from other data processing devices (step 823). When the route changes (step 824), each router calculates the shortest path from itself to each of the networks (step 825) and sets its routing table according to the paths (step 826).

Figure 5 shows an example of the use of the OSPF in the networks of Figure 1. In this figure, all the data processing devices (3a, 3b, 3c, 3d, 3e, 4a, 4b) interchange data by using the OSPF protocol and thus have a control add-on 91 for executing the OSPF basic algorithm. Alternatively, the OSPF can be used only in a subset of the networks 1, 2, 5 or in a subset of the data processing devices (3a, 3b, 3c, 3d, 3e, 4a, 4b), or in both subsets.

In a special case, interconnected networks include not only parallel processing devices but also massively parallel processing devices and workstation clusters. These parallel processing devices contain a plurality of nodes that are interconnected by networks. Examples of such machines are Fujitsu's AP3000, IBM's RS/6000 SP, and Digital Corp.'s Tru-Cluster. The case of a parallel processing device is shown in Figure 1, in which the data processing devices connected to the second network 2 are the nodes of the parallel processing device 6. In this configuration, the parallel processing device has multiple interfaces for other networks to improve the reliability and the networking performance. The data processing devices (4a, 4b) of other networks (1, 5) are mainly clients which access the services provided by the parallel processing device 6.

### Summary of the Invention

In the above examples (in the case of using Fig. 2 and Fig. 3), all communication between the network 1 and the data processing devices (3c, 3d, 3e) pass through only the data processing device 3a. Therefore, the data processing device 3a is a potential bottleneck

for this communication. This bottleneck can be alleviated by assigning a part of the communication to the router 3a (the data processing device 3a) and by assigning other part of the communication to the router 3b, so that the traffic (communication) passes through two routers. Herein, the data processing device is a normal computer which has a processing unit, a memory, a disk, a cache, an interface for the network, and so on, and an operating system; and an application program stored in the memory and the disk are executed by the processing unit. However, the network traffic changes with time, and if most of the network traffic flows through the data processing device assigned as one of the routers, at a certain time, the bottleneck persists. Hence, balance of the traffic among multiple routers must be controlled dynamically, rather than statically, in order to be effective.

The explicit routing table setup and the Proxy ARP setup are static and thus can provide static traffic balancing, but not dynamic traffic balancing.

The OSPF protocol definition (RFC 1247) states that "when there exist several equivalent routes to a destination, traffic is distributed equally among them," but does not mention the method of distributing the traffic. The OSPF protocol leaves the distribution of the traffic to the execution of the routing algorithm. Therefore, although in the OSPF the route changes dynamically, the OSPF by itself cannot provide dynamic traffic balancing.

Data processing devices which execute user programs such as personal computers and workstations, are often used as routers. Because the routing generates processing load, the processing performance of such data processing devices is decreased by the processing load created by the routing. Hence, when selecting a route so as to balance the network traffic, it is necessary to additionally consider the processing load to improve the networking performance and the processing throughput. As in the network traffic, the processing load changes with time and thus the route must be changed dynamically with the processing load. As described earlier, the explicit routing table setup and the Proxy ARP setup are static and therefore cannot provide such dynamic traffic balancing. The OSPF protocol selects the routers that give the shortest path to the target network but this selection does not consider the processing load of the routers. Therefore, the OSPF protocol cannot provide dynamic traffic balancing according to the processing load of the routers.

It is desirable that the routing be changed in the event of a network failure to provide reliability. Such a feature is provided by the OSPF and is desirable also for Proxy ARP. However, because the Proxy ARP setup is static, it cannot provide such a dynamic traffic change.

The problem of traffic balancing among a plurality of routers between two networks is solved by choosing for each data processing device, a router which transfers the network traffic of this data processing device,

based on the network traffic of each data processing device. Next, the routing tables of the data processing devices, the Proxy ARP association between physical and network addresses, and the routes to be used by the OSPF protocol are set by the above selection. This procedure is cyclically repeated to provide dynamic traffic balancing.

The problem of the processing load in traffic balancing is solved by using an equivalent network traffic, which is appropriately converted from the processing load of the router, in the above procedure of selecting the routers.

The problem of reliability of the Proxy ARP is solved by executing the above traffic balancing procedure and by not distributing the network traffic to those routers whose normal operation is prevented by some malfunction. Furthermore, when a router failure or a router recovery is detected, the routes should also be selected.

### **Brief Description of the Drawings**

Figure 1 is an example of a network in which this invention is embodied.

Figure 2a is an explanatory diagram of a routing table setup of a data processing device 3a in Figure 1.

Figure 2b is an explanatory diagram of a routing table setup of a data processing device 3b in Figure 1.

Figure 2c is an explanatory diagram of a routing table setup of a data processing device 3c in Figure 1.

Figure 2d is an explanatory diagram of a routing table setup of a data processing device 3d in Figure 1.

Figure 2e is an explanatory diagram of a routing table setup of a data processing device 3e in Figure 1.

Figure 3 is an explanatory diagram of an ARP cache setup for Proxy ARP.

Figure 4 is a flowchart of the basic operation of the OSPF algorithm.

Figure 5 is a block diagram of a control add-on for the OSPF.

Figure 6 is an explanatory diagram of a global routing table.

Figure 7 is a flowchart of an algorithm for selecting routers.

Figure 8 is a diagram correlating the processing load and the network traffic.

Figure 9 is a flowchart of an algorithm for distributing the traffic.

Figure 10 is a flowchart of a procedure for balancing the traffic when OSPF is not used.

Figure 11 is a block diagram of an embodiment that does not use the OSPF.

Figure 12 is an explanatory diagram of a routing table setup for balancing traffic.

Figure 13 is an explanatory diagram of an ARP cache setup for distributing the traffic.

Figure 14a is an explanatory diagram of a routing table setup of a data processing device 3a in an inter-

face failure.

Figure 14b is an explanatory diagram of a routing table setup of a data processing device 3c in an interface failure.

Figure 14c is an explanatory diagram of a routing table setup of a data processing device 3d in an interface failure.

Figure 14d is an explanatory diagram of a routing table setup of a data processing device 3e in an interface failure.

Figure 15 is an explanatory diagram of an ARP cache setup in an interface failure.

Figure 16a is an explanatory diagram of a routing table setup of a data processing device 3c in multiple router failures.

Figure 16b is an explanatory diagram of a routing table setup of a data processing device 3d in multiple router failures.

Figure 16c is an explanatory diagram of a routing table setup of a data processing device 3e in multiple router failures.

Figure 17 is a flowchart of a procedure executed by an OSPF master device.

Figure 18 is a flowchart of a detailed procedure executed by an OSPF master device.

Figure 19 is a flowchart of a procedure executed by an OSPF slave device.

Figure 20 is a block diagram showing an embodiment when an OSPF is only used.

Figure 21 is a block diagram showing an embodiment when an OSPF is used with the Proxy ARP.

### **Detailed Description of the Preferred Embodiment**

The whole process of traffic balancing includes the following three steps: (1) choosing (selecting) the routers, (2) calculating the distribution of traffic among these routers, and (3) changing the route according to the distribution, in that order. These three steps are described in this order and some examples are given in the explanation of the three steps. All the following explanations concerns a single network hereinafter referred to as "the network in question." Other networks are henceforth referred to as "target networks."

The explicit routing table setup and the OSPF are used to set the routes for communication from the network in question to the target network. The Proxy ARP is used to set the routes for communication from the target network to the network in question. Hence, when the data processing devices connected to the second network 2 are nodes of the parallel processing device 6, the explicit routing table setup and the OSPF are used to set the routes for communication from the parallel processing device 6 to the clients 4a, 4b. The Proxy ARP is used to set the routes for communication in the reverse direction.

## (1) Selection of routers

Selection of routers is made to provide reliability. This is done by selecting normally operating routers from among all possible routers to the target network. This selection process is done in OSPF. In the case of the explicit routing setup and the Proxy ARP, the following procedure is used. First, the network administrator has to prepare a global routing table for the network in question.

Figure 6 shows an example 73 of a global routing table for the network 2, i.e., the network in question. For each target network 731, this table shows a router (ID) 732 to this network. For each router, this table shows a physical address 733 of the interface to the target network, a network address 734 in the network in question, the next hop 735 to the target network after this router, and a priority 736 of the route. This priority is set by the network administrator according to the number of hops up to the target network and according to the network processing capability. Hence, the best route will be selected preferentially. In Figure 6 priority is 1 when the number of hop is 1 and priority is 2 when the number of hops are 2.

The global routing table 73 in Figure 6 is used as an input to the algorithm for selecting routers in Figure 7. At a first step (step 812), the algorithm reads all entries in the global routing table (73) in a memory. At the next step (step 813), the algorithm verifies the status of all routers (732), i.e., if the routers (732) are active or not, and if the indicated network interfaces (733, 734, 735) are active or not. At the next step (step 814), the algorithm removes all router entries (732, 733, 734, 735, 736) from the table in the memory for which the routers or interfaces are not active. At the next step (step 815), the algorithm selects router entries (732, 733, 734, 735, 736) with the lowest value of priority (736) and eliminates other router entries (732, 733, 734, 735, 736) from the table in the memory. At step 815, for each network, the data processing devices that are to operate as routers to the target network 731 appear in the column 732.

## (2) Calculating the distribution of traffic among the routers

Once the routers have been chosen, the routes can be selected. First, the network administrator has to prepare for the network in question a list of the data processing devices involved in the traffic balancing. Figure 8 shows an example of such a list for the network 2. The list of the data processing devices 72 includes the identification code (ID) 721 for discriminating all data processing devices for which traffic balancing is applied to and the factors  $a(k)$  722 for correlating the processing load of this device and the routing load. The factors  $a(k)$  will be described below in more detail.

The algorithm for the calculation (84) of the routes

to provide traffic balancing is shown in Figure 9. Herein, "i" is an index of a data processing device that does not operate as a router, "j" is an index of the target network (731 in Figure 6), and "k" is an index of a data processing device that acts as a router (732 in Figure 6). The traffic (i, j) is the traffic between the data processing device i and the network j. A load (k) is the processing load of a data processing device, and it is equal to or larger than zero. The load (k) increases with the processing load of the data processing device when this data processing device can execute user programs, and when it cannot, is equal to zero. The algorithm also uses the value of route (k) representing the amount of traffic (i, j) assigned to the router k. The algorithm basically distributes the traffic (i, j) among the routes (k) in such a way that the values of the routes (k) among different routers k are balanced.

The algorithm for selecting a route is given in Figure 9.

At the first step (step 842), the algorithm obtains the total traffic (i, j) and the load (k) from other data processing devices.

At the next step (step 843), the algorithm initializes the value of the route (k) to  $a(k) \times \text{load}(k)$  (route (k) =  $a(k) \times \text{load}(k)$ ), where  $a(k)$  (722 in Figure 8) is taken from the data processing device list (72). Hence, the processing load of the router is incorporated into the calculation as a networking load. The "constant  $a(k)$ " provides the tradeoff between the network traffic balancing and the processing load balancing among the routers. Lower overall values of  $a(k)$  give priority to the network traffic balancing and higher overall values give priority to the processing load balancing. The relative values of  $a(k)$  represent the performance of a data processing device, wherein lower values of  $a(k)$  represent higher performance of the data processing devices (i.e., a lighter processing load is produced by routing the same amount of network traffic).

At the next step (step 844), the algorithm calculates  $\text{route}(k) = \text{route}(k) + \text{traffic}(i, j)$  for all k in which all the traffic between the networks can flow only through this router k. All the values of traffic (i, j) used at this stage (step 844) are set to zero.

At the next step (step 845), the algorithm obtains i and j so that the traffic (i, j) that is selected is the highest value of traffic (i, j) which is left.

At the next step (step 846), the algorithm finds k that has the smallest value of route (k) among the routers for the target network j. This selection means that the network traffic between the network j and the data processing device i is to pass through the router k.

At the next step (step 847), the traffic is assigned to the router k found in step 846, i.e.,  $\text{route}(k) = \text{route}(k) + \text{traffic}(i, j)$  for the i and j selected in step 845, whose traffic (i, j) is then set to zero.

At the final step (step 848), if there is any non-zero value of the traffic (i, j), the algorithm returns to the step (step 845).



### (3) Changing the route according to the distribution

Once the correspondence between the data processing devices and the routers is obtained, the corresponding routes are set. The process of setting the routes varies depending on whether the OSPF is used or not.

When the OSPF is not used, the whole process of traffic balancing (i.e., three steps of (1) selecting routers, (2) calculating the distribution of traffic among these routers, and (3) changing the route according to the distribution) is shown in Figure 10. After selecting the router (step 81) and traffic distribution (step 84), the algorithm executes an explicit routing table setup (step 852) and Proxy ARP setup (step 853). The process is periodically repeated to provide dynamic load balancing.

The setup for routes for the explicit route setup (852) is done as follows. The route for the target network (731 in Figure 6), which is set in the data processing device to be used as a router, is taken from the column of the next hop (735 in Figure 6) in the global routing table 73 in Figure 6. In this column 735, the next hop corresponds to the above mentioned router. In this case, the flag 323 is set to "interface." The routes for other data processing devices and for the data processing devices that can work as routers but whose network interfaces for the target network is inactive, are taken from the column of network 734. In this column 734, the network corresponds to the router assigned to the data processing device at step 847. In this case, the flag is set to "gateway." The routes are not set for the inactive data processing devices or for the network in question whose network interfaces are not active.

The setup (853) for routes for the Proxy ARP is as follows. The network address of the interface of a data processing device is related to the physical address of the interface of the router selected for this data processing device. The physical address is taken from the physical address column (733) of the global routing table 73. If the data processing device with the Proxy ARP linkage is not operational, the Proxy ARP linkage should be set in another data processing device connected to the target network (731). The Proxy ARP setup is only performed when the target network (731) is adjacent to the network in question. When the physical address entry (733) of the global routing table 73 is blank, the interconnection using the Proxy ARP is not possible. Hence the Proxy ARP setup is not performed. When the ARP linkage changes, the data processing device having this linkage can broadcast an ARP reply having a new ARP linkage. In this case, other data processing devices in the target network (731) receive this reply. The other data processing devices compare the linkage contained in the reply with the content of their own ARP cache. Then, if any change is found, the other data processing devices change the linkage in the ARP cache to the linkage contained in the reply, and the other data process-

ing devices transmit the traffic to the new physical address.

Therefore, if an ARP reply is broadcast when the linkage changes, the new linkage is reflected on the data processing devices, realizing the dynamic traffic balance of the network traffic among routers. When the ARP reply is not broadcast, the new linkage is reflected on the data processing devices that send ARP queries to the target network (731) later.

In any of the data processing device (3, 4) that execute user programs, the algorithm of Figure 10 can be executed by running a control add-on which uses hardware and/or software.

Figure 11 shows a preferred embodiment of this invention. A control add-on (92) is installed in the same data processing device in which an ARP cache (71) is provided, to facilitate the setup of the ARP cache. Alternatively, the control add-on (92) can be located in any of the data processing devices (3a, 3b, 3c, 3d, 3e, 4a and 4b). In the case of a parallel processing device (6), a preferred embodiment for the control add-on (92) is in one of the nodes (3a, 3b, 3c, 3d and 3e) of the parallel processing device. This is because special features of the operating system of the parallel processing device (6) can be used to gather data from other nodes and execute the setup in other nodes.

The values of the traffic (i, j) and the load (k) can be taken from a data processing device in which the control add-on (92) is located by a system call. The network traffic and the traffic (i, j) of other data processing devices can be obtained, for example, by the Simple Network Management Protocol (SNMP, see RFC 1157). The processing load of other data processing devices can be obtained by, for instance, the rwho protocol or the Internet Systat Service. The routes and the Proxy ARP linkage can be changed by a system call in a data processing device in which the control add-on (92) is running and by the SNMP protocol in other data processing devices.

An example of the algorithm of Figure 10 applied to the examples of Figure 1 and Figure 6 is as follows. It is assumed that the network traffic associated with the network 1 of the data processing devices 3c, 3d, 3e is 1 M byte/sec for each device and the traffic (i, j) is the network traffic measured at M byte/sec. It is also assumed that the data processing devices 3a, 3b, 3c and their interfaces operate without malfunction, that the data processing devices have the ability to execute user process, that the data processing devices 3a and 3c are not executing user processes and the data processing device 3b is executing one user process, and that the load (k) in a data processing device k is measured by the number of user processes in the data processing device k.

The algorithm (81) for selecting routers reads the global routing table 73 at the first step (step 812) and, at the next step (step 813), verifies that the data processing devices 3a, 3b, 3c and their interfaces operate with-

out malfunction. At the next step (step 814), the entries of these data processing devices (3a, 3b, 3c) are not eliminated from the table as they (3a, 3b, 3c) are active. At the step 815, the data processing devices 3a and 3b are selected because they have the smallest value of priority. Hence, the data processing devices 3a and 3b are selected as routers between the network 1 and the network 2.

Next, at the first step (step 842), the algorithm (84) for balancing the traffic obtains the processing loads of the data processing devices 3a and 3b and the network traffic between the network 1 and the data processing devices 3c, 3d, 3e. At step 843, the route (3a) is assigned 0 and the route (3b) is assigned 1.5. At step 844, no traffic is assigned to the network 1.

Next, the algorithm repeats the steps 845, 846 and 847 three times and assigns the traffic to the routers. The data processing devices 3c, 3d, 3e are assumed to have been selected in that order at step 845. Thus, in the first repetition, the data processing device 3c is assigned to the router 3a and the route (3a) is updated to 1 (= (traffic = 1) + (route (3a) = 0)). In the second repetition, the data processing device 3d also is assigned to the router 3a and the route (3a) is updated to 2 (= (traffic = 1) + (route(3a) = 1)). In the third repetition, the data processing device 3e is assigned to the router 3b and the route (3b) is updated to 2.5 (= (traffic = 1) + (route(3b) = 1.5)). Therefore, the network traffic of the data processing devices 3c, 3d, 3e is distributed between the data processing devices 3a, 3b, because the data processing device 3b has a heavier processing load, greater traffic is given to the data processing device 3a. The route to and from the data processing device 3e is changed to the router 3b in this way.

The routing table of the data processing device 3e after the explicit routing table setup (step 852) is executed and is shown in Figure 12. The ARP cache after executing the Proxy ARP setup (step 853) is shown in Figure 13. The entry encircled by a dotted line is a changed one.

Two examples for explaining reliability will be described below.

The first example is one in which the network interface of the data processing device 3a (31a) in the network 2 fails, but data processing device 3a continues operating. In the algorithm (81 in Figure 7) for selecting routers, the failure of the data processing device 3a is detected at step 813 and the entry of this failed router is eliminated at step 814. The entry of the data processing device 3b is selected at step 815 because it has the lowest priority value. Thus, only the data processing device 3b is used as the router. Next, in the traffic distribution algorithm (84 in Figure 9), the traffic of the data processing devices 3a, 3c, 3d, 3e are assigned to the router 3b. The routing tables (32a, 32c, 32d, 32e) of the data processing devices 3a, 3c, 3d, 3e after the explicit routing table setup (step 852) is performed are shown in Figs. 14a - 14d. The ARP cache (71) after the Proxy

ARP setup (step 853) is shown in Figure 15. The entry encircled by a dotted line is a changed one. In this example, the communication among the data processing devices (3a, 3b, 3c, 3d and 3e) in the second network is not affected by failures of the network interfaces. Similarly, in the case of the parallel processing device (6), the communication between the parallel processing device and the clients (4a, 4b) continues even after the route has been changed.

Another example concerning reliability is a case that both data processing devices 3a and 3b totally fail. In this case, the algorithm (81) for selecting routers detects the failures of the data processing devices 3a and 3b at step 813. And the entries of these routers are eliminated at step 814. Thus, only the entry of the data processing device 3c remains and is selected as the router. Next, in the traffic distribution algorithm (84), the traffic of the data processing devices 3d and 3e is assigned to the router 3c. The routing tables (32c, 32d, 32e) of the data processing devices 3c, 3d, 3e after the explicit routing table setup (852) are shown in Figure 16a - 16c. All of the entries in the ARP cache (71) have been eliminated because the physical address (733) corresponding to the router 3c are not set. In this example, the communication between the data processing devices (3c, 3d, 3e) that are still active in the network in question and the data processing devices (4a, 4b) in the target network are interrupted. This is because only the route from the network 2 to the network 1 is set and the route from the network 1 to the network 2 is not set. However, if the same system is also applied to the network 1, the traffic from the network 1 to the network 2 can flow through the data processing device 4b, and therefore the communication between the networks 1 and 2 is not interrupted. The same also holds for the parallel processing device (6).

When the OSPF is used in the network in question, two major differences arise. First, the router is selected by the OSPF processing, and therefore the OSPF is used in place of the router selection algorithm (81). Second, while the router exchange routes by the OSPF operate in such a way as to perform distribution, the router selection algorithm (81) and the traffic distribution algorithm (84) should be executed in a centralized fashion, to avoid inconsistencies among the routes selected by various data processing devices, and to prevent the processing load that may be produced if all of the data processing devices query the processing load and networking traffic of one another.

Thus, the OSPF (Open System Path First) is extended as follows. One of the data processing devices that exchanges data by the OSPF (hereinafter referred to as "master") is responsible for centralized tasks such as that of selecting the routes and broadcasting these routes to other data processing devices that execute the OSPF. Other data processing devices (hereinafter referred to as "slaves") set these routes accordingly. A detailed description of the procedure per-

formed by the master and the slaves is as follows.

The basic algorithm of the master is shown in Figure 17. Step 822 to step 826 are shown in Figure 4. The OSPF basic algorithm is extended by an additional step (step 831) and executed periodically. This step 831 is detailed in Figure 18. In Figure 18, first, a traffic balance calculation algorithm (step 84) is executed. The step 84 is explained in Figure 9. Next, the routes in the master data processing device are set (step 8311). Then, the selected routes are broadcast to the slave data processing devices (step 8312). Next, the routes are set in the data processing devices that do not exchange data by OSPF (step 8313). Finally, when Proxy ARP is used along with OSPF, the Proxy ARP correspondence is set (step 853). The basic algorithm for the slave is shown in Figure 19. Step 822 to step 826 are shown in Figure 4. Here the basic OSPF algorithm is extended by two additional steps (step 832 and step 833) and also periodically executed. In the first step of the two additional steps, the slave data processing device receives the selected routes broadcast by the master and at the next step (step 833), sets its own routes accordingly. It should be noted that the special exchange of data is performed independently of the OSPF protocol and thus there is no need to change the OSPF protocol in the master and slave, when the router to be used changes as a result of the status change in the router, the master and slave should select a route arbitrarily from those available until a traffic balance calculation result is obtained and broadcast. Otherwise, the previous route, if still valid, should be retained. This route then changes depending on the result of the traffic distribution algorithm (84).

Two preferred embodiments of control add-on for the master 93 and slave 94 are shown in Figure 20 and 21. In the embodiment of Figure 20, the OSPF is used in all networks (1, 2, 5). In Figure 20, the control add-on of the master 93 is located in the data processing device 3a. Alternatively, it may be located any of the data processing devices (3a, 3b, 3c, 3d, 3e, 4a, 4b) that exchange data by the OSPF. In the embodiment of Figure 21, the OSPF is used in one or more networks (in the figure, network 2), and Proxy ARP is used to set routes for the traffic coming from one or more networks (in the figure, network 1) adjacent to the network that uses the OSPF. The control add-on of the master 93 is located in the same data processing device in which the ARP cache (71) is located, in order to facilitate the setup of the ARP cache (71). Alternatively, the master may be located in any network using the OSPF, and the ARP cache (71) may be located in any data processing device (3a, 3b, 4a, 4b) in the target network.

The three examples for the explicit routing table setup and the Proxy ARP operate in a similar way to the case of the OSPF, in that the same routes may be selected by the algorithm (84) similarly to the algorithm (81) which selects routers through the shortest path selection (825), and in that the routes (32a, 32b, 32c,

32d, 32e) set in the routing table of the data processing devices in the network of interest and the linkage set in the ARP cache (71) may be the same. The same also applies to the parallel processing device (6).

This invention has five advantages.

First, good networking performance is provided by dynamically balancing the network traffic among routers between two networks.

Second, the invention takes the routing traffic from a data processing device having a higher processing load, thereby utilizing the data processing devices more effectively and increasing the job processing capability.

Third, in the case of failure of one or more routers, the invention transparently changes the routing of the Proxy ARP to improve the networking reliability.

Fourth, this invention has compatibility with the existing network protocols and thus these protocols need not be changed. In the case of the explicit route setup and the Proxy ARP, the operating system of the data processing devices does not need any change. When there are two or more equivalent paths when the invention is applied to the OSPF protocol, only the control add-on for executing the OSPF protocol needs to be changed.

Fifth, in the case of parallel processing devices, because not all nodes (data processing devices) require external network interfaces to communicate with other networks efficiently, the cost of parallel processing devices can be reduced.

While preferred embodiments have been set forth with specific details, further embodiments, modifications and variations are contemplated according to the broader aspects of the present invention, all as determined by the spirit and scope of the following claims.

## Claims

1. In a first network and a second network, each interconnecting a plurality of data processing devices and interconnected by a subset of the data processing devices, a networking method to select routes to balance network traffic among the data processing devices that interconnect the first and second networks, comprising:

a first step of obtaining an amount of network traffic flow between the data processing devices of the first and second networks; and a second step of selecting for each data processing device routes according to the amount of network traffic flow, in a way that network traffic flow between the data processing devices interconnecting the first and second networks is distributed among the data processing devices of the second network, so that the network traffic flow among the data processing devices interconnecting the first and second networks is balanced.

2. A networking method according to claim 1, further comprising:
- a third step of obtaining a processing load of the data processing devices interconnecting the first and second networks,
- wherein the second step uses the processing load of the data processing devices, and the routes are selected according to the network traffic and the processing load in such a way as to balance the network traffic and the processing load.
3. A networking method according to claim 2, wherein the second step uses the processing load of the data processing devices which are converted to network traffic.
4. A networking method according to claim 1, wherein an Address Resolution Protocol (ARP) and a Proxy ARP method are used in the first network and dynamically balances the network traffic from the first network to the second network among the data processing devices interconnecting the first and second networks, the networking method further comprising:
- a fourth step of correlating, by the Proxy ARP, a network addresses of the data processing devices of the second network with a physical addresses of the interfaces connected to the first network of the data processing devices interconnecting the first and second networks, the data processing devices being selected as the routers for the data processing devices of the second network, wherein the first, second, third and fourth steps are repeated.
5. A networking method according to claim 2, wherein an Address Resolution Protocol (ARP) and a Proxy ARP method are used in the first network and which dynamically balances the network traffic from the first network to the second network among the data processing devices interconnecting the first and second networks, the networking method further comprising:
- a fourth step of correlating, by the Proxy ARP, a network addresses of the data processing devices of the second network with a physical addresses of the interfaces connected to the first network of the data processing devices interconnecting the first and second networks, the data processing devices being selected as the routers for the data processing devices of the second network, wherein the first, second, third and fourth steps are repeated.
6. A networking method according to claim 4, wherein at the fourth step, the network traffic of the data processing devices from the first network to the second network is distributed among only data processing devices interconnecting the first and second networks which have no malfunction that prevents the transfer of the network traffic between the first and second networks.
7. A networking method according to claim 5, wherein at the fourth step, the network traffic of the data processing devices from the first network to the second network is distributed among only data processing devices interconnecting the first and second networks which have no malfunction that prevents the transfer of the network traffic between the first and second networks.
8. A networking method according to claim 1, further comprising:
- a step of changing routes to the first network in a routing table of each of the data processing devices of the second network to data processing devices interconnecting the first and second networks that are selected to act as the routers for the data processing devices of the second network,
- wherein the network traffic from the second network to the first network is balanced dynamically among the data processing device interconnecting the first and second networks.
9. A networking method according to claim 2, further comprising:
- a step of changing routes to the first network in a routing table of each of the data processing devices of the second network to data processing devices interconnecting the first and second networks that are selected to act as the routers for the data processing devices of the second network,
- wherein the network traffic from the second network to the first network is balanced dynamically among the data processing device interconnecting the first and second networks.
10. A networking method according to claim 1, wherein an Open Shortest Path First (OSPF) protocol is used in the second network and dynamically balances the network traffic from the second to the first network among the data processing devices interconnecting the first and second networks, the networking method further comprising:
- a step of deciding routes of other data processing devices by one data processing device,

according to the amount of network traffic flow,  
and  
a step of broadcasting the decided routes to  
the other data processing devices by using the  
OSPF protocol.

5

11. A networking method according to claim 1, wherein  
an Open Shortest Path First (OSPF) protocol is  
used in the second network and dynamically bal-  
ances the network traffic from the second to the first  
network among the data processing devices inter-  
connecting the first and second networks, the net-  
working method further comprising:

10

a step of receiving routes broadcast by other  
data processing devices; and  
a step of setting routes on the data processing  
devices which receive the broadcast routes,  
according to the received routes.

15

20

12. A networking method according to claim 2, wherein  
an Open Shortest Path First (OSPF) protocol is  
used in the second network and dynamically bal-  
ances the traffic to the network traffic from the sec-  
ond to the first network among the data processing  
devices interconnecting the first and second net-  
works, the networking method further comprising:

25

a step of deciding routes of other data process-  
ing devices by one data processing device,  
according to the amount of network traffic flow,  
and  
a step of broadcasting the decided routes to  
the other data processing devices by using the  
OSPF protocol.

30

35

13. A networking method according to claim 2, wherein  
an Open Shortest Path First (OSPF) protocol is  
used in the second network and dynamically bal-  
ances the network traffic from the second to the first  
network among the data processing devices inter-  
connecting the first and second networks, the net-  
working method further comprising:

40

a step of receiving routes broadcast by other  
data processing devices; and  
a step of setting routes on the data processing  
devices which receive the broadcast routes,  
according to the received routes.

45

50

14. A networking method according to claim 1, wherein  
the data processing devices connected to the sec-  
ond network are nodes of a parallel processing  
device, and the first network connects the parallel  
processing device to other data processing  
devices.

55

15. A networking method according to claim 4, wherein

the data processing devices connected to the sec-  
ond network are nodes of a parallel processing  
device, and the first network connects the parallel  
processing device to other data processing  
devices.

16. A networking method according to claim 8, wherein  
the data processing devices connected to the sec-  
ond network are nodes of a parallel processing  
device, and the first network connects the parallel  
processing device to other data processing  
devices.

17. A networking method according to claim 10,  
wherein the data processing devices connected to  
the second network are nodes of a parallel process-  
ing device, and the first network connects the paral-  
lel processing device to other data processing  
devices.

18. A networking method according to claim 11,  
wherein the data processing devices connected to  
the second network are nodes of a parallel process-  
ing device, and the first network connects the paral-  
lel processing device to other data processing  
devices.

19. In a first network and a second network, each inter-  
connecting a plurality of data processing devices,  
both networks being interconnected by routers  
which are a subset of the data processing devices,  
a networking method on one of the data processing  
devices comprising:

a step of gathering an amount of a network traf-  
fic flow among the data processing devices of  
the first network and the second network;  
a step of computing routes to provide network  
traffic balancing among the data processing  
devices which interconnect the first and the  
second networks, according to the amount of  
the network traffic flow;  
a step of setting routes on one data processing  
device based on the computed routes; and  
a step of sending the computed routes from the  
one data processing device to other data  
processing devices so that the other data  
processing devices set routes according to the  
sent routes.

20. In a first network and a second network, each inter-  
connecting a plurality of data processing devices,  
both networks being interconnected by routers  
which are a subset of the data processing devices,  
a networking method on one of the data processing  
devices comprising:

a first step of gathering routes from other rout-

ers;

a second step of setting routes on one data processing device based on the gathered routes;

a third step of gathering an amount of a network traffic flow on the data processing devices that interconnect the first network and the second network;

a fourth step of computing routes to provide network traffic balancing among data processing devices which interconnect the first and the second networks, according to the amount of the network traffic flow;

a fifth step of setting routes on one data processing device, based on the computed routes; and

a sixth step of sending the computed routes to other data processing devices, so that the other data processing devices set routes according to the sent routes.

21. A networking method according to claim 20, wherein said first and second steps are part of the OSPF protocol.

5

10

15

20

25

30

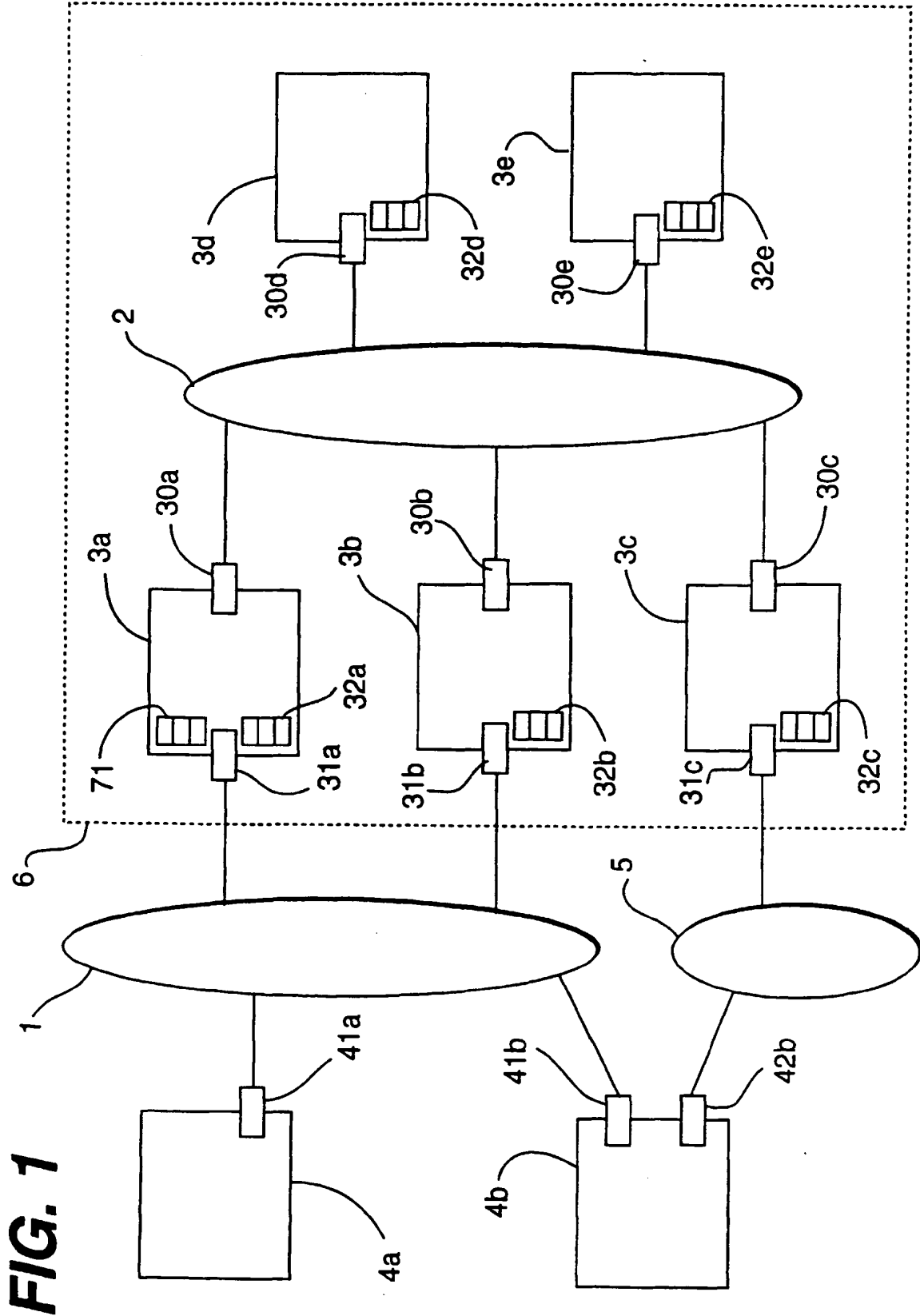
35

40

45

50

55



**FIG. 1**

32a	321a	TARGET NETWORK	322a	NEXT HOP	323a	FLAGS
		...		...		...
		NETWORK 1		31a-N		INTERFACE
		...		...		...

**FIG. 2a**

32b	321b	TARGET NETWORK	322b	NEXT HOP	323b	FLAGS
		...		...		...
		NETWORK 1		31b-N		INTERFACE
		...		...		...

**FIG. 2b**

32c	321c	TARGET NETWORK	322c	NEXT HOP	323c	FLAGS
		...		...		...
		NETWORK 1		30a-N		GATEWAY
		...		...		...

**FIG. 2c**

32d	321d	TARGET NETWORK	322d	NEXT HOP	323d	FLAGS
		...		...		...
		NETWORK 1		30a-N		GATEWAY
		...		...		...

**FIG. 2d**

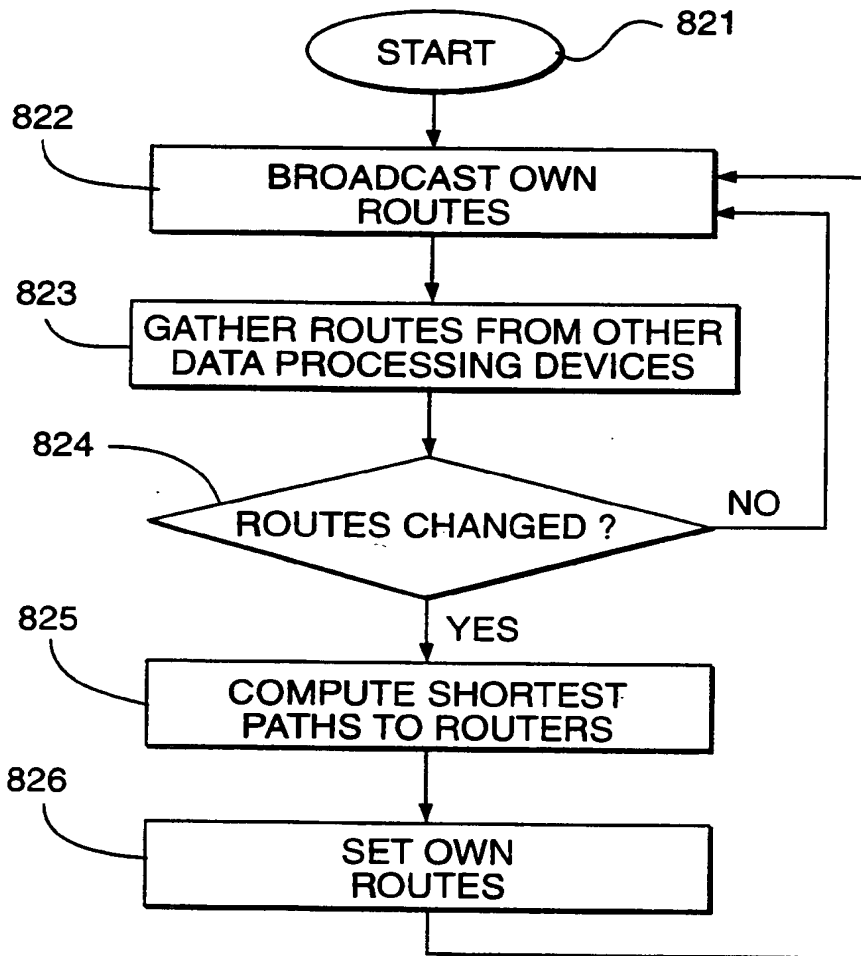
32e	321e	TARGET NETWORK	322e	NEXT HOP	323e	FLAGS
		...		...		...
		NETWORK 1		30a-N		GATEWAY
		...		...		...

**FIG. 2e**

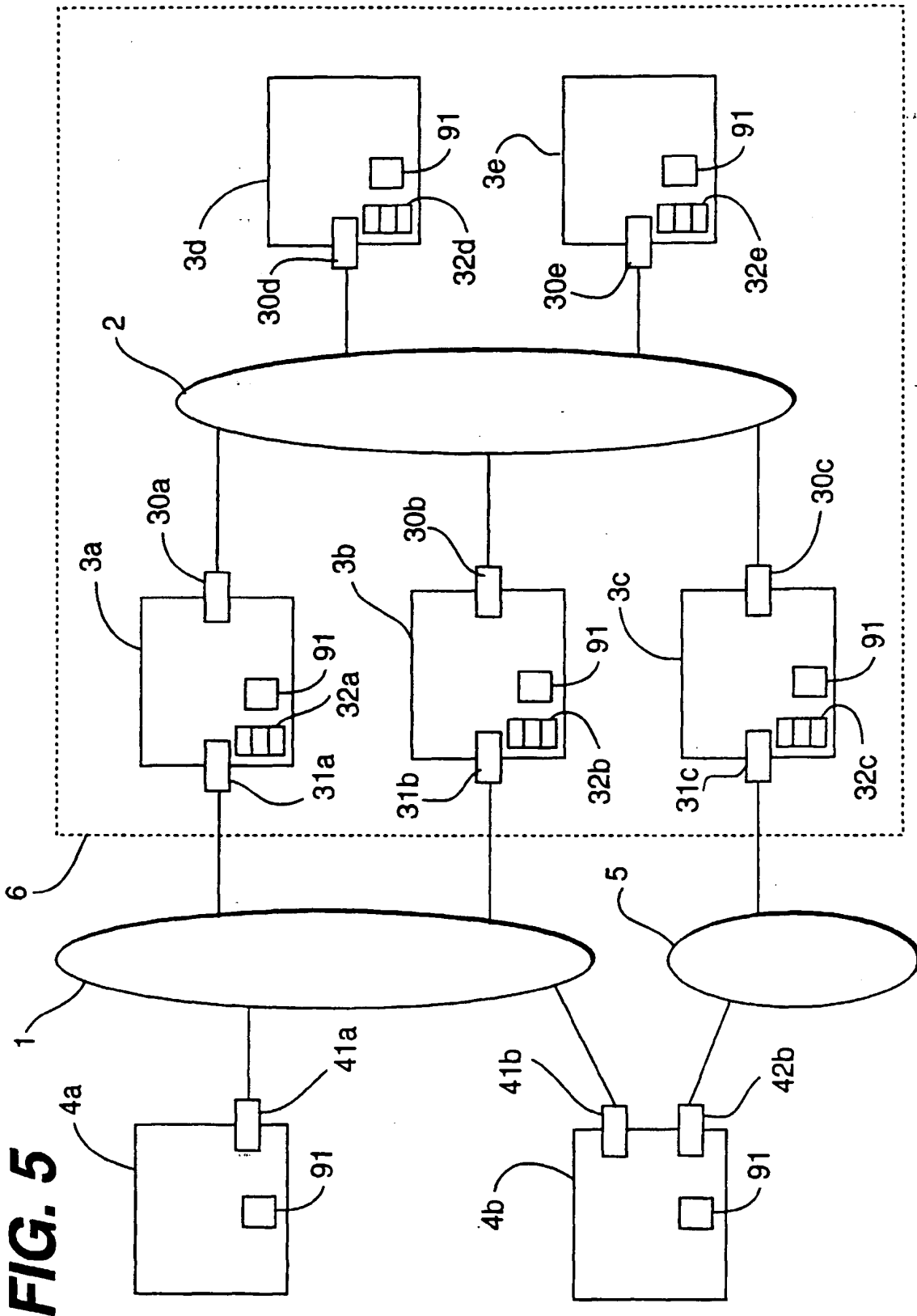


NETWORK ADDRESS	PHYSICAL ADDRESS	FLAGS
...	...	...
30c-N	31a-P	PUBLIC
30d-N	31a-P	PUBLIC
30e-N	31a-P	PUBLIC
...	...	...

**FIG. 3**



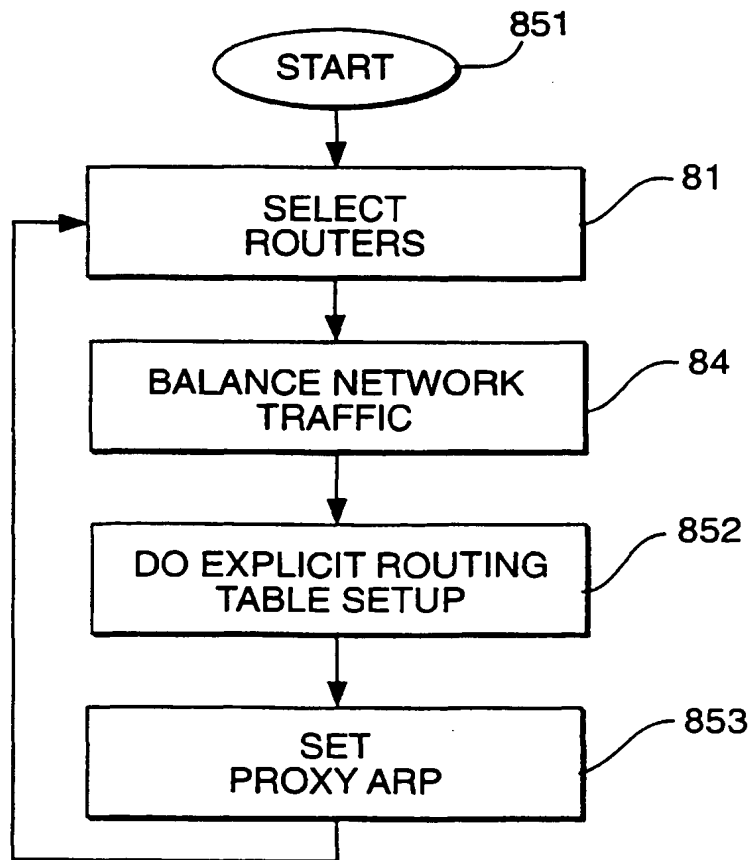
**FIG. 4**



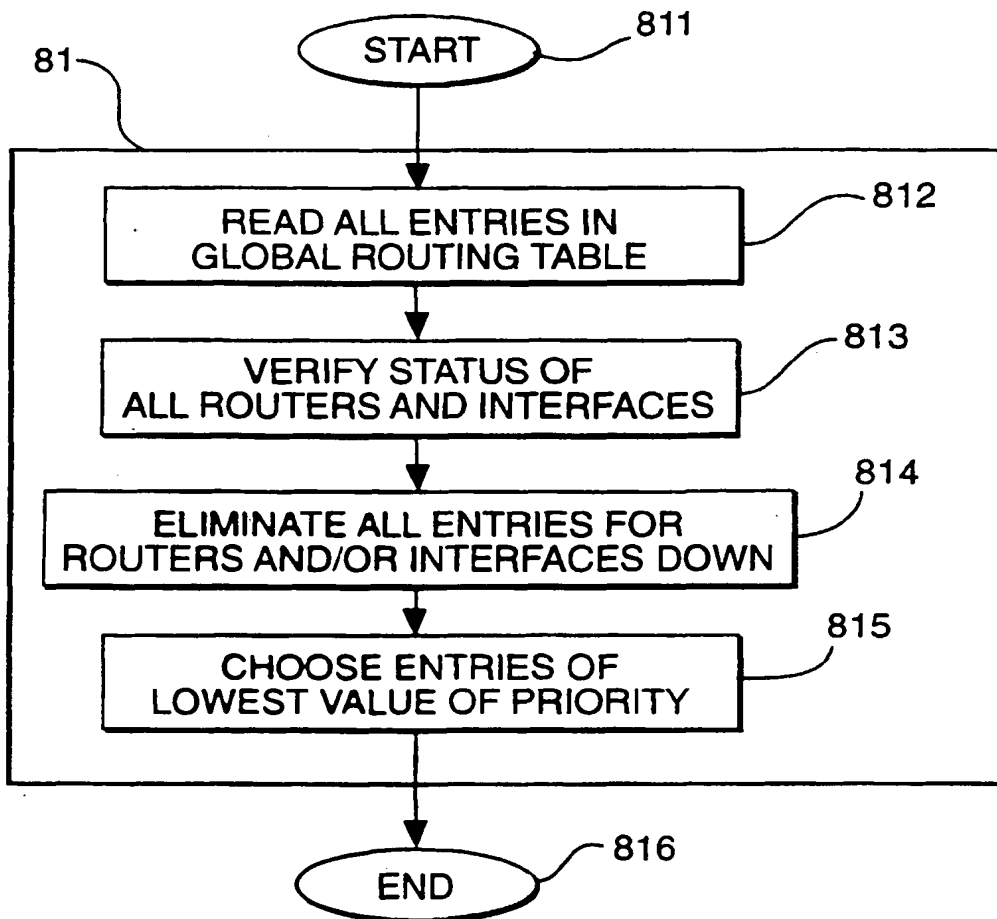
**FIG. 5**

TARGET	ROUTER			NEXT HOP	PRIORITY
	ID	PHYSICAL	NETWORK		
...	...	...	...	...	...
NETWORK 1	3a	31a-P	30a-N	31a	1
	3b	31b-P	30b-N	31b	1
	3c	—	30c-N	42b	2
...	...	...	...	...	...

**FIG. 6**



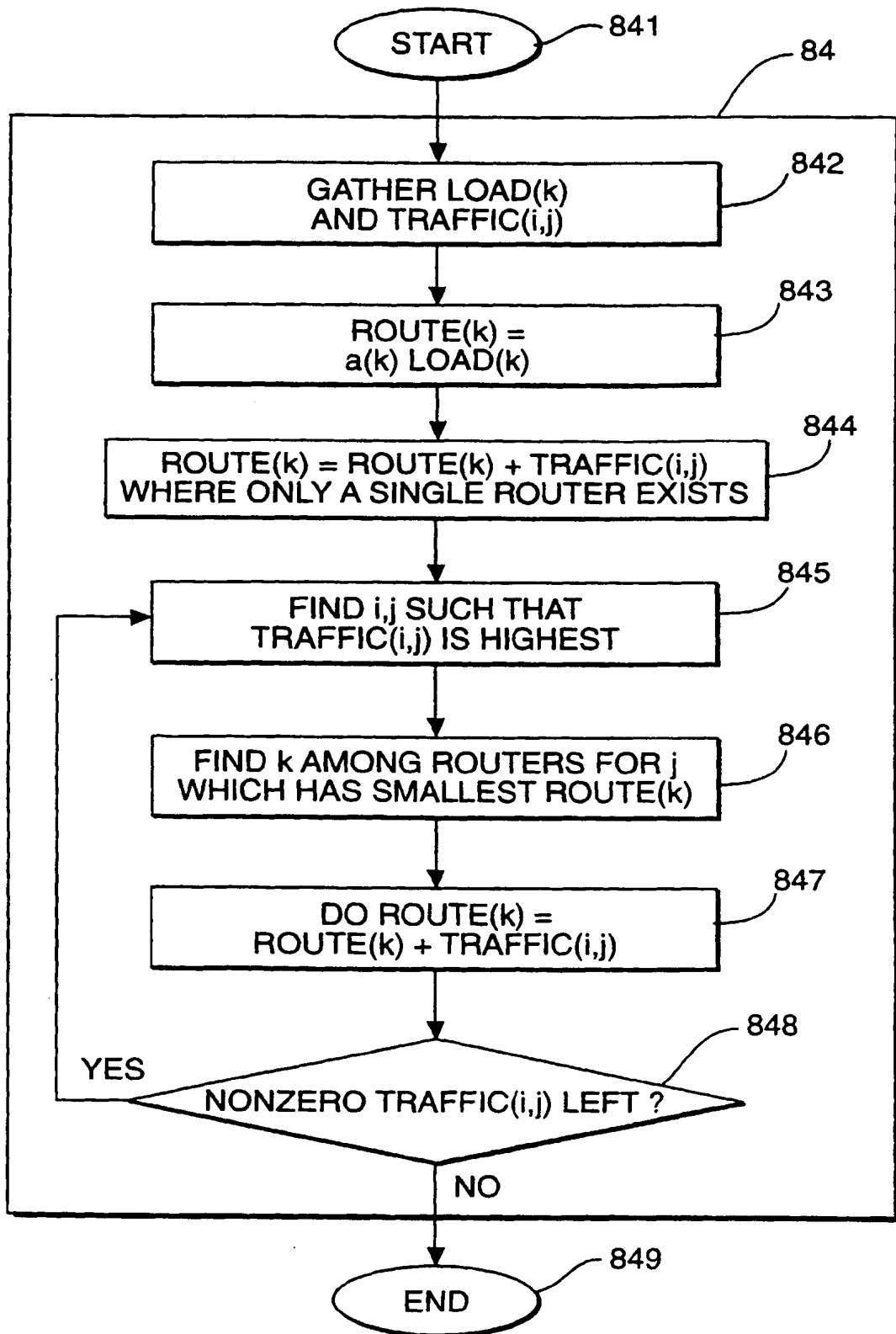
**FIG. 10**



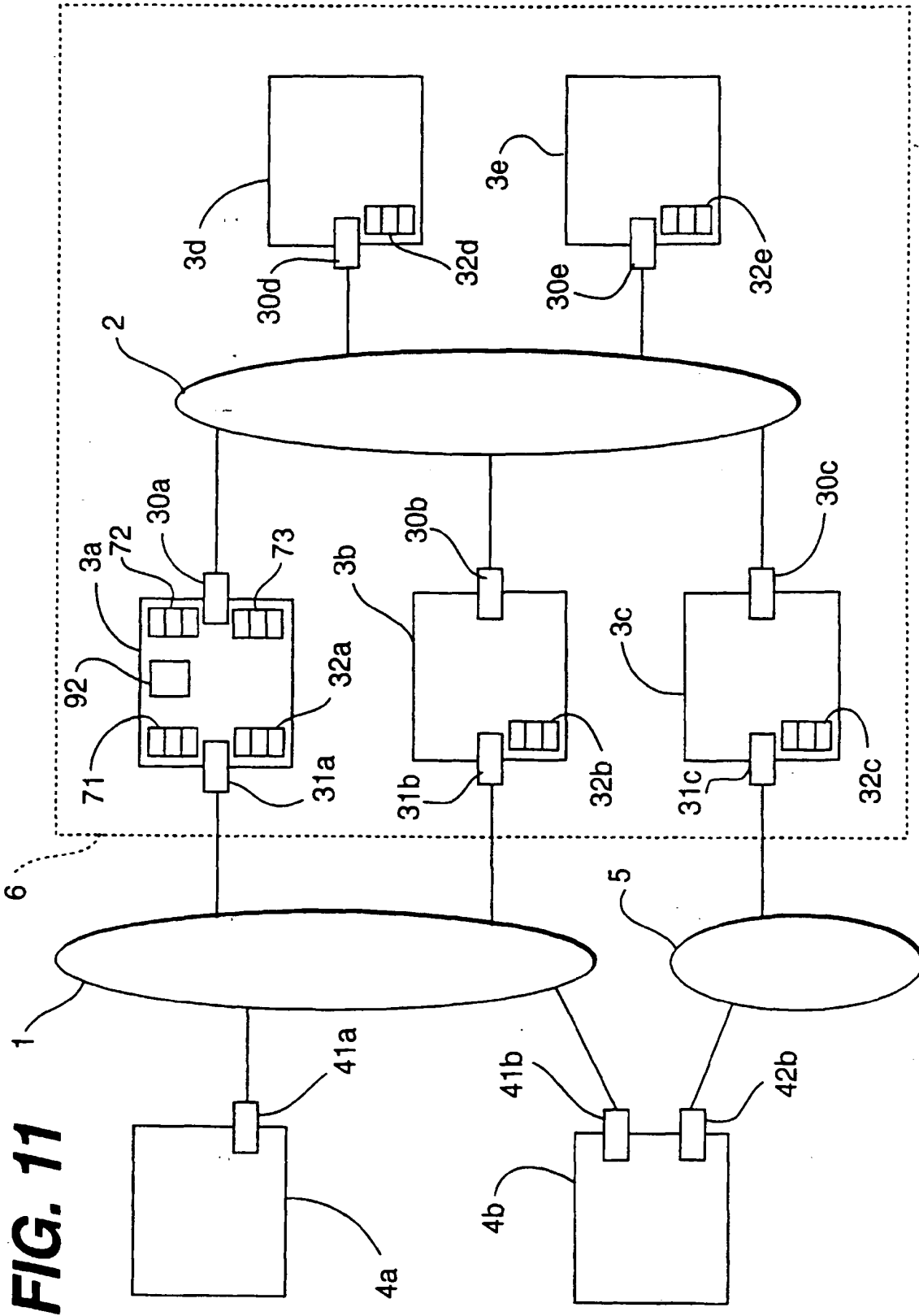
**FIG. 7**

721	722
ID	a(k)
3a	1.5
3b	1.5
3c	1.5
3d	0
3e	0

**FIG. 8**



**FIG. 9**



**FIG. 11**

321e TARGET NETWORK	322e NEXT HOP	323e FLAGS
...	...	...
NETWORK 1	30b-N	GATEWAY
...	...	...

32e

**FIG. 12**

711 NETWORK ADDRESS	712 PHYSICAL ADDRESS	713 FLAGS
...	...	...
30c-N	31a-P	PUBLIC
30d-N	31a-P	PUBLIC
30e-N	31b-P	PUBLIC
...	...	...

71

**FIG. 13**

TARGET NETWORK	NEXT HOP	FLAGS
...	...	...
NETWORK 1	30b-N	GATEWAY
...	...	...

**FIG. 14a**

TARGET NETWORK	NEXT HOP	FLAGS
...	...	...
NETWORK 1	30b-N	GATEWAY
...	...	...

**FIG. 14b**

TARGET NETWORK	NEXT HOP	FLAGS
...	...	...
NETWORK 1	30b-N	GATEWAY
...	...	...

**FIG. 14c**

TARGET NETWORK	NEXT HOP	FLAGS
...	...	...
NETWORK 1	30b-N	GATEWAY
...	...	...

**FIG. 14d**



71	711	712	713
	NETWORK ADDRESS	PHYSICAL ADDRESS	FLAGS
	...	...	...
	30a-N	... 31b-P	PUBLIC
	30c-N	31b-P	PUBLIC
	30d-N	31b-P	PUBLIC
	30e-N	31b-P	PUBLIC
	...	...	...

**FIG. 15**

32c	321c	322c	323c
	TARGET NETWORK	NEXT HOP	FLAGS
	...	...	...
	NETWORK 1	31c-N	INTERFACE
	...	...	...

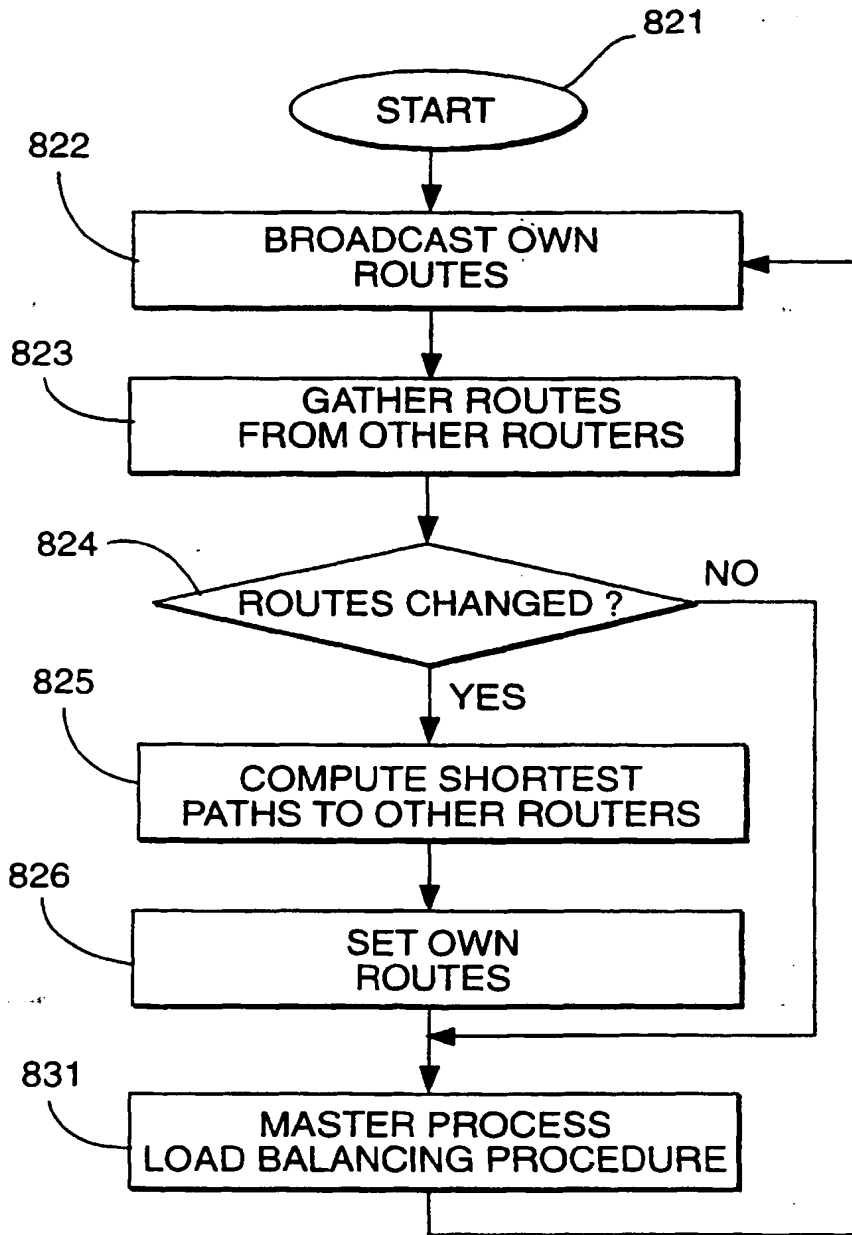
**FIG. 16a**

32d	321d	322d	323d
	TARGET NETWORK	NEXT HOP	FLAGS
	...	...	...
	NETWORK 1	30c-N	GATEWAY
	...	...	...

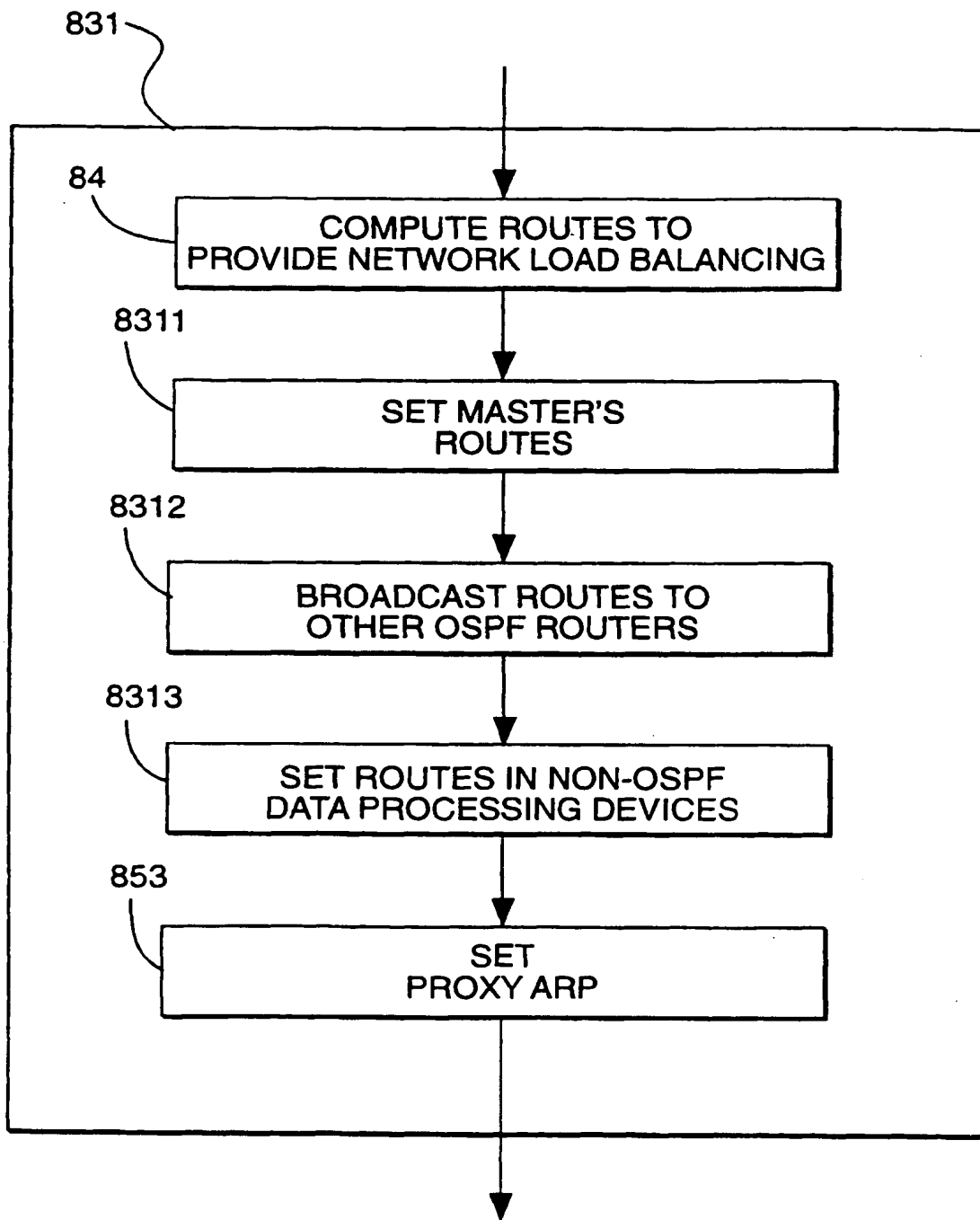
**FIG. 16b**

32e	321e	322e	323e
	TARGET NETWORK	NEXT HOP	FLAGS
	...	...	...
	NETWORK 1	30c-N	GATEWAY
	...	...	...

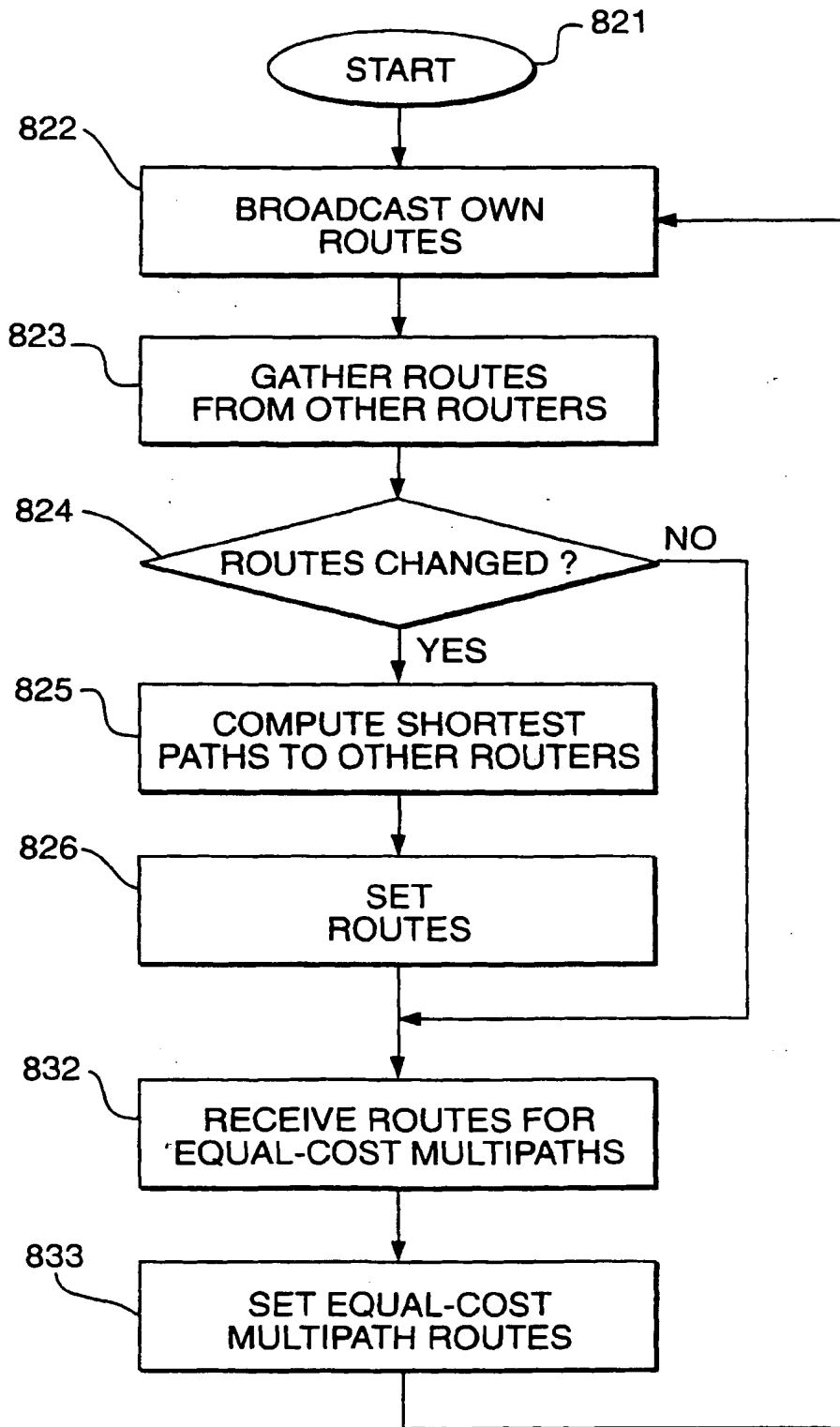
**FIG. 16c**



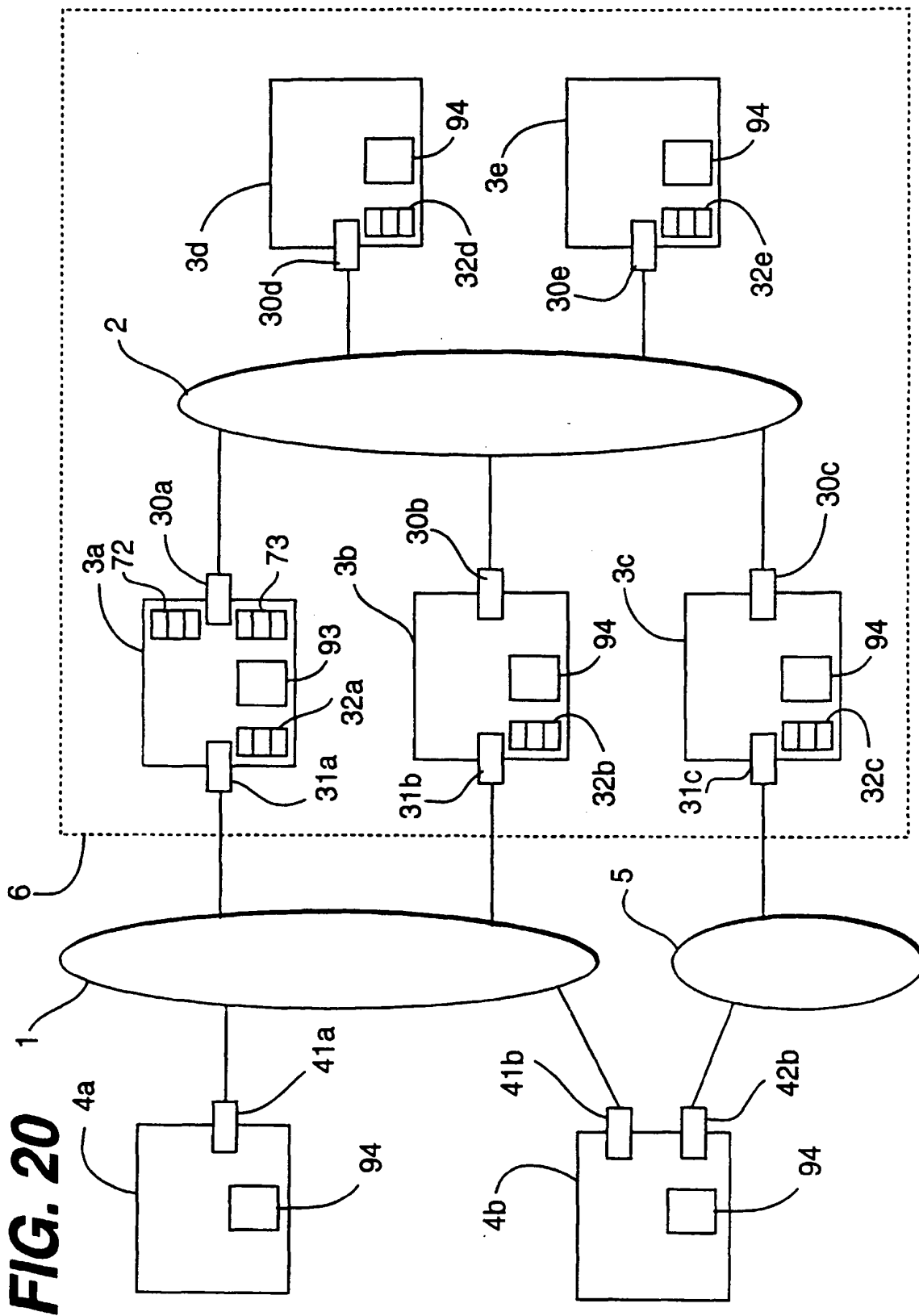
**FIG. 17**



**FIG. 18**

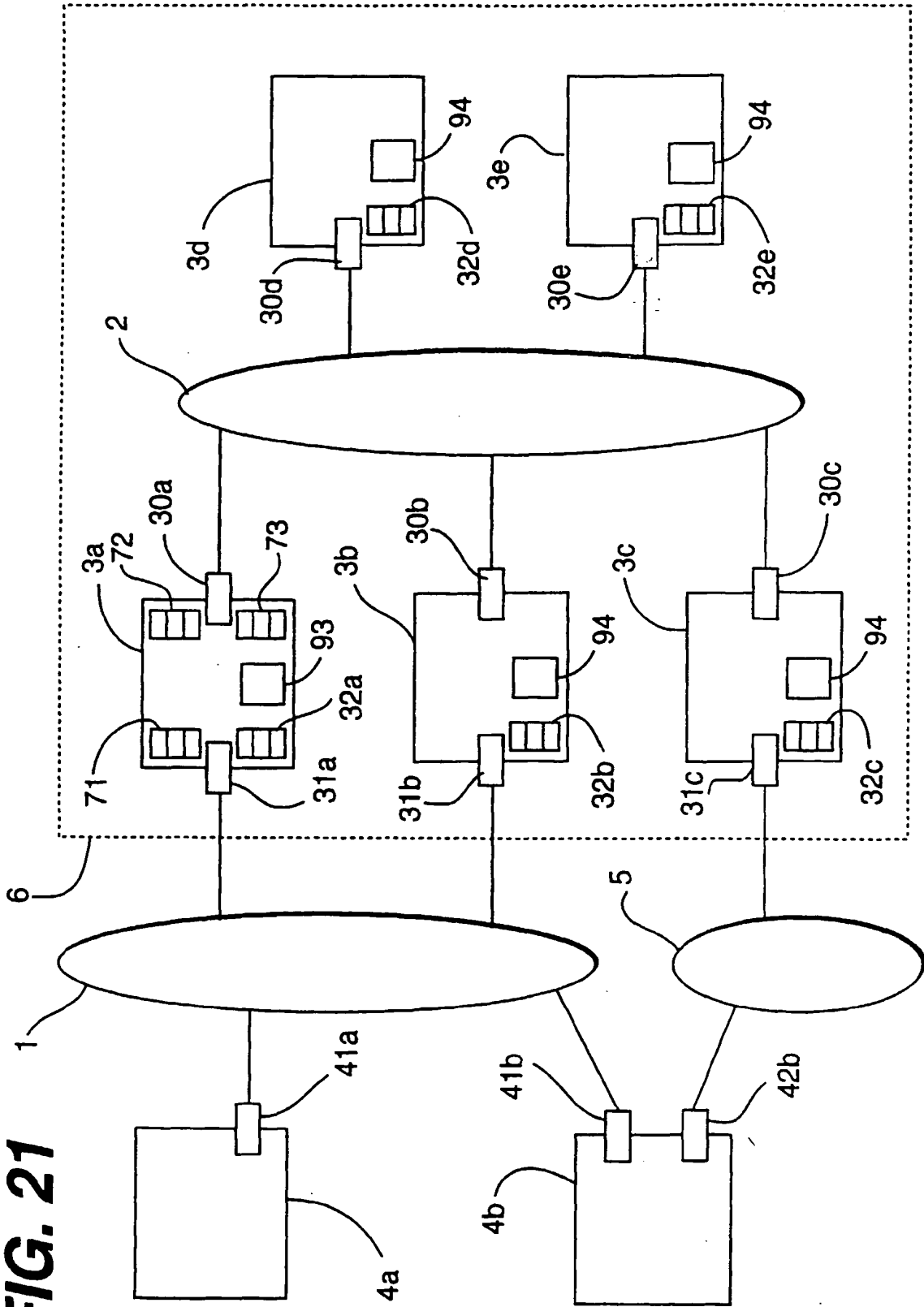


**FIG. 19**



**FIG. 20**

**FIG. 21**



(19) World Intellectual Property Organization  
International Bureau



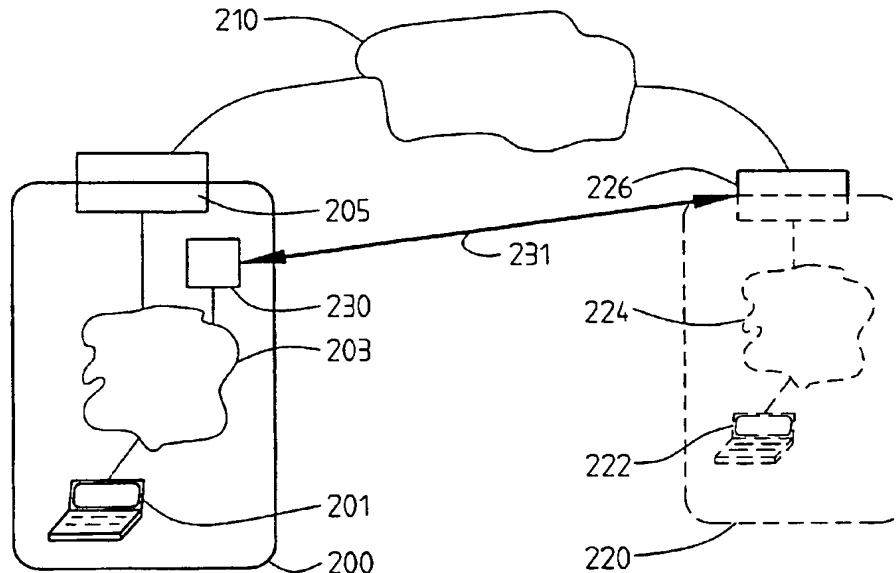
(43) International Publication Date  
12 July 2001 (12.07.2001)

PCT

(10) International Publication Number  
**WO 01/50688 A1**

- (51) International Patent Classification<sup>7</sup>: **H04L 12/46**, 12/56, 9/00
  - (21) International Application Number: PCT/SE00/02565
  - (22) International Filing Date:  
18 December 2000 (18.12.2000)
  - (25) Filing Language: English
  - (26) Publication Language: English
  - (30) Priority Data:  
9904841-5 29 December 1999 (29.12.1999) SE
  - (71) Applicant: **TELEFONAKTIEBOLAGET LM ERICSSON (publ.)** [SE/SE]; S-126 25 Stockholm (SE).
  - (72) Inventor: **KRIENS, Peter**; Finnasandsvägen 22, S-439 33 Onsala (SE).
  - (74) Agents: **BERGENTALL, Annika** et al.; Cegumark AB, P.O. Box 53047, S-400 14 Göteborg (SE).
  - (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.
  - (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- Published:**  
— *With international search report.*
- For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

(54) Title: METHOD AND SYSTEM FOR COMMUNICATION



(57) Abstract: A method and a system for establishing a connection between a first computer of a first computer network and a resource of a second computer network via a third network through a gateway intervening between the second computer network and the third network. A requester issues a request for a connection to the resource by specifying a name of the resource. A temporary IP number is returned to the first computer in answer to the request. The temporary IP number is mapped to a tunnel to the gateway. The gateway administrates the handling of data packets such that data packets addressed by the first computer to the temporary IP number, arriving through the tunnel, are routed to the resource and data packets arriving from the resource destined to the first computer, are routed through the tunnel to the first computer.

WO 01/50688 A1

Method and system for communication

5

## FIELD OF THE INVENTION

The present invention relates generally to a method and a system for communicating between different networks, especially from one network to a host within a private network.

10

## BACKGROUND TO THE INVENTION

The Internet is a collection of networks that can interwork. Clients connected to one network can access resources on other networks because data packets are routed from one network to the other. The Internet Protocol (IP) makes this possible. The same  
15 protocol can also be used to create private networks that are not directly connected to the Internet. These networks are called intranets. These intranets can be extended over a large area to remote offices using private lines. They are in a way the same intranet because there is a single authority that controls the network. Instead of private lines, an intranet can also be extended using the public internet as a tunneling  
20 medium. Instead of coupling an intranet directly to the Internet, the data traffic for a remote office is encapsulated and encrypted before being forwarded over the internet to the remote office. At the remote office, the reverse is done and the data package is placed in the local network. This is usually called a Virtual Private Network (VPN). For the end users it looks like a single private network, but the public Internet is used  
25 to securely transport data traffic between remote places.

In the Internet Protocol (IP) routing decisions are made on addresses. An address in IP is a 32-bit number. On the Internet, every host requires at least one unique number to be able to communicate. This unique number cannot be used by any other host on the  
30 Internet. A special official body allocates these IP numbers, and Internet routers all over the world must know how to map these IP numbers to the correct hosts. To simplify the routing and due to some original design choices the 4294967296 possible



numbers are running out. For this reason there are a number of number ranges that are reserved which anybody can use privately in e.g. a private intranet. However, IP packets cannot be routed over the Internet with a number within these ranges, and consequently must remain within the private intranet. This creates a problem when users of such an intranet with host numbers in these number ranges want to access the Internet.

There are two basic very close solutions to this problem, one is the use of a firewall and the other is using network address translation (NAT). Using a firewall, all access to the Internet is terminated at a firewall computer that is connected both to the Internet and to the intranet. This firewall then looks at the access from the intranet and acts as a proxy to the Internet using its own public IP number that is valid on the internet. However, a proxy requires a program that knows about the protocol. The other solution using NAT has a computer acting as a gateway between the Internet and the intranet. Every packet directed to the Internet is processed by a program that replaces/translates the address and port of the packet, and keeps a track of on who's behalf this translation is done. If the return packet comes, the address is translated back to the original address. NAT is a very transparent solution but unfortunately has some problems with some protocols, which then requires special measures.

20

A user does not have to use IP numbers to address a packet. When a user uses a name as an address then a special application, a name server, is used to translate the name into an IP number. On the Internet the Domain Name System (DNS) is used for naming. This is a hierarchical scheme where a DNS server can provide the translation for a domain or it can look up the name via/in another name server. If a DNS server comprises tables for a domain, then it is authoritative for that domain. Each DNS server is registered in a parent DNS server, this is done recursively until the root DNS servers are reached. Private intranets also require special handling of the DNS. A host on the inside of the intranet should not be visible on the outside, i.e. on the Internet, because it has a private number. However, when NAT is used, hosts on the outside of

30

the intranet are required to be present in the local intranet DNS. This is called a split universe DNS.

The real problems start when someone on the Internet wants private access to a host on an intranet with a private numbering scheme, or when two intranets with private numbering schemes want to connect privately. For example, assume that two companies, each with their own private intranet, decide to co-operate on a project and that they therefore want to share a number of resources on their respective intranets. This will cause a number of problems. The intranets cannot directly be routed to each other because the IP numbers used potentially overlap. Most probably the respective DNS of both companies are set-up as split universe DNSs and thus have no knowledge of each other's hosts. The normal forwarding to the internet DNS does not help since the domain of the other company does not expose the internal hosts with private IP numbers. Thus, since the internal hosts cannot see each other, it is impossible to route anything between them.

There have been a number of different solutions put forward. Unfortunately the known solutions either does not work for all protocols or they require complex administration or suffer from both disadvantages. For example, proxying is a solution to the problem. For each service that the companies want to share they have a publicly addressable host that contains a proxy for this service. This proxy does the mapping from the outside to the inside. A disadvantage of proxying is that it requires a significant amount of administration to set them up and then to keep them aligned with the original resources. Another disadvantage is that not all protocols are easy to proxy or have existing proxies. Another solution to the problem is to renumber the intranets so that a non-overlapping address space is created. A single DNS can then be used. However, this is a very complicated and heavy operation making it virtually impossible if the companies only co-operate on a project basis. This solution also requires a significant amount of trust between the parties in question.

30

A suggestion has also been disclosed in US patent number 5,898,830 to Wesinger, Jr. et al. (Wesinger). The Wesinger patent discloses a method of setting up virtual hosts in firewalls and using name based routing. The solution allegedly provides a full transparency for the users. However, this solution also only forwards hosts and not networks and it also requires quite a bit of administration.

There is thus a need to improve the methods of providing access to one or more hosts of a private intranet from the outside of the intranet with full transparency to users and a simple administration.

#### SUMMARY OF THE INVENTION

An object of the invention is to define a method and a system for transparently accessing hosts within a private intranet.

Another object of the invention is to define a method and a system for transparently accessing a host within private intranet by name.

A further object of the invention is to define a method and a system for accessing hosts within a private intranet with minimal administration.

A still further object of the invention is to define a method and a system for accessing hosts within a private intranet with security control and access control administration at the private intranet.

The aforementioned objects are achieved according to the invention by a method and a system for establishing a connection between a first computer of a first computer network and a resource, such as a second computer, of a second computer network via a third network through a gateway, such as a firewall, intervening between the second computer network and the third network. A requester issues a request for a connection from the first computer to the resource by specifying a name of the resource. A

temporary IP number is returned to the first computer in answer to the request. The temporary IP number is mapped to a tunnel to the gateway. The gateway administrates the handling of data packets such that data packets addressed by the first computer to the temporary IP number, arriving through the tunnel, are routed to the resource and  
5 data packets arriving from the resource destined to the first computer, are routed through the tunnel to the first computer.

The aforementioned objects are also achieved according to the invention by a method of establishing a connection between a first computer of a first computer network and a  
10 resource of a second computer network via a third network. The connection is established along a route through an intermediate system having an interface to the first computer network, and through a gateway intervening between the second computer network and the third network. The resource belongs to the domain of the gateway. According to the invention the method comprises a number of steps. A first step  
15 configuring the intermediate system with a tunnel from the intermediate system to the gateway. A second step mapping the tunnel with a requester and a domain name of the gateway. A third step wherein the requester issues a request for a connection from the first computer to the resource by specifying a name of the resource. A fourth step receiving the request at the intermediate system via the interface. A fifth step using a  
20 rule for matching the name of the resource with the gateway. A sixth step mapping the name of the resource to the tunnel. A seventh step returning a temporary IP number to the first computer in answer to the request. An eighth step mapping the temporary IP number to the name of the resource. A ninth step wherein the gateway administrates the handling of data packets such that data packets addressed by the first computer to the  
25 temporary IP number, arriving through the tunnel, are routed to the resource. And a tenth step wherein the gateway administrating the handling of data packets such that data packets arriving from the resource destined to the first computer, are routed through the tunnel to the first computer via the intermediate system. It is to be understood that the steps according to the invention do not indicate any sequential  
30 execution, but is merely a manner to distinguish them.

The method can advantageously further comprise the step of transmitting a message with the mapping of the temporary IP number to the gateway by means of the tunnel.

5 Preferably the step of the gateway administrating the handling of data packets such that data packets addressed by the first computer to the temporary IP number, arriving through the tunnel, are routed to the resource, comprises the substep of directing the intermediate system to translate source addresses of data packets addressed to the temporary IP number to be sent through the tunnel. The step of the gateway  
10 administrating the handling of data packets such that data packets addressed by the first computer to the temporary IP number, arriving through the tunnel, are routed to the resource, can comprise the substep of directing the intermediate system to translate destination addresses of data packets addressed to the temporary IP number to be sent through the tunnel, by means of at least a partial DNS function in the intermediate  
15 system.

Advantageously the step of the gateway administrating the handling of data packets such that data packets addressed by the first computer to the temporary IP number, arriving through the tunnel, are routed to the resource, can comprise the substep of the  
20 gateway translating source addresses of data packets arriving through the tunnel addressed to the temporary IP number and routing these data packets to the resource. The step of the gateway administrating the handling of data packets such that data packets addressed by the first computer to the temporary IP number, arriving through the tunnel, are routed to the resource, can comprise the substep of the gateway  
25 translating destination addresses of data packets arriving through the tunnel addressed to the temporary IP number and routing these data packets to the resource. The step of the gateway administrating the handling of data packets such that data packets arriving from the resource destined to the first computer, are routed through the tunnel to the first computer via the intermediate system, can comprise the substep of the gateway  
30 translating source and destination addresses of data packets arriving from the resource

destined to the first computer, and routing these data packets through the tunnel to the first computer via the intermediate system.

In some versions the step of the gateway administrating the handling of data packets such that data packets arriving from the resource destined to the first computer, are routed through the tunnel to the first computer via the intermediate system, can comprise the substep of directing the intermediate system to translate source and destination addresses of data packets arriving from the resource via the tunnel destined to the first computer.

10

In some versions the third network is a telecommunications network, in other versions it is the Internet, i.e. a computer network.

Advantageously the rule for matching the name of the resource with the gateway can be based on a mapping, and/or based on a list of hosts, and/or based on a regular or wildcard expression, and/or based on matching a domain name of the name of the resource with the domain name of the gateway.

15

Preferably the method further comprises the step of authenticating the requester at the first computer for access to the tunnel.

20

In some versions the name of the resource corresponds to a second computer within the second computer network, the second computer belonging to the domain of the gateway and comprising the resource. Then preferably the gateway administrates the handling of data packets such that data packets addressed by the first computer to the temporary IP number, arriving through the tunnel, are routed to the resource residing on the second computer. Otherwise in other versions the gateway administrates the handling of data packets such that data packets addressed by the first computer to the temporary IP number, arriving through the tunnel, are routed to the resource, the resource residing on a proxy of the second computer. Advantageously the proxy to

25

30

which the gateway routes data packets addressed by the first computer to the temporary IP number, is in dependence on an identity of the requester.

5 One or more of the features of the above described different methods according to the invention can be combined in any desired manner, as long as the features are not contradictory.

The aforementioned objects are achieved in accordance with the invention also by a device arranged to establish a connection between a first computer of a first computer network and a resource of a second computer network via a third network. The connection being established along a route through the device having an interface to the first computer network, and through a gateway intervening between the second computer network and the third network. The resource belongs to the domain of the gateway. According to the invention the device comprises a number of means arranged to carry out the invention. A first means arranged to configure a tunnel from the device to the gateway. A second means arranged to map the tunnel with a requester and a domain name of the gateway. A third means arranged to receive a request, issued by the requester, via the interface for a connection from the first computer to the resource by specifying a name of the resource. A fourth means arranged to use a rule for matching the name of the resource with the gateway. A fifth means arranged to map the name of the resource to the tunnel. A sixth means arranged to return a temporary IP number to the first computer in answer to the request. A seventh means arranged to map the temporary IP number to the name of the resource. An eighth means arranged to cooperate with the gateway administrating the handling of data packets such that data packets addressed by the first computer to the temporary IP number, arriving through the tunnel at the gateway, are routed to the resource. A ninth means arranged to cooperate with the gateway administrating the handling of data packets such that data packets arriving from the resource destined to the first computer, are at the gateway routed through the tunnel to the first computer via the device.

10  
15  
20  
25  
30

Different embodiments of the device according to the invention can be reached according to additional features mentioned above in connection with the description of the method according to the invention. The features of the above described different  
5      embodiments of a device according to the invention can be combined in any desired manner, as long as no conflict occurs.

By providing a device and a method for accessing one or more hosts within a private intranet, a plurality of advantages over prior art systems are obtained. According to the  
10     invention a route process/connection is made within a requesters network, which could also be a private intranet. Complete transparency is achieved; there is no restriction as to what protocol is used. The requester/user does not have to have any understanding of the set-up, such as the use of special ports or hosts and other network issues. The routing is name based; a requester/user requests access to a name of a host and will get  
15     an IP number in return to be used for access to the requested host. A requester is totally unaware that the request was intercepted and a route was set-up to respond to the IP number that was returned to the requester. All authentication and security issues such as access control can be handled by the private intranet to which access is desired. All the set-up at the requester's side that is required is some means of intercepting DNS  
20     requests before they are transferred to the internet. This means can, for example, be located in a gateway to the internet or at some other point logically before the gateway. This intercept means will have one or more tunnels configured to one or more private intranets and will determine if a DNS request is for one of the private intranets or not. If it determines that the DNS request is for one of the private intranets then a route  
25     process is set-up with an arbitrary but for the requestor valid IP number and a mapping to the corresponding tunnel is made. All access control can be handled at the other end of the tunnel, but in some embodiments some authentication and security is handled by the intercept means. Preferably all address translation is also done at the private intranet side of the tunnel, but in some embodiments at least some of the address  
30     translations can be handled directly by the intercept means, preferably under complete



control of the private intranet side of the tunnel. Further advantages and variations of the invention will become apparent from the following.

#### DESCRIPTION OF THE FIGURES

5 The invention will now be described in more detail for explanatory, and in no sense limiting, purposes, with reference to the following figures, in which

Fig. 1 shows a diagram of communication situation to which the invention is suitable,

10

Fig. 2 shows a diagram of an implementation of the invention,

Fig. 3 shows a flow chart of an example of an intermediate system processing,

15 Fig. 4 shows a flow chart of an example of a firewall/gateway processing when receiving from a tunnel,

Fig. 5 shows a flow chart of an example of a firewall/gateway processing when transferring a data packet from a second computer to a first computer.

20

#### DESCRIPTION OF PREFERRED EMBODIMENTS

In order to clarify the system according to the invention, some examples of its use will now be described in connection with Figures 1 to 5.

25 Figure 1 shows a diagram of a communication situation to which the invention is suitable. A user/requestor which is situated at a first computer 101 connected to a first computer network 103, which network can comprise several computer networks, within a first domain 100, which can be open or private, desires to communicate/gain  
30 access to a second computer 122, a destination host, connected to a second computer network 124, which network can also comprise several networks, which in turn is

within a second domain 120 which is private. A private domain is a domain which uses a private numbering scheme, i.e. hosts within the domain are not visible from the outside and can thus have the same number as a host on the internet. The first computer 101 and the second computer 122 are interconnected via, for example, an internet 110, a third computer network, a network, which will most likely comprise many networks, by means of a gateway/firewall 105 between the first computer network 103 and the third computer network 110, and a firewall/gateway 126 between the second computer network 124 and the third computer network 110. Other types of interconnections between the gateway/firewall 105 of the first computer network and the firewall/gateway 126 of the second computer network 124 are possible according to the invention. However, any direct ways of ordinary connection between the first computer 101 and the second computer 122 is not possible. The second computer 122 is not visible to the first computer 101 or to an internet 110, and if it is not visible then it is not ordinarily possible to route data packages from the first computer 101 to the second computer 122. Several known, less suitable, solutions to this situation have been discussed previously.

Figure 2 shows a diagram of an implementation of the invention. The set-up is the same as in figure 1 with a first computer 201, with a user/requestor, connected to a first computer network 203, which can comprise several computer networks, which in turn is connected to a gateway/firewall 205, all 201, 203, 205 of a first domain 200 which can be open or private. The gateway/firewall 205 is connected between the first computer network 203 and a third computer network 210. The third computer network 210, for example the Internet, will most likely comprise many networks. There is also a second computer 222, a desired destination, which is connected to a second computer network 224, which can comprise several networks, which in turn is connected to a firewall/gateway 226, all of a second domain 220 which is a private domain. The firewall/gateway 226 is connected between the third computer network 210 and the second computer network 224.

30

According to the invention there is also an intermediate system 230, an intercept means, connected somewhere into the first computer network 203. The intermediate system can be placed anywhere in the first domain 200, as long as it can intercept any DNS request from the first computer 201 before the request reaches the third computer network 210. To give a few examples, the intermediate system 230 can be a process running on the gateway/firewall 205, an intelligent connection box logically connected between the first computer 201 and the gateway/firewall 205, or even a process running on the first computer 201. The intermediate system 230 is preferably implemented as close as possible to, if not within, the gateway/firewall 205 to enable as many users/computers in the first domain 200 to have access to it, and thus have the possibilities of the invention. The intermediate system 230 will configure at least one tunnel 231 from the intermediate system to the firewall/gateway 226 of the second domain 220. A tunnel is a logical network connection between two processes, encapsulating the traffic during transport. Traffic over such a connection is traditionally encrypted to prevent eavesdropping. The tunnel or tunnels are preferably authenticated at regular, or irregular, intervals.

The intermediate system 230 will intercept DNS requests at least from the user or users and associate connection points/connected computers for which the intermediate system is set-up, in this example the first computer 201. The intermediate system must at least intercept DNS requests from the first computer 201 before the requests leave the domain 200. A user wanting a permitted access from the first computer 201 to the second computer 222 requests this by naming the second computer 222. The DNS request will then be intercepted by the intermediate system 230 which will determine if the requested name has any association with any tunnel 231 that is previously set-up. The determination can be based on a mapping, a list of hosts, or a regular or wildcard expression. In a preferred method the intermediate system 230 will try to match a domain name suffix of the second domain 220 to a domain name suffix of the DNS request for a match to the tunnel 231 of the example. As can be seen, the intermediate system does not have to be set-up with any details as to exactly which host or hosts are

requested for within the second domain 220. If there is a match the intermediate system will set-up a route to the second domain 220 via a tunnel 231 in view of the match, in this case the described tunnel 231. An IP number, a temporary random IP number, will be generated/made and associated to the route. The generated/given temporary random IP number must at least be valid within the first domain 200 so that communication addressed to that temporary random IP number will be correctly routed to the associated tunnel 231 of the intermediate system 230. The first computer 201 will get the temporary random IP number back as an answer to its DNS request and then use this temporary random IP number for all communication to the second computer 222, at least during this session. The communication will end up at the route interface, which in turn will send it down the tunnel for correct routing to the desired destination, the second computer 222. The temporary random IP number is mapped to the complete name of the DNS request and sent as a message to the gateway/firewall 226 at the other end of the tunnel. The gateway/firewall 226 at the other end of the tunnel 231 will deal with all the details of routing packages to and from the correct desired host, in this case the second computer 222. Return communications will either have the correct destination, the first computer 201, when they emerge from the tunnel 231, or there has been some address translation in the intermediate system 230, governed by the gateway/firewall 226 of the second domain 220, in which case the intermediate system 230 will retranslate the communication so that it will be routed correctly within the first domain 200 to the first computer 201.

For an even better understanding of the invention, it will be explained in relation to flow diagrams of a specific implementation of the invention. Flow diagrams describe something as a string of events, one after another. The different processes according to the invention are mostly independent event-driven processes. The major difference is that the processes of the invention might not appear in the order described below, but it is believed that the flow diagrams can however provide an easier understanding of the invention.

30

Figure 3 shows a flow chart of an example of the processes of an intermediate system according to one specific implementation of the invention. In a first step 340 one or more predetermined tunnels are configured and tables/mappings are generated/set-up. A table can, for example, be set-up in a matrix where each line comprises; a user (optionally), a source IP number, a destination domain (e.g. \*.ericsson.se), access time or times to the destination domain (optionally), a tunnel to the destination domain. The amount of information comprised in a table and the manner it is stored and mutually associated will vary in dependence of an implementation in question. A table/mapping can preferably be dynamically updated, i.e. information/entries added or deleted from for example a destination domain. In a second step 341 after the first step 340, authentication of the configured tunnel(s) and of configured users/requesters is done, for example, from which source IP number(s), e.g. the first computer, when, and to which domains access is allowed. In a third step 342 after the second step 341 it is determined if there is any communication to intercept or not, if there is none then it simply returns to itself. If there is some communication to intercept, the procedure continues with a fourth step 343 after the third step 342. The fourth step 343 determines if the communication was a DNS request or not. If the communication was determined to be a DNS request, then the procedure continues with a fifth step 344 after the fourth step 343. The fifth step 344 determines if the DNS request is from a configured user, e.g. the first computer, or not. If the DNS request is determined to have originated from a configured user then the procedure continues with a sixth step 345 after the fifth step 344. The sixth step 345 tries to match domains, in the configured user's map/table, with the domain of the DNS request. Thereafter the procedure continues with a seventh step 346 after the sixth step 345. The seventh step 346 determines if there is a match or not. If there is a match, then the procedure continues with an eighth step 347 after the seventh step 346. The eighth step 347 retrieves the entries of the user's map/table which correspond to the match of the seventh step 346 and also generates a temporary IP number, a temporary random IP number, which is a valid IP number in view of the place of the intermediate system. The intermediate system dynamically allocates a temporary IP number. Thereafter the procedure continues with

a ninth step 348 after the eighth step 347. The ninth step 348 maps the temporary random IP number to a tunnel according to the retrieved entries in the user's map/table. Thereafter the procedure continues with a tenth step 349 after the ninth step 348. The tenth step 349 will send a message through the tunnel with a mapping of the temporary  
5 random IP number with the complete DNS request, i.e. the complete name of the desired destination, e.g. the second computer. Thereafter the procedure continues with an eleventh step 350 after the tenth step 349. The eleventh step 350 returns the temporary random IP number to the requester, e.g. the first computer, in answer to the DNS request.

10

If in the fourth step 343 it was determined that it was not a DNS request, then the procedure continues with a twelfth step 351 after the fourth step 343. The twelfth step determines if the communication is a data packet or not. If it is determined to be a data packet then the procedure continues with a thirteenth step 352 after the twelfth step  
15 351. The thirteenth step 352 determines if the destination IP number of the data packet matches with any temporary random IP number which is mapped with the source IP number of the data packet. If there is a match, then the procedure continues with a fourteenth step 353 after the thirteenth step 352. The fourteenth step 353 sends the data packet in a tunnel according to the match and corresponding mapping/table entry. If it  
20 was determined in the twelfth step 351 that it was not a data packet, then the procedure continues with a fifteenth step 354 after the twelfth step 351. The fifteenth step 354 will ensure that the communication gets attention by means of some other processing. If it was determined in the thirteenth step 352 that there was no match, then the procedure continues with a sixteenth step 355 after the thirteenth step 352. The  
25 sixteenth step 355 provides normal routing of the data packet. If it was determined in the fifth step 344 that the DNS request was not from a configured user or if it was determined in the seventh step 346 that there is no match in the users domain name table, then the procedure continues with a seventeenth step 356 after the fifth step 344 or after the seventh step 346. The seventeenth step 356 provides a normal DNS request  
30 processing.

What happens next? We have opened a route interface process at the intermediate system and are now sending data packets and messages down a tunnel. Figure 4 shows a flow chart of an example of a second domain firewall/gateway processing when receiving from a tunnel. In a first step 460 the procedure waits for some communication received from a tunnel, and returns to itself as long as there is none. However when there is some communication received from a tunnel then the procedure continues with a second step 461 after the first step 460. The second step 461 determines if the communication is a message with a mapping of a temporary random IP number with a DNS request, or not, e.g a message sent by the tenth step 349 of Figure 3. If it is determined that it is not a message with a mapping then the procedure continues with a third step 462 after the second step 461. The third step 462 determines if the communication is a data packet to be routed or not. If it is determined that it is a data packet to be routed then the procedure continues with a fourth step 463 after the third step 462. The fourth step 463 determines if there exists a mapping/table or not for the destination IP number, i.e. a temporary random IP number, of the data packet. If there exists a mapping/table for the destination IP number then the procedure continues with a fifth step 464 after the fourth step 463. The fifth step 464 determines if security control of the tunnel through which the communication came is OK and still valid. If it is determined that the security of the tunnel is satisfactory, then the procedure continues with a sixth step 465 after the fifth step 464. The sixth step 465 determines if, according to the table/map, the source IP number, e.g. the IP number of the first computer, of the data packet have allowed access to the destination IP number, i.e. the temporary random IP number, of the data packet. If it is determined that the data packet from the source IP number has access to the destination IP number then the procedure continues with a seventh step 466 after the sixth step 465. The seventh step 466 translates/re-maps the source IP number, e.g. the IP number of the first computer, to a temporary locally valid IP number, a temporary local IP number. This is done so that the packet can be routed properly in the second domain. After the seventh step 466 the procedure continues with an eighth step 467 which lookups the real local IP number

of the destination, e.g. the second computer, by doing a DNS request in the second domain on the name received with the mapping to the temporary random IP number. The procedure then continues with a ninth step 468 after the eighth step 467. The ninth step 468 translates/re-maps the destination IP number, i.e. the temporary random IP number, of the data packet to the real local IP number of the destination, e.g. the second computer. Thereafter the procedure continues with a tenth step 469 after the ninth step 468. The tenth step 469 routes the data packet in the second domain to the destination, e.g. the second computer, with the real local IP number as destination and the temporary local IP number as the source.

10

If it was determined in the second step 461 that the communication was a map/table message then the procedure continues with an eleventh step 470 after the second step 461. The eleventh step 470 receives a mapping of a temporary random IP number with a DNS name, e.g. the second computer, of the second domain, and adds this to its mapping. If it was determined in the third step 462 that it was not a data packet to be routed that was received through the tunnel, then the procedure continues with a twelfth step 471 after the third step 462. The twelfth step 471 does other appropriate processing. If it was determined in the fifth step 464 that the security of the tunnel is not valid then the procedure could continue with a thirteenth step 472 after the fifth step 464. The thirteenth step 472 will then try to authenticate the tunnel, and then return and continue with the fifth step. If it was determined in the fourth step 463 that there does not exist a mapping/table or if it was determined in the sixth step 465 that the source IP number is not allowed access to the destination IP number, then the procedure continues with a fourteenth step 473 after either the fourth step 463 or the sixth step 465. The fourteenth will reject request, and not route the data packet, the "destination is unknown". Preferably security will also be alerted of an attempted breach of security.

As mentioned, packets must be able to be sent back to the original requester. Figure 5 shows a flow chart of an example of firewall/gateway processing when transferring a

30



data packet from a second computer to a first computer. In a first step 580 it is checked if there is any communication from within the second computer network, and if not then just return to itself. If there is communication from within the second computer network, then the procedure continues with a second step 581 after the first step 580.

5 The second step 581 determines if it is a data packet that should be routed. If it is a data packet to be routed then the procedure continues with a third step 582 after the second step 581. The third step 582 determines if the destination IP number of the data packet is equal to any valid temporary local IP number. If the destination IP number is matched then the procedure continues with a fourth step 583 after the third step 582.

10 The fourth step retrieves the mapping/table that corresponds to the matched temporary local IP number to thereby find out where, which tunnel, to route the data package. After the fourth step 583 the procedure continues with a fifth step 584 which translates (re-maps) the source IP number, the IP number of the second computer, of the data packet to the temporary random IP number according to table (map). After the fifth

15 step 584 the procedure continues with a sixth step 585 which translates (re-maps) the destination IP number, the temporary local IP number, of the data packet to the IP number of the first computer according to the table (map). Thereafter in a seventh step 586 after the sixth step 585 the data packet is transferred in an appropriate tunnel according to the table (map). If it was determined in the second step 581 that it is not a

20 data packet that is to be routed then the procedure continues with an eighth step 587 after the second step 581 and does some other processing. If it was determined in the third step 582 that the destination IP number of the data packet is not equal to any valid temporary local IP number then the procedure continues with a ninth step 588 after the third step 582 and does a normal routing of the data packet.

25

The present invention can be put into apparatus-form either as pure hardware, as pure software or as a combination of both hardware and software. If the method according to the invention is realized in the form of software, it can be completely independent or it can be one part of a larger program. The software can suitably be located in a

30 general-purpose computer or in a dedicated computer.

As a summary, the invention can basically be described as a method of accessing one or more hosts within a private network by means of a route interface process.

- 5 The invention is not limited to the embodiments described above but may be varied within the scope of the appended patent claims.

- FIG 1 a diagram of communication situation to which the invention is suitable,  
5 100 open or private first domain  
101 user/requestor, a first computer,  
103 a first computer network, can comprise several computer networks,  
105 gateway/firewall between the first computer network and a third  
computer network,  
10 110 internet, the third network, will most likely comprise many networks  
120 private second domain,  
122 a second computer, a destination,  
124 a second computer network, can comprise several networks,  
126 a firewall/gateway between the second computer network and the third  
15 computer network.
- FIG 2 a diagram of an implementation of the invention,  
200 open or private first domain,  
201 user/requestor, a first computer, a source,  
20 203 a first computer network, can comprise several computer networks,  
205 gateway/firewall between the first computer network and a third  
computer network,  
210 internet, the third computer network, will most likely comprise many  
networks,  
25 220 private second domain,  
222 a second computer, a destination,  
224 a second computer network, can comprise several networks, to which  
the second computer is connected,  
226 a firewall/gateway between the third computer network and the second  
30 computer network, the second computer,

230 an intermediate system between the third computer network and the first  
computer, the source,  
231 a tunnel from the intermediate system to the firewall.

5 FIG 3 flow chart of an example of intermediate system processing,  
340 : configure tunnels and generate tables/mappings  
341 from 340: authentication of tunnel(s) and of users/requesters, for  
example from which source IP number(s), e.g. the first computer, when,  
and to which domains,  
10 342 from 341 or no from itself: any communication ?  
343 yes from 342: is it a DNS request ?  
344 yes from 343: is it from a configured user, e.g. the first computer ?  
345 yes from 344: try to match domains, in the configured user's table, with  
the domain of the DNS request,  
15 346 from 345: is there a match,  
347 yes from 346: get map/table and also generate a temporary IP number, a  
temporary random IP number, which is a valid IP number in view of the  
place of the intermediate system,  
348 from 347: map the temporary IP number to a tunnel according to the  
20 retrieved map/table,  
349 from 348: send message through tunnel with mapping of temporary  
random IP number with the DNS request,  
350 from 349: return temporary random IP number to requester, e.g. the first  
computer, in answer to the DNS request,  
25 351 no from 343: is it a data packet ?  
352 yes from 351: does destination IP number of the data packet match with  
any temporary random IP number which is mapped with the source IP  
number of the data packet,  
353 yes from 352: send data packet in a tunnel according to mapping/table  
30 entry,

- 354 no from 351: other processing,
- 355 no from 352: normal routing of data packet,
- 356 no from 344 or no from 346: do a normal DNS request processing.
- 5 FIG 4 flow chart of an example of firewall processing when receiving from a tunnel,
- 460 no from itself: communication received from a tunnel?
- 461 yes from 460: is the communication a map/table message?
- 462 no from 461: is the communication a data packet to be routed?
- 10 463 yes from 462: does there exist a mapping/table for the destination IP number, i.e. a temporary random IP number, of the data packet?
- 464 yes from 463 or from 472: security control of tunnel, through which the communication came, is it OK, still valid ?
- 465 yes from 464: does, according to the table/map, the source IP number, e.g. the IP number of the first computer, of the data packet have allowed access to the destination IP number, i.e. the temporary random IP number, of the data packet ?
- 15 466 yes from 465: translate/remap source IP number, e.g. the IP number of first computer, to a temporary locally valid IP number, a temporary local IP number,
- 20 467 yes from 466: lookup of real local IP number of destination, e.g. the second computer, by DNS in the second domain,
- 468 yes from 467: translate/remap destination IP number, i.e. the temporary random IP number, of the data packet to the real local IP number of the destination, e.g. the second computer,
- 25 469 yes from 468: route the data packet in the second domain to the destination, e.g. the second computer, with the real local IP number as destination and the temporary local IP number as the source,
- 470 yes from 461: receive a mapping of a temporary random IP number with a DNS name, e.g. the second computer, of the second domain,
- 30

- 471 no from 462: do other processing,  
472 no from 464: authenticate tunnel,  
473 no from 463 or no from 465: reject request, do not route data packet,  
“destination unknown”, alarm security of an attempted break in.
- 5
- FIG 5 flow chart of an example of firewall processing when transferring a data  
packet from a second computer to a first computer,
- 580 no from itself: communication from within the second computer  
network ?
- 10 581 yes from 580: is it a data packet that should be routed ?  
582 yes from 581: is the destination IP number of the data packet equal any  
valid temporary local IP number ?  
583 yes from 582: get mapping/table to find out where, which tunnel, to  
route the data package,
- 15 584 from 583: translate (remap) the source IP number, the IP number of the  
second computer, of the data packet to temporary random IP number  
according to table (map),  
585 from 584: translate (remap) the destination IP number, the temporary  
local IP number, of the data packet to the IP number of the first  
20 computer according to the table (map),  
586 from 585: transfer data packet in appropriate tunnel according to table  
(map)  
587 no from 581: other processing,  
588 no from 582: normal routing.

## CLAIMS

5

1. A method of establishing a connection between a first computer of a first computer network and a resource of a second computer network via a third network, along a route through an intermediate system having an interface to the first computer network, and through a gateway intervening between the second computer network and the third network, the resource belonging to the domain of the gateway  
10 **characterized in that** the method comprises the following steps:
- configuring the intermediate system with a tunnel from the intermediate system to the gateway;
  - mapping the tunnel with a requester and a domain name of the gateway;
  - 15 - the requester issuing a request for a connection from the first computer to the resource by specifying a name of the resource;
  - receiving the request at the intermediate system via the interface;
  - using a rule for matching the name of the resource with the gateway;
  - mapping the name of the resource to the tunnel;
  - 20 - returning a temporary IP number to the first computer in answer to the request;
  - mapping the temporary IP number to the name of the resource;
  - the gateway administrating the handling of data packets such that data packets addressed by the first computer to the temporary IP number, arriving through the tunnel, are routed to the resource;
  - 25 - the gateway administrating the handling of data packets such that data packets arriving from the resource destined to the first computer, are routed through the tunnel to the first computer via the intermediate system.

2. The method according to claim 1, **characterized in that** the method  
30 further comprises the step of:

- transmitting a message with the mapping of the temporary IP number to the gateway by means of the tunnel.
3. The method according to claim 1 or 2, **characterized in that** the step  
5 of the gateway administrating the handling of data packets such that data packets addressed by the first computer to the temporary IP number, arriving through the tunnel, are routed to the resource, comprises the substep of:
- directing the intermediate system to translate source addresses of data packets addressed to the temporary IP number to be sent through the tunnel.
- 10
4. The method according to any one of claims 1 to 3, **characterized in that** the step of the gateway administrating the handling of data packets such that data packets addressed by the first computer to the temporary IP number, arriving through the tunnel, are routed to the resource, comprises the substep of:
- 15 - directing the intermediate system to translate destination addresses of data packets addressed to the temporary IP number to be sent through the tunnel, by means of at least a partial DNS function in the intermediate system.
5. The method according to claim 1 or 2, **characterized in that** the step  
20 of the gateway administrating the handling of data packets such that data packets addressed by the first computer to the temporary IP number, arriving through the tunnel, are routed to the resource, comprises the substep of:
- the gateway translating source addresses of data packets arriving through the tunnel addressed to the temporary IP number and routing these data packets to  
25 the resource.
6. The method according to claim 1, 2, 3 or 5, **characterized in that** the  
step of the gateway administrating the handling of data packets such that data packets  
addressed by the first computer to the temporary IP number, arriving through the  
30 tunnel, are routed to the resource, comprises the substep of:



- the gateway translating destination addresses of data packets arriving through the tunnel addressed to the temporary IP number and routing these data packets to the resource.

5 7. The method according to any one of claims 1 to 6, **characterized in that** the step of the gateway administrating the handling of data packets such that data packets arriving from the resource destined to the first computer, are routed through the tunnel to the first computer via the intermediate system, comprises the substep of:

- the gateway translating source and destination addresses of data packets  
10 arriving from the resource destined to the first computer, and routing these data packets through the tunnel to the first computer via the intermediate system.

8. The method according to any one of claims 1 to 6, **characterized in that** the step of the gateway administrating the handling of data packets such that data  
15 packets arriving from the resource destined to the first computer, are routed through the tunnel to the first computer via the intermediate system, comprises the substep of:

- directing the intermediate system to translate source and destination addresses of data packets arriving from the resource via the tunnel destined to the first  
20 computer.

9. The method according to any one of claims 1 to 8, **characterized in that** the third network is a telecommunications network.

10. The method according to any one of claims 1 to 8, **characterized in that** the third network is the Internet.  
25

11. The method according to any one of claims 1 to 10, **characterized in that** the rule for matching the name of the resource with the gateway is based on a mapping.  
30

12. The method according to any one of claims 1 to 10, **characterized in that** the rule for matching the name of the resource with the gateway is based on a list of hosts.
- 5 13. The method according to any one of claims 1 to 10, **characterized in that** the rule for matching the name of the resource with the gateway is based on a regular or wildcard expression.
- 10 14. The method according to any one of claims 1 to 10, **characterized in that** the rule for matching the name of the resource with the gateway is based on matching a domain name of the name of the resource with the domain name of the gateway.
- 15 15. The method according to any one of claims 1 to 14, **characterized in that** the method further comprises the step of:
- authenticating the requester at the first computer for access to the tunnel.
- 20 16. The method according to any one of claims 1 to 15, **characterized in that** the name of the resource corresponds to a second computer within the second computer network, the second computer belonging to the domain of the gateway and comprising the resource.
- 25 17. The method according to claim 16, **characterized in that** the gateway administrating the handling of data packets such that data packets addressed by the first computer to the temporary IP number, arriving through the tunnel, are routed to the resource residing on the second computer.
18. The method according to claim 16, **characterized in that** the gateway administrating the handling of data packets such that data packets addressed by

the first computer to the temporary IP number, arriving through the tunnel, are routed to the resource, the resource residing on a proxy of the second computer.

19. The method according to claim 18, **characterized in that** the proxy  
5 to which the gateway routes data packets addressed by the first computer to the temporary IP number, is in dependence on an identity of the requester.

20. A device arranged to establish a connection between a first computer  
of a first computer network and a resource of a second computer network via a third  
10 network, along a route through the device having an interface to the first computer network, and through a gateway intervening between the second computer network and the third network, the resource belonging to the domain of the gateway **characterized in that** the device comprises:

- means arranged to configure a tunnel from the device to the gateway,
- 15 - means arranged to map the tunnel with a requester and a domain name of the gateway,
- means arranged to receive a request, issued by the requester, via the interface for a connection from the first computer to the resource by specifying a name of the resource,
- 20 - means arranged to use a rule for matching the name of the resource with the gateway,
- means arranged to map the name of the resource to the tunnel,
- means arranged to return a temporary IP number to the first computer in answer to the request,
- 25 - means arranged to map the temporary IP number to the name of the resource,
- means arranged to cooperate with the gateway administrating the handling of data packets such that data packets addressed by the first computer to the temporary IP number, arriving through the tunnel at the gateway, are routed to the resource,

- means arranged to cooperate with the gateway administrating the handling of data packets such that data packets arriving from the resource destined to the first computer, are at the gateway routed through the tunnel to the first computer via the device.

1/4

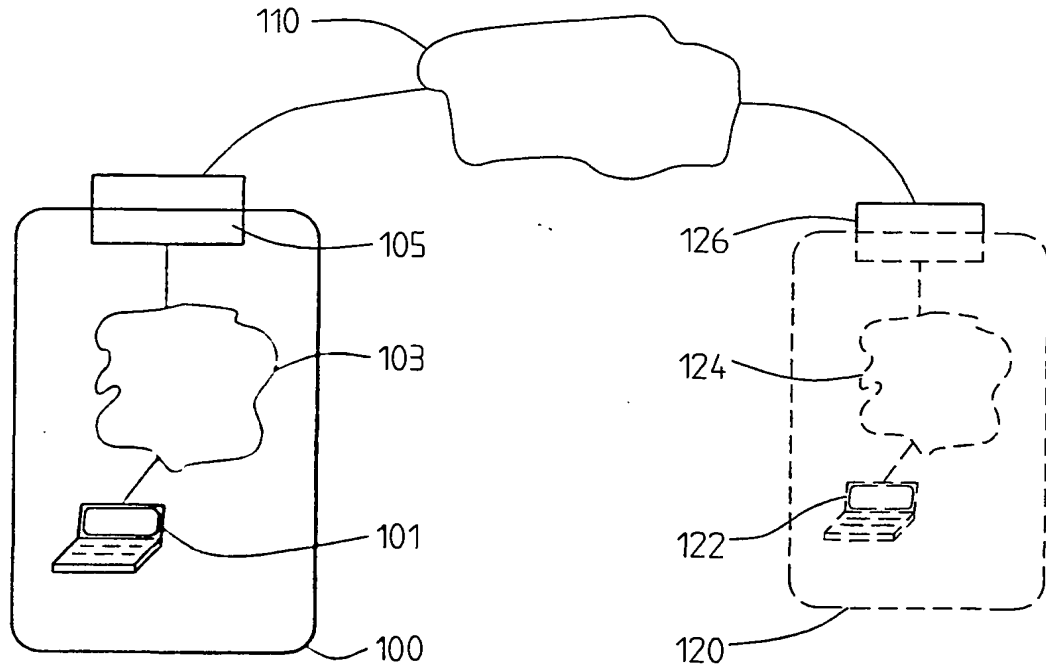


Fig. 1

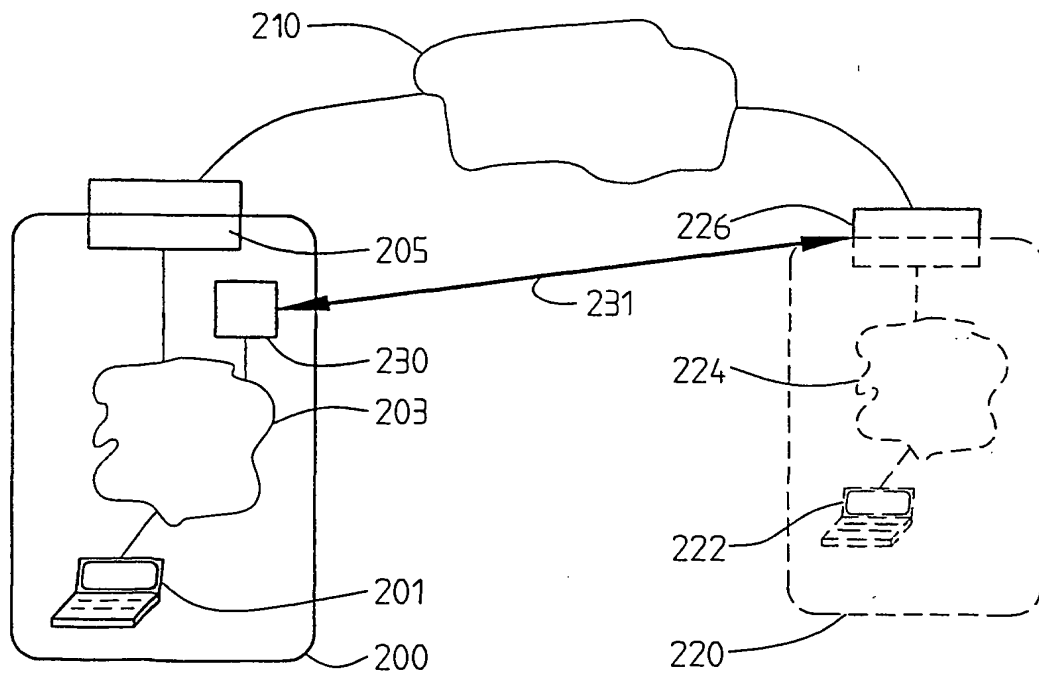


Fig. 2

2/4

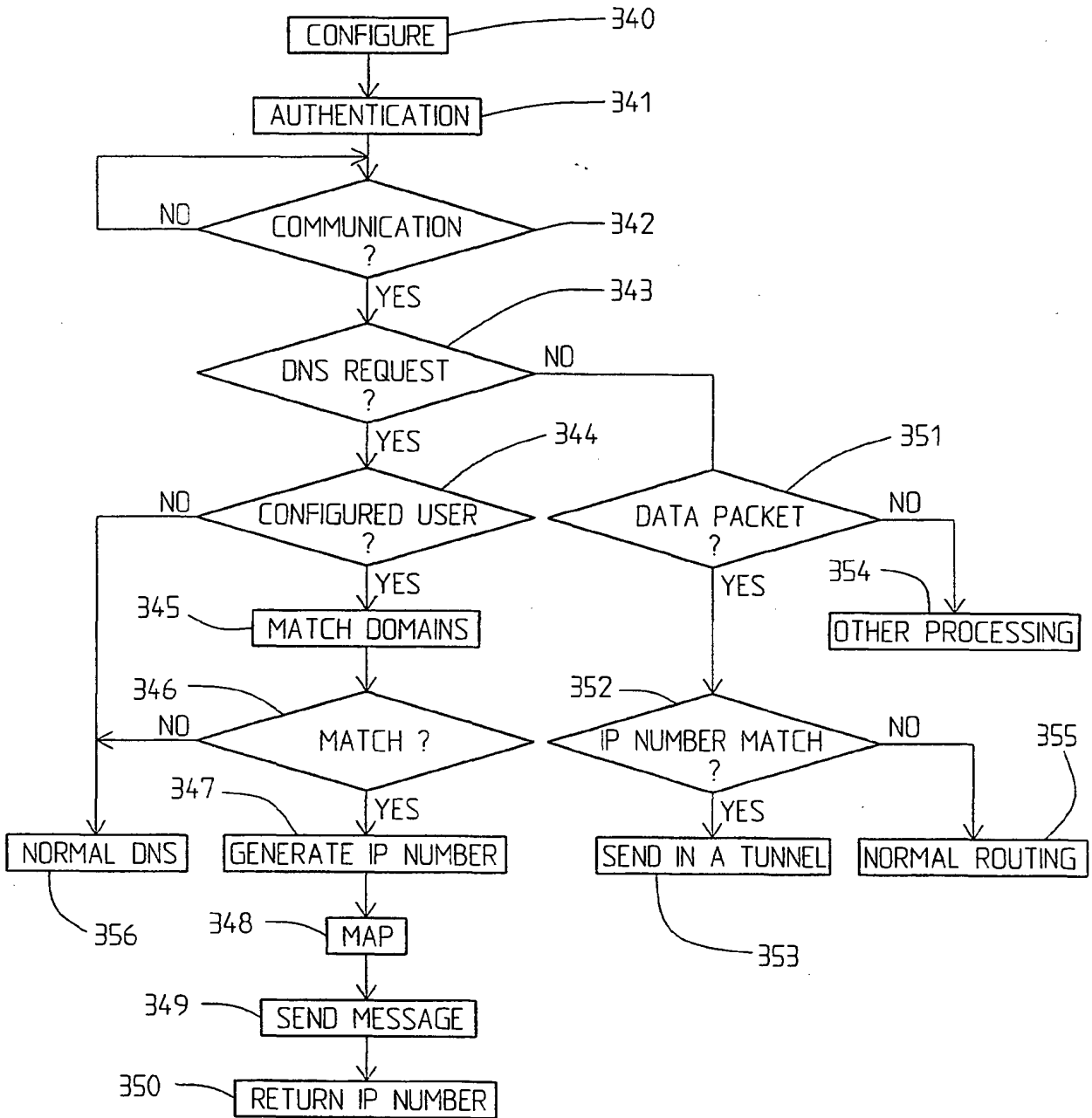


Fig. 3

3/4

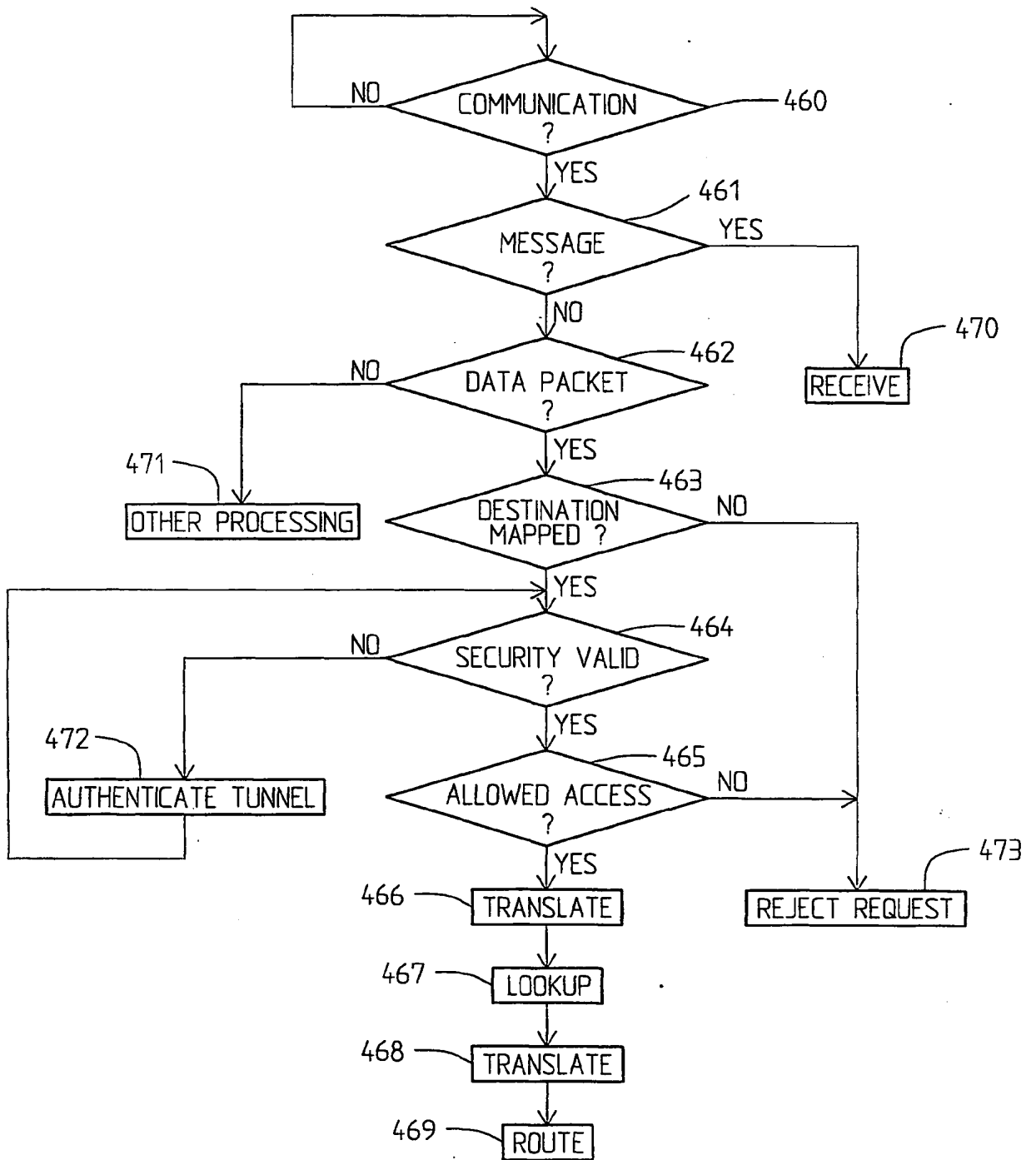


Fig. 4

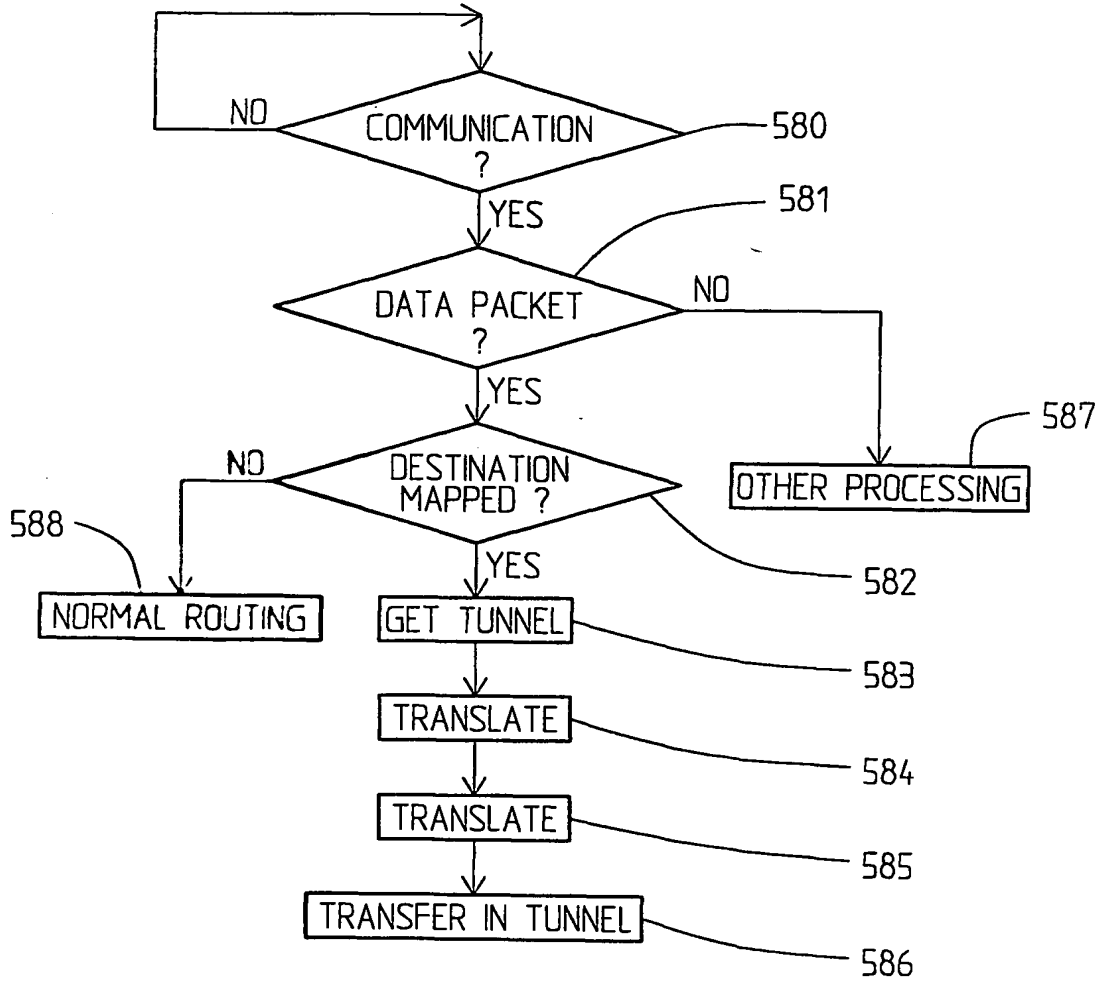


Fig. 5

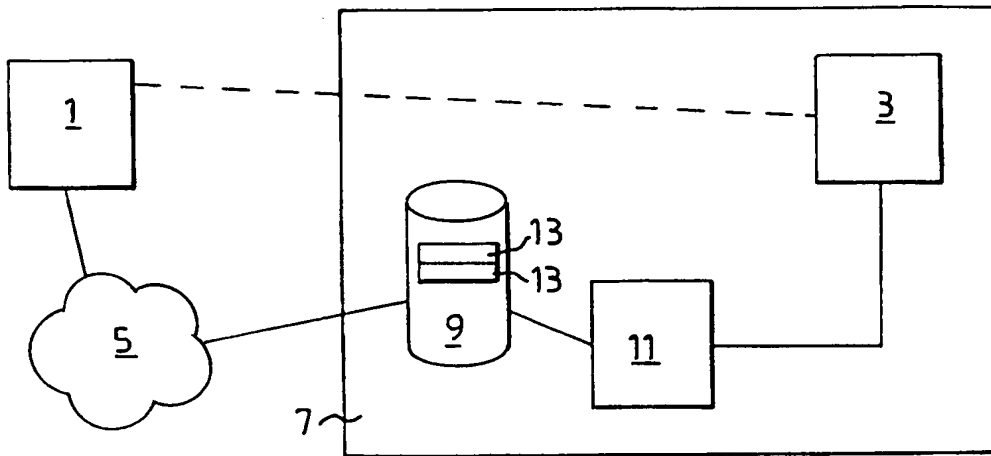




INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(51) International Patent Classification <sup>6</sup> : <b>H04L 12/56</b></p>	<p><b>A2</b></p>	<p>(11) International Publication Number: <b>WO 98/59470</b> (43) International Publication Date: 30 December 1998 (30.12.98)</p>
<p>(21) International Application Number: PCT/SE98/01217 (22) International Filing Date: 23 June 1998 (23.06.98) (30) Priority Data: 9702385-7 23 June 1997 (23.06.97) SE (71) Applicants (for all designated States except US): TELEFON-AKTIEBOLAGET LM ERICSSON (publ) [SE/SE]; S-126 25 Stockholm (SE). TELIA AB [SE/SE]; S-123 86 Farsta (SE). (72) Inventors; and (75) Inventors/Applicants (for US only): KANTER, Theo [NL/SE]; Rönninge skolväg 35E, S-144 62 Rönninge (SE); FOGELHOLM, Rabbe [SE/SE]; Turesåsen 54 B, S-191 47 Sollen-tuna (SE). (74) Agents: HERBJØRNSSEN, Rur et al., Albinus Patentbyrå Stockholm AB, P.O. Box 3137, S-103 62 Stockholm (SE).</p>		<p>(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, GM, GW, HU, ID, 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, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, 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, CY, 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).</p> <p><b>Published</b> <i>Without international search report and to be republished upon receipt of that report.</i></p>

(54) Title: METHOD AND APPARATUS TO ENABLE A FIRST SUBSCRIBER IN A LARGER NETWORK TO RETRIEVE THE ADDRESS OF A SECOND SUBSCRIBER IN A VIRTUAL PRIVATE NETWORK



(57) Abstract

The present invention relates to an apparatus and a method for use in a virtual private network, VPN, (7, 7'), or a network domain forming part of a larger network, such as the Internet, to enable a first subscriber (1; 1') in the larger network to retrieve the address of a second subscriber (3; 3') in the VPN. The address may be returned to the first subscriber (1; 1') or a connection means (11) may set up the connection between the subscribers (1, 3; 1', 3') automatically.

**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		Republic of Korea	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		

METHOD AND APPARATUS TO ENABLE A FIRST SUBSCRIBER IN A LARGER NETWORK TO RETRIEVE THE ADDRESS OF A SECOND SUBSCRIBER IN A VIRTUAL PRIVATE NETWORK

### Technical Field

The present invention relates to the communication between terminals connected to data or multimedia networks, such as the Internet.

### Background

Internet Protocol (IP) type networks are used to an increasing degree for data, video and audio communication. It is a problem for subscribers in such networks to find the physical addresses, or IP addresses, of subscribers in other networks or subnetworks.

### Summary of the Invention

It is an object of the present invention to enable a subscriber in any part of an IP based network to locate other subscribers in the same or other parts of the IP based network.

It is another object of the invention to enable subscribers in any part of an IP based network to connect to other subscribers in the same or other parts of the IP based network, for any kind of communication according to any known protocol.

It is yet another object of the invention to enable a subscriber to move between different locations in the network and still be reached.

The objects are achieved in a network by using a name server means according to the invention for each Virtual Private Network (VPN) connected to the network, the name server means being adapted to

- resolve a logical address in the VPN to the real IP address of hosts and user terminals for a specific service, such as e-mail or communication according to the H.323 protocol,

- function as a look-up table between the logical E.164 addresses in the VPN and the real IP addresses of the hosts and users
- cooperate with connection means for call set-up.

- 5 The solution according to the invention offers the following advantages:  
As it is based on known solutions, it may be implemented at a relatively low cost.  
It involves the separation of an internal and an external number plan, thus increasing the flexibility in the network.  
It enables the connection between an H.323 domain and an Internet domain.

10

### **Brief Description of the Drawings**

Figure 1 is a schematic drawing of a connection between two user terminals set up according to a first embodiment of the invention.

- 15 Figure 2 is a flow chart of the actions performed when a connection between two user terminals is set up according to the first embodiment.

Figure 3 is a schematic drawing of a connection between two subscribers set up according to a second embodiment of the invention.

- 20 Figure 4 is a flow chart of the actions performed when a connection between two user terminals is set up according to the second embodiment.

### **Detailed Description of Embodiments**

- The dotted line in Figure 1 shows a connection between a first 1 and a second 3 user terminal. The terminals 1, 3 may be any kind of terminals which may be used for  
25 communication, for example personal computers (PCs) or telephones. The first user terminal 1 is connected to a data or telecommunications network 5 via a leased line, a modem a corporate network, or in any other way. The network 5 may be any network allowing communication between two end points on a logical connection, which may be packet switched or circuit switched. A common network today, in  
30 which the teachings of the invention may become particularly useful, is the Internet.

In the following discussion, therefore, the network 5 will be referred to as the Internet.

The second user terminal is found in a Virtual Private Network (VPN) 7, which  
5 functions as an Internet domain. A name server 9 in the VPN 7 is connected to the Internet 5 and to a connection unit 11. In TCP/IP networks the name server 9 might be a Domain Name Server (DNS) well known in the art. If the H.323 protocol for data, audio and video communication is used, the connection unit 11 might be a gatekeeper, of a kind well known in the art. The connection unit 11 is connected to  
10 the second user terminal 3 with a semi-permanent connection.

The name server 9 is a database comprising, in addition to the information found in prior art name servers, an MX record 13 for each user terminal in the VPN 7. The MX record comprises information about the IP addresses of all user terminals in the  
15 VPN 7 for different types of communication, for example, e-mail, H.323, or telnet connections.

Figure 2 shows the actions taken when the first user 1 in the first embodiment wishes to establish a connection to the second subscriber 3.  
20

Step S11: The first user 1 connects to the name server 9 and requests the gate number for H.323 and enters the known address of the second user 3.

Step S12: The name server 9 determines what type of connection is wanted and  
25 forwards the request to the connection unit 11, together with the address of the first user 1.

Step S13: The connection unit 11 retrieves the appropriate IP address of the second user 3 for the type of connection, in this case, the H.323 address.  
30 The type of connection may be determined, for example, by the port of the name server at which the connection is made.

Step S14: The connection unit 11 establishes the connection between the users 1, 3.

5 Figure 3 shows a second embodiment of the invention. In this embodiment a first user terminal 1' is connected to a second user terminal 3' as shown by the dotted line. The second user terminal is found in a VPN 7', which also comprises a name server 9', identical to the name server 9 in Figure 1. A user directory 11' is connected to the name server 9'. The user directory 9' comprises information about  
10 the physical addresses of the user terminals 3' in the VPN 7'. In a TCP/IP network, the name server will be a Domain Name Server (DNS) and the user directory will be a Lightweight Directory Access Protocol (LDAP) server of the kinds known in the art.

15 Figure 4 shows the actions taken when the first user 1' in the second embodiment wishes to establish a connection to the second subscriber 3'.

Step S21: The first user 1' connects to the name server 9' and transmits the known, logical address of the second user 3' to the name server 9'.

20

Step S22: The name server 9' determines what type of connection is wanted and forwards the logical address of the second user 3' to the user directory 11' of the VPN 7'.

25 Step S23: The user directory 11' retrieves the physical address corresponding to the logical address entered.

Step S24: The user directory 11' returns the physical address of the second user 3' to the first user 1' via the name server 9'.

30

Step S25: The first user 1' initiates the connection to the second user 3' in a conventional manner.

If the first user 1' knows the address to the user directory 11', he can go directly to  
5 the user directory 11' instead of connecting via the name server 9'.

**Claims**

- 1. A name server means (9; 9') for use in a virtual private network (7; 7'), or a network domain, forming part of a compound network,  
 5 said means (9; 9') being **characterized** by means (13;13') for receiving a request for the physical address of a user terminal (3; 3') from another user terminal (1; 1') and forwarding said request to a connection means (11; 11') in the virtual private network (7; 7') or network domain.
- 10 2. A name server means according to claim 1, **characterized** in that the logical addresses comprise IP addresses, addresses according to the E.164 protocol and/or other logical identities according to the appropriate numbering plan.
- 15 3. A name server means according to claim 1 or 2, **characterized** by means (11, 11') for initiating the connection between two subscribers (1, 3).
- 4. A connection means (11; 11') for use in a virtual private network (7; 7') or a network domain, forming part of a compound network, said connection means being **characterized** in that it is adapted to return, upon a request comprising a logical  
 20 address of a user (3;3') in the virtual private network (7; 7'), a physical address of said user (3; 3').
- 5. A connection means (11; 11') according to claim 4, **characterized** in that it is adapted, upon a request originating from a user (1; 1') in said compound network,  
 25 said request comprising a logical address of a user (3; 3') in the virtual private network (7; 7'), to establish a connection between said users (1, 3; 1' 3').
- 6. A telecommunications or data communications network, forming part of a compound network, **characterized** by at least one connection means, according to  
 30 claim 4 or 5.



7. A network according to claim 6, characterized by at least one name server means according to any one of claims 1-3.

5 8. A method for enabling a user (1; 1') in compound network to retrieve the IP address of a second user (3; 3') in a virtual private network (7; 7') or a network domain, forming part of said compound network, characterized by the following steps:

- 10 - transmitting a request for a physical address, the request comprising a logical address of the second user (3; 3')
- forwarding the logical address of the second user (3; 3') to a connection means (11);
- returning the logical address to the first user (1; 1') or automatically establishing a connection between the first (1; 1') and the second (3; 3') user.

15

9. A method according to claim 8, characterized by

- automatically establishing a connection between the first (1) user and the second user (3).

20 10. A method according to claim 8, characterized by

- returning the address of the second user (3') to the first user (1').

11. A method according to any one of claims 8-10, characterized by determining the type of address to be used in dependence of the port of the name server (9; 9') on  
25 which the request was received.

12. A method according to any one of claims 6-11, characterized in that the physical address may be an e-mail-address, and/or an E.164 address.

**SUBSTITUTE SHEET (RULE 26)**

1/2

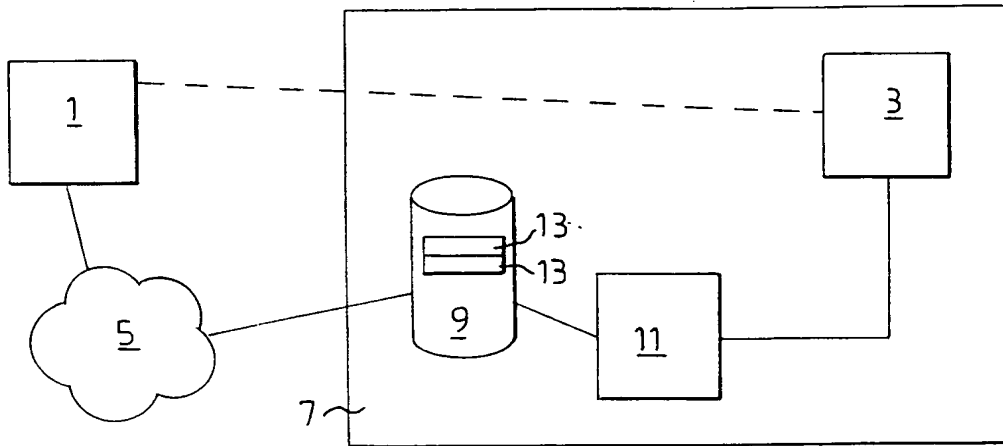


FIG. 1

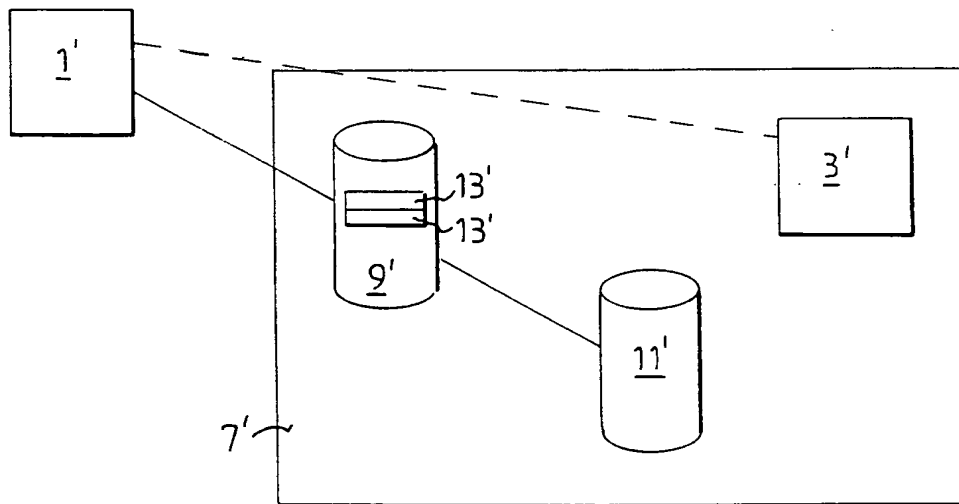


FIG. 3

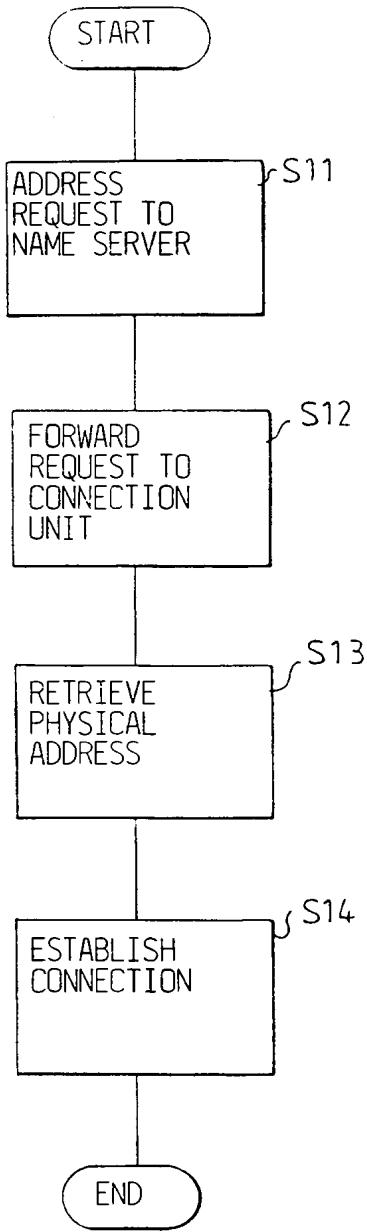


FIG. 2

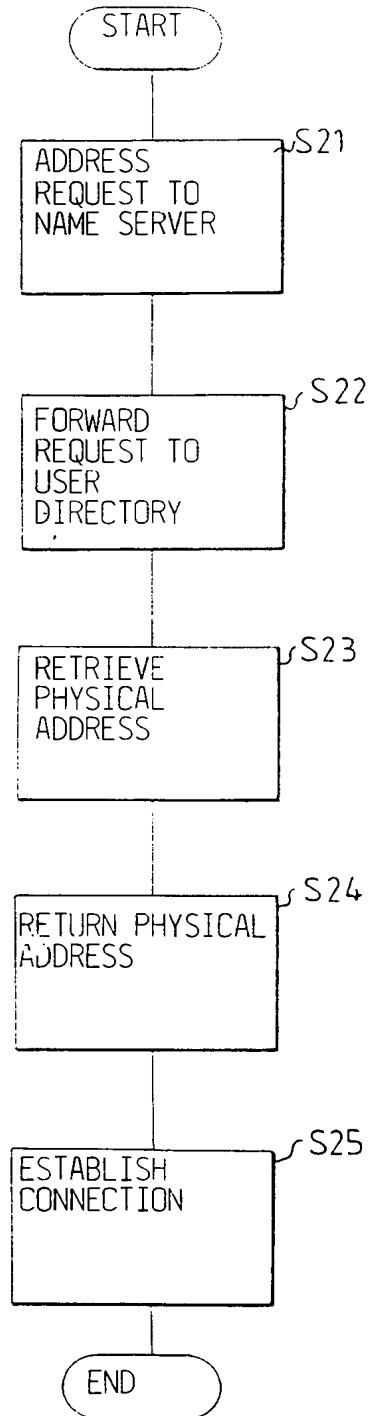


FIG. 4



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification <sup>6</sup> : <b>H04Q</b>	<b>A2</b>	(11) International Publication Number: <b>WO 99/48303</b>
		(43) International Publication Date: 23 September 1999 (23.09.99)

(21) International Application Number: PCT/US99/05900

(22) International Filing Date: 18 March 1999 (18.03.99)

(30) Priority Data:  
09/040,898 18 March 1998 (18.03.98) US

(71) Applicant: CISCO TECHNOLOGY, INC. [US/US]; 170 West Tasman Drive, San Jose, CA 95134 (US).

(72) Inventors: COX, Dennis; 6800 McNeil Drive #828, Austin, TX 78729 (US). MCCLANAHAN, Kip; 3112 Kerbey Lane, Austin, TX 78703 (US).

(74) Agent: SHOWALTER, Barton, E.; Baker & Botts, L.L.P., 2001 Ross Avenue, Dallas, TX 75201-2980 (US).

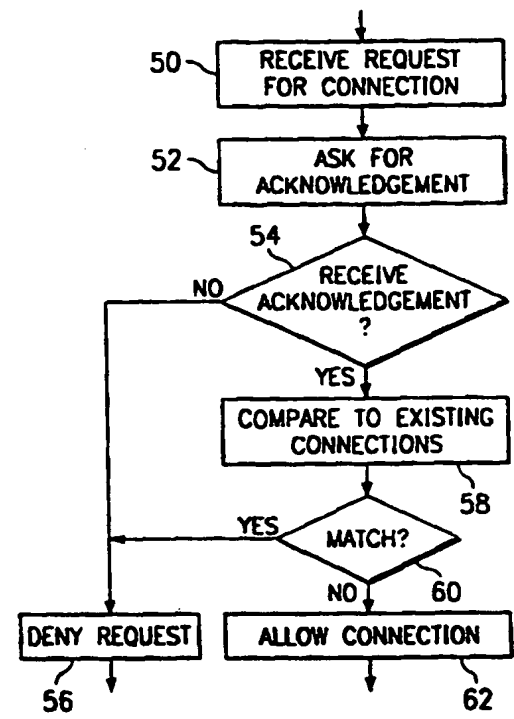
(81) Designated States: AE, AL, AM, AT, AT (Utility model), AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, CZ (Utility model), DE, DE (Utility model), DK, DK (Utility model), EE, EE (Utility model), ES, FI, FI (Utility model), GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK,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

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

(54) Title: METHOD FOR BLOCKING DENIAL OF SERVICE AND ADDRESS SPOOFING ATTACKS ON A PRIVATE NETWORK

(57) Abstract

A method is provided for blocking attacks on a private network (12). The method is implemented by a routing device (10) interconnecting the private network (12) to a public network (14). The method includes analyzing an incoming data packet from the public network (14). The incoming data packet is then matched against known patterns where the known patterns are associated with known forms of attack on the private network (12). A source of the data packet is then identified as malicious or non-malicious based upon the matching. In one embodiment, one of the known forms of attack is a denial of service attack and an associated known pattern in unacknowledged data packets. In another embodiment, one of the known forms of attack is an address spoofing attack and an associated known pattern is a data packet having a source address matching an internal address of the private network (12).



**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	Kazakistan	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						

METHOD FOR BLOCKING DENIAL OF SERVICE AND  
ADDRESS SPOOFING ATTACKS ON A PRIVATE NETWORK

TECHNICAL FIELD OF THE INVENTION

This invention relates in general to communication systems, and more particularly to a method for blocking denial of service and address spoofing attacks on a private network.

BACKGROUND OF THE INVENTION

Corporate and other private networks often provide external access outward and inward through Internet gateways, firewalls or other routing devices. It is important for these routing devices to defend the private network against attackers from the outside as well as to allow access to the private network by authorized users. However there are numerous forms of attack on conventional routing device that can incapacitate the devices and interfere with an associated private network. The problem of keeping unauthorized persons from accessing data is a large problem for corporate and other information service management. Routing devices, such as gateways, firewalls and network routers lack important safeguards to block or prevent attacks. In particular, the number of denial service attacks have risen dramatically in recent years. Further, IP spoofing incidents occur with increasing frequency.

A denial of service attack consists of repeatedly sending requests for connections to different hosts through and/or behind the routing device. Typically, the host will wait for acknowledgment from the requester.

Because a host can only handle a finite number of requests (for example, 1 to n, where n depends on the resources available to the host), the attacker can crash or "flood" a host with requests to the point of  
5 disrupting network service (host/server/port) to users.

Another form of attack is address spoofing which can be used by unauthorized third parties to gain access to a private network. This attack involves the attacker identifying a valid internal network address within the  
10 private network. The attacker then requests access to the private network through the routing device by spoofing that internal network address. Conventional routing devices typically are not sophisticated enough to determine that such a request should be denied (i.e.,  
15 because an external request can not originate from an internal address) and will allow access to the attacker. Address spoofing attacks can be carried out against various types of networks and network protocols such as IPX/SPX, MAC layer, Netbios, and IP.

20 It is therefore advantageous to provide facilities within a routing device that block denial of service, address spoofing and other attacks on an associated private network.

#### 25 SUMMARY OF THE INVENTION

In accordance with the present invention, a method for blocking denial of service and address spoofing attacks on a private network is disclosed that provides significant advantages over conventional network routing  
30 devices.

According to one aspect of the present invention, the method is implemented by a routing device interconnecting the private network to a public network. The method includes analyzing an incoming data packet  
35 from the public network. The incoming data packet is

then matched against known patterns where the known patterns are associated with known forms of attack on the private network. A source of the data packet is then identified as malicious or non-malicious based upon the matching. In one embodiment, one of the known forms of attack is a denial of service attack and an associated known pattern is unacknowledged data packets. In another embodiment, one of the known forms of attack is an address spoofing attack and an associated known pattern is a data packet having a source address matching an internal address of the private network.

A technical advantage of the present invention is the enabling of a routing device to identify a denial of service attack and to block such an attack from tying up the routing device.

Another technical advantage of the present invention is enabling a routing device to identify an address spoofing attack and to block such an attack.

A further technical advantage of the present invention is an ability for the routing device to track information about the attacker to allow preventive measures to be taken.

Other technical advantages should be readily apparent to one skilled in the art from the following figures, description, and claims.

#### BRIEF DESCRIPTION OF THE DRAWINGS

A more complete understanding of the present invention and advantages thereof may be acquired by referring to the following description taken in conjunction with the accompanying drawings, in which like reference numbers indicate like features, and wherein:

FIGURE 1 is a block diagram of an communication system including a routing device and an associated private network;



FIGURE 2 is a flow chart of one embodiment of a method for blocking attacks on a private network according to the present invention;

5 FIGURE 3 is a flow chart of one embodiment of a method for blocking an address spoofing attack according to the present invention; and

FIGURE 4 is a flow chart of one embodiment of a method for blocking a denial of service attack according to the present invention.

10

#### DETAILED DESCRIPTION OF THE INVENTION

FIGURE 1 is a block diagram of an communication system including a routing device 10 and an associated private network 12. Routing device 10 provides a connection between corporate private network 12 and an Internet cloud 14. Routing device 10 can include a gateway, firewall or other device interconnecting private network 12 and Internet cloud 14. In operation, routing device 10 allows internal users within private network 12 to gain access to Internet cloud 14. Routing device 10 also allows external users connected to Internet cloud 14 to gain access to private network 12. A significant and growing problem is that an attacker 16 may try to gain access to or disrupt private network 12 through Internet cloud 14.

Denial of service and address spoofing are two common forms of attack that might be used by attacker 16. In general, a denial service attack is one in which attacker 16 attempts to prevent others from using private network 12. A denial service attack works if routing device 10 spends all of its time processing requests and cannot respond quickly enough to satisfy additional requests. An Address spoofing attack is one in which attacker 16 fakes an internal address to get around or into standard address filtering schemes. According to

35

the present invention, routing device 10 is enabled with a method for blocking these and other types of attacks by analyzing incoming data packets.

Thus, one possible occurrence is that attacker 16 will try to get into private network 12 by spoofing an address that exists inside private network 12. This is intended to allow attacker 16 to gain access and impersonate an internal user. When a packet from attacker 16 reaches routing device 12, an attack blocking component, according to the present invention, will notice that the address matches one that exists within private network 12. Because incoming packets should not be the same as outgoing packets, the attack blocking component can deny access to private network 12 and record the information about the attack for use by the system administrator. Attacker 16 can also try to deny access to all external users by conducting a denial of service attack. This involves attacker 16 flooding private network 12 or routing device 10 by sending an extremely large number of packets. For example, attacker 16 may send 30,000 or more packets. According to the present invention, the attack blocking component of routing device 10 can notice that the first packet is spoofed or that it cannot be acknowledged and ignore all other packets. Further, routing device 10 can use diagnostic detection tools (e.g., trace root, ping, NS lookup) to pinpoint attacker 16 and notify the system administrator. In general, according to the present invention, routing device 10 can be enabled to intelligently analyze incoming packets, match the packets against known patterns for attack strategies and respond accordingly to malicious packets.

FIGURE 2 is a flow chart of one embodiment of a method for blocking attacks on a private network according to the present invention. As shown, an

incoming packet is analyzed by the routing device in step 20. In step 22, the routing device analyzes the incoming packet against known patterns. Based upon this pattern matching, in step 24, the routing device can identify the data packet and its source as malicious or non-malicious. The known patterns used in step 22 can be built using knowledge about various types of attacks. This knowledge can be recorded in the form of patterns that are then stored in a database or other storage device accessible by the routing device. The routing device can then match the analyzed packets against the patterns to determine whether or not some type of attack is being made. If an attack is identified, the routing device can identify the source of that packet as malicious and treat the source accordingly.

In particular, the routing device can implement methods for blocking denial of service attacks and address spoofing attacks as shown, for example, in FIGURES 3 and 4. FIGURE 3 is a flow chart of one embodiment of a method for blocking an address spoofing attack according to the present invention. This method is applicable to address spoofing attacks on various types of networks, but is described specifically with respect to an IP network.

As shown in step 30 of FIGURE 3, the routing device receives a packet. In step 32, the routing device compares the IP address of the packet against known internal IP addresses of the associated private network. In step 34, the routing device determines if the source IP address matches an internal address. If not, in step 36, the routing device routes the packet as appropriate for the packet. However, if the source IP address matches an internal address, then the routing device identifies that there is an attempt to spoof an internal address. The addressed is known to be spoofed because an

internal IP address of the private network cannot be accessing the private network from an external point. Consequently, in step 38, the routing device drops the packet and does not route it to the network. In step 40, the routing device analyzes the packet header for the history of the packet in order to obtain some information about the source of the packet. Then, in step 42, the routing device takes an appropriate defensive action against that packet. For example, the routing device can refuse to accept any more packets from the real source of the packet. In this case, the defensive action can include adding the offending IP address to a cache of IP addresses and then not allowing access to the router device for any IP address in the cached list. Further, the routing device can store information about the attack for later use and for analysis for administrators of the private network. For example, information concerning the packet origination, destination or content can be stored internally to the router device or sent to a syslog server for later analysis.

FIGURE 4 is a flow chart of one embodiment of a method for blocking a denial of service attack according to the present invention. As shown, in step 50, the routing device receives a request for a connection. Then, in step 52, the routing device asks for an acknowledgment from the requestor. In step 54, the routing device checks whether or not an acknowledgment has been received. If one is not received within a specified period of time, the routing device moves to step 56 and denies the request. This denial ensures that the routing device does not churn on pending requests even though acknowledgments have not been received within reasonable amounts of time.

If an acknowledgment is received in step 54, the routing device moves to step 58 and compares the

requested connection to existing connections. Then, in step 60, the routing device determines if there is a match between the requested connection and one of the existing connections. If so, the routing device moves to  
5 step 46 and denies the request. The request is denied because one source should not have more than one connection through the routing device to the private network. If, in step 60, there is no match, then the routing device can allow the connection in step 62. The  
10 method of FIGURE 4 prevents the routing device from being tied up by multiple requests from one source and thereby blocks the denial of service attack.

In general, the method of the present invention can be integrated as a component of a gateway, firewall or  
15 other routing device. In one implementation, the present invention can work off of a variable size cache file that holds network addresses. For blocking spoofing, each incoming address can be held in the cache file and checked to see if the incoming address matches an network  
20 address that is on the private network. If the incoming address matches, then the request can be denied. Also, a message can be sent to a system log which, rather than being written to a file, can be written to a console to prevent the log from getting overloaded and crashing the  
25 routing device. Further, an optional E-mail message or page can be sent to a specified address or number in the case of an attack. If an attack happens more than once on the same address in the span of a certain period of time (for example, five minutes), then the number of  
30 messages can be limited to prevent overloading of the E-mail or paging service. An optional shutdown mechanism can also be in place that will enable the routing device to automatically shut down certain services if attacks continued.

Denial of service attacks are generally easier to trace. However, when such an attack is also spoofed, the problem becomes very difficult to stop. According to the present invention, an incoming address can be checked  
5 against the cache file and a quick search can be performed to see if the address is already in a list of pending addresses. If so, the request packet can be discarded. An address is removed from the list if a successful acknowledge packet is sent back or a variable  
10 time limit is reached. The number of matching addresses that are allowed in the list can be a variable set by the system administrator.

Although the present invention has been described in detail, it should be understood that various changes,  
15 substitutions and alterations can be made thereto without departing from the sphere and scope of the invention as defined by the appended claims.

WHAT IS CLAIMED IS:

1. A method for blocking attacks on a private network implemented by a routing device interconnecting the private network to a public network, comprising:

5 analyzing an incoming data packet from the public network;

matching the incoming data packet against known patterns, the known patterns associated with known forms of attack on the private network; and

10 identifying a source of the data packet as malicious or non-malicious based upon the matching.

2. The method of Claim 1, wherein one of the known forms of attack is a denial of service attack and an associated known pattern is unacknowledged data packets.

3. The method of Claim 1, wherein one of the known forms of attack is an address spoofing attack and an associated known pattern is a data packet having a source address matching an internal address of the private network.

4. The method of Claim 1, wherein the public network is the Internet.

25 5. The method of Claim 4, wherein the routing device is a firewall providing access to the Internet.

6. A method for blocking an address spoofing attack on a private network implemented by a routing device interconnecting the private network to a public network, comprising:

5 receiving an incoming data packet from the public network;

comparing a source address of the data packet against known internal addresses of the private network;

10 determining if the source address matches a known internal address;

if there is no match, routing the data packet to the private network;

if there is a match, dropping the data packet.

15 7. The method of Claim 6, further comprising, if there is a match, analyzing a header of the data packet for a history of the data packet and taking defensive action against the data packet based upon the history.

20 8. The method of Claim 7, wherein the defensive action comprises refusing to accept any more data packets from a real source of the data packet.

25 9. The method of Claim 7, wherein the defensive action comprises storing information about the data packet for use and analysis by a system administrator.

10. The method of Claim 6, wherein the public network is the Internet.

30

11. The method of Claim 10, wherein the routing device is a firewall providing access to the Internet.



12. A method for blocking a denial of service attack on a private network implemented by a routing device interconnecting the private network to a public network, comprising:

5 receiving a request for a connection from the public network;

requesting an acknowledgment from an initiator of the request;

10 determining whether an acknowledgment has been received;

if an acknowledgment is not received, denying the request;

if an acknowledgment is received, comparing the request to existing connections;

15 if there is a match between the request and an existing connection, denying the request;

if there is not match between the request and an existing connection, allowing the connection and routing packets to the private network.

20

13. The method of Claim 12, wherein the public network is the Internet.

25 14. The method of Claim 13, wherein the routing device is a firewall providing access to the Internet.

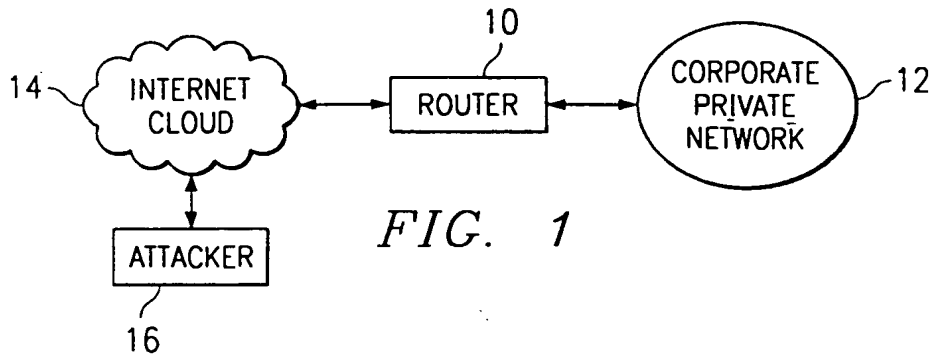


FIG. 1

FIG. 2

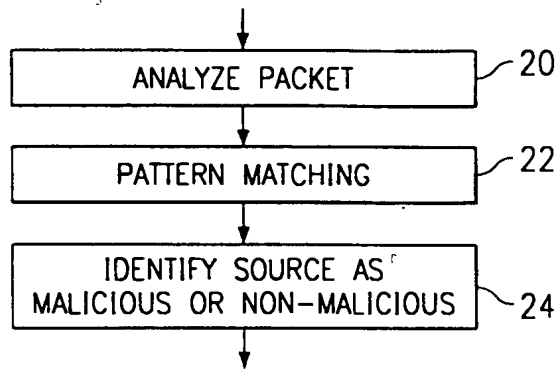


FIG. 3

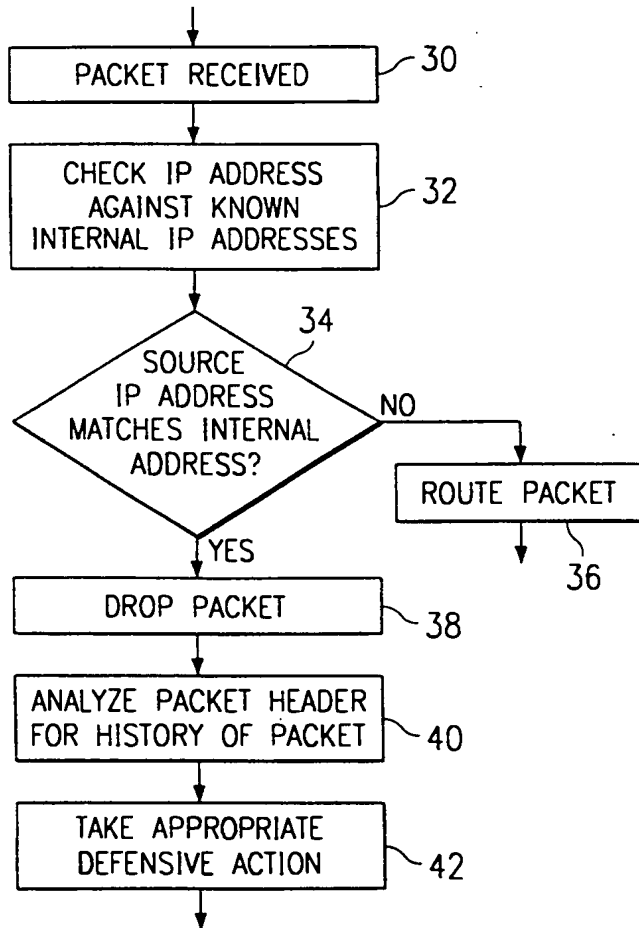
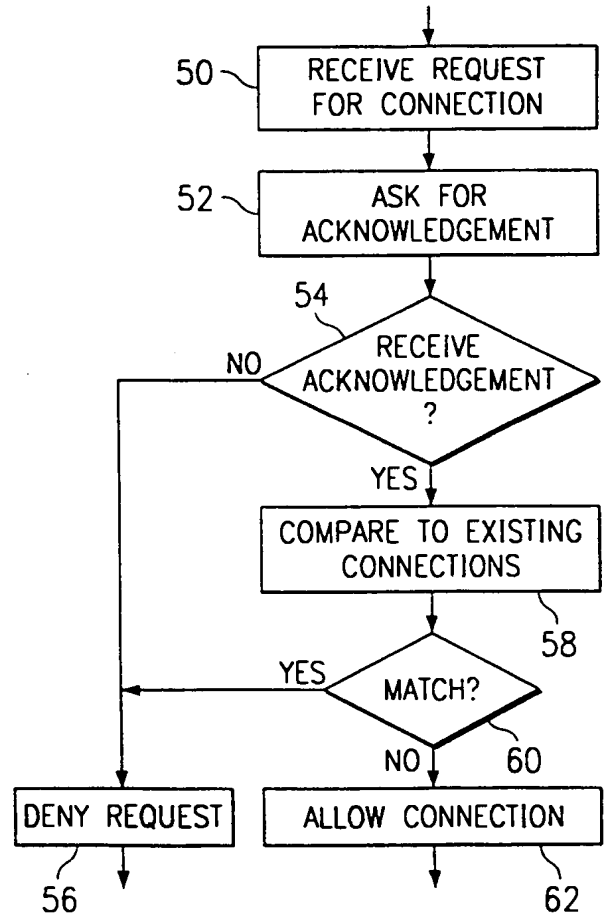


FIG. 4

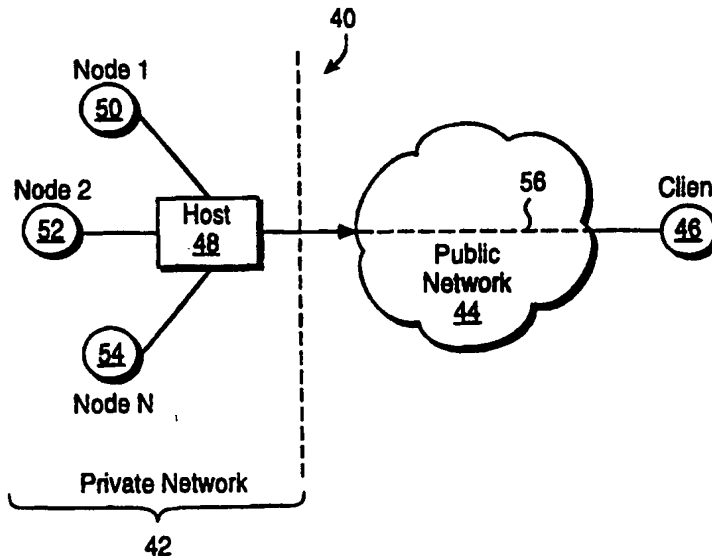




INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(51) International Patent Classification <sup>6</sup> : G06F 13/00, H04L 9/30</p>	<p>A1</p>	<p>(11) International Publication Number: <b>WO 99/38081</b> (43) International Publication Date: 29 July 1999 (29.07.99)</p>
<p>(21) International Application Number: PCT/US99/01583 (22) International Filing Date: 26 January 1999 (26.01.99) (30) Priority Data: 09/013,122 26 January 1998 (26.01.98) US (71) Applicant: ASCEND COMMUNICATIONS, INC. [US/US]; One Ascend Plaza, 1701 Harbor Bay Parkway, Alameda, CA 94502 (US). (72) Inventors: PAULSEN, Gaige, B.; 513 Springvale Road, Great Falls, VA 22066 (US). WALKER, Amanda; 2230 Cedar Cove Court, Reston, VA 20191 (US). (74) Agent: LOHSE, Timothy, W.; Gray Cary Ware &amp; Freidenrich, 400 Hamilton Avenue, Palo Alto, CA 94301 (US).</p>		<p>(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, GM, HR, HU, ID, 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, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, 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, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).</p> <p><b>Published</b> <i>With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i></p>

(54) Title: VIRTUAL PRIVATE NETWORK SYSTEM AND METHOD



(57) Abstract

A system and method for remote users to access a private network (42) having a first communications protocol via a public network (44), such as any TCP/IP network having a second different communications protocol, in a secure manner so that the remote user appears to be connected directly to the private network (42) and appears to be a node on that private network (42). A host (48) connected to the private network (42) may execute a host software application which establishes and provides a communications path for secure access of the remote client computer (46). An encrypted data stream may be communicated between the host (48) and the client (46) representing traffic and commands on the network.

**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						

VIRTUAL PRIVATE NETWORK SYSTEM AND METHODBackground of the Invention

This invention relates generally to apparatus and methods for accessing computer networks and in particular to establishing a secure connection between a remote computer and a private computer network using a public computer network.

In the past, organizations and companies have used private (internal) computer data networks to connect its users to each other. These private networks are not accessible to the public and permit sensitive data to be transferred between users within the company. However, due to the increasing numbers of people who need access to the private computer data network and the disparate locations of these people, there are several disadvantages of these conventional private computer networks.

As the number of people in a company grows, the workforce becomes more dispersed among different locations and there are more employees who are mobile, such as salespeople who travel around a region of the United States. For example, some employees may telecommute which requires dial-up access to the private computer data network. The dispersed workforce and the mobile workforce make a private computer data network unmanageable because this mobility requires at least two network connections for each user. In addition, since cellular telephone access has also become more available, additional connections to the network for this access is needed. In addition, full-time telecommuters dramatically increase the number of permanent "remote offices" a company must interconnect which further complicates

the private computer data network administration and topology. In addition, as companies increase in size, due to acquisitions, mergers and expansion, the private computer data network must support more remote offices and more network nodes. Thus, as a organization expands, the private computer data network of the organization becomes unwieldy and unmanageable.

Recently, it has become necessary and desirable to permit employees of the company to interact "on-line" with customers and suppliers. This function adds a new dimension of complexity to the private computer data network since multiple private computer data networks must be interfaced together in a delicate balance of integration while maintaining some isolation due to security concerns. The individual networks that are being integrated together typically use different data transfer protocols, different software applications, different data carriers and different network management systems. Thus, interfacing these private computer data networks is a major challenge.

There is also a desire to consolidate and simplify the user interface to the computer network as well as to the software applications being executed by the computer network since it is often difficult to keep on top of each new software application. Thus, the costs of implementing and maintaining a private computer data network is high and is expected to increase in the future as the factors set forth above continue to drive up the costs of the private computer data networks. These high costs are compounded by the high costs for long distance telephone charges for leased lines and switched services. The number of support staff necessary to manage the complex

topologies of these private computer data networks also further increases the costs to manage the private computer data networks. In addition, software applications which execute over the private network require separate backup equipment which further complicates the topology and increases the cost of the private computer data network. Thus, the costs and complexity of these private computer data networks are continuing to spiral upwards and there is no foreseeable end in sight.

A typical private computer data network may be used by a organization for some of its communications needs and may carry exclusively data traffic or a mix of voice/video and data traffic. The private computer data network may be constructed with a variety of wide area network (WAN) services that often use the public switched telephone network (PSTN) as a communications medium. A typical network may use high speed leased lines that carry voice, facsimile, video and data traffic between major facilities. These leased lines may include integrated services digital network (ISDN) lines or conventional T1 telephone lines. Because these leased lines are point-to-point connections, a mesh topology is necessary to interconnect multiple facilities. In addition, each leased line must be dedicated to a particular interconnection. A remote office may use switched services over the PSTN, such as ISDN or frame relay. For individual mobile employees, an analog modem may be the best solution for connection to the private computer data network. The private computer data network with all of these different connections, therefore, is very expensive to implement and maintain for the reasons set forth above.

A virtual private network (VPN), on the other hand, may offer the same capabilities as a private computer data network, but at a fraction of the cost. A virtual private network is a private data network that uses a public data network, instead of leased lines, to carry all of the traffic. The most accessible and less expensive public data network currently is the Internet which can be accessed worldwide with a computer and a modem. An Internet-based virtual private network (VPN) is virtual because although the Internet is freely accessible to the public, the Internet appears to the organization to be a dedicated private network. In order to accomplish this, the data traffic for the organization may be encrypted at the sender's end and then decrypted at the receiver's end so that other users of the public network can intercept the data traffic, but cannot read it due to the encryption.

A VPN can replace an existing private data network, supplement a private data network by helping relieve the load on the private data network, handle new software applications without disturbing the existing private data network or permit new locations to be easily added to the network. A typical VPN connects one or more private networks together through the Internet in which the network on each side of the Internet has a gateway and a leased line connecting the network to the Internet. In these typical VPNs, the same protocol for each private network, such as TCP/IP, is used which makes it easier to communicate data between the two networks. To create the VPN, a secure communications path between the two gateways is formed so that the two private networks may communicate with each other. In this configuration, however, each network is aware that the other network is at some other location and is



connected via a router. As an example, if a company has a central private network in California and a remote office in Hong Kong, these two private networks may be connected via the VPN which reduces long distance telephone call charges. However, if a single individual is traveling in Hong Kong and want to connect to the private network in California, the individual must incur long distance telephone charges or, if there is a remote office in Hong Kong, then the entire private network must be connected via the VPN to the California private network to communicate data. In addition, with the conventional VPN described, the individual in Hong Kong is aware that he is connected to the Hong Kong network which is in turn connected, via the gateway and the VPN, to the network in California so that the person in Hong Kong cannot, for example, easily use the network resources of the California network, such as a printer.

Thus, a conventional VPN requires the expense of a leased line and a gateway at each end of the VPN and cannot adequately address the needs of a individual who needs access to the private network. In addition, these conventional VPNs cannot easily connect networks which have different networking protocols. In addition, these conventional VPNs cannot be easily used for connecting an individual who needs remote access to the private network since the entire network with a gateway is needed.

Thus, the invention provides a virtual private network (VPN) which avoids these and other problems with conventional VPNs and it is to this end that the invention is directed.

### Summary of the Invention

In accordance with the invention, a virtual private network system is provided which connects a private data network and a remote client which does not require expensive leased lines or gateways to establish a secure communications path. The system also permits an individual to access the private data network without incurring any long distance telephone charges. In addition, the system permits a private data network and remote client that use one communications protocol to communicate with each other over a public data network that uses a different communications protocol. The system also permits an individual to easily connect to the private data network without a remote private network and the individual appears to be a node on the private network, once connected, so that the individual may access any resources on the private data network.

In accordance with the invention, a system and method for forming a communications path between a public access network and a private access network where the two networks have substantially incompatible transmission protocols is provided. The method comprises establishing a secure communications path over the public access network between a host computer connected to the private network and a remote client computer, encrypting data and commands of the host computer and the client computer, and formatting the encrypted data and commands into a format compatible for transmission over the public access network. The formatted data and commands are then transmitted over the public access network. Once the formatted data and commands has reached its destination, it is decrypted to establish the client

computer as a virtual node on the private network. In accordance with another aspect of the invention, a data structure for communicating data for a private data network having a first communications protocol over a public access network having a second communications protocol is provided.

### Brief Description of the Drawings

Figure 1 is a block diagram illustrating a conventional virtual private network;

Figure 2 is a block diagram illustrating a virtual private network in accordance with the invention;

Figure 3 is a block diagram illustrating more details of the host computer of Figure 1; and

Figure 4 is a flowchart illustrating a method for establishing a virtual private network and communicating secure data over the virtual private network in accordance with the invention.

### Detailed Description of a Preferred Embodiment

The invention is particularly applicable to a system and method for providing a virtual private network which permits remote users to access a private network, such as an AppleTalk network, via a public TCP/IP network, such as the Internet, in a secure manner as if the remote user was one of the nodes on that private network. It is in this

context that the invention will be described. It will be appreciated, however, that the system and method in accordance with the invention has greater utility. Before describing the invention, a brief description of a conventional virtual private network (VPN) will be provided.

Figure 1 is a block diagram illustrating a conventional virtual private network (VPN) 20. The VPN includes a first private network 22 and a second private network 24 connected together through a public computer network 26, such as the Internet. The communications protocols for the first and second private networks as well as the public network may be the standard Transmission Control Protocol/Internet Protocol (TCP/IP). Thus, the communications protocols for the private networks are the same as the public network. Each private network 22, 24 includes a gateway 28, 30 which interfaces between the respective private network and the public network. Each gateway encrypts data traffic from the private network which is going to enter the public network and decrypts encrypted data received from the public network. In normal operation, a secure communications path 32, referred to as a tunnel, is formed over the public network that connects the first and second private networks through the respective gateways. The combination of the two private networks and the tunnel over the public network forms the virtual private network (VPN). The VPN is virtual since it is actually using a public network for the connection, but due to the encryption both private networks believe that they have a private network over which data may be sent. For example, a node 34 of the first private network 22 may send data which is encrypted by the gateway 28 through the tunnel 32, and the data is received by the

second gateway 30 which decrypts the data and routes it to the appropriate node in the second private network. This conventional VPN, however, does not adequately provide an individual remote user with a system for remotely accessing the private network because the conventional VPN connects two networks with a tunnel and would require the individual to be connected to one of the private networks to utilize the VPN. In addition, this conventional VPN does not connect a remote individual directly to the private network so that a remote user with a VPN connection cannot directly access resources, such as a printer, connected to the private network. This conventional system also does not handle computer networks which have different communications protocols. Now, the virtual private network system in accordance with the invention will be described which overcomes these problems with a conventional VPN.

Figure 2 is a block diagram illustrating a virtual private network (VPN) 40 in accordance with the invention. The VPN may include a private network 42 which communicates data using a first communications protocol, a public network 44 which communicates data using a second communications protocol, and a client node 46 that is connected for secure communications to the private network 42 through the public network 44 as described below. The private network 42 may be any type of computer network, such as an AppleTalk network. The public network may be any type of publicly accessible computer network such as the Internet.

The private network 42 may include a host computer 48, and a plurality of network nodes, such as a first node (NODE\_1) 50, a second node (NODE\_2) 52, and

an nth node (NODE\_N) 54 which are all connected to the host computer. In normal operation any node of the private network may share resources with any other node on the network. For example, any node of the private network may share a printer which is attached to the private network. The host computer 48 establishes a secure communications path 56, referred to as a tunnel, through the public network 44 with the remote client 46 by negotiating the communications protocol with the client 46 and authenticating the identity of the client. Once the secure tunnel has been established between the private network 42 through the host computer 48 and the public network 44 with the remote client 46, the remote client is treated as a node of the private network and uses the communications protocol of the private network even though the public network uses a different protocol. Thus, the remote client 46 may access resources connected to the private network, such as a printer, as if the remote client were directly connected to the private network. Therefore, with the VPN in accordance with the invention, the various connections between the remote client and the private network are transparent to the user of the remote client since the user can use the private network in any manner that a user directly connected to the private network can.

With the VPN in accordance with the invention, a gateway at each end of the virtual private network is not required. In addition, data traffic for the private network which has a first data communications protocol may be communicated over a public computer network which has a different communications protocol. In particular, the system encapsulates the data destined for the private data network having a first

protocol in a data packet that may be sent over the public network, as described in more detail below. Thus, once the secure virtual private network connection has been established, the remote client may interact with the private network as if the remote client was directly connected to the private network. The virtual private network in accordance with the invention also permits an individual remote user to easily establish a connection with a distant private network without the need for a remote private network and a leased line or long distance telephone charges. Now, more details about the host computer 48 and the remote client 46 in accordance with the invention will be described.

Figure 3 illustrates more details of the host computer 48 and the remote client 46 in accordance with the invention. The host computer 48 may include a central processing unit (CPU) 60, a memory 62 and a host 64 stored in the memory 62. The host may be a software application which is executed by the CPU 60 of the host computer. When a remote client contacts the private network 42 to establish a secure connection, the host 64 may negotiate and establish the secure virtual connection to the remote client 46, as described below. Once the secure connection has been established, the host 64 accepts unencrypted data from the private network, combines the data with a header containing information about the protocol of the private data network, encrypts the data and the header, and communicates the encrypted data and header, over the secure communications path, to the remote client. The host also receives encrypted data with a header from the remote client, decrypts the data and the

header, and passes the data traffic onto the appropriate node in the private network based on the header information, as described below.

Similarly, at the remote client 46, a client software application 66 stored in a memory 68 in the client computer 46 is executed by a central processing unit (CPU) 70 in the client computer 46. The client 66 negotiates and establishes the secure communications path with the host computer, combines the data with an appropriate header, encrypt the data traffic and the header destined for the client computer, and communicate the encrypted data to the host computer. The client also receives encrypted data traffic from the host computer, decrypts it, and passes the data traffic onto other software application which are being executed by the CPU 70. Thus, the virtual private network in accordance with the invention is software application based so that expensive hardware, such as a gateway and leased lines, are not necessary. The software applications also permit the data between the client and host, which have a first communications protocol, to be communicated over a public computer network which has a second different communications protocol. Now, a method for establishing and communicating data traffic over the virtual private network in accordance with the invention will be described.

Figure 4 is a flowchart illustrating a method 100 for establishing and communicating data over the virtual private network in accordance with the invention.

An example of the phases and data formats for the communications between an AppleTalk network host and an AppleTalk remote client over the Internet will be described below, but the invention is not limited to that example and may be used to



communicate data between any hosts and remote clients having a different communications protocol than the public data network. To begin the method, the remote client may request a connection to the host by any conventional method.

In step 102, once the initial unsecure connection has been established between the host and the client, a protocol negotiation phase occurs in which the host and the client negotiate the parameters that will govern the subsequent communications between the host and the client. The negotiated parameters may include the protocol version, the compression level, and the encryption technique. Each of these parameters has a default setting that must be available for either the host or the remote client to request so that there is a minimum set of functionality which may be implemented. To ensure backwards compatibility of any host or remote client, each host or client will implement at least a first protocol version so that there is backwards compatibility for future versions. These parameters will be described in more detail below. In addition, for the encryption parameter, each host and remote client must be able to support both data encryption standard (DES) type encryption as well as some form of non-DES encryption to permit communications between hosts and clients that are licensed for use within the United States as well as outside of the United States. The invention may use a plurality of different well-known non-DES encryption methods and these encryption methods will not be described here. The protocol negotiation phase is started when the connection is established and is initiated by the remote client sending the host a Protocol Request in which it communicates which protocol version it would like to use and any options, such as the encryption, that it would like to use. The host

then sends the remote client a Protocol Response verifying the protocol version number and any options. An example of the data formats of the Protocol Request and Protocol Response in the context of an AppleTalk network are provided below.

Once the protocol has been negotiated, it is determined, in step 103, if an optional session key negotiation phase 104 is going to occur. In the first protocol version, the session key negotiation phase is optional, but later versions of the protocol will require the session key negotiation phase. The session key negotiation phase is thus entered if a session key bit in the Protocol Request is set during the protocol negotiation phase. During the session key negotiation phase, data is exchanged between the host and remote client for the purpose of setting up an encryption key that is used for the remainder of the communication. In a preferred embodiment, a well known Diffie-Hellman key exchange method is used, but any other conventional key exchange method may be used. If the session key phase and the Diffie-Hellman key exchange method are not being used, the encryption key is chosen during an authentication phase 106, as described below. The data communicated during the session key negotiation phase may include a length word indicating the length of the data and the data. The data flow is bi-directional and is completed when the host and the remote client have agreed on a session key. If the system determines, in step 105, that a session key has been established, an authentication phase 106 is entered. In the event that a session key is not successfully negotiated during the session key negotiation phase, the method proceeds to a teardown phase 110 in which the

communications between the host and the remote client is terminated and the methods ends.

During the authentication phase 106, the remote client and the host negotiate what type of authentication is used for the communications and then provides challenges and responses to authenticate the identity of the remote client. Due to the wide variety of security requirements and methods, the host must, at a minimum, send a request with at least one default authentication type identifier and an associated challenge. However, if the host has the ability to use more than one authentication method, then the host may send the remote client, in a Authentication Request, more than one authentication type identifier and their associated challenges as described below. Thus, to start the authentication phase, the host may communicate an authentication request, as described below, to the remote client. The authentication request may include one or more authentication type/authentication challenge data pairs. In response to the authentication request, the remote client communicates an authentication response back to the host which includes exactly one authentication type/response data pair. If the host sends more than one authentication type/challenge pair, the remote client selects a particular authentication type and responds with the authentication type/response pair for only that particular authentication type. An example of the types of authentication methods is set forth below.

If the session key negotiation phase is not used, then, during a successful authentication phase, an implicit session key may be generated by the remote client. In a preferred embodiment, the session key may be generated by the following steps.

First, a Unicode string containing the password from the client is concatenated with the challenge from the authentication request. Next, a SHA-1 hash value over the resultant concatenated data is calculated and the initial bytes of the hash value may then be used as the session key which may be communicated back to the host.

In response to the authentication response, the host determines if the response was successful or not in step 107. If the response was successful (i.e., an appropriate response to the challenge was received which verifies the identity of the remote client), a success data structure is sent to the remote client and the method goes to an established phase 108, as described below. If the response was not successful (i.e., an appropriate response to the challenge was not received so that the identity of the remote client can not be verified), then an error code is sent to the remote client and the teardown phase 110 is entered.

During a typical successful secure communications session, most of the time is spent in the established phase 108 in which encrypted data including the header is communicated between the remote client and the host. The header, as described below, contains information required by the communications protocol of the private network (i.e., the host and the remote client) to appropriately route data. Thus, the communications protocol information for the private network is embedded in the encrypted data packet so that the data destined for the private data network may be communicated over the public network having a different communications protocol. For each piece of encrypted data sent during the established phase, the data may be preceded by a length and flag word which contains the length of the data in bytes and

six bits of flags. Since the data is typically sent over a TCP/IP based public network, a PUSH bit in the flag bits must be set to accelerate the processing of the transactions once a complete unit of data has been received.

If an unsuccessful session key negotiation, an unsuccessful authentication, or the end of the established phase occurs, then the tear down phase 110 is begun. During the tear down phase, there is no data traffic between the remote client and the host and the communications channel is forcibly closed by either the remote client or the host.

During the teardown phase, when one side shuts down the communications channel, an acknowledgment from the other side may consist of shutting down the connection from that side as well so nothing remains of the communications path. After the teardown phase, the method has been completed. The method, therefore sets up a communication session as needed and then tears down the communications path once the communications have been completed.

Now, an example of the data formats for a system and method in accordance with the invention for communicating AppleTalk data between a remote client and a host over a TCP/IP public network, such as the Internet, will be described. As described above, the virtual private network in accordance with the invention may connect any private network having a first communications protocol to a public network having a second different communications protocol securely to permit remote users to access the private network in a secure manner wherein the remote user appears to be one of the nodes in the private network. In this example, the data formats for each of the communications phases are set forth and explained. For each different

private data network with a different communications protocol, these data formats will vary slightly. The bytes of these data formats are sent across the network connection path over the Internet using a Network Byte Order protocol in which the most significant byte is communicated first.

To better understand the utility of the invention in the context of a connection between an AppleTalk private network and a AppleTalk remote client over the TCP/IP-based Internet, the differences between the protocol for the AppleTalk network and the Internet will be described before describing the data formats for this example. AppleTalk is a proprietary suite of networking protocols which is designed for plug-and-play operation whereas TCP/IP is designed to be administered. In particular, the Internet or any other TCP/IP network has been designed such that each node on the Internet is permanently assigned a unique IP address by a quasi-governmental entity. AppleTalk, on the other hand, assigns a node or device number to a node or device when the nodes or devices are actually placed on the network to provide the plug-and-play functionality. Therefore, the two networking protocols assigns network numbers in different manners.

AppleTalk also has a smaller network number range than the Internet and is not centrally administered so that AppleTalk networks can not be arbitrarily connected to each other without substantial planning to ensure that the connected nodes do not have overlapping network numbers. In AppleTalk, there is also a service location protocol that permits users to locate servers and network devices, such as printers, and AppleTalk has the concept of a "zone" which provide a level of scoping for the service

location protocol. In order to access the network services on a particular network, you must have access to the particular zone. One advantage of the invention is that the remote client can avoid the network number and zone addressing by connecting the user of the remote client directly on the AppleTalk network as a virtual node in the zone of the host computer in a secure manner. Thus, once the user of the remote client is securely connected to the AppleTalk network over the Internet, the user sees all of the devices of the AppleTalk network, such as printers and file servers, in a familiar manner which permits them to access any device on the private network. Now, an example of the data formats for the invention when connecting an AppleTalk private network and a remote client over the Internet will be described.

During the protocol negotiation phase, as described above, there is a protocol request from the host and a protocol response from the remote client. The data formats of the protocol request and protocol response are set forth in Tables 1 - 3 below.

**Table 1- Protocol Request**

Byte Offset	Width	Contents
0	2 bytes	Total Bytes: Total number of bytes in the transaction (excluding this field)
2	2 bytes	Protocol Version: Protocol version requested
4	2 bytes	Options Bytes: Length of the following data bytes
6	specified by the previous field	Options: Any options to be requested

In version 1 of the protocol, the Total Bytes in the protocol request is 6, the Protocol Version is 1, the Options Bytes is 2, and the Options field will contain two bytes which represent 16 individual flag bits. For other versions of the protocol, these fields may contain different values. The meanings of the flag bits in the protocol request data format are set forth below in Table 2.

**Table 2 - Option Flag Bits Format**

Byte Location	Meaning
15-2	Reserved for future options. These must be 0 in the first version of the protocol.
1	Use session key negotiation. If this bit is set, the requester wants to use the Session Key Negotiation phase. If not, it is requested that the phase be omitted.
0	Use DES encryption. If this bit is set, the requester wants to use DES encryption. If it is not set, an alternate encryption method is to be used.

Thus, using the options fields in the first version of the protocol, the session key negotiation phase and the type of encryption may be chosen. With future versions of the protocol, additional options may be selected. The format of the Protocol Response will now be described with reference to Table 3.



**Table 3 - Protocol Response**

Byte Offset	Width	Contents
0	2 bytes	Total Bytes: Total number of bytes in the transaction (excluding this field)
2	2 bytes	Protocol Version: Protocol version to be used
4	2 bytes	Options Bytes: Length of the following data bytes
6	specified in Options Bytes	Options: Any options that are in use

The protocol response data uses a similar data format to the Protocol request, and contains the same data. However, when returned from the Host to the Client in the Protocol Negotiation phase, this data establishes the actual communication protocol and data format to be followed during the Established phase. The data communicated during the protocol negotiation phase is unencrypted since the secure communications path has not yet been established. Now, the data formats for the optional session key negotiation phase will be described.

The session key negotiation phase, as described above, may include the session negotiation request and the session negotiation response. The data format for both of these pieces of data are identical for all responses and requests. In particular, each data packet contains a 2 byte length field followed by the data used for the negotiation of the session key for use in the well-known Diffie-Hellman key exchange method. Once

again, the data is sent unencrypted since no secure communications channel has been established.

The authentication phase, as described above, may include an authentication request and an authentication response, whose data formats are set forth below in Tables 4-6.

**Table 4 - Authentication Request**

Byte Offset	Width	Contents
0	2 bytes	Total Bytes: Total number of bytes in the transaction (excluding this field)
2	2 bytes	Authentication Type: Identifies the authentication type
4	2 bytes	Challenge Bytes: The number of bytes that follow for the challenge (0 or more)
6	specified in Challenge Bytes	Challenge: The data for the challenge in the authentication. The exact contents vary based on the authentication method.

As described above, this data must contain at least one authentication type/challenge pair, but may contain more than one authentication type/challenge pair if the host supports more than one type of authentication. In version 1 of the protocol, the Authentication Type must be one of types set forth in Table 5.

**Table 5 - Authentication Types**

Authentication Type	Description
0	No authentication.
1 - Clear Text authentication.	No bytes follow for the challenge (may not be supported by any server). A 0-length response is expected by Hosts which request this method.
2	There is no challenge (may not be supported by any server). A 0-length challenge is sent, and the Host expects the user name and password of the client to be sent in clear text.
3	Challenge-Handshake Authentication Protocol (CHAP) - There is an 8-byte encrypted challenge. A 24-byte response is expected by the Host. This method <b>MAY</b> be supported by Hosts and Clients.
	NT RAS compatible CHAP - There is an 8-byte encrypted challenge. A 16-byte response is expected by the Host. This method <b>MUST</b> be supported by all Hosts and Clients.

As shown, there are several different authentication methods which may be used. The default authentication method is the NT RAS compatible CHAP with an 8 byte challenge and a 16 byte response. Again, since no secure communications path has been established, this data is sent unencrypted. Now, the data format of the authentication response is described with reference to Table 6.

**Table 6 - Authentication Response**

Byte Offset	Width	Contents
0	2 bytes	Total Bytes: Total number of bytes in the transaction (excluding this field)
2	2 bytes	Authentication Type: Identifies the authentication type
4	2 bytes	Response Bytes: Number of bytes in the authentication response
6	specified in Response Bytes	Response: The data which responds to the Challenge. The length and exact contents vary based on the authentication type and the challenge.
Response Bytes +6	up to 32	User Name: The clear text version of the user name. The name is terminated by the end of the data (based on Total Bytes).

This authentication response data must contain exactly one response to one of the Authentication Type/Challenge pairs in the preceding Authentication Request. The Client may choose which of the pairs to respond to if more than one appears in the Authentication Request. The User Name in the response specifies which user is requesting access and is used in conjunction with the Response to authenticate the user.

This data is also sent unencrypted, unless a session key has been negotiated previously in the Session Key Negotiation phase, in which case it is encrypted.

During the initial portion of the established phase, there may be a success data structure or a failure data structure and then during the actual established phase there may be a data structure for data communicated to the remote client and a data structure for data communicated to the host. These data structures are set forth below

in Tables 7 - 11. If a successful secure connection is established, then a connections success data structure, as set forth in Table 7 is sent to the remote client.

**Table 7 - Connection Success**

Byte Offset	Width	Contents
0	2 bytes	Total Bytes: Total number of bytes in the transaction (excluding this field)
2	2 bytes	Success: always contains 0
4	2 bytes	Client Network Number: the assigned network number for the Client
6	1 byte	Client Node Number: the node number of the Client for the nearest AppleTalk Bridge
7	1 byte	Bridge Node Number: the node number of the nearest AppleTalk Bridge
8	2 bytes	Bridge Network Number: the network number of the nearest AppleTalk Bridge
10	2 bytes	Network Range Start: The start of the network range for the AppleTalk network connected to the Host
12	2 bytes	Network Range End: The end of the network range for the AppleTalk network connected to the Host

This successful connection data is sent by the Host when a connection is successfully established between the Client and the Host. It contains the data necessary to configure the AppleTalk connection on the Client side. The connection success data structure thus contains the embedded information about the private data network communications protocol so that private network data may be communicated over the public network which has a different communications protocol. For example, the Bridge Node Number and Bridge Network Number specify AppleTalk specific

network information, such as the AppleTalk default Bridge (or Router) on the network that the Host resides on. This embedded private data network information permits the client and the host to format their data formats, as set forth in Tables 10 and 11, for the particular connection to the particular type of private data network. This embedded information also permits the remote client to be treated as a virtual node of the AppleTalk network so that any devices, such as printers or file servers, on the private network may be accessed by the user of the remote client. The connection success data structure is sent unencrypted, unless a session key has been negotiated in the Session Key Negotiation phase, in which case it is encrypted. The connection failure data format is set forth in Table 8.

**Table 8 - Connection Failure**

Byte Offset	Width	Contents
0	2 bytes	Total Bytes: Total number of bytes in the transaction (excluding this field)
2	2 bytes	Error Code: Contains the error code sent by the Host

This connection failure data is sent by the Host when a connection cannot be successfully established between the Client and the Host. It contains a length field and only one other field, an Error Code field. The error code field contains an optional representation of why the connection failed. As a default, the host may always return an "Undefined Error" message, which gives no information on why it rejected the request. An example of the error codes are set forth below in Table 9.

**Table 9 - Error Codes**

Error Code	Description
1	Unsupported Authentication. This is returned when the Client sent an Authentication Response for an Authentication type which was not in the Authentication Request.
2	Failed Authentication. The specified User Name and Response were not valid for the authentication type and Challenge specified. Note: This could be any kind of error from unknown user to invalid password.
3	No Free Ports. The Host does not have any available ports.
4	Already Logged On. The specified User Name is already in use on this server, and multiple logins of the same user are disallowed.
0xFFFF	Undefined Error. An error prevented the connection from succeeding.

This error data is sent unencrypted, unless a session key has been negotiated in the Session Key Negotiation phase, in which case it is encrypted. If the connection failure data structure is sent, then the communications session ends. If a successful connection is established, then data is communicated between the host and the client using the data format for established data to the remote client as set forth in Table 10.

**Table 10 - Established Data (To Client)**

Byte Offset	Width	Contents
0	2 bytes	Length and Flags: contains the length of the following data in the low 10 bits and a set of reserved flags in the upper 6 bits.
2	2 bytes	Source Network: the network number that sent the packet.
4	1 byte	Source Node: the node number that sent the packet.
5	1 byte	Destination Socket: the socket that the packet is being sent to.
6	1 byte	Source Socket: the socket that sent the packet.
7	1 byte	Type: the AppleTalk type of the packet.
8	Specified by the Length	Payload: the data from the original packet.

This data is sent from the Host to the Client during the established phase. As shown, the data contains the AppleTalk specific information to route the data packet to the client. This data is always encrypted. The basic format (with no flags set) contains data from one packet on the AppleTalk network that is destined for the Client. An example of the data format for data from the remote client to the host is set forth in Table 11.



**Table 11 - Established Data (From Client)**

Byte Offset	Width	Contents
0	2 bytes	Length and Flags: contains the length of the following data in the low 10 bits and a set of reserved flags in the upper 6 bits.
2	2 bytes	Destination Network: the network number the packet is being sent to.
4	1 byte	Destination Node: the node number the packet is being sent to.
5	1 byte	Destination Socket: the socket that the packet is being sent to.
6	1 byte	Source Socket: the socket that sent the packet.
7	1 byte	Type: the AppleTalk type of the packet.
8	Specified by the Length	Payload: the data for the packet.

This data is sent from the remote client to the host during the established phase in order to communicate data packets. The data includes AppleTalk specific information to route the client's data packets to the appropriate node on the private data network. The established data from the remote client to the host is always encrypted to ensure a secure communications channel. The basic format (without any flags set) contains data from one data packet that the remote client is sending to the host which is the AppleTalk network. There are not any special data formats for the teardown phase since no data is communicated between the remote client and the host during the teardown phase.

In summary, the invention provides a virtual private network system between a private data network and a remote client which does not require expensive leased lines

or gateways to establish a secure communications path in which the remote client becomes a virtual node of the private network. The system also permits an individual to access the private data network without incurring any long distance telephone charges. In addition, the system permits a private data network and remote client that use a first communications protocol to communicate with each other over a public data network that uses a different communications protocol. The system also permits an individual to easily connect to the private data network as a virtual node without a remote private network and the individual appears to be a node on the private network, once connected, so that the individual may access any resources on the private data network.

In operation, a user of the remote client establishes a secure connection with the host of the private computer network through the authentication process so that the remote client is a virtual node of the private network. The user may then transmit data and commands in the private network's communication protocol over the public network through the secure communications path and receive data and commands back from the private network. For example, the user of the remote client may issue a print command to a printer attached to the private network, that print command is encapsulated in an encrypted data packet sent over the public access network, the host computer decrypts the print command and passes the print command on to the printer attached to the private network. Thus, the remote client is a virtual node of the private network and the user of the remote client may access any of the resources of the private network as if the remote client was an actual physical node of the private network.

While the foregoing has been with reference to a particular embodiment of the invention, it will be appreciated by those skilled in the art that changes in this embodiment may be made without departing from the principles and spirit of the invention, the scope of which is defined by the appended claims.

**Claims:**

1. A method for forming a virtual node for a private access network having a private access communications protocol over a public access network having a public access communications protocol, the virtual node being a remote client computer and the method comprising:

establishing a secure communications path over the public access network between a host computer connected to the private network and a remote client computer to establish the remote client computer as a virtual node of the private network;

generating a data packet to be transmitted over the secure communications path, the data packet including data and information about routing the data in the data packet in accordance with the private access communications protocol;

encrypting said data packet;

encapsulating said encrypted data packet into second data packet having a format compatible with the public access communications protocol;

transmitting the second data packet over the public access network;

unpacking the encrypted data packet from said second data packet; and

decrypting the data packet received from the public access network to route the data in the data packet over the private access network using the information about the private access communications protocol.

2. The method of Claim 1, wherein said establishing further comprises negotiating a communications protocol compatible with the private network between the host computer connected to the public access network and the remote client computer, and authenticating the identity of the remote client computer.

3. The method of Claim 2, wherein the authentication comprises generating a challenge at the host computer, communicating said challenge to the remote client computer, and receiving a challenge response from the remote client computer.

4. The method of claim 1 further comprising negotiating a session key for communicating between the host and the client.

5. The method of Claim 1, wherein generating the information in the data packet comprises generating a network node identification number for the remote client node.

6. The method of Claim 5, wherein said private access network comprises an AppleTalk communications network.

7. The method of Claim 6, wherein said public access network comprises the Internet.

8. A virtual node for a private access network having a private access communications protocol over a public access network having a public access communications protocol, the virtual node being a remote client computer and comprising:

means for establishing a secure communications path over the public access network between a host computer connected to the private network and a remote client computer to establish the remote client computer as a virtual node of the private network;

means for generating a data packet to be transmitted over the secure communications path, the data packet including data and information about routing the data in the data packet in accordance with the private access communications protocol;

means for encrypting said data packet;

means for encapsulating said encrypted data packet into second data packet having a format compatible with the public access communications protocol;

means for transmitting the second data packet over the public access network;

means for unpacking the encrypted data packet from said second data packet;

and

means for decrypting the data packet received from the public access network to route the data in the data packet over the private access network using the information about the private access communications protocol.

9. The virtual node of Claim 8, wherein said establishing means further comprises means for negotiating a communications protocol compatible with the private network between the host computer connected to the public access network and the remote client computer, and means for authenticating the identity of the remote client computer.

10. The virtual node of Claim 9, wherein the authentication means comprises means for generating a challenge at the host computer, means for communicating said challenge to the remote client computer, and means for receiving a challenge response from the remote client computer.

11. The virtual node of claim 8 further comprising negotiating a session key for communicating between the host and the client.

12. The virtual node of Claim 8, wherein said means for generating the information in the data packet comprises means for generating a network node identification number for the remote client node.

13. The virtual node of Claim 12, wherein said private access network comprises an AppleTalk communications network.

14. The virtual node of Claim 13, wherein said public access network comprises the Internet.



# PCT

## INTERNATIONAL SEARCH REPORT

(PCT Article 18 and Rules 43 and 44)

Applicant's or agent's file reference <b>00479.00028</b>	<b>FOR FURTHER ACTION</b> see Notification of Transmittal of International Search Report (Form PCT/ISA/220) as well as, where applicable, item 5 below.	
International application No. <b>PCT/US 01/13260</b>	International filing date (day/month/year) <b>25/04/2001</b>	(Earliest) Priority Date (day/month/year) <b>30/10/1998</b>
Applicant <b>SCIENCE APPLICATIONS INTERNATIONAL CORPORATION</b>		

This International Search Report has been prepared by this International Searching Authority and is transmitted to the applicant according to Article 18. A copy is being transmitted to the International Bureau.

This International Search Report consists of a total of 6 sheets.

It is also accompanied by a copy of each prior art document cited in this report.

**1. Basis of the report**

a. With regard to the **language**, the international search was carried out on the basis of the international application in the language in which it was filed, unless otherwise indicated under this item.

the international search was carried out on the basis of a translation of the international application furnished to this Authority (Rule 23.1(b)).

b. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, the international search was carried out on the basis of the sequence listing :

contained in the international application in written form.

filed together with the international application in computer readable form.

furnished subsequently to this Authority in written form.

furnished subsequently to this Authority in computer readable form.

the statement that the subsequently furnished written sequence listing does not go beyond the disclosure in the international application as filed has been furnished.

the statement that the information recorded in computer readable form is identical to the written sequence listing has been furnished

2.  **Certain claims were found unsearchable** (See Box I).

3.  **Unity of invention is lacking** (see Box II).

4. With regard to the **title**,

the text is approved as submitted by the applicant.

the text has been established by this Authority to read as follows:

**SECURE DOMAIN NAME SERVICE**

5. With regard to the **abstract**,

the text is approved as submitted by the applicant.

the text has been established, according to Rule 38.2(b), by this Authority as it appears in Box III. The applicant may, within one month from the date of mailing of this international search report, submit comments to this Authority.

6. The figure of the **drawings** to be published with the abstract is Figure No.

26

as suggested by the applicant.

None of the figures.

because the applicant failed to suggest a figure.

because this figure better characterizes the invention.

**INTERNATIONAL SEARCH REPORT**

International Application No

PCT/US 01/13260

**A. CLASSIFICATION OF SUBJECT MATTER**  
 IPC 7 H04L29/12 H04L29/06 G06F17/60

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)  
 IPC 7 H04L G06F

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 practical, search terms used)

EPO-Internal, WPI Data, PAJ, IBM-TDB

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	DONALD E. EASTLAKE 3RD: "<draft-ietf-dnssec-secext2-05.txt> Domain Name System Security Extensions" INTERNET DRAFT, 'Online! April 1998 (1998-04), XP002199931 Retrieved from the Internet: <URL:ftp://ftp.inet.no/pub/ietf/internet-drafts/draft-ietf-dnssec-secext2-05.txt> 'retrieved on 2002-05-23! 1. Overview of the contents 2.3 Data origin authentication and integrity 2.4 DNS transaction and request authentication --- -/--	1,4-6

Further documents are listed in the continuation of box C.

Patent family members are listed in 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 but 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

12 August 2002

Date of mailing of the international search report

23. 08. 2002

Name and mailing address of the ISA  
 European Patent Office, P.B. 5818 Patentlaan 2  
 NL - 2280 HV Rijswijk  
 Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
 Fax: (+31-70) 340-3016

Authorized officer

Bertolissi, E

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	CHAPMAN D.B.; ZWICKY E.D.: " Building Internet Firewalls" O'REILLY, November 1995 (1995-11), XP002199932 pag 278-296 pag 351-375	1,4-6
P,X	DE 199 24 575 A (SUN MICROSYSTEMS INC) 2 December 1999 (1999-12-02) abstract column 2, line 48 -column 3, line 6	1,4,5,7
A	P. SRISURESH, G. TSIRTSIS, P. AKKIRAJU, A. HEFFERNAN: "<draft-ietf-nat-dns-alg-00.txt> DNS extensions to Network Address Translators (DNS_ALG)" INTERNET DRAFT, 'Online! July 1998 (1998-07), XP002199933 Retrieved from the Internet: <URL:ftp://ftp.inet.no/pub/ietf/internet-drafts/draft-ietf-nat-dns-alg-00.txt> 'retrieved on 2002-05-23! 1. Introduction 2. Requirement for DNS extensions fig 3 5.3 Incoming name lookup queries 8. Security considerations	1-12
A	EP 0 838 930 A (DIGITAL EQUIPMENT CORP) 29 April 1998 (1998-04-29) abstract column 9, line 11 -column 10, line 34	1-12
X	JAMES E. BELLAIRE: "Subject: New Statement of Rules - Naming Internet Domains " INTERNET NEWSGROUP, 'Online! 30 July 1995 (1995-07-30), XP002209580 comp.dcom.telecom Retrieved from the Internet: <URL:http://groups.google.com/> 'retrieved on 2002-08-12! page 1, paragraph 8 page 2, paragraph 4	13,15
A	CLARK D: "US CALLS FOR PRIVATE DOMAIN-NAME SYSTEM" COMPUTER, IEEE COMPUTER SOCIETY, LONG BEACH., CA, US, US, vol. 31, no. 8, 1 August 1998 (1998-08-01), pages 22-25, XP000780513 ISSN: 0018-9162 page 22	13-16

## INTERNATIONAL SEARCH REPORT

International Application No

PCT/US 01/13260

## C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	BEQUAI A: "Balancing Legal Concerns Over Crime and Security in Cyberspace" COMPUTERS & SECURITY. INTERNATIONAL JOURNAL DEVOTED TO THE STUDY OF TECHNICAL AND FINANCIAL ASPECTS OF COMPUTER SECURITY, ELSEVIER SCIENCE PUBLISHERS. AMSTERDAM, NL, vol. 17, no. 4, 1998, pages 293-298, XP004129224 ISSN: 0167-4048 pag 296-297 Lanham Act -----	13-16
A	RICH WINKEL : "CAQ: NETWORKING WITH SPOOKS: THE NET & THE CONTROL OF INFORMATION " INTERNET NEWSGROUP, 'Online! 21 June 1997 (1997-06-21), XP002209581 misc.activism.progressive Retrieved from the Internet: <URL:http://groups.google.com/> 'retrieved on 2002-08-12! the whole document -----	13-16

INTERNATIONAL SEARCH REPORT

International application No.  
PCT/US 01/13260

**Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)**

This International Search Report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

- 1.  Claims Nos.:  
because they relate to subject matter not required to be searched by this Authority, namely:
  
- 2.  Claims Nos.:  
because they relate to parts of the International Application that do not comply with the prescribed requirements to such an extent that no meaningful International Search can be carried out, specifically:
  
- 3.  Claims Nos.:  
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

**Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)**

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

- 1.  As all required additional search fees were timely paid by the applicant, this International Search Report covers all searchable claims.
  
- 2.  As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.
  
- 3.  As only some of the required additional search fees were timely paid by the applicant, this International Search Report covers only those claims for which fees were paid, specifically claims Nos.:
  
- 4.  No required additional search fees were timely paid by the applicant. Consequently, this International Search Report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

**Remark on Protest**

- The additional search fees were accompanied by the applicant's protest.
- No protest accompanied the payment of additional search fees.

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. Claims: 1-12

A portal for authenticating a query for a secure computer network address

2. Claims: 13-16

A method and a computer readable storage medium for registering a secure domain name

1. The present communication is an Annex to the invitation to pay additional fees (Form PCT/ISA/206). It shows the results of the international search established on the parts of the international application which relate to the invention first mentioned in claims Nos.:

**1-12**

2. This communication is not the international search report which will be established according to Article 18 and Rule 43.

3. If the applicant does not pay any additional search fees, the information appearing in this communication will be considered as the result of the international search and will be included as such in the international search report.

4. If the applicant pays additional fees, the international search report will contain both the information appearing in this communication and the results of the international search on other parts of the international application for which such fees will have been paid.

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	DONALD E. EASTLAKE 3RD : "<draft-ietf-dnssec-secext2-05.txt> Domain Name System Security Extensions" INTERNET DRAFT , [Online] April 1998 (1998-04), XP002199931 Retrieved from the Internet: <URL:ftp://ftp.inet.no/pub/ietf/internet-drafts/draft-ietf-dnssec-secext2-05.txt> [retrieved on 2002-05-23] 1. Overview of the contents 2.3 Data origin authentication and integrity 2.4 DNS transaction and request authentication ---	1,4-6
Y	CHAPMAN D.B.; ZWICKY E.D.: " Building Internet Firewalls " O'REILLY, November 1995 (1995-11), XP002199932 pag 278-296 pag 351-375 ---	1,4-6
P,X	DE 199 24 575 A (SUN MICROSYSTEMS INC) 2 December 1999 (1999-12-02) abstract column 2, line 48 -column 3, line 6 --- -/--	1,4,5,7

Further documents are listed in the continuation of box C.

Patent family members are listed in 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 but 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

2

**INTERNATIONAL SEARCH REPORT**

information on patent family members

International Application No

PCT/US 01/13260

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
DE 19924575	A	02-12-1999	DE 19924575 A1	02-12-1999
			FR 2782873 A1	03-03-2000
			GB 2340702 A ,B	23-02-2000
			JP 2000049867 A	18-02-2000
-----				
EP 0838930	A	29-04-1998	US 6101543 A	08-08-2000
			EP 0838930 A2	29-04-1998
			JP 10178450 A	30-06-1998
-----				



AUG 29 2002

From the INTERNATIONAL SEARCHING AUTHORITY

**PCT** BANNER WITCOFF

To:  
**BANNER & WITCOFF, LTD.**  
 Attn. Curtin, Joseph P.  
 1001 G Street, N.W.  
 Eleventh Floor  
 Washington, DC 20001-4597 K9  
 UNITED STATES OF AMERICA

NOTIFICATION OF TRANSMITTAL OF  
 THE INTERNATIONAL SEARCH REPORT  
 OR THE DECLARATION

*000479.00028*  
**DOCKETED**

(PCT Rule 44.1)

AUG 29 2002

*Art. 19 Amend 10/23/02*

*000479.00028*  
**DOCKETED**

Date of mailing  
 (day/month/year) 23/08/2002

Applicant's or agent's file reference  
 00479.00028

**FOR FURTHER ACTION** See paragraphs 1 and 4 below

International application No.  
 PCT/US 01/ 13260

International filing date  
 (day/month/year) 25/04/2001

Applicant  
**SCIENCE APPLICATIONS INTERNATIONAL CORPORATION**

1.  The applicant is hereby notified that the International Search Report has been established and is transmitted herewith.

**Filing of amendments and statement under Article 19:**

The applicant is entitled, if he so wishes, to amend the claims of the International Application (see Rule 46):

**When?** The time limit for filing such amendments is normally 2 months from the date of transmittal of the International Search Report; however, for more details, see the notes on the accompanying sheet.

**Where?** Directly to the International Bureau of WIPO  
 34, chemin des Colombettes  
 1211 Geneva 20, Switzerland  
 Facsimile No.: (41-22) 740.14.35

**For more detailed instructions,** see the notes on the accompanying sheet.

2.  The applicant is hereby notified that no International Search Report will be established and that the declaration under Article 17(2)(a) to that effect is transmitted herewith.

3.  **With regard to the protest** against payment of (an) additional fee(s) under Rule 40.2, the applicant is notified that:

the protest together with the decision thereon has been transmitted to the International Bureau together with the applicant's request to forward the texts of both the protest and the decision thereon to the designated Offices.

no decision has been made yet on the protest; the applicant will be notified as soon as a decision is made.

4. **Further action(s):** The applicant is reminded of the following:

Shortly after **18 months** from the priority date, the international application will be published by the International Bureau. If the applicant wishes to avoid or postpone publication, a notice of withdrawal of the international application, or of the priority claim, must reach the International Bureau as provided in Rules 90bis.1 and 90bis.3, respectively, before the completion of the technical preparations for international publication.

Within **19 months** from the priority date, a demand for international preliminary examination must be filed if the applicant wishes to postpone the entry into the national phase until 30 months from the priority date (in some Offices even later).

Within **20 months** from the priority date, the applicant must perform the prescribed acts for entry into the national phase before all designated Offices which have not been elected in the demand or in a later election within 19 months from the priority date or could not be elected because they are not bound by Chapter II.

Name and mailing address of the International Searching Authority



European Patent Office, P.B. 5818 Patentlaan 2  
 NL-2280 HV Rijswijk  
 Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
 Fax: (+31-70) 340-3016

Authorized officer

Claude Berthon

## NOTES TO FORM PCT/ISA/220

These Notes are intended to give the basic instructions concerning the filing of amendments under article 19. The Notes are based on the requirements of the Patent Cooperation Treaty, the Regulations and the Administrative Instructions under that Treaty. In case of discrepancy between these Notes and those requirements, the latter are applicable. For more detailed information, see also the PCT Applicant's Guide, a publication of WIPO.

In these Notes, "Article", "Rule", and "Section" refer to the provisions of the PCT, the PCT Regulations and the PCT Administrative Instructions, respectively.

### INSTRUCTIONS CONCERNING AMENDMENTS UNDER ARTICLE 19

The applicant has, after having received the international search report, one opportunity to amend the claims of the international application. It should however be emphasized that, since all parts of the international application (claims, description and drawings) may be amended during the international preliminary examination procedure, there is usually no need to file amendments of the claims under Article 19 except where, e.g. the applicant wants the latter to be published for the purposes of provisional protection or has another reason for amending the claims before international publication. Furthermore, it should be emphasized that provisional protection is available in some States only.

#### What parts of the international application may be amended?

Under Article 19, only the claims may be amended.

During the international phase, the claims may also be amended (or further amended) under Article 34 before the International Preliminary Examining Authority. The description and drawings may only be amended under Article 34 before the International Examining Authority.

Upon entry into the national phase, all parts of the international application may be amended under Article 28 or, where applicable, Article 41.

#### When?

Within 2 months from the date of transmittal of the international search report or 16 months from the priority date, whichever time limit expires later. It should be noted, however, that the amendments will be considered as having been received on time if they are received by the International Bureau after the expiration of the applicable time limit but before the completion of the technical preparations for international publication (Rule 46.1).

#### Where not to file the amendments?

The amendments may only be filed with the International Bureau and not with the receiving Office or the International Searching Authority (Rule 46.2).

Where a demand for international preliminary examination has been/is filed, see below.

#### How?

Either by cancelling one or more entire claims, by adding one or more new claims or by amending the text of one or more of the claims as filed.

A replacement sheet must be submitted for each sheet of the claims which, on account of an amendment or amendments, differs from the sheet originally filed.

All the claims appearing on a replacement sheet must be numbered in Arabic numerals. Where a claim is cancelled, no renumbering of the other claims is required. In all cases where claims are renumbered, they must be renumbered consecutively (Administrative Instructions, Section 205(b)).

**The amendments must be made in the language in which the international application is to be published.**

#### What documents must/may accompany the amendments?

##### Letter (Section 205(b)):

The amendments must be submitted with a letter.

The letter will not be published with the international application and the amended claims. It should not be confused with the "Statement under Article 19(1)" (see below, under "Statement under Article 19(1)").

**The letter must be in English or French, at the choice of the applicant. However, if the language of the international application is English, the letter must be in English; if the language of the international application is French, the letter must be in French.**

## NOTES TO FORM PCT/ISA/220 (continued)

The letter must indicate the differences between the claims as filed and the claims as amended. It must, in particular, indicate, in connection with each claim appearing in the international application (it being understood that identical indications concerning several claims may be grouped), whether

- (i) the claim is unchanged;
- (ii) the claim is cancelled;
- (iii) the claim is new;
- (iv) the claim replaces one or more claims as filed;
- (v) the claim is the result of the division of a claim as filed.

**The following examples illustrate the manner in which amendments must be explained in the accompanying letter:**

1. [Where originally there were 48 claims and after amendment of some claims there are 51]:  
"Claims 1 to 29, 31, 32, 34, 35, 37 to 48 replaced by amended claims bearing the same numbers; claims 30, 33 and 36 unchanged; new claims 49 to 51 added."
2. [Where originally there were 15 claims and after amendment of all claims there are 11]:  
"Claims 1 to 15 replaced by amended claims 1 to 11."
3. [Where originally there were 14 claims and the amendments consist in cancelling some claims and in adding new claims]:  
"Claims 1 to 6 and 14 unchanged; claims 7 to 13 cancelled; new claims 15, 16 and 17 added." or  
"Claims 7 to 13 cancelled; new claims 15, 16 and 17 added; all other claims unchanged."
4. [Where various kinds of amendments are made]:  
"Claims 1-10 unchanged; claims 11 to 13, 18 and 19 cancelled; claims 14, 15 and 16 replaced by amended claim 14; claim 17 subdivided into amended claims 15, 16 and 17; new claims 20 and 21 added."

### **"Statement under article 19(1)" (Rule 46.4)**

The amendments may be accompanied by a statement explaining the amendments and indicating any impact that such amendments might have on the description and the drawings (which cannot be amended under Article 19(1)).

The statement will be published with the international application and the amended claims.

**It must be in the language in which the international application is to be published.**

It must be brief, not exceeding 500 words if in English or if translated into English.

It should not be confused with and does not replace the letter indicating the differences between the claims as filed and as amended. It must be filed on a separate sheet and must be identified as such by a heading, preferably by using the words "Statement under Article 19(1)."

It may not contain any disparaging comments on the international search report or the relevance of citations contained in that report. Reference to citations, relevant to a given claim, contained in the international search report may be made only in connection with an amendment of that claim.

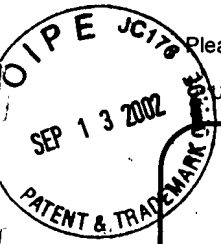
### **Consequence if a demand for international preliminary examination has already been filed**

If, at the time of filing any amendments and any accompanying statement, under Article 19, a demand for international preliminary examination has already been submitted, the applicant must preferably, at the time of filing the amendments (and any statement) with the International Bureau, also file with the International Preliminary Examining Authority a copy of such amendments (and of any statement) and, where required, a translation of such amendments for the procedure before that Authority (see Rules 55.3(a) and 62.2, first sentence). For further information, see the Notes to the demand form (PCT/IPEA/401).

### **Consequence with regard to translation of the international application for entry into the national phase**

The applicant's attention is drawn to the fact that, upon entry into the national phase, a translation of the claims as amended under Article 19 may have to be furnished to the designated/elected Offices, instead of, or in addition to, the translation of the claims as filed.

For further details on the requirements of each designated/elected Office, see Volume II of the PCT Applicant's Guide.



Please type a plus sign (+) inside this box

Approved for use through 10/31/2002. OMB 0651-0031  
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

PTO/SE/21 (08-00)

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

SV \$2153 #6 19

<b>TRANSMITTAL FORM</b> <i>(to be used for all correspondence after initial filing)</i>	<b>Application Number</b>	09/504,783
	<b>Filing Date</b>	February 15, 2000
	<b>First Named Inventor</b>	Edmond Colby Munger
	<b>Group Art Unit</b>	2153
	<b>Examiner Name</b>	Krisna Lim
<b>Total Number of Pages in This Submission</b>		<b>Attorney Docket Number</b> 000479.85672

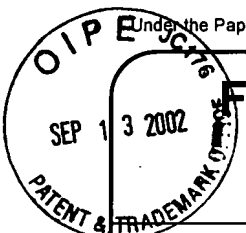
ENCLOSURES <i>(check all that apply)</i>		
<input checked="" type="checkbox"/> Fee Transmittal Form  <input type="checkbox"/> Fee Attached  <input type="checkbox"/> Amendment / Response <input type="checkbox"/> After Final  <input type="checkbox"/> Affidavits/declaration(s)  <input type="checkbox"/> Extension of Time Request  <input type="checkbox"/> Express Abandonment Request  <input checked="" type="checkbox"/> Information Disclosure Statement  <input type="checkbox"/> Certified Copy of Priority Document(s)  <input type="checkbox"/> Response to Missing Parts/ Incomplete Application  <input type="checkbox"/> Response to Missing Parts under 37 CFR 1.52 or 1.53	<input type="checkbox"/> Assignment Papers <i>(for an Application)</i>  <input type="checkbox"/> Drawing(s)  <input type="checkbox"/> Licensing-related Papers  <input type="checkbox"/> Petition  <input type="checkbox"/> Petition to Convert to a Provisional Application  <input type="checkbox"/> Power of Attorney, Revocation Change of Correspondence Address  <input type="checkbox"/> Terminal Disclaimer  <input type="checkbox"/> Request for Refund  <input type="checkbox"/> CD, Number of CD(s) _____	<input type="checkbox"/> After Allowance Communication to Group  <input type="checkbox"/> Appeal Communication to Board of Appeals and Interferences  <input type="checkbox"/> Appeal Communication to Group <i>(Appeal Notice, Brief, Reply Brief)</i>  <input type="checkbox"/> Proprietary Information  <input type="checkbox"/> Status Letter  <input type="checkbox"/> Other Enclosures <i>(please identify below)</i>
Remarks		<b>RECEIVED</b> <b>SEP 18 2002</b> <b>Technology Center 2100</b>

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT	
Firm or Individual name	Bradley C. Wright, Reg. No. 38,061
Signature	<b>Reg. No. 49,024</b>
Date	September 13, 2002

CERTIFICATE OF MAILING			
I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on this date: <input style="width: 150px;" type="text"/>			
Typed or printed name			
Signature		Date	

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be send to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

Match & Return



# FEE TRANSMITTAL for FY 2002

*Patent fees are subject to annual revision.*

Applicant claims small entity status. See 37 CFR 1.27

**TOTAL AMOUNT OF PAYMENT** (\$) 180

Complete if Known	
Application Number	09/504,783
Filing Date	February 15, 2000
First Named Inventor	Edmond Colby Munger
Examiner Name	Krisna Lim
Group / Art Unit	2153
Attorney Docket No.	000479.85672

**METHOD OF PAYMENT (check all that apply)**

Check    Credit card    Money    Other    None Order

Deposit Account:

Deposit Account Number: 19-0733

Deposit Account Name: Banner & Witcoff, Ltd.

**The Commissioner is authorized to: (check all that apply)**

Charge fee(s) indicated below    Credit any overpayments

Charge any additional fee(s) during the pendency of this application

Charge fee(s) indicated below, except for the filing fee to the above-identified deposit account.

**FEE CALCULATION**

**1. BASIC FILING FEE**

Large Entity		Small Entity		Fee Description	Fee Paid
Fee Code	Fee (\$)	Fee Code	Fee (\$)		
101	740	201	370	Utility filing fee	
106	330	206	165	Design filing fee	
107	510	207	255	Plant filing fee	
108	740	208	370	Reissue filing fee	
114	160	214	80	Provisional filing fee	
<b>SUBTOTAL (1)</b>					(\$ 0)

**2. EXTRA CLAIM FEES**

Total Claims:  - \*\* =  X  =

Independent Claims:  - \*\* =  X  =

Multiple Dependent: X  =

Large Entity		Small Entity		Fee Description	Fee Paid
Fee Code	Fee (\$)	Fee Code	Fee (\$)		
103	18	203	9	Claims in excess of 20	
102	84	202	42	Independent claims in excess of 3	
104	280	204	140	Multiple dependent claim, if not paid	
109	84	209	42	** Reissue independent claims over original patent	
110	18	210	9	** Reissue claims in excess of 20 and over original patent	
<b>SUBTOTAL (2)</b>					(\$ 0)

\*\*or number previously paid, if greater; For Reissues, see above

**3. ADDITIONAL FEES**

Large Entity		Small Entity		Fee Description	Fee Paid
Fee Code	Fee (\$)	Fee Code	Fee (\$)		
105	130	205	65	Surcharge - late filing fee or oath	
127	50	227	25	Surcharge - late provisional filing fee or cover sheet.	
139	130	139	130	Non-English specification	
147	2,520	147	2,520	For filing a request for reexamination	
112	920*	112	920*	Requesting publication of SIR prior to Examiner action	
113	1,840*	113	1,840*	Requesting publication of SIR after Examiner action	
115	110	215	55	Extension for reply within first month	
116	400	216	200	Extension for reply within second month	
117	920	217	460	Extension for reply within third month	
118	1,440	218	720	Extension for reply within fourth month	
128	1,960	228	980	Extension for reply within fifth month	
119	320	219	160	Notice of Appeal	
120	320	220	160	Filing a brief in support of an appeal	
121	280	221	140	Request for oral hearing	
138	1,510	138	1,510	Petition to institute a public use proceeding	
140	110	240	55	Petition to revive - unavoidable	
141	1,280	241	640	Petition to revive - unintentional	
142	1,280	242	640	Utility issue fee (or reissue)	
143	460	243	230	Design issue fee	
144	620	244	310	Plant issue fee	
122	130	122	130	Petitions to the Commissioner	
123	50	123	50	Processing fee under 37 CFR 1.17 (q)	
126	180	126	180	Submission of Information Disclosure Stmt	180
581	40	581	40	Recording each patent assignment per property (times number of properties)	
146	740	246	370	Filing a submission after final rejection (37 CFR § 1.129(a))	
149	740	249	370	For each additional invention to be examined (37 CFR § 1.129(b))	
179	740	279	370	Request for Continued Examination (RCE)	
169	900	169	900	Request for expedited examination of a design application	
Other fee (specify) _____					
*Reduced by Basic Filing Fee Paid				<b>SUBTOTAL (3)</b>	(\$ 180)

**RECEIVED**  
SEP 18 2002  
Technology Center 2100

SUBMITTED BY		Complete (if applicable)			
Name (Print/Type)	Bradley C. Wright	Registration No. Attorney/Agent)	38,061	Telephone	(202) 508-9160
Signature		Reg. No. 49,024	Date	September 13, 2002	

**WARNING:** Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.



# 15

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	)	Group Art Unit: 2153
	)	
Edmond Colby Munger et al.	)	Examiner: Krisna Lim
	)	
Serial No. 09/504,783	)	Attorney Docket No. 000479.85672
	)	
Filed: February 15, 2000	)	

For: IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

INFORMATION DISCLOSURE STATEMENT

RECEIVED

SEP 18 2002

Technology Center 2100

Assistant Commissioner of Patents  
Washington, D.C. 20231

Sir:

In accordance with Applicants' duty of disclosure, and pursuant to 37 C.F.R. § 1.97(d), the following information is submitted for consideration by the United States Patent and Trademark Office in connection with the above-captioned application. The information is identified on the attached PTO 1449 form.

Applicants do not waive any right to take appropriate action to establish patentability over the listed documents should they be applied as references against the claims of the present application.

The undersigned certifies under 37 C.F.R. § 1.97(e)(1) that each item of information contained in this information disclosure statement was first cited in a communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of this statement. A copy of the foreign search report is attached.

09/13/2002 09:08:51 00000055 190733 09504783

01 00:00 100.00 CH

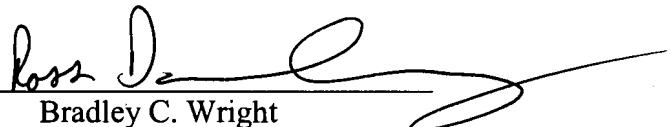
**Information Disclosure Statement**

Serial No. 09/504,783

The Commissioner is authorized to charge the \$180 fee to our Deposit Account No. 19-0733. No additional fees are believed due to ensure consideration of the attached documents by the Examiner. However, if any fees are required or an overpayment of fees made, the Commissioner is hereby authorized to debit or credit our Deposit Account No. 19-0733, as necessary.

Respectfully submitted,

Date: September 13, 2002

By:   
Bradley C. Wright  
Registration No. 38,061

Banner & Witcoff, LTD  
1001 G Street, N.W.  
Washington, D.C. 20001-4597  
(202) 508-9100

**Reg. No. 49,024**

BCW/RAD/mmd



PTO-1449 (Modified) U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE INFORMATION DISCLOSURE STATEMENT BY APPLICANT	ATTY. DOCKET NO. 000479.85672	SERIAL NUMBER 09/504,783
	APPLICANT Edmond Colby Munger et al.	
	FILING DATE February 15, 2000	GROUP ART UNIT 2153

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO
<i>K</i>	199 24 575 -	12/2/99	DE			
<i>K</i>	0 838 930	4/29/98	EPO			

RECEIVED  
 SEP 18 2002  
 Technology Center 2100

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

<i>K</i>	Search Report (dated 8/23/02), International Application No. PCT/US01/13260
<i>K</i>	Donald E. Eastlake, 3 <sup>rd</sup> , "Domain Name System Security Extensions", INTERNET DRAFT, April 1998, pages 1-51
<i>K</i>	D. B. Chapman et al., "Building Internet Firewalls", November 1995, pages 278-375
<i>K</i>	P. Srisuresh et al., "DNS extensions to Network address Translators (DNS_ALG)", INTERNET DRAFT, July 1998, pages 1-27
<i>K</i>	James E. Bellaire, "New Statement of Rules - Naming Internet Domains", Internet Newsgroup, July 30, 1995, 1 page
<i>K</i>	D. Clark, "US Calls for Private Domain-Name System", Computer, IEEE Computer Society, August 1, 1998, pages 22-25
<i>K</i>	August Bequai, "Balancing Legal Concerns Over Crime and Security in Cyberspace", Computer & Security, Vol. 17, No. 4, 1998, pages 293-298
<i>K</i>	Rich Winkel, "CAQ: Networking With Spooks: The NET & The Control Of Information", Internet Newsgroup, June 21, 1997, 4 pages

EXAMINER <i>KRISNA LIM</i>	DATE CONSIDERED <i>10/7/02</i>
----------------------------	--------------------------------

EXAMINER: Initial citation if reference was considered. Draw line through citation if not in conformance to MPEP 609 and not considered. Include copy of this form with next communication to applicant.

IDS w/1449 form filed: September 13, 2002





19 BUNDESREPUBLIK  
DEUTSCHLAND



DEUTSCHES  
PATENT- UND  
MARKENAMT

# Offenlegungsschrift DE 199 24 575 A 1

51 Int. Cl.<sup>6</sup>:  
**H 04 L 29/06**  
H 04 L 12/22  
G 06 F 13/00  
G 06 F 12/14  
// H04L 9/00

21 Aktenzeichen: 199 24 575.4  
22 Anmeldetag: 28. 5. 99  
43 Offenlegungstag: 2. 12. 99

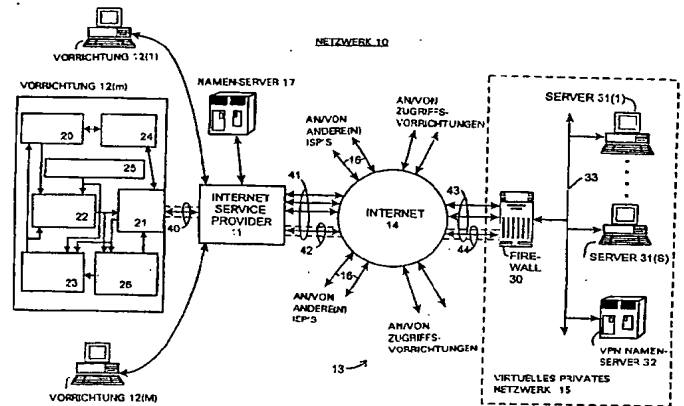
DE 199 24 575 A 1

30 Unionspriorität:  
087823 29. 05. 98 US  
71 Anmelder:  
Sun Microsystems, Inc., Palo Alto, Calif., US  
74 Vertreter:  
Samson & Partner, Patentanwälte, 80538 München

72 Erfinder:  
Provino, Joseph E., Cambridge, Mass., US

### Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

54 Kommunikationssystem und -Verfahren  
57 Das erfindungsgemäße System umfaßt ein virtuelles privates Netzwerk (15) und eine externe Vorrichtung (12(m)), welche durch ein digitales Netzwerk (14) miteinander verbunden sind. Das virtuelle private Netzwerk (15) weist eine Firewall (30), wenigstens eine interne Vorrichtung (31(s)) und einen Namen-Server (32) auf, welche jeweils eine Netzwerkadresse besitzen. Die interne Vorrichtung (31(s)) besitzt auch eine Sekundäradresse, und der Namen-Server (32) ist derart konfiguriert, daß er eine Zuordnung zwischen der Sekundäradresse und der Netzwerkadresse bereitstellt. In Reaktion auf eine Anfrage von der externen Vorrichtung (12(m)) zum Aufbau einer Verbindung zur Firewall (30) übermittle die Firewall (30) der externen Vorrichtung (12(m)) die Netzwerkadresse des Namen-Servers (32). In Reaktion auf eine Anfrage von einem Bediener oder ähnlichem, welche die Sekundäradresse der internen Vorrichtung (31(s)) enthält und einen Zugriff an die interne Vorrichtung (31(s)) anfordert, erzeugt die externe Vorrichtung (12(m)) eine Netzwerkadressen-Anfragennachricht zur Übertragung über die Verbindung an die Firewall (30), welche eine Auflösung der Netzwerkadresse, die der Sekundäradresse zugeordnet ist, anfordert. Die Firewall (30) übermittle die Adressenauflösungsanfrage an den Namen-Server (32), und der Namen-Server (32) übermittle die Netzwerkadresse, welche der Sekundäradresse zugeordnet ist, an die Firewall (30). Daraufhin stellt die Firewall (30) die Netzwerkadresse in einer ...



DE 199 24 575 A 1

Die Erfindung betrifft allgemein das Gebiet der digitalen Kommunikationssysteme und -verfahren, und insbesondere Systeme und Verfahren zum Vereinfachen der Kommunikation zwischen Vorrichtungen, welche mit öffentlichen Netzwerken verbunden sind, z. B. dem Internet, und Vorrichtungen, welche mit privaten Netzwerken verbunden sind.

Digitale Netzwerke wurden entwickelt, um die Übertragung von Information, welche auch Daten und Programme umfaßt, über digitale Computersysteme und andere Digitalvorrichtungen zu ermöglichen. Es wurde eine Vielzahl von Arten von Netzwerken entwickelt und realisiert, einschließlich sog. Fernverbindungsnetze (Wide-Area Networks, nachfolgend "WAN" genannt) und lokale Netzwerke (Local Area Networks, nachfolgend "LAN" genannt), welche eine Information unter Verwendung verschiedener Informationsübertragungsmethoden übermitteln. Im allgemeinen werden LANs innerhalb kleiner geographischer Bereiche realisiert, z. B. innerhalb eines einzelnen Bürogebäudes oder ähnlichem, zum Übertragen von Information innerhalb eines bestimmten Büros, einer Firma oder einer ähnlichen Art von Organisationseinheit. Andererseits werden WANs im allgemeinen auf relativ großen geographischen Bereichen realisiert und können verwendet werden, um Information sowohl zwischen LANs als auch zwischen Vorrichtungen, welche nicht mit LANs verbunden sind, zu übertragen. Derartige WANs umfassen auch öffentliche Netzwerke, z. B. das Internet, welche zur Informationsübertragung zwischen einer Anzahl von Unternehmen verwendet werden können.

Es sind mehrere Probleme im Zusammenhang der Kommunikation über ein Netzwerk aufgetreten, insbesondere in einem großen öffentlichen WAN, wie es z. B. das Internet ist. Im allgemeinen werden Informationen über ein Netzwerk in Nachrichtepaketeten übertragen, welche ausgehend von einer Vorrichtung, als Quelle bzw. Quellenvorrichtung, zu einer anderen Vorrichtung, als Ziel bzw. Zielvorrichtung, über einen oder mehrere Router oder allgemein Schaltungsknoten im Netzwerk übertragen werden. Jedes Nachrichtepaket enthält eine Zieladresse, welche von den Schaltungsknoten verwendet wird, um das jeweilige Nachrichtepaket an die geeignete Zielvorrichtung zu leiten. Z.B. im Internet haben solche Adressen die Form von "n"-Bit Zahlen (wobei "n" 32 oder 128 sein kann), wobei solche Zahlenkolonnen für einen Benutzer schwierig sind zu merken und einzugeben, wenn die oder der Benutzer die Übertragung eines Nachrichtepakets veranlassen möchte. Um einen Benutzer von der Notwendigkeit zu befreien, sich solche spezifische Zahlen-Internetadressen zu merken und einzugeben, stellt das Internet einen zweiten Adressierungsmechanismus bereit, der durch Benutzer der jeweiligen Vorrichtungen einfacher handzuhaben ist. Bei diesem Adressierungsmechanismus werden Internet-Domains, wie etwa LANs, Internet-Service-Provider (nachfolgend "ISP" genannt) und ähnliche, welche im Internet verbunden sind, durch für einen Benutzer relativ einfach les- und merkbare Namen identifiziert, die nachfolgend als "Klartextnamen" bezeichnet werden. Um den Einsatz von solchen Klartextnamen umzusetzen, werden Namen-Server, auch als DNS-Server für "Domain Name Server" bezeichnet, bereitgestellt, um die Klartextnamen in die geeigneten Internetadressen umzuwandeln. Wenn ein Bediener einer Vorrichtung, der die Übertragung eines Nachrichtepakets an eine andere Vorrichtung wünscht, den Klartextnamen der anderen Vorrichtung eingibt, nimmt die Vorrichtung zuerst Kontakt mit einem Namen-Server auf. Im allgemeinen kann der Namen-Server ein Teil des ISP selbst sein oder er kann eine spezielle Vorrichtung sein, welche durch den ISP über das Internet zugäng-

lich ist; in jedem Fall wird der ISP den Namen-Server identifizieren, welcher für die Vorrichtung zu verwenden ist, wenn sich die Vorrichtung beim ISP einloggt, d. h. anmeldet. Falls der Namen-Server, nachdem die Vorrichtung einen Kontakt hergestellt hat, eine Zahlen-Internetadresse für den Klartext-Domainnamen besitzt oder erhalten kann, übermittelt der Namen-Server die Zahlen-Internetadresse, welche dem Klartext-Domainnamen entspricht, zu der Vorrichtung des Bedieners. Die Vorrichtung kann sodann die Zahlen-Internetadresse, welche von dem Namen-Server zurückgesendet wurde, in das Nachrichtepaket einfügen und das Nachrichtepaket an den ISP für die Übertragung über das Internet auf konventioneller Weise liefern. Die Internet-Schaltungsknoten verwenden die Zahlen-Internetadresse, um das Nachrichtepaket an die gewünschte Zielvorrichtung zu übermitteln.

Andere Probleme treten insbesondere in Verbindung mit der Übertragung von Information über ein öffentliches WAN, z. B. das Internet, auf. Ein Problem besteht darin, sicherzustellen, daß die über das WAN übertragene Information, welche die Quellenvorrichtung und die Zielvorrichtung vertraulich behalten möchten, auch tatsächlich vertraulich bleibt gegenüber möglichen Lauschern, welche die Information abfangen können. Um die Vertraulichkeit zu wahren, wurden verschiedene Formen von Verschlüsselung entwickelt und werden verwendet, um die Information vor der Übertragung durch die Quellenvorrichtung zu verschlüsseln und die Information nach deren Empfang durch die Zielvorrichtung zu entschlüsseln. Falls gewünscht wird, daß beispielsweise die gesamte Information, welche zwischen einer bestimmten Quellenvorrichtung und einer bestimmten Zielvorrichtung übertragen wird, vertraulich bleiben soll, können die Vorrichtungen einen sog. "Sicherheitstunnel" zwischen den Vorrichtungen einrichten, der im wesentlichen sicherstellt, daß die gesamte Information, welche von der Quellenvorrichtung an die Zielvorrichtung übertragen wird, vor der Übertragung verschlüsselt wird (mit Ausnahme von bestimmten Protokollinformationen, wie Adresseninformation, welche den Fluß von Netzpaketen über das Netzwerk zwischen der Quellen- und Zielvorrichtung steuert), und daß die verschlüsselte Information vor der Verwendung durch die Zielvorrichtung entschlüsselt wird. Die Quellen- und Zielvorrichtungen können jeweils für sich eine Verschlüsselung bzw. Entschlüsselung durchführen, oder die Verschlüsselung und Entschlüsselung kann durch andere Vorrichtungen durchgeführt werden, bevor die Nachrichtepakete über das Internet übertragen werden.

Ein weiteres Problem, welches insbesondere im Zusammenhang mit Unternehmen, Regierungsämtern und privaten Organisationen auftritt, deren private Netzwerke, welche LANs, WANs oder etwaige Kombinationen derselben sein können, mit öffentlichen WANs, z. B. dem Internet, verbunden sind, besteht darin, sicherzustellen, daß deren private Netzwerke sicher sind gegenüber anderen Netzwerken, zu welchen z. B. die Unternehmen keinen Zugriff haben möchten, oder einen Zugriff durch andere zu regulieren und zu kontrollieren, zu welchen z. B. die jeweiligen Organisationen einen begrenzten Zugriff haben möchten. Um dies umzusetzen, verbinden die Organisationen in der Regel ihre privaten Netzwerke mit öffentlichen WANs über eine begrenzte Anzahl von Gateways, welche manchmal als "Firewalls" bezeichnet werden, durch welche der gesamte Netzwerkverkehr zwischen dem internen und dem öffentlichen Netzwerk läuft. In der Regel sind Netzwerkadressen von Domains und Vorrichtungen in dem privaten Netzwerk "hinter" der Firewall den Namen-Servern bekannt, welche in den privaten Netzwerken vorgesehen sind; sie sind aber nicht zugänglich für Namen-Server oder andere Vorrichtungen au-

ßerhalb der privaten Netzwerke, was die Kommunikation zwischen einer Vorrichtung außerhalb des privaten Netzwerkes und einer Vorrichtung innerhalb des privaten Netzwerkes schwierig macht.

Ein Ziel der vorliegend Erfindung ist es, hier Abhilfe zu schaffen.

Dieses Ziel erreicht die Erfindung durch die Gegenstände der Ansprüche 1, 7 und 13. Bevorzugte Ausführungsbeispiele der Erfindung sind in den jeweils abhängigen Ansprüchen beschrieben.

Danach schafft die Erfindung ein neuartiges und verbessertes System und ein Verfahren zum Vereinfachen von Kommunikation zwischen Vorrichtungen, welche mit öffentlichen Netzwerken, z. B. dem Internet, verbunden sind, und Vorrichtungen, welche mit privaten Netzwerken verbunden sind, wobei die Auflösung von Sekundäradressen, wie etwa Text- bzw. Klartextnamen im Internet, in die zugehörigen Netzwerkadressen durch Namen-Server oder ähnliche Vorrichtungen, die mit den privaten Netzwerken verbunden sind, ermöglicht wird.

Hierfür stellt die Erfindung ein System zur Verfügung mit einem virtuellen Privaten Netzwerk und einer externen Vorrichtung, welche durch ein digitales Netzwerk miteinander verbunden sind, sowie ein Kommunikationsverfahren und ein Computerprogrammprodukt zum gemeinsamen Verwenden mit einem derartiges System. Das virtuelle private Netzwerk weist eine Firewall bzw. ein Firewall-System, wenigstens eine interne Vorrichtung und einen Namen-Server auf, welche jeweils eine Netzwerkadresse besitzen. Die interne Vorrichtung besitzt ferner eine Sekundäradresse, und der Namen-Server ist derart konfiguriert, daß er eine Zuordnung zwischen der Sekundäradresse und der Netzwerkadresse bereitstellt. In Reaktion auf eine Anfrage von der externen Vorrichtung zum Aufbau einer Verbindung zur Firewall übermittelt die Firewall der externen Vorrichtung die Netzwerkadresse des Namen-Servers. In Reaktion auf eine Anfrage von einem Bediener oder ähnlichem, welche die Sekundäradresse der internen Vorrichtung enthält und einen Zugriff an die interne Vorrichtung anfordert, erzeugt die externe Vorrichtung eine Netzwerkadressen-Anfragennachricht zur Übertragung über die Verbindung an die Firewall, welche eine Auflösung der Netzwerkadresse, die der Sekundäradresse zugeordnet ist, anfordert. Die Firewall übermittelt die Adressenauflösungsanfrage an den Namen-Server und der Namen-Server übermittelt die Netzwerkadresse, welche der Sekundäradresse zugeordnet ist, an die Firewall. Daraufhin stellt die Firewall die Netzwerkadresse in einer Netzwerkadressenantwortnachricht zur Übertragung über die Verbindung an die externe Vorrichtung bereit. Die externe Vorrichtung kann sodann die auf diese Weise bereitgestellte Netzwerkadresse in nachfolgenden an die interne Vorrichtung gerichtete Kommunikationen mit der Firewall verwenden.

Weitere Vorteile und Ausgestaltungen der Erfindung ergeben sich aus der nachfolgenden detaillierten Beschreibung eines bevorzugten Ausführungsbeispiels. In der Beschreibung wird auf die beigefügte schematische Zeichnung Bezug genommen. Darin zeigt:

**Fig. 1** ein funktionelles Blockdiagramm eines erfindungsgemäßen Netzwerkes.

**Fig. 1** zeigt ein funktionelles Blockdiagramm eines Netzwerkes **10**, welches gemäß der vorliegenden Erfindung aufgebaut ist. Das Netzwerk **10** gemäß **Fig. 1** umfaßt einen Internet-Service-Provider (nachfolgend "ISP") **11**, welcher die Übertragung von Nachrichtenpaketen zwischen einer oder mehreren Vorrichtungen **12(1)** bis **12(M)** (nachfolgend allgemein mit dem Bezugszeichen **12(m)** identifiziert), welche mit dem ISP **11** verbunden sind, und anderen Vorrich-

tungen, welche allgemein durch ein Bezugszeichen **13** gekennzeichnet sind, über das Internet **14** ermöglicht, wobei die Übertragung von Information in Nachrichtenpaketen zwischen den Vorrichtungen **12(m)** und **13** realisiert wird.

Der ISP **11** verbindet das Internet **14** über eine oder mehrere logische Verbindungen oder Gateways oder ähnlichem (im vorliegenden allgemein als "Verbindungen" bezeichnet), welche allgemein durch das Bezugszeichen **41** gekennzeichnet sind. Der ISP **11** kann ein öffentlicher ISP sein, welcher in diesem Falle die Verbindung mit Vorrichtungen **12(m)** herstellt, welche durch Bediener betrieben werden können, die der allgemeinen Öffentlichkeit angehören, so daß diese Bediener Zugang zu dem Internet erlangen. Alternativ dazu kann der ISP **11** ein privater ISP sein. In diesem Falle werden die damit verbundenen Vorrichtungen **12(m)** im allgemeinen beispielsweise durch Angestellte eines bestimmten Unternehmens oder einer Regierungseinrichtung, Mitgliedern von einer privaten Organisation oder ähnlichen betrieben, um diesen Angestellten oder Mitglieder einen Zugang in das Internet bereit zu stellen.

In an sich konventioneller Weise weist das Internet ein Netz von Schaltungsknoten auf (welche nicht separat dargestellt sind), welche die ISPs **11** und die Vorrichtungen **13** miteinander verbinden, um dazwischen die Übertragung von Nachrichtenpaketen zu ermöglichen. Die Nachrichtenpakete, welche über das Internet **14** übertragen werden, stimmen mit denjenigen überein, welche durch das sog. Internetprotokoll (IP) definiert werden, und umfassen einen Kopfabschnitt, einen Datenabschnitt und können einen Fehlererfassungs- und/oder Korrekturabschnitt aufweisen. Der Kopfabschnitt enthält Information, welche verwendet wird, um das Nachrichtenpaket über das Internet **14** zu übertragen, beispielsweise eine Zieladresse, welche die Vorrichtung identifiziert, welche das Nachrichtenpaket als Zielvorrichtung empfangen soll, und eine Quellenadresse, welche diejenige Vorrichtung identifiziert, welche das Nachrichtenpaket erzeugt hat. In jedem Nachrichtenpaket haben die Ziel- und Quellenadresse jeweils die Form einer Zahl, welche eindeutig die jeweilige Ziel- bzw. Quellenvorrichtung identifiziert. Die Schaltungsknoten im Internet **14** verwenden wenigstens die Zieladresse eines jeweiligen Nachrichtenpaketes, um das jeweilige Nachrichtenpaket an die Zielvorrichtung zu übermitteln, wenn die Zielvorrichtung an das Internet angeschlossen ist, oder an einen ISP **11** oder andere Vorrichtungen, welche an das Internet **14** angeschlossen sind, welche sodann das Nachrichtenpaket an das geeignete Ziel senden werden. Der Datenabschnitt eines jeden Nachrichtenpakets enthält die in dem Nachrichtenpaket übertragenen Daten; und der Fehlererfassungs- und/oder Korrekturabschnitt enthält Fehlererfassungs- und/oder Korrekturinformationen, welche verwendet werden können, um zu verifizieren, daß das Nachrichtenpaket in korrekter Weise von der Quelle zu der Zielvorrichtung übertragen wurde (im Fall der Fehlererfassungsinformation), und um ausgewählte Arten von Fehlern zu korrigieren, falls das Nachrichtenpaket nicht korrekt übertragen wurde (im Falle der Fehlerkorrekturinformation).

Die Vorrichtungen **12(m)**, welche mit dem ISP **11** verbunden sind, können jede beliebige Anzahl von Arten von Vorrichtungen umfassen, welche über das Internet **14** mit anderen Vorrichtungen **13** kommunizieren, umfassend z. B. Personalcomputer, Computer-Workstations und ähnliches. Jede Vorrichtung **12(m)** kommuniziert mit dem ISP **11**, um Nachrichtenpakete für die Übertragung über das Internet **14** an diesen zu übertragen, oder um Nachrichtenpakete, welche durch den ISP **11** über das Internet empfangen werden, von diesem zu empfangen. Dabei kann jedes geeignete Protokoll verwendet werden, z. B. das bekannte Point-to-Point Proto-

koll (allgemein mit "PPP" abgekürzt), falls die Vorrichtung **12(m)** über eine Point-to-Point Verbindung mit dem ISP **11** verbunden ist, oder irgendein konventionelles "Multi-Drop" Protokoll, falls die Vorrichtung **12(m)** mit dem ISP **11** über ein "Multi-Drop"-Netzwerk, z. B. das Ethernet, verbunden ist, oder ähnliches. Die Vorrichtungen **12(m)** sind im allgemeinen entsprechend der üblichen Computerarchitektur mit gespeicherten Programmen aufgebaut, welche z. B. eine Systemeinheit, eine Bildschirmanzeigeeinheit und Bedieneingabeeinrichtungen, wie etwa eine Tastatur oder eine Maus, umfaßt. Eine Systemeinheit weist im allgemeinen eine oder mehrere Prozessor-, Speicher-, Massenspeichereinrichtungen, z. B. Festplatten- und/oder Bandspeicherelemente, oder andere Elemente (nicht separat gezeigt) auf, wie etwa Netzwerk- und/oder Telefonschnittstelleneinrichtungen, um die jeweilige Vorrichtung an den ISP **11** anzukoppeln. Die Prozessor- bzw. Verarbeitungseinrichtungen verarbeiten Programme, einschließlich Anwendungsprogramme, unter der Steuerung eines Betriebssystemes, um verarbeitete Daten zu erzeugen. Die Bildschirmeinheit ermöglicht es der Vorrichtung, die verarbeiteten Daten und einen Verarbeitungsstatus der Daten dem Benutzer anzuzeigen, und die Bedieneingabeeinrichtung ermöglicht es dem Bediener, Daten einzugeben und die Verarbeitung zu steuern.

Diese Elemente der Vorrichtung **12(m)** arbeiten in Verbindung mit einer geeigneten Programmierung so zusammen, um eine Vorrichtung **12(m)** mit einer Anzahl von funktionellen Elementen bereit zustellen, beispielsweise eine Bedienerchnittstelle **20**, eine Netzwerkschnittstelle **21**, einen Nachrichtenpaketgenerator **22**, einen Nachrichtenpaketempfänger und -prozessor **23**, eine ISP Einloggsteuerung bzw. Anmeldesteuerung **24**, einen Internetparameterspeicher **25** und im Zusammenhang mit der vorliegenden Erfindung einen Sicherheits-Nachrichtenpaketprozessor **26**. Die Bedienerchnittstelle **20** ermöglicht, daß die Vorrichtung **12(m)** Eingabeinformationen von der/den Bedieneingabeeinrichtung(en) der Vorrichtung **12(m)** empfängt und die Ausgabeinformationen dem Bediener auf der/den Bildschirmeinrichtung(en) der Vorrichtung **12(m)** angezeigt werden. Die Netzwerkschnittstelle **21** ermöglicht eine Verbindung der Vorrichtung **12(m)** mit dem ISP **11** unter Verwendung des geeigneten PPP oder Netzwerkprotokolls, um Nachrichtenpakete an den ISP **11** zu übertragen und von diesem Nachrichtenpakete zu empfangen. Die Netzwerkschnittstelle **21** kann eine Verbindung mit dem ISP **11** über das öffentliche Telefonnetz vorsehen, um einen Wählverbindungsnetzwerkbetrieb (sog. Dial-Up Betrieb) der Vorrichtung **12(m)** über das öffentliche Telefonnetz zu ermöglichen. Alternativ oder zusätzlich dazu kann die Netzwerkschnittstelle **21** eine Verbindung durch den ISP **11** über beispielsweise ein konventionelles LAN ermöglichen, wie etwa das Ethernet. In Reaktion auf eine durch die Bedienerchnittstelle **20** gelieferte Eingabe und/oder in Reaktion auf Anfragen aus Programmen (nicht gezeigt), welche durch die Vorrichtung **12(m)** verarbeitet werden, kommuniziert die ISP Einloggsteuerung **24** über die Netzwerkschnittstelle **21**, um die Initialisierung (sog. "Log-On") einer Kommunikationssitzung zwischen der Vorrichtung **12(m)** und dem ISP **11** zu ermöglichen. Während dieser Kommunikationssitzung kann die Vorrichtung **12(m)** Information in der Form von Nachrichtenpaketen an andere Vorrichtungen über das Internet **14** sowie an andere Vorrichtungen **12(m')** (wobei  $m' \neq m$ ), welche mit der ISP **11** oder mit anderen ISPs verbunden sind, übertragen. Während eines Log-On-Betriebs empfängt die ISP Einloggsteuerung **24** die Internetprotokollparameter (IP-Parameter), welche im Zusammenhang mit einer Nachrichtenpaketezeugung während der Kommunikationssitzung verwendet werden.

Während einer Kommunikationssitzung erzeugt der Nachrichtenpaketgenerator **22** Nachrichtenpakete zur Übertragung durch die Netzwerkschnittstelle **21** in Reaktion auf eine Eingabe, welche durch den Bediener über die Bedienerchnittstelle **20** geliefert wird und/oder in Reaktion auf Anfragen aus Programmen (nicht separat gezeigt), welche durch die Vorrichtung **12(m)** verarbeitet werden. Die Netzwerkschnittstelle **21** empfängt auch Nachrichtenpakete aus dem ISP **11** und liefert diese an den Nachrichtenpaketempfänger und -prozessor **23** zur Verarbeitung und Bereitstellung an die Bedienerchnittstelle **20** und/oder anderen Programmen (nicht gezeigt), welche durch die Vorrichtung **12(m)** verarbeitet werden. Falls die empfangenen Nachrichtenpakete eine Information enthalten, z. B. Web-Seiten oder ähnliches, welche dem Bediener angezeigt werden soll, kann die Information der Bedienerchnittstelle **20** geliefert werden, damit die Information auf der Bildschirmeinheit der Vorrichtung angezeigt wird. Zusätzlich oder alternativ dazu kann die Information an andere Programme (nicht gezeigt) zur Verarbeitung geliefert werden, welche durch die Vorrichtung **12(m)** verarbeitet werden.

Im allgemeinen können die Elemente, wie die Bedienerchnittstelle **20**, der Nachrichtenpaketgenerator **22**, der Nachrichtenpaketempfänger und -prozessor **23**, die ISP Einloggsteuerung **24** und der Internetparameterspeicher **25** Elemente eines konventionellen Internet-Browsers enthalten, wie die von Mosaic, Netscape Navigator und Microsoft Internet Explorer.

Wie es oben erwähnt wurde, weist die Vorrichtung **12(m)** im Zusammenhang mit der vorliegenden Erfindung einen Sicherheits-Nachrichtenpaketprozessor **26** auf. Der Sicherheits-Nachrichtenpaketprozessor **26** ermöglicht den Aufbau und Verwendung eines "Sicherheitstunnels" zwischen der Vorrichtung **12(m)** und anderen Vorrichtungen **12(m')** (wobei  $m' \neq m$ ) oder **13**, wie es welches weiter unten beschrieben wird. Im allgemeinen wird in einem solchen Sicherheitstunnel Information in wenigstens dem Datenabschnitt der zwischen der Vorrichtung **12(m)** und einer spezifischen anderen Vorrichtung **12(m')** (wobei  $m' \neq m$ ) oder **13** übertragenen Nachrichtenpakete geheimgehalten, beispielsweise durch Verschlüsselung des Datenabschnittes vor der Übertragung durch die Quellenvorrichtung. Die Information in anderen Abschnitten eines derartigen Nachrichtenpakets kann ebenfalls geheimgehalten werden, mit Ausnahme der Information, welche benötigt wird, um die Übertragung des jeweiligen Nachrichtenpakets zwischen den Vorrichtungen zu ermöglichen, also z. B. wenigstens die Zielinformation, damit die Schaltungsknoten des Internets und die ISPs die Vorrichtung identifizieren können, welche das Nachrichtenpaket empfangen soll.

Zusätzlich zu dem ISP **11** kann eine Vielzahl von anderen ISPs die Verbindung zum Internet herstellen, wie es durch die Pfeile **16** angedeutet ist, um eine Kommunikation zwischen Vorrichtungen, welche an diesen anderen ISPs angeschlossen sind, mit anderen Vorrichtungen über das Internet zu ermöglichen, welche die Vorrichtungen **12(n)**, welche an dem ISP **11** angeschlossen sind, umfassen können.

Die Vorrichtungen **13**, auf welche die Vorrichtungen **12(m)** zugreifen und mit welchen diese kommunizieren, können auch von jeder beliebigen Anzahl von Arten von Vorrichtungen sein, einschließlich Personalcomputer, Computer-Workstations und ähnliches, oder auch Minicomputer und Großrechner, Großspeichersysteme, Rechnerserver, lokale Netzwerke (LANs) und Fernverbindungsnetzwerke (WANs), welche derartige Vorrichtungen und zahlreiche andere Arten von Vorrichtungen enthalten, die direkt oder indirekt mit den Netzwerken verbunden werden können. Nach der vorliegenden Erfindung umfaßt wenigstens eine der Vor-

richtungen wenigstens ein privates Netzwerk, welches als virtuelles privates Netzwerk 15 gekennzeichnet ist und z. B. die Form eines LAN oder eines WAN haben kann. Das virtuelle private Netzwerk 15 kann jede der Vorrichtungen 12(m') (wobei  $m' \neq m$ ) aufweisen (wobei die Verbindung zu dem Internet 14 über einen ISP erfolgt) oder der Vorrichtungen 13 (wobei die Verbindung zu dem Internet 14 unmittelbar erfolgt). Bei dem vorliegend beschriebenen Ausführungsbeispiel wird angenommen, daß das virtuelle Netzwerk 15 eine Vorrichtung 13 aufweist. Das virtuelle private Netzwerk 15 umfaßt selbst mehrere Vorrichtungen, welche hier als eine Firewall bzw. ein Firewall-System 30, mehrere Server 31(1) bis 31(S) (im nachfolgenden allgemein mit dem Bezugszeichen 31(s) angegeben) und ein Namen-Server 32 gekennzeichnet sind, wobei allesamt durch eine Übertragungsverbindung 33 miteinander verbunden sind. Die Firewall 30 und die Server 31(s) können ähnlich sein wie jede der verschiedenen Arten von Vorrichtungen 12(m) und 13, die hier beschrieben sind, und können daher beispielsweise umfassen Personalcomputer, Computer-Workstations und ähnliches, aber auch Minicomputer und Großrechner, Großspeichersysteme, Rechnerserver, lokale Netzwerke (LANs) und Fernverbindungsnetzwerke (WANs), welche derartige Vorrichtungen und zahlreiche andere Arten von Vorrichtungen umfassen, welche direkt oder indirekt mit den Netzwerken verbunden werden können.

Wie oben ausgeführt wurde, kommunizieren diese Vorrichtungen einschließlich der Vorrichtungen 12(m) und der Vorrichtungen 13 durch Übertragung von Nachrichtenpaketen über das Internet. Die Vorrichtungen 12(m) und 13 können Information in einem Peer-to-Peer bzw. gleichrangigem Modus, in einem Client-Server Modus oder nach beiden dieser Modi übertragen. Im allgemeinen überträgt eine Vorrichtung in einer Peer-to-Peer Nachrichtenpaketübertragung Information in einem oder mehreren Nachrichtenpaketen an die andere Vorrichtung. Andererseits kann eine Vorrichtung, welche in einem Client-Server Modus als Client fungiert, ein Nachrichtenpaket an eine andere Vorrichtung übertragen, welche als Server fungiert, um beispielsweise einen Dienst durch die andere Vorrichtung auszulösen. Mehrere Arten derartiger Dienste sind dem Fachmann bekannt, beispielsweise das Wiedergewinnen bzw. Auslesen von Information aus der anderen Vorrichtung, damit diese aktiviert wird, um Verarbeitungsoperationen und dergleichen durchzuführen. Falls der Server dazu dient, dem Client vor allem Informationen zu liefern, kann dieser allgemein als ein Speicherserver bezeichnet werden. Falls der Server andererseits Verarbeitungsoperationen auf Anfrage des Client ausführen soll, kann dieser allgemein als ein Rechnerserver bezeichnet werden. Andere Arten von Servern zum Ausführen von anderen Arten von Diensten und Operationen auf Anfrage von Clients sind dem Fachmann ebenfalls bekannt.

Wenn in einer Client-Server Anordnung eine Vorrichtung 12(m) einen Dienst durch beispielsweise eine Vorrichtung 13 ausgeführt haben möchte, erzeugt die Vorrichtung 12(m) eines oder mehrere Anfragenachrichtenpakete zur Übertragung an die Vorrichtung 13, welche den benötigten Dienst anfordern. Das Anfragenachrichtenpaket enthält die Internetadresse der Vorrichtung 13, welche als die Zielvorrichtung das Nachrichtenpaket empfängt und den Dienst ausführt. Die Vorrichtung 12(m) überträgt das/die Anfragenachrichtenpaket(e) an den ISP 11. Der ISP 11 überträgt daraufhin das Nachrichtenpaket über das Internet an die Vorrichtung 13.

Falls die Vorrichtung 13 die Form eines WAN oder LAN hat, empfängt das WAN oder LAN das/die Nachrichtenpaket(e) und leitet dieses/diese zu einer dort angeschlossenen Vorrichtung weiter, welche den angeforderten Dienst aus-

führen soll.

In jedem Fall wird die Vorrichtung 13, welche den angeforderten Dienst ausführen soll, nach Empfang des/der Anfragenachrichtenpaket(e) die Anfrage bearbeiten. Falls die Vorrichtung 12(m), welche das/die Anfragenachrichtenpaket(e) erzeugt hat, oder deren Bediener die notwendigen Befugnisse hat, um den Dienst von der Vorrichtung 13 anzufordern, und falls der angeforderte Dienst die Einleitung einer Informationsübertragung aus der Vorrichtung 13 als ein Speicherserver an die Vorrichtung 12(m) als ein Client umfaßt, erzeugt die Vorrichtung 13 eines oder mehrere Antwortnachrichtenpakete, welche die angeforderten Information enthalten, und überträgt das/die Paket(e) über das Internet 14 an den ISP 11. Daraufhin überträgt der ISP 11 das/die Nachrichtenpaket(e) an die Vorrichtung 12(m). Falls andererseits der angeforderte Dienst die Einleitung eines Verarbeitungsvorganges durch die Vorrichtung 13 als ein Rechnerserver beinhaltet, wird die Vorrichtung 13 den/die angeforderten Rechendienst(e) ausführen. Falls die Vorrichtung 13 verarbeitete Daten, welche während den Rechenvorgängen erzeugt wurden, an die Vorrichtung 12(m) als Client zurücksenden soll, erzeugt die Vorrichtung 13 zusätzlich eines oder mehrere Antwortnachrichtenpakete, welche die verarbeiteten Daten enthalten und überträgt das/die Paket(e) über das Internet 14 an den ISP 11. Der ISP 11 überträgt daraufhin das/die Nachrichtenpaket(e) an die Vorrichtung 12(m). Entsprechende Operationen können durch die Vorrichtungen 12(m) und 13, dem ISP 11 und dem Internet 14 in Verbindung mit anderen Arten von Diensten ausgeführt werden, welche durch die Server-Vorrichtungen 13 bereitgestellt werden können.

Wie oben angemerkt wurde, enthält jedes Nachrichtenpaket, welches durch die Vorrichtungen 12(m) und 13 zur Übertragung über das Internet 14 erzeugt wird, eine Zieladresse, welche von den Schaltungsknoten verwendet wird, um das jeweilige Nachrichtenpaket an die geeignete Zielvorrichtung zu leiten. Adressen im Internet haben die Form von "n"-Bit Zahlen (wobei "n" beim gegenwärtigen Standard 32 oder 128 sein kann). Um insbesondere einen Bediener einer Vorrichtung 12(m) von der Notwendigkeit zu befreien, sich spezifische Zahlenkolonnen bzw. Zahlen-Internetadressen zu merken und diese der Vorrichtung 12(m) einzugeben, um die Erzeugung eines Nachrichtenpakets zur Übertragung über das Internet einzuleiten, stellt das Internet einen zweiten Adressierungsmechanismus zur Verfügung, welcher einfacher durch menschliche Bediener der jeweiligen Vorrichtungen handhabbar ist. Bei diesem Adressierungsmechanismus werden Internet-Domains, wie etwa LANs, Internet-Service-Provider (ISPs) und ähnliche, welche in bzw. mit dem Internet verbunden sind, durch relativ einfach les- und merkbare Namen, sog. Klartextnamen, identifiziert. Dabei soll sich hier die Bezeichnung "Klartextname" auf jede Art von Namenstext beziehen, z. B. auch auf Abkürzungen, generische Bezeichnungen, Phantasiebegriffe, etc. Um das System der Klartext-Domainnamen umzusetzen, ist der ISP 11 mit einem Namen-Server 17 (der auch als ein DNS Server (Domain Name Server) bezeichnet werden kann) verbunden, welcher die Klartext-Domainnamen auflösen bzw. in eine gültige Internetadresse umwandeln kann, um die geeignete Internetadresse für das in dem jeweiligen Klartextnamen angegebene Ziel bereitzustellen. Im allgemeinen kann der Namen-Server ein Teil des ISP 11 oder damit direkt verbunden sein, wie es in Fig. 1 gezeigt ist, oder er kann eine bestimmte Vorrichtung sein, welche durch den ISP über das Internet zugänglich ist. Jednfalls wenn sich die Vorrichtung 12(m) bei dem ISP 11 während einer Kommunikationssitzung einloggt, wird der ISP 11, wie oben hingewiesen wurde, verschiedene Internet-Proto-

kollparameter (IP-Parameter) zuordnen, welche die Vorrichtung 12(m) während der Kommunikationssitzung verwendet, und welche in dem Internetparameterspeicher 25 gespeichert sind. Diese IP-Parameter enthalten Informationen, wie

- (a) eine Internetadresse für die Vorrichtung 12(m), welche die Vorrichtung 12(m) während der Kommunikationssitzung identifiziert; und
- (b) die Identifizierung eines Namen-Servers 17, welchen die Vorrichtung 12(m) während der Kommunikationssitzung verwendet.

Wenn die Vorrichtung 12(m) Nachrichtenpakete zur Übertragung erzeugt, fügt sie ihre Internetadresse (oberer Punkt (a)) als die Quellenadresse ein. Die Vorrichtung(en) 13, welche die jeweiligen Nachrichtenpakete empfängt/empfangen, kann/können die Quellenadresse aus den Nachrichtenpaketen, welche von der Vorrichtung 12(m) empfangen werden, in Nachrichtenpaketen verwenden, welche die Vorrichtung(en) 13 zur Übertragung an die Vorrichtung 12(m) erzeugt/erzeugen, so daß das Internet in der Lage ist, die durch die jeweilige Vorrichtung 13 erzeugten Nachrichtenpakete an die Vorrichtung 12(m) zu leiten. Falls die Vorrichtung 12(m) auf den Namen-Server 17 über das Internet 14 zugreift, hat die durch den ISP 11 bereitgestellte Identifizierung des Namen-Servers 17 (siehe oben unter (b)) die Form einer Zahlen-Internetadresse, welche es der Vorrichtung 12(m) ermöglicht, für den Namen-Server 17 Nachrichten zu erzeugen, welche eine Auflösung der Klartext-Internetadressen in Zahlen-Internetadressen anfordern. Der ISP 11 kann der Vorrichtung 12(m) auch andere IP-Parameter zuordnen, wenn diese sich beim ISP 11 einloggt, beispielsweise die Identifizierung einer Verbindung zu dem Internet 14, welche für Nachrichten zu verwenden ist, die durch die Vorrichtung 12(m) übersandt werden, insbesondere falls der ISP 11 Mehrfach-Gateways aufweist. In der Regel speichert die Vorrichtung 12(m) die Internetparameter im Internetparameterspeicher 25 für die Verwendung während der Kommunikationssitzung.

Wenn ein Bediener die Vorrichtung 12(m) veranlassen möchte, daß sie ein Nachrichtenpaket an eine Vorrichtung 13 überträgt gibt der oder die Bediener(in) die Internetadresse der Vorrichtung 13 an die Vorrichtung 12(m) über die Bedienerchnittstelle 20 ein, sowie eine Information oder die Identifizierung der in der Vorrichtung 12(m) aufbewahrten Information, welche in der Nachricht übertragen werden sollen. Die Bedienerchnittstelle 20 aktiviert daraufhin den Paketgenerator 22 zur Freigabe der benötigten Pakete zur Übertragung durch den ISP 11 über das Internet 14. Falls

- (i) der Bediener die Zahlen-Internetadresse bereitgestellt hat, oder
- (ii) der Bediener die Klartext-Internetadresse bereitgestellt hat, aber der Paketgenerator 22 bereits die Zahlen-Internetadresse besitzt, welche der durch den Bediener eingegebenen Klartext-Internetadresse entspricht,

kann der Paketgenerator 22 unmittelbar nach Aktivierung durch die Bedienerchnittstelle 20 die Pakete erzeugen und diese an die Netzwerkschnittstelle 21 zur Übertragung an den ISP 11 liefern.

Falls aber der Bediener die Klartext-Internetadresse der Vorrichtung 13, an welche die Pakete zu übertragen sind, eingegeben hat, und falls der Paketgenerator 22 die entsprechende Zahlen-Internetadresse davon nicht bereits besitzt,

ermöglicht es der Paketgenerator 22, daß die Netzwerkadresse von dem Namen-Server 17, der in dem IP-Parameterspeicher 25 identifiziert ist, erhalten wird.

Bei diesem Vorgang wird der Paketgenerator 22 anfänglich den Namen-Server 17 kontaktieren, um zu versuchen, die geeignete Zahlen-Internetadresse von dem Namen-Server 17 zu erhalten. Bei diesem Vorgang wird die Vorrichtung 12(m) geeignete Nachrichtenpakete zur Übertragung an den Namen-Server 17 unter Verwendung der Zahlen-Internetadresse des Namen-Servers 17 erzeugen, welche durch den ISP 11 bereitgestellt wird, wenn sich die Vorrichtung 12(m) zu Beginn der Kommunikationssitzung einloggt. Jedenfalls wenn der Namen-Server 17 die Zahlen-Internetadresse für den Klartextnamen besitzt oder erhalten kann, wird der Namen-Server 17 die Zahlen-Internetadresse an die Vorrichtung 12(m) übermitteln. Die Zahlen-Internetadresse wird durch den Paketgenerator 22 über die Netzwerkschnittstelle 21 und den Paketempfänger und -prozessor 23 empfangen. Nachdem der Paketgenerator 22 die Zahlen-Internetadresse empfangen hat, kann er die notwendigen Nachrichtenpakete zur Übertragung an die Vorrichtung 13 durch die Netzwerkschnittstelle 21 und den ISP 11 erzeugen.

Wie oben ausgeführt wurde, ist in Fig. 1 eine der Vorrichtungen 13, welche an das Internet 14 angeschlossen sind, ein virtuelles privates Netzwerk 15, wobei das virtuelle private Netzwerk 15 eine Firewall bzw. ein Firewall-System 30, mehrere als Server 31(s) gekennzeichnete Vorrichtungen und einen Namen-Server 32 aufweist, die durch eine Übertragungsverbindung 33 miteinander verbunden sind. Die Server 31(s), die Firewall 30 und der Namen-Server 32 können als z. B. in einem LAN oder WAN verbundene Vorrichtungen untereinander Information in Form von Nachrichtenpaketen austauschen. Da die Firewall 30 mit dem Internet 14 verbunden ist und darüber Nachrichtenpakete empfangen kann, hat sie auch eine Internetadresse. Zusätzlich haben wenigstens die Server 31(s), welche über das Internet zugänglich sind, auch jeweilige Internetadressen. Dabei dient der Namen-Server 32 der Umwandlung von Klartext-Internetadressen für die Server 31(s) innerhalb des virtuellen privaten Netzwerkes 15 in die jeweiligen Zahlen-Internetadressen.

Im allgemeinen wird das virtuelle private Netzwerke 15 von einem Unternehmen, einem Regierungsamt, einer Organisation oder ähnlichem gehalten, welche möchten, daß die Server 31(s) Zugriff auf andere Vorrichtungen außerhalb des virtuellen privaten Netzwerkes 15 haben und an diese Information über das Internet 14 übertragen können, aber welche ebenfalls möchten, daß der Zugriff an die Server 31(s) durch Vorrichtungen 12(m) und andere externe Vorrichtungen über das Internet 14 in einer kontrollierten Weise begrenzt ist. Die Firewall 30 dient dazu, den Zugriff durch Vorrichtungen außerhalb des virtuellen privaten Netzwerkes 15 auf Server 31(s) innerhalb des virtuellen privaten Netzwerkes 15 zu kontrollieren. Bei diesem Vorgang stellt die Firewall 30 auch die Verbindung zum Internet 14 her und empfängt Nachrichtenpakete darüber zur Übertragung an einen Server 31(s). Falls das Nachrichtenpaket angibt, daß die Quelle des Nachrichtenpaketes einen Zugriff auf einen bestimmten Server 31(s) anfordert, und falls die Quelle für den Zugriff an den Server 31(s) autorisiert ist, sendet die Firewall 30 das Nachrichtenpaket über die Übertragungsverbindung 33 an den Server 31(s). Falls andererseits die Quelle nicht autorisiert ist, auf den Server 31(s) zuzugreifen, wird die Firewall 30 das Nachrichtenpaket nicht an den Server 31(s) übersenden, und kann anstelle ein Antwortnachrichtenpaket an die Quellenvorrichtung übermitteln, welches angibt, daß die Quelle nicht für den Zugriff an den Server 31(s) autorisiert ist. Die Firewall kann ähnlich aufgebaut sein wie die ande-



ren Vorrichtungen 31(s) in dem virtuellen privaten Netzwerk 15, wobei zusätzlich eine oder mehrere Verbindungen mit dem Internet vorhanden sind, welche allgemein durch das Bezugszeichen 43 gekennzeichnet sind.

Kommunikationen zwischen Vorrichtungen außerhalb des virtuellen privaten Netzwerkes 15, z. B. der Vorrichtung 12(m), und einer Vorrichtung, z. B. einem Server 31(s), innerhalb des virtuellen privaten Netzwerkes 15 kann über einen Sicherheitstunnel zwischen der Firewall 30 und der externen Vorrichtung, wie es oben beschrieben ist, erreicht werden, damit die ausgetauschten Information geheim bleiben, während diese über das Internet 14 und durch den ISP 11 übertragen werden. Ein Sicherheitstunnel zwischen der Vorrichtung 12(m) und dem virtuellen privaten Netzwerk 15 ist in Fig. 1 durch logische Verbindungen dargestellt, welche durch die Bezugszeichen 40, 42 und 44 gekennzeichnet sind; es versteht sich, daß die logische Verbindung 42 eine der logischen Verbindungen 41 zwischen dem ISP 11 und dem Internet 14 und die logische Verbindung 44 eine der logischen Verbindungen 43 zwischen dem Internet 14 und der Firewall 30 umfaßt.

Der Aufbau eines Sicherheitstunnels kann durch eine Vorrichtung 12(m), die extern zu dem virtuellen privaten Netzwerk 15 ist, ausgelöst werden. Bei diesem Vorgang erzeugt die Vorrichtung 12(m) in Reaktion auf eine Aufforderung durch deren Bediener ein Nachrichtenpaket zur Übertragung durch den ISP 11 und das Internet 14 an die Firewall 30, welches den Aufbau eines Sicherheitstunnels zwischen der Vorrichtung 12(m) und der Firewall 30 anfordert. Das Nachrichtenpaket kann an eine bestimmte Zahlen-Internetadresse gerichtet sein, welche der Firewall 30 zugeordnet ist und welche für Sicherheitstunnelaufbauanfragen reserviert ist, und welche ferner der Vorrichtung 12(m) bekannt ist und durch den Namen-Server 17 bereitgestellt wird. Falls die Vorrichtung 12(m) autorisiert ist, auf einen Server 31(s) in dem virtuellen privaten Netzwerk 15 zuzugreifen, nehmen die Vorrichtung 12(m) als Client und die Firewall 30 einen Dialog auf, welcher den Austausch von einem oder mehreren Nachrichtenpaketen über das Internet 14 umfaßt. Während des Dialogs kann die Firewall 30 der Vorrichtung 12(m) die Identifizierung eines Entschlüsselungsalgorithmus und einen zugehörigen Entschlüsselungsschlüssel bereitstellen, welche die Vorrichtung 12(m) beim Entschlüsseln der verschlüsselten Abschnitte der Nachrichtenpakete zu verwenden hat, welche das virtuelle private Netzwerk an die Vorrichtung 12(m) überträgt. Zusätzlich dazu kann die Firewall 30 der Vorrichtung 12(m) auch die Identifizierung eines Verschlüsselungsalgorithmus und einen zugehörigen Verschlüsselungsschlüssel bereitstellen, welche die Vorrichtung 12(m) beim Verschlüsseln der Abschnitte der Nachrichtenpakete zu verwenden hat, welche die Vorrichtung 12(m) an das virtuelle private Netzwerk 15 überträgt und welche verschlüsselt werden sollen. Alternativ dazu kann die Vorrichtung 12(m) die Identifizierung des Verschlüsselungsalgorithmus und des Verschlüsselungsschlüssels, welche die Vorrichtung 12(m) verwenden wird, an die Firewall 30 während des Dialogs liefern. Die Vorrichtung 12(m) kann in ihrem IP-Parameterspeicher 25 Informationen betreffend den Sicherheitstunnel speichern, einschließlich der Information in Verbindung mit der Identifizierung der Firewall 30 und der Identifizierungen der Verschlüsselungs- und Entschlüsselungsalgorithmen und dazugehöriger Schlüssel für Nachrichtenpakete, welche durch den Sicherheitstunnel übertragen werden.

Sodann können die Vorrichtung 12(m) und die Firewall 30 Nachrichtenpakete über den Sicherheitstunnel übertragen. Beim Erzeugen von Nachrichtenpaketen zur Übertragung über den Sicherheitstunnel verwendet die Vorrichtung

12(m) den Sicherheits-Paketprozessor 26, um die Abschnitte der Nachrichtenpakete zu verschlüsseln, welche vor der Übertragung durch die Netzwerkschnittstelle 21 an den ISP 11 zur Übertragung über das Internet 14 an die Firewall 30 verschlüsselt werden sollen, und um die verschlüsselten Abschnitte der Nachrichtenpakete zu entschlüsseln, welche durch die Vorrichtung 12(m) empfangen werden und welche verschlüsselt sind. Insbesondere nachdem der Paketgenerator 22 ein Nachrichtenpaket zur Übertragung an die Firewall 30 über den Sicherheitstunnel erzeugt hat, liefert er das Nachrichtenpaket an den Sicherheits-Paketprozessor 26. Der Sicherheits-Paketprozessor 26 verschlüsselt daraufhin die Abschnitte des Nachrichtenpakets, welche verschlüsselt werden sollen, unter Verwendung des Verschlüsselungsalgorithmus und des Verschlüsselungsschlüssels. Nachdem die Firewall 30 ein Nachrichtenpaket von der Vorrichtung 12(m) über den Sicherheitstunnel empfangen hat, wird sie dieses entschlüsseln und, falls der beabsichtigte Empfänger des Nachrichtenpakets eine andere Vorrichtung, z. B. ein Server 31(s), in dem virtuellen privaten Netzwerk 15 ist, wird die Firewall 30 das Nachrichtenpaket an diese andere Vorrichtung über die Übertragungsverbindung 33 übertragen.

Wenn ein Nachrichtenpaket von einer Vorrichtung, z. B. einem Server 31(s), in dem virtuellen privaten Netzwerk 15 an die Vorrichtung 12(m) über den Sicherheitstunnel übertragen werden soll, empfängt die Firewall 30 ein solches Nachrichtenpaket über die Übertragungsverbindung 33 und verschlüsselt das Nachrichtenpaket zur Übertragung über das Internet 14 an den ISP 11. Der ISP 11 sendet daraufhin das Nachrichtenpaket an die Vorrichtung 12(m), insbesondere an deren Netzwerkschnittstelle 21. Die Netzwerkschnittstelle 21 liefert das Nachrichtenpaket an den Sicherheits-Paketprozessor 26, welcher die verschlüsselten Abschnitte des Nachrichtenpakets unter Verwendung des Entschlüsselungsalgorithmus und -schlüssels entschlüsselt.

Ein Problem tritt auf im Zusammenhang mit Zugriffen durch eine Vorrichtung, z. B. einer Vorrichtung 12(m), welche extern zum virtuellen privaten Netzwerk 15 ist, und einer Vorrichtung, z. B. einem Server 31(s), welche extern zu der Firewall ist, nämlich dann, wenn dem Namen-Server 17 keine Zahlen-Internetadressen für die Server 31(s) und andere Vorrichtungen bereitgestellt sind, die sich innerhalb des virtuellen privaten Netzwerkes 15 befinden – mit Ausnahme der Zahlen-Internetadressen, welche der Firewall 30 zugeordnet sind. Folglich wird die Vorrichtung 12(m) nach Eingabe der Klartext-Internetadresse durch den Bediener nicht in der Lage sein, die Zahlen-Internetadresse des Servers 31(s) zu erhalten, wenn er auf den Namen-Server 17 zugreift.

Wenn die Vorrichtung 12(m) und die Firewall 30 zusammenarbeiten, um einen dazwischenliegenden Sicherheitstunnel aufzubauen, liefert die Firewall 30 zur Behebung des obigen Problems an die Vorrichtung 12(m) zusätzlich zu möglichen Identifikationen der Verschlüsselungs- und Entschlüsselungsalgorithmen und -schlüsseln, welche im Zusammenhang mit der Übertragung der Nachrichtenpakete über den Sicherheitstunnel zu verwenden sind, an die Vorrichtung 12(m) auch die Identifizierung eines Namen-Servers, z. B. eines Namen-Servers 32, innerhalb des virtuellen privaten Netzwerkes 15, auf welchen die Vorrichtung 12(m) zugreifen kann, um die geeigneten Zahlen-Internetadressen für die Klartext-Internetadressen zu erhalten, welche durch den Bediener einer Vorrichtung 12(m) eingegeben werden. Die Identifizierung des Namen-Servers 32 wird ebenfalls in dem IP-Parameterspeicher 25 gespeichert, zusammen mit der Identifizierung des Namen-Servers 17, welche durch den ISP 11 bereitgestellt wurde, sobald die Vorrichtung 12(m)

beim ISP 11 zu Beginn einer Kommunikationssitzung eingeloggt wurde. Wenn daher die Vorrichtung 12(m) ein Nachrichtenpaket an eine Vorrichtung, z. B. einen Server 31(s), in dem virtuellen privaten Netzwerk 15 unter Verwendung einer Klartext-Internetadresse übertragen möchte, welche z. B. durch einen Bediener bereitgestellt bzw. eingegeben wurde, greift die Vorrichtung 12(m) zu Beginn auf den Namen-Server 17 zu, wie es oben beschrieben wurde, um zu versuchen, die zu der Klartext-Internetadresse zugehörige Zahlen-Internetadresse zu erhalten. Da der Namen-Server 17 außerhalb des virtuellen privaten Netzwerkes 15 ist und die durch die Vorrichtung 12(m) angeforderten Information nicht besitzt, sendet er ein entsprechend lautendes Antwortnachrichtenpaket. Die Vorrichtung 12(m) wird sodann ein Anfragennachrichtenpaket zur Übertragung an den Namen-Server 32 durch die Firewall 30 und über den Sicherheitstunnel erzeugen. Falls der Namen-Server 32 eine Zahlen-Internetadresse besitzt, welche zu der Klartext-Internetadresse in dem Anfragennachrichtenpaket gehört, welches durch die Vorrichtung 12(m) geliefert wird, stellt er die Zahlen-Internetadresse in einer Weise bereit, welche im allgemeinen derjenigen ähnlich ist, welche oben im Zusammenhang mit dem Namen-Server 17 beschrieben wurde mit der Ausnahme, daß die Zahlen-Internetadresse durch den Namen-Server 32 in einem an die Firewall 30 gerichteten Nachrichtenpaket geliefert wird, und die Firewall 30 sodann das Nachrichtenpaket über den Sicherheitstunnel an die Vorrichtung 12(m) übermittelt. Es versteht sich, daß sich in dem Nachrichtenpaket, welches durch die Firewall 30 übertragen wird, die Zahlen-Internetadresse in dem Nachrichtenpaket im Datenabschnitt des Nachrichtenpakets befindet, welches über den Sicherheitstunnel übertragen wird und entsprechend verschlüsselt sein wird. Das Nachrichtenpaket wird durch die Vorrichtung 12(m) in einer ähnlichen Weise verarbeitet, wie sie oben im Zusammenhang mit anderen Nachrichtenpaketen beschrieben wurde, welche durch die Vorrichtung 12(m) über den Sicherheitstunnel empfangen werden. Das heißt, daß das Nachrichtenpaket durch den Sicherheits-Paketprozessor 26 vor dem Übermitteln an den Paketempfänger und -prozessor 23 zur Verarbeitung entschlüsselt wird. Die Zahlen-Internetadresse für den Server 31(s) kann in einem Cache in einer Zugriffskontrollliste (ACL) in dem IP-Parameterspeicher 25 gespeichert werden, zusammen mit der Zuordnungsinformation bezüglich der zugehörigen Klartext-Internetadresse, einer Angabe, daß der Server 31(s), der dieser Klartext-Internetadresse zugeordnet ist, über die Firewall 30 des virtuellen privaten Netzwerkes 15 zugänglich ist, und die Identifizierungen der Verschlüsselungs- und Entschlüsselungsalgorithmen und -schlüssel, welche für eine Verschlüsselung und Entschlüsselung der geeigneten Abschnitte der Nachrichtenpakete zu verwenden sind, welche an den Server 31(s) übertragen und von diesem erhalten werden.

Es versteht sich, daß in Reaktion auf ein Nachrichtenpaket von der Vorrichtung 12(m), welches beim Namen-Server 32 die Bereitstellung einer Zahlen-Internetadresse für eine durch die Vorrichtung 12(m) angegebene Klartext-Internetadresse anfordert, falls der Namen-Server 32 keine Zuordnungsinformation zwischen der Klartext-Internetadresse und einer Zahlen-Internetadresse besitzt, der Namen-Server 32 ein Antwortnachrichtenpaket, das entsprechend lautet, übertragen kann. Falls die Vorrichtung 12(m) eine Identifizierung von anderen Namen-Servern besitzt, welche z. B. mit anderen virtuellen privaten Netzwerken (nicht gezeigt) verbunden sein können und zu welchen die Vorrichtung 12(m) Zugriff hat, dann kann die Vorrichtung 12(m) versuchen, auf die anderen Namen-Server in einer ähnlichen Weise, wie es oben beschrieben ist, zuzugreifen. Falls die

Vorrichtung 12(m) nicht in der Lage ist, eine Zahlen-Internetadresse, welche der Klartext-Internetadresse zugeordnet ist, von irgendeinem der Namen-Server zu erhalten, zu welchem sie Zugriff hat und welche im allgemeinen im IP-Parameterspeicher 25 der Vorrichtung 12(m) identifiziert sind, wird sie allgemein nicht in der Lage sein, auf eine Vorrichtung mit der vorgegebenen Klartext-Internetadresse zuzugreifen und wird den Bediener oder ein Programm, welche den Zugriff angefordert haben, dementsprechend unterrichten.

Mit diesem Hintergrund werden nun Operationen, welche durch die Vorrichtung 12(m) und das virtuelle private Netzwerk 15 in Verbindung mit der vorliegenden Erfindung durchgeführt werden, im Detail beschrieben. Im allgemeinen laufen die Operationen in zwei Phasen ab. In einer ersten Phase arbeiten die Vorrichtung 12(m) und das virtuelle private Netzwerk 15 zusammen, um einen Sicherheitstunnel durch das Internet 14 aufzubauen. In dieser ersten Phase liefert das virtuelle private Netzwerk 15, insbesondere die Firewall 30, die Identifizierung eines Namen-Servers 32, und es kann auch die den Verschlüsselungs- und Entschlüsselungsalgorithmus und -schlüssel betreffende Information bereitstellen, wie es oben beschrieben wurde. In der zweiten Phase, nachdem der Sicherheitstunnel eingerichtet wurde, kann die Vorrichtung 12(m) die während der ersten Phase gelieferten Information im Zusammenhang mit der Erzeugung und Übertragung von Nachrichtenpaketen an einen oder mehrere Server 31(s) in dem virtuellen privaten Netzwerk 15 und bei dem notwendigen Umwandlungsvorgang der Klartext-Internetadressen zu Zahlen-Internetadressen aus dem Namen-Server 32, welcher durch die Firewall 30 während der ersten Phase identifiziert wurde, verwenden.

Folglich erzeugt die Vorrichtung 12(m) in der ersten (Sicherheitstunnelaufbau)phase zu Beginn ein Nachrichtenpaket zur Übertragung an die Firewall 30, welches einen Aufbau eines Sicherheitstunnels anfordert. Das Nachrichtenpaket enthält eine Zahlen-Internetadresse für die Firewall, (welche durch den Bediener der Vorrichtung oder ein Programm bereitgestellt werden kann, welches durch die Vorrichtung 12(m) verarbeitet wird, oder durch den Namen-Server 17 bereitgestellt werden kann, nachdem eine Klartext-Internetadresse durch den Bediener oder ein Programm bereitgestellt wurde), und welche insbesondere dazu dient, die Firewall 30 zu veranlassen, mit der Vorrichtung 12(m) einen Sicherheitstunnel aufzubauen. Falls die Firewall 30 die Anfrage bezüglich des Sicherheitstunnelaufbaus akzeptiert und falls die Firewall 30 die Verschlüsselungs- und Entschlüsselungsalgorithmen und -schlüssel bereitstellt, so wie es oben angegeben wurde, erzeugt die Firewall 30 ein Antwortnachrichtenpaket zur Übertragung an die Vorrichtung 12(m), welches die Verschlüsselungs- und Entschlüsselungsalgorithmen und -schlüssel identifiziert. Wie oben beschrieben, wird dieses Antwortnachrichtenpaket nicht verschlüsselt. Wenn die Vorrichtung 12(m) die Antwort empfängt, werden die Identifizierungen der Verschlüsselungs- und Entschlüsselungsalgorithmen und -schlüssel in dem IP-Parameterspeicher 25 gespeichert.

Zu einem späteren Zeitpunkt in der ersten Phase erzeugt die Firewall 30 auch ein Nachrichtenpaket zur Übertragung an die Vorrichtung 12(m), welches die Zahlen-Internetadresse des Namen-Servers 32 enthält. Bei diesem Nachrichtenpaket wird der Abschnitt des Nachrichtenpakets, welcher die Zahlen-Internetadresse des Namen-Servers 32 enthält, unter Verwendung eines Verschlüsselungsalgorithmus und Verschlüsselungsschlüssels verschlüsselt, und dies kann unter Verwendung des Entschlüsselungsalgorithmus und -schlüssels, die durch das zuvor beschriebene Antwortnachrichtenpaket geliefert wurden, wieder entschlüsselt



werden. Diese Nachricht hat im allgemeinen die folgende Struktur:

```
"<IIA(FW),IIA(DEV_12(m))><SEC_TUN>
<ENCR<<IIA(FW),IIA(DEV_12(m))><DNS_ADRS:IIA(NS_2)>>>"
```

wobei

- (i) "IIA(FW)" die Quellenadresse darstellt, d. h. eine Zahlen-Internetadresse der Firewall 30,
- (ii) "IIA(DEV\_12(m))" die Zieladresse darstellt, d. h. die Zahlen-Internetadresse der Vorrichtung 12 (m),
- (iii) "DNS\_ADRS:IIA(NS)" angibt, daß "IIA(NS\_32)" die Zahlen-Internetadresse des Namen-Servers 32 darstellt, für dessen Benutzung die Vorrichtung 12(m) autorisiert ist, und
- (iv) "ENCR<...>" bedeutet, daß die Information, zwischen den Klammern "<" und ">" verschlüsselt ist.

Der Anfangsabschnitt der Nachricht "IIA(FW),IIA(DEV\_12(m))>" bildet wenigstens einen Teil des Kopfabschnitts der Nachricht, und "<ENCR<<IIA(FW),IIA(DEV\_12(m))><IIA(NS)>>>" stellt wenigstens einen Teil des Datenabschnitts der Nachricht dar. "<SEC\_TUN>" stellt einen Hinweis in dem Kopfabschnitt dar, welcher angibt, daß die Nachricht über den Sicherheitstunnel übertragen wird, wodurch auch angezeigt wird, daß der Datenabschnitt der Nachricht verschlüsselte Information enthält.

Nachdem die Vorrichtung 12(m) die Nachricht von der Firewall 30 empfängt, wie es oben beschrieben wurde, und weil das Nachrichtenpaket den <SEC\_TUN> Hinweis enthält, überträgt deren Netzwerkschnittstelle 21 den verschlüsselten Abschnitt "<ENCR<<IIA(FW),IIA(DEV\_12(m))><DNS\_ADRS:IIA(NS\_32)>>>" an den Sicherheits-Paketprozessor 26 zur Verarbeitung. Der Sicherheits-Paketprozessor 26 entschlüsselt den verschlüsselten Abschnitt, bestimmt weiter, daß der Abschnitt "IIA(NS\_32)" die Zahlen-Internetadresse des Namen-Servers darstellt, insbesondere des Namen-Servers 32, für dessen Benutzung die Vorrichtung 12(m) autorisiert ist, und speichert diese Adresse in dem IP-Parameterspeicher 25 zusammen mit einer Angabe, daß die dorthin gerichteten Nachrichtenpakete zu der Firewall 30 zu übertragen sind, und daß die Daten in den Nachrichtenpaketen unter Verwendung des Verschlüsselungsalgorithmus und -schlüssels, die davor durch die Firewall 30 übermittelt wurden, zu verschlüsseln sind. Es versteht sich, daß aufgrund der Tatsache, daß die Zahlen-Internetadresse des Namen-Servers 32 von der Firewall an die Vorrichtung 12(m) in verschlüsselter Form übertragen wird, diese vertraulich bleibt, selbst wenn das Paket durch einen Dritten abgefangen wird.

In Abhängigkeit des speziellen Protokolls, welches für den Aufbau des Sicherheitstunnels verwendet wird, können die Firewall 30 und die Vorrichtung 12(m) auch Nachrichtenpakete austauschen, welche andere Information enthalten als die oben beschriebenen.

Wie oben erwähnt wurde, kann die Vorrichtung 12(m) in der zweiten Phase nach der Einrichtung des Sicherheitstunnels die Information, welche während der ersten Phase bereitgestellt wurde, im Zusammenhang mit dem Erzeugen und Übertragen von Nachrichtenpaketen zu einem oder mehreren der Server 31(s) in dem virtuellen privaten Netzwerk 15 nutzen. Falls bei diesen Operationen der Bediener einer Vorrichtung 12(m) oder ein Programm, welches durch eine Vorrichtung 12(m) verarbeitet wird, möchte, daß die Vorrichtung 12(m) ein Nachrichtenpaket an einen Server

31(s) in dem virtuellen privaten Netzwerk 15 überträgt, und falls der Bediener durch die Bedienerchnittstelle 20 oder das Programm eine Klartext-Internetadresse bereitstellt, wird zunächst die Vorrichtung 12(m), insbesondere der Paketgenerator 22, bestimmen, ob der IP-Parameterspeicher 25 dort in einem Cache eine Zahlen-Internetadresse gespeichert hat, welche zu der Klartext-Internetadresse gehört. Falls dies nicht der Fall ist, erzeugt der Paketgenerator 22 ein Anfragennachrichtenpaket zur Übertragung an den Namen-Server 17, um von diesem die zu der Klartext-Internetadresse gehörige Zahlen-Internetadresse anzufordern. Falls der Namen-Server 17 eine zu der Klartext-Internetadresse gehörige Zahlen-Internetadresse besitzt, wird dieser die Zahlen-Internetadresse an die Vorrichtung 12(m) liefern. Es versteht sich, daß dies nur erfolgen kann, wenn die Klartext-Internetadresse im Anfragennachrichtenpaket sowohl einer Vorrichtung 13 außerhalb des virtuellen privaten Netzwerkes 15 als auch einem Server 32(s) in dem virtuellen privaten Netzwerk 15 zugeordnet wurde. Danach kann die Vorrichtung 12(m) die Zahlen-Internetadresse verwenden, um Nachrichtenpakete zur Übertragung über das Internet zu erzeugen, wie es oben beschrieben wurde.

Falls andererseits angenommen wird, daß der Namen-Server 17 keine der Klartext-Internetadresse zugeordnete Zahlen-Internetadresse besitzt, wird der Namen-Server 17 ein entsprechend lautendes Antwortnachrichtenpaket an die Vorrichtung 12(m) übermitteln. Sodann erzeugt der Paketgenerator 22 der Vorrichtung 12(m) ein Anfragennachrichtenpaket zur Übertragung an den nächsten Namen-Server, der in ihrem IP-Parameterspeicher 25 identifiziert ist, um von diesem Namen-Server die der Klartext-Internetadresse zugeordnete Zahlen-Internetadresse anzufordern. Falls dieser nächste Namen-Server der Namen-Server 32 ist, liefert der Paketgenerator 22 das Nachrichtenpaket an den Sicherheits-Paketprozessor 26 zur weiteren Verarbeitung. Der Sicherheits-Paketprozessor 26 erzeugt daraufhin ein Anfragennachrichtenpaket zur Übertragung über den Sicherheitstunnel an die Firewall 30. Diese Nachricht hat im allgemeinen folgende Struktur:

```
"<IIA(DEV_12(m)),IIA(FW)><SEC_TUN>
<ENCR<<IIA(DEV_12(m)),IIA(NS_32)>><IIA_REQ>>>"
```

wobei

- (i) "IIA(DEV\_12(m))" die Quellenadresse darstellt, d. h. die Zahlen-Internetadresse der Vorrichtung 12(m),
- (ii) "IIA(FW)" die Zieladresse darstellt, d. h. die Zahlen-Internetadresse der Firewall 30,
- (iii) "IIA(NS\_32)" die Adresse des Namen-Servers 32 darstellt,
- (iv) "<<IIA(DEV\_12(m)),IIA(NS\_32)>><IIA\_REQ>>>" das Anfragennachrichtenpaket darstellt, welches durch den Paketgenerator 22 erzeugt wird, wobei "<IIA(DEV\_12(m)),IIA(NS\_32)>" den Kopfabschnitt des Anfragennachrichtenpakets und "<IIA\_REQ>" den Datenabschnitt des Anfragennachrichtenpakets darstellt,
- (v) "ENCR<...>" angibt, daß die Information zwischen den Klammern "<" und ">" verschlüsselt ist, und
- (vi) "<SEC\_TUN>" einen Hinweis in dem Kopfabschnitt des Nachrichtenpakets darstellt, welches durch den Sicherheitspaketgenerator 26 erzeugt wird und angibt, daß die Nachricht über den Sicherheitstunnel übertragen wird, wobei hierdurch angegeben wird, daß der Datenabschnitt der Nachricht verschlüsselte Information enthält.

Wenn die Firewall 30 das durch den Sicherheitspaketgenerator 26 erzeugte Anfragennachrichtenpaket empfängt, wird diese den verschlüsselten Abschnitt des Nachrichtenpakets entschlüsseln, um " <<IIA(DEV\_12(m)),IIA(NS\_32)>><IIA\_REQ>>" zu erhalten. Dies stellt das Anfragennachrichtenpaket dar, welches durch den Paketgenerator 22 erzeugt wird. Nachdem das Anfragennachrichtenpaket erhalten wurde, überträgt die Firewall 30 dieses über die Übertragungsverbindung 33 an den Namen-Server 32. In Abhängigkeit von dem Protokoll zur Übertragung von Nachrichtenpaketen über die Übertragungsverbindung 33 kann es bei diesem Prozeß für die Firewall 30 notwendig sein, das Anfragennachrichtenpaket zu modifizieren, damit es dem Protokoll der Übertragungsverbindung 33 entspricht.

Nachdem der Namen-Server 32 das Anfragennachrichtenpaket erhalten hat, wird dieser das Anfragennachrichtenpaket verarbeiten, um zu bestimmen, ob er eine der Klartext-Internetadresse, welche in dem Anfragennachrichtenpaket gesendet wird, zugeordnete Zahlen-Internetadresse besitzt. Falls der Namen-Server feststellt, daß er eine solche Zahlen-Internetadresse aufweist, wird dieser ein Antwortnachrichtenpaket zur Übertragung an die Firewall erzeugen, welches die Zahlen-Internetadresse enthält. Im allgemeinen hat das Antwortnachrichtenpaket die folgende Struktur:

```
"<<IIA(NS_32),IIA(DEV_12(m))>><IIA_RESP>>"
```

wobei

- (i) "IIA(NS\_32)" die Quellenadresse darstellt, d. h. die Zahlen-Internetadresse des Namen-Servers 32,
- (ii) "IIA(DEV\_12(m))" die Zieladresse darstellt, d. h. die Zahlen-Internetadresse der Vorrichtung 12(m), und
- (iii) "IIA\_RESP" die Zahlen-Internetadresse darstellt, welche der Klartext-Internetadresse zugeordnet ist.

Nachdem die Firewall 30 das Antwortnachrichtenpaket empfangen hat, und weil die Kommunikation mit der Vorrichtung 12(m) über den dazwischenliegenden Sicherheitstunnel stattfindet, verschlüsselt die Firewall 30 das von dem Namen-Server 32 empfangene Antwortnachrichtenpaket und erzeugt ein Nachrichtenpaket zur Übertragung an die Vorrichtung 12(m), welches das verschlüsselte Antwortnachrichtenpaket enthält. Im allgemeinen hat das durch die Firewall 30 erzeugte Nachrichtenpaket die folgende Struktur:

```
"<IIA(FW),IIA(DEV_12(m))><SEC_TUN>  
<ENCR<<IIA(NS_32),IIA(DEV_12(m))>><IIA_RESP>>>"
```

wobei

- (i) "IIA(FW)" die Quellenadresse darstellt, d. h. die Zahlen-Internetadresse der Firewall 30,
- (ii) "IIA(DEV\_12(m))" die Zieladresse darstellt, d. h. die Zahlen-Internetadresse der Vorrichtung 12(m),
- (iii) "SEC\_TUN" einen Hinweis in dem Kopfabschnitt des Nachrichtenpakets darstellt, welches durch den Sicherheitspaketgenerator 26 erzeugt wird, und angibt, daß die Nachricht über den Sicherheitstunnel übertragen wird, und wobei auch angegeben wird, daß der Datenabschnitt der Nachricht verschlüsselte Information enthält,
- (iv) "ENCR<. . . >" angibt, daß die Information zwischen den Klammern "<" und ">" (was dem von dem Namen-Server 32 empfangenen Antwortnachrichten-

paket entspricht) verschlüsselt ist.

Zusätzlich kann es je nach dem Protokoll zur Übertragung von Nachrichtenpaketen über die Übertragungsverbindung 33 für die Firewall 30 notwendig sein, das Nachrichtenpaket zu bearbeiten und/oder zu modifizieren, damit dieses dem Protokoll des Internets 14 entspricht.

Wenn die Vorrichtung 12(m) das Nachrichtenpaket von der Firewall 30 empfängt, wird das Nachrichtenpaket an den Sicherheits-Paketprozessor 26 geliefert. Der Sicherheitspaketprozessor 26 entschlüsselt daraufhin den verschlüsselten Abschnitt des Nachrichtenpakets, um die der Klartext-Internetadresse zugeordnete Zahlen-Internetadresse zu erhalten und lädt diese Information in den IP-Parameterspeicher 25. Danach kann die Vorrichtung diese Zahlen-Internetadresse beim Erzeugen von Nachrichtenpaketen zur Übertragung an den Server 31(s) verwenden, welcher zu der Klartext-Internetadresse gehört.

Es versteht sich, daß, falls der Namen-Server 32 keine Zahlen-Internetadresse besitzt, welche der durch die Vorrichtung 12(m) in dem Anfragennachrichtenpaket gelieferte Klartext-Internetadresse zugeordnet ist, dies der Namen-Server 32 in dem durch ihn erzeugten Antwortnachrichtenpaket entsprechend anzeigen. Die Firewall 30 erzeugt dann in Reaktion auf das durch den Namen-Server 32 gelieferte Antwortnachrichtenpaket auch ein Nachrichtenpaket zur Übertragung an die Vorrichtung 12(m), welches einen verschlüsselten Abschnitt enthält, der das Antwortnachrichtenpaket umfaßt, das durch den Namen-Server 32 erzeugt wurde. Nachdem die Vorrichtung 12(m) das Nachrichtenpaket empfangen hat, wird der verschlüsselte Abschnitt durch den Sicherheitspaketprozessor 26 entschlüsselt, welcher daraufhin den Paketgenerator 22 darüber informiert, daß der Namen-Server 32 keine der Klartext-Internetadresse zugeordnete Zahlen-Internetadresse besitzt. Falls der IP-Parameterspeicher 25 die Identifizierung eines anderen Namen-Servers enthält, erzeugt sodann der Paketgenerator 22 der Vorrichtung 12(m) ein Anfragennachrichtenpaket zur Übertragung an den nächsten Namen-Server, der in deren IP-Parameterspeicher 25 identifiziert ist, um von diesem Namen-Server die Zahlen-Internetadresse anzufordern, welche der Klartext-Internetadresse zugeordnet ist. Falls andererseits der IP-Parameterspeicher 25 keine Identifizierung eines anderen Namen-Servers enthält, kann der Paketgenerator 22 die Bedienerschnittstelle 20 oder ein Programm darüber informieren, daß er nicht in der Lage ist, ein Nachrichtenpaket zur Übertragung an eine Vorrichtung zu erzeugen, welche der Klartext-Internetadresse zugeordnet ist, welche durch die Bedienerschnittstelle 20 oder ein Programm eingegeben bzw. bereitgestellt wurde.

Die Erfindung liefert eine Anzahl von Vorteilen. Insbesondere schafft die Erfindung ein System zum Vereinfachen der Kommunikation zwischen Vorrichtungen, welche mit einem öffentlichen Netzwerk verbunden sind, z. B. mit dem Internet 14, und Vorrichtungen, welche mit privaten Netzwerken verbunden sind, z. B. mit dem virtuellen privaten Netzwerk 15, indem die Umwandlung von Klartextadressen in Netzwerkadressen durch einen Namen-Server, der bevorzugt über einen Sicherheitstunnel mit den privaten Netzwerken verbunden ist, ermöglicht wird.

Es versteht sich, daß eine Vielzahl von Modifikationen an der im Zusammenhang mit Fig. 1 beschriebenen Anordnung durchgeführt werden können. Obwohl das Netzwerk 10 so beschrieben wurde, daß die Identifizierung der Verschlüsselungs- und Entschlüsselungsalgorithmen und -schlüssel durch die Vorrichtung 12(m) und die Firewall 30 während des Dialogs, währenddessen der Sicherheitstunnel eingerichtet wird, ausgetauscht wird, versteht es sich, daß bei-

spielsweise Information durch die Vorrichtung **12(m)** und die Firewall **30** getrennt von dem Aufbau eines solchen Sicherheitstunnels bereitgestellt werden können.

Obwohl die Erfindung im Zusammenhang mit dem Internet beschrieben wurde, versteht es sich ferner, daß die Erfindung in Verbindung mit jedem, insbesondere globalen, Netzwerk verwendet werden kann. Obwohl die Erfindung im Zusammenhang mit einem Netzwerk beschrieben wurde, welches ein System von Klartext-Netzwerkadressen bereitstellt, versteht es sich ferner, daß die Erfindung nicht darauf beschränkt ist sondern in Verbindung mit jedem Netzwerk verwendet werden kann, welches irgendeine Form einer – den systemeigenen Netzwerkadressen übergeordnete – Sekundär-Netzwerkadresseneinrichtung oder vergleichbare nicht-formeller Netzwerkadresseneinrichtung vorsieht.

Es versteht sich ferner, daß ein erfindungsgemäßes System als ganzes oder in Teilen aus speziell hierfür geeigneter Hardware oder einem allgemein geeigneten Computersystem oder jeder Kombination davon aufgebaut werden kann, wobei jeder Abschnitt davon durch ein geeignetes Programm gesteuert werden kann. Jedes Programm kann als ganzes oder in Teilen einen Teil des Systems umfassen oder auf dem System in einer konventionellen Weise gespeichert sein, oder es kann als ganzes oder in Teilen in das System über ein Netzwerk oder andere Mechanismen zur Übertragung von Information in einer konventionellen Weise bereitgestellt werden. Zusätzlich versteht es sich, daß das System betrieben und/oder auf andere Art und Weise mittels Information gesteuert werden kann, welche durch einen Bediener mittels Bedieneingabelementen (nicht gezeigt) bereitgestellt wird, welche direkt an das System angeschlossen sein können oder welche die Information über ein Netzwerk oder andere Mechanismen zur Übertragung von Information in einer konventionellen Weise übertragen können.

Die vorstehende Beschreibung hat sich auf ein spezifisches Ausführungsbeispiel der Erfindung bezogen. Es versteht sich jedoch, daß verschiedene Variationen und Modifikationen der Erfindung gemacht werden können, bei welchen einige oder alle der Vorteile der Erfindung erreicht werden. Diese und andere Variationen und Modifikationen fallen in den Schutzbereich der vorliegenden Erfindung, der durch die nachfolgenden Ansprüche bestimmt ist.

#### Patentansprüche

1. System umfassend ein virtuelles privates Netzwerk (**15**) und eine externe Vorrichtung (**12(m)**), welche über ein digitales Netzwerk (**14**) kommunizieren, wobei:  
das virtuelle private Netzwerk (**15**) eine Firewall (**30**), wenigstens eine interne Vorrichtung (**31(s)**) und einen Namen-Server (**32**) aufweist, welche jeweils eine Netzwerkadresse besitzen, wobei die interne Vorrichtung (**31(s)**) auch eine Sekundäradresse besitzt und der Namen-Server (**32**) derart konfiguriert ist, daß er eine Zuordnung zwischen der Sekundäradresse und der Netzwerkadresse bereitstellt,  
die Firewall (**30**) derart konfiguriert ist, daß sie der externen Vorrichtung (**12(m)**) in Reaktion auf deren Anfrage zum Aufbau einer Verbindung zur Firewall (**30**) die Netzwerkadresse des Namen-Servers (**32**) liefert, und  
die externe Vorrichtung (**12(m)**) derart konfiguriert ist, daß sie in Reaktion auf eine Anfrage zum Zugriff auf die interne Vorrichtung (**31(s)**), welche die Sekundäradresse der internen Vorrichtung (**31(s)**) enthält, eine Netzwerkadressen-Anfragennachricht zur Übertragung über die Verbindung an die Firewall (**30**) erzeugt, wel-

che eine Auflösung der der Sekundäradresse zugeordneten Netzwerkadresse anfordert, wobei die Firewall (**30**) derart konfiguriert ist, daß sie die Adressenauflosungsanfrage an den Namen-Server (**32**) übermittelt, der Namen-Server (**32**) derart konfiguriert ist, daß er die der Sekundäradresse zugeordnete Netzwerkadresse bereitstellt, und die Firewall (**30**) daraufhin die Netzwerkadresse in einer Netzwerkadressen-Antwortnachricht zur Übertragung über die Verbindung an die externe Vorrichtung (**12(m)**) bereitstellt.

2. System nach Anspruch 1, bei welchem die externe Vorrichtung (**12(m)**) derart konfiguriert ist, daß sie die in der Netzwerkadressen-Antwortnachricht bereitgestellte Netzwerkadresse beim Erzeugen von wenigstens einer Nachricht zur Übertragung an die interne Vorrichtung (**31(s)**) verwendet.

3. System nach Anspruch 1 oder 2, bei welchem die externe Vorrichtung (**12(m)**) derart konfiguriert ist, daß sie mit dem Netzwerk (**14**) durch einen Netzwerk-Service-Provider (**11**) verbunden wird.

4. System nach Anspruch 3, bei welchem die externe Vorrichtung (**12(m)**) derart konfiguriert ist, daß sie eine Kommunikationssitzung mit dem Netzwerk-Service-Provider (**11**) aufbaut, wobei der Netzwerk-Service-Provider (**11**) der externen Vorrichtung (**12(m)**) die Identifizierung eines weiteren Namen-Servers übermittelt, wobei der weitere Namen-Server derart konfiguriert ist, daß er eine Zuordnung zwischen einer Sekundäradresse und einer Netzwerkadresse für wenigstens eine Vorrichtung bereitstellt.

5. System nach einem der vorstehenden Ansprüche, bei welchem die externe Vorrichtung (**12(m)**) derart konfiguriert ist, daß sie eine Liste von Namen-Servern erhält, welche der externen Vorrichtung (**12(m)**) identifiziert wurden, und die externe Vorrichtung (**12(m)**) die Namen-Server in der Liste nacheinander in Reaktion auf eine Anfrage zum Zugriff auf eine andere Vorrichtung abfragt, wobei die Anfrage eine Sekundäradresse der anderen Vorrichtung enthält, solange bis die externe Vorrichtung (**12(m)**) eine Netzwerkadresse empfängt, wobei die externe Vorrichtung (**12(m)**) in jedem Abfragevorgang eine Netzwerkadressen-Anfragennachricht zur Übertragung über das Netzwerk (**14**) erzeugt, welche durch einen der Namen-Server in der Liste zu beantworten ist, und von diesem eine Netzwerkadressen-Antwortnachricht empfängt.

6. System nach einem der vorstehenden Ansprüche, bei welchem die Verbindung zwischen der externen Vorrichtung (**12(m)**) und der Firewall (**30**) ein Sicherheitstunnel ist, in welchem wenigstens ein der zwischen der externen Vorrichtung (**12(m)**) und der Firewall (**30**) übertragenen Nachrichten verschlüsselt ist.

7. Verfahren zum Betreiben eines Systems umfassend ein virtuelles privates Netzwerk (**15**) und eine externe Vorrichtung (**12(m)**), welche durch ein digitales Netzwerk (**14**) miteinander verbunden sind, wobei das virtuelle private Netzwerk (**15**) eine Firewall (**30**), wenigstens eine interne Vorrichtung (**31(s)**) und einen Namen-Server (**32**) aufweist, welche jeweils eine Netzwerkadresse besitzen, wobei die interne Vorrichtung (**31(s)**) auch eine Sekundäradresse besitzt, und der Namen-Server (**32**) derart konfiguriert ist, daß er eine Zuordnung zwischen der Sekundäradresse und der Netzwerkadresse bereitstellt, wobei:

A. in Reaktion auf eine Anfrage der externen Vorrichtung (**12(m)**) zum Aufbau einer Verbindung zur Firewall (**30**) die Firewall (**30**) der externen Vorrichtung (**12(m)**) die Netzwerkadresse des

Namen-Server (32) übermittelt; und

B. (i) in Reaktion auf eine Anfrage zum Zugriff auf die interne Vorrichtung (31(s)), welche die Sekundäradresse der internen Vorrichtung (31(s)) enthält, die externe Vorrichtung (12(m)) eine Netzwerkadressen-Anfragennachricht zur Übertragung über die Verbindung an die Firewall (30) erzeugt, welche eine Auflösung der Netzwerkadresse, welche der Sekundäradresse zugeordnet ist, anfordert,

(ii) die Firewall (30) die Adressenauflösungsanfrage an den Namen-Server (32) übermittelt, (iii) der Namen-Server (32) die der Sekundäradresse zugeordnete Netzwerkadresse bereitstellt, und

(iv) die Firewall (30) die Netzwerkadresse in einer Netzwerkadressen-Antwortnachricht zur Übertragung über die Verbindung an die externe Vorrichtung (12(m)) bereitstellt.

8. Verfahren nach Anspruch 7, bei welchem die externe Vorrichtung (12(m)) ferner die in der Netzwerkadressen-Antwortnachricht bereitgestellte Netzwerkadresse beim Erzeugen von wenigstens einer Nachricht zur Übertragung an die interne Vorrichtung (31(s)) verwendet.

9. Verfahren nach Anspruch 7 oder 8, bei welchem die externe Vorrichtung (12(m)) mit dem Netzwerk (14) durch einen Netzwerk-Service-Provider (11) verbunden werden kann.

10. Verfahren nach Anspruch 9, bei welchem die externe Vorrichtung (12(m)) eine Kommunikationssitzung mit dem Netzwerk-Service-Provider (11) aufbaut, wobei der Netzwerk-Service-Provider (11) der externen Vorrichtung (12(m)) die Identifizierung eines weiteren Namen-Servers übermittelt, wobei der weitere Namen-Server eine Zuordnung zwischen einer Sekundäradresse und einer Netzwerkadresse für wenigstens eine Vorrichtung bereitstellt.

11. Verfahren nach einem der Ansprüche 7 bis 10, bei welchem die externe Vorrichtung (12(m)) eine Liste von Namen-Servern erhält, welche der externen Vorrichtung (12(m)) identifiziert wurden, und die externe Vorrichtung (12(m)) die Namen-Server in der Liste nacheinander in Reaktion auf eine Anfrage zum Zugriff auf eine andere Vorrichtung abfragt, wobei die Anfrage eine Sekundäradresse der anderen Vorrichtung enthält, solange bis die externe Vorrichtung (12(m)) eine Netzwerkadresse empfängt, wobei die externe Vorrichtung (12(m)) in jedem Abfragevorgang eine Netzwerkadressen-Anfragennachricht zur Übertragung über das Netzwerk (14) erzeugt, welche durch einen der Namen-Server in der Liste zu beantworten ist, und von diesem eine Netzwerkadressen-Antwortnachricht empfängt.

12. Verfahren nach einem der Ansprüche 7 bis 11, bei welchem die Verbindung zwischen der externen Vorrichtung (12(m)) und der Firewall (30) ein Sicherheitstunnel ist, in welchem wenigstens ein Abschnitt der zwischen der externen Vorrichtung (12(m)) und der Firewall (30) übertragenen Nachrichten verschlüsselt ist.

13. Computerprogramm-Produkt zur gemeinsamen Verwendung mit einem virtuellen privaten Netzwerk (15) und einer externen Vorrichtung (12(m)), welche durch ein digitales Netzwerk (14) miteinander verbunden sind, wobei das virtuelle private Netzwerk eine Firewall (30), wenigstens eine interne Vorrichtung (31(s)) und einen Namen-Server (32) aufweist, welche jeweils eine Netzwerkadresse besitzen, wobei die interne Vorrichtung (31(s)) auch eine Sekundäradresse

besitzt, und der Namen-Server (32) derart konfiguriert ist, daß er eine Zuordnung zwischen der Sekundäradresse und der Netzwerkadresse bereitstellt, wobei das Computerprogrammprodukt ein maschinenlesbares Medium mit folgenden Codes aufweist:

A. ein Namen-Server-Identifizierungscodemodul, welches veranlaßt, daß die Firewall (30) der externen Vorrichtung (12(m)) in Reaktion auf deren Anfrage zum Aufbau einer Verbindung zur Firewall (30) die Netzwerkadresse des Namen-Servers (32) übermittelt,

B. ein Codemodul zur Erzeugung einer Netzwerkadressen-Anfragennachricht, welches veranlaßt, daß die externe Vorrichtung (12(m)) in Reaktion auf eine Anfrage zum Zugriff auf die interne Vorrichtung (31(s)), welche die Sekundäradresse der internen Vorrichtung (31(s)) enthält, eine Netzwerkadressen-Anfragennachricht zur Übertragung über die Verbindung an die Firewall (30) erzeugt, welche die Auflösung der der Sekundäradresse zugeordneten Netzwerkadresse anfordert,

C. ein Modul zur Übermittlung einer Adressenauflösungsanfrage, welches veranlaßt, daß die Firewall (30) die Adressenauflösungsanfrage an den Namen-Server (32) übermittelt,

D. ein Namen-Server-Steuerungsmodul, welches veranlaßt, daß der Namen-Server (32) die der Sekundäradresse zugeordnete Netzwerkadresse bereitstellt, und

E. ein Modul zur Übermittlung einer Netzwerkadressen-Antwortnachricht, welches veranlaßt, daß die Firewall (30) die Netzwerkadresse in einer Netzwerkadressen-Antwortnachricht zur Übertragung über die Verbindung an die externe Vorrichtung (12(m)) bereitstellt.

14. Computerprogramm-Produkt nach Anspruch 13, welches ferner ein Netzwerkadressenverwendungsmodul aufweist, welches veranlaßt, daß die externe Vorrichtung (12(m)) die in der Netzwerkadressen-Antwortnachricht übermittelte Netzwerkadresse beim Erzeugen von wenigstens einer Nachricht zur Übertragung an die interne Vorrichtung (31(s)) verwendet.

15. Computerprogramm-Produkt nach Anspruch 13 oder 14, welches ferner ein Netzwerk-Service-Provider-Steuerungsmodul aufweist, welches veranlaßt, daß die externe Vorrichtung (12(m)) mit dem Netzwerk (14) durch einen Netzwerk-Service-Provider (11) verbunden wird.

16. Computerprogramm-Produkt nach Anspruch 15, bei welchem das Netzwerk-Service-Provider-Steuerungsmodul ein Kommunikationssitzungsaufbaumodul umfaßt, welches veranlaßt, daß die externe Vorrichtung (12(m)) mit dem Netzwerk-Service-Provider (11) eine Kommunikationssitzung aufbaut und von diesem eine Identifizierung von einem weiteren Namen-Server empfängt.

17. Computerprogramm-Produkt nach einem der Ansprüche 13 bis 16, welches ferner ein Namen-Server-Abfragesteuerungsmodul aufweist, welches veranlaßt, daß die externe Vorrichtung (12(m)) eine Liste von Namen-Servern erhält, welche der externen Vorrichtung (12(m)) identifiziert wurden, und die Namen-Server in der Liste nacheinander in Reaktion auf eine Anfrage zum Zugriff auf eine andere Vorrichtung abfragt, wobei die Anfrage eine Sekundäradresse der anderen Vorrichtung enthält, solange bis die externe Vorrichtung (12(m)) eine Netzwerkadresse empfängt, und wobei die externe Vorrichtung (12(m)) in jedem Abfragevor-

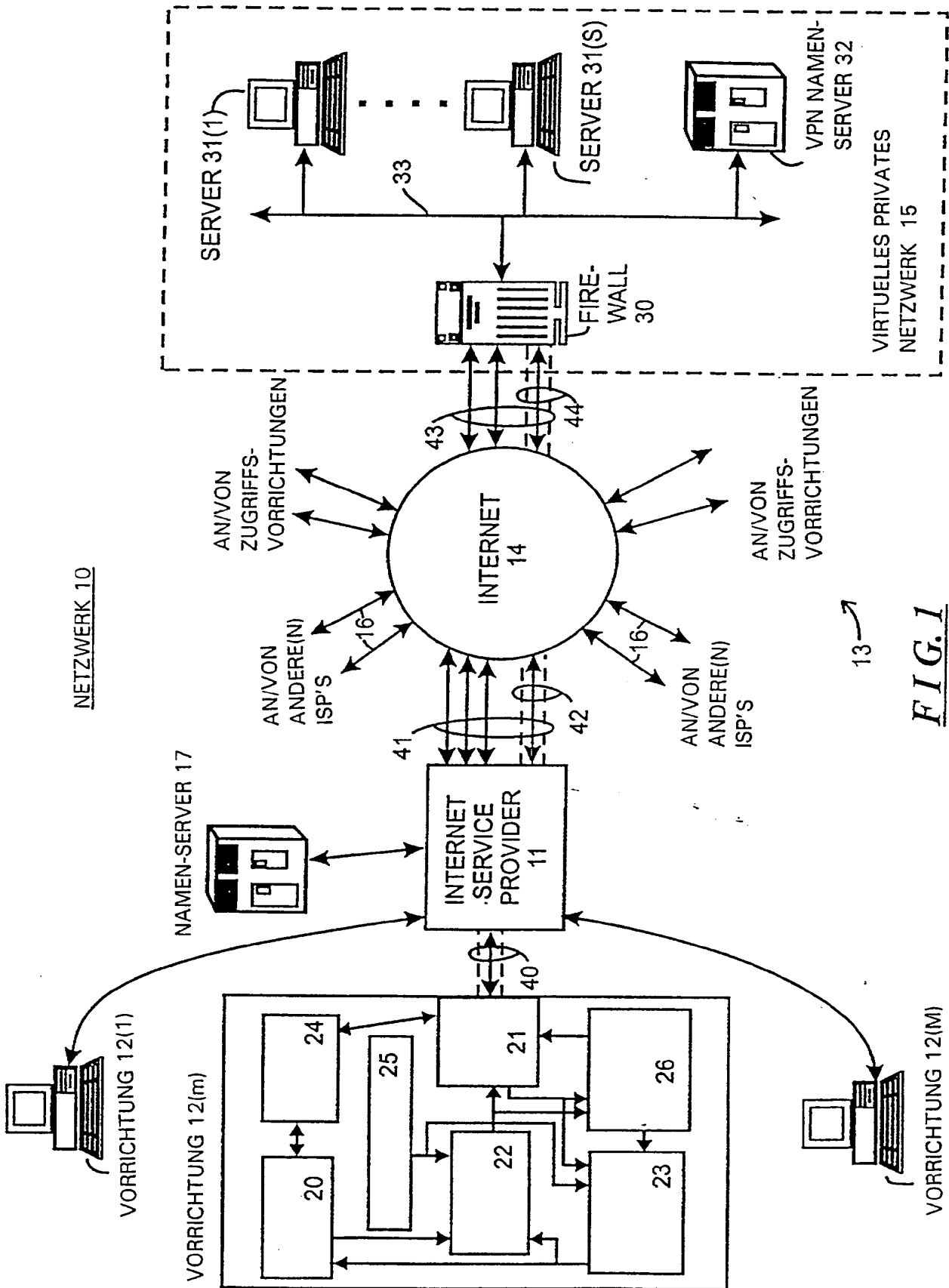


FIG. 1

gang eine Netzwerkadressen-Anfragesnachricht zur Übertragung über das Netzwerk (14) erzeugt, welche durch einen der Namen-Server in der Liste zu beantworten ist, und von diesem eine Netzwerkadressen-Antwortnachricht empfängt.

5

18. Computerprogramm-Produkt nach einem der Ansprüche 13 bis 17, bei welchem die Verbindung zwischen der externen Vorrichtung (12(m)) und der Firewall (30) ein Sicherheitstunnel ist, in welchem wenigstens ein Abschnitt der zwischen der externen Vorrichtung (12(m)) und der Firewall (30) übertragenen Nachrichten verschlüsselt ist.

10

Hierzu 1 Seite(n) Zeichnungen

15

20

25

30

35

40

45

50

55

60

65

**PART B - FEE(S) TRANSMITTAL**

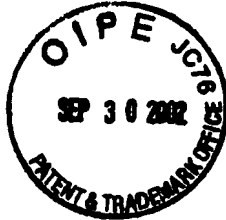
Complete and send this form, together with applicable fee(s), to: **Mail Box ISSUE FEE**  
**Commissioner for Patents**  
**Washington, D.C. 20231**  
**Fax (703)746-4000**

**INSTRUCTIONS:** This form should be used for transmitting the **ISSUE FEE** and **PUBLICATION FEE** (if required). Blocks 1 through 4 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as 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" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Legibly mark-up with any correction of this Block 1)

7590 07/03/2002

Banner & Witcoff, Ltd  
 1001 G Street, NW  
 Washington, DC 20001-4597



Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must 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 United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Box Issue Fee address above, or being facsimile transmitted to the USPTO, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

09/30/02

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/504,783	02/15/2000	Edmund Colby Munger	00479.85672	8308

TITLE OF INVENTION: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

APPLN. TYPE	SMALL ENTITY	ISSUE FEE	PUBLICATION FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	\$1280	\$0	\$1280	10/03/2002

EXAMINER	ART UNIT	CLASS-SUBCLASS
LIM, KRISNA	2153	709-225000

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 BANNER & WITCOFF, LTD.
- 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. Inclusion of assignee data is only appropriate when an assignment has been previously submitted to the USPTO or is being submitted under separate cover. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE: Science Applications International Corporation  
 (B) RESIDENCE: (CITY and STATE OR COUNTRY): San Diego, CA

Please check the appropriate assignee category or categories (will not be printed on the patent)  Individual  corporation or other private group entity  government

4a. The following fee(s) are enclosed:

4b. Payment of Fee(s):

- Issue Fee
- Publication Fee
- Advance Order - # of Copies 10

- A check in the amount of the fee(s) is enclosed.
- Payment by credit card. Form PTO-2038 is attached.
- The Commissioner is hereby authorized by charge the required fee(s), or credit any overpayment, to Deposit Account Number 19-0733 (enclose an extra copy of this form).

Commissioner for Patents 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.

(Authorized Signature) Ross A. Dannenberg (Date) 9/30/02  
 Ross A. Dannenberg, Reg. No. 69,024

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 party in interest as shown by the records of the United States Patent and Trademark Office.

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, Washington, D.C. 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, Washington, DC 20231.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

10/01/2002 BFELEKE2 00000265 190733 09504783  
 01 FC:142 1280.00 CH  
 02 FC:561 30.00 CH

TRANSMIT THIS FORM WITH FEE(S)



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER OF PATENTS AND TRADEMARKS  
Washington, D.C. 20231  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/504,783	02/15/2000	Edmund Colby Munger	00479.85672	8308

7590 10/08/2002

Banner & Witcoff, Ltd  
1001 G Street, NW  
Washington, DC 20001-4597

EXAMINER

LIM, KRISNA

ART UNIT	PAPER NUMBER
2153	#14

DATE MAILED: 10/08/2002

Please find below and/or attached an Office communication concerning this application or proceeding.





UNITED STATES DEPARTMENT OF COMMERCE  
Patent and Trademark Office  
ASSISTANT SECRETARY AND COMMISSIONER OF  
PATENTS AND TRADEMARKS  
Washington, D.C. 20231

In re Application Serial No: 09/504,783 : DECISION ON PETITION UNDER  
Inventor: Edmond Colby Munger et al. : 37 CFR § 1.97 FOR  
Filed: February 15, 2000 : CONSIDERATION  
For: Improvements To An Agile Network : OF INFORMATION DISCLOSURE  
Protocol For Secure Communications : STATEMENT AFTER FINAL  
With Assured System Availability : REJECTION OR ALLOWANCE

The petition under 37 CFR § 1.97(d) for consideration of an information disclosure statement filed September 13, 2002 after allowance as been:

GRANTED.

DENIED.

The petition lacks:

The required fee under 37 CFR §§ 1.97(d)(2)(iii) and 1.17(l)(1).

A proper certification as specified in 37 CFR §§ 1.97(d)(2)(l) and 1.97(e).

The information disclosure statement has been considered and placed in the file as a record.

  
Glenton Burgess  
Supervisory Patent Examiner TC 2153

ADDRESS:

Bradley C. Wright  
Banner & Witcoff, LTD  
1001 G Street, N.W.  
Washington, D.C. 20001-4597  
(202) 508-9100

  
KRISNA LIM  
PRIMARY EXAMINER



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER OF PATENTS AND TRADEMARKS  
Washington, D.C. 20231  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/504,783	02/15/2000	Edmund Colby Munger	00479.85672	8308

7590 11/08/2002

Banner & Witcoff, Ltd  
1001 G Street, NW  
Washington, DC 20001-4597

EXAMINER

LIM, KRISNA

ART UNIT PAPER NUMBER

2153

DATE MAILED: 11/08/2002

# 15 ✓

Please find below and/or attached an Office communication concerning this application or proceeding.



UNITED STATES DEPARTMENT OF COMMERCE  
Patent and Trademark Office  
ASSISTANT SECRETARY AND COMMISSIONER OF  
PATENTS AND TRADEMARKS  
Washington, D.C. 20231

In re Application Serial No: 09/504,783 : DECISION ON PETITION UNDER  
Inventor: Edmond Colby Munger et al. : 37 CFR § 1.97 FOR  
Filed: February 15, 2000 : CONSIDERATION  
For: Improvements To An Agile Network : OF INFORMATION DISCLOSURE  
Protocol For Secure Communications : STATEMENT AFTER FINAL  
With Assured System Availability : REJECTION OR ALLOWANCE

The petition under 37 CFR § 1.97(d) for consideration of an information disclosure statement filed August 23, 2002 and September 12, 2002 after allowance as been:

GRANTED.

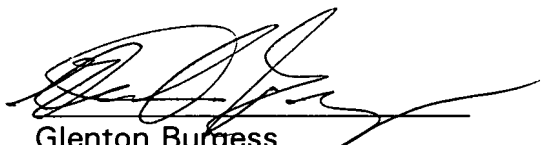
DENIED.

The petition lacks:

The required fee under 37 CFR §§ 1.97(d)(2)(iii) and 1.17(l)(1).

A proper certification as specified in 37 CFR §§ 1.97(d)(2)(l) and 1.97(e).

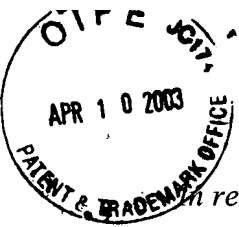
The information disclosure statement has been considered and placed in the file as a record.

  
Glenton Burgess  
Supervisory Patent Examiner TC 2153

ADDRESS:

Bradley C. Wright  
Banner & Witcoff, LTD  
1001 G Street, N.W.  
Washington, D.C. 20001-4597  
(202) 508-9100

  
KRISNA LIM  
PRIMARY EXAMINER



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

copy C  
PATENT #20  
ECW  
1 of 2

in re U.S. Patent No. 6,502,135 ) Serial No. 09/504,783  
)  
) Inventors: Edmund Colby MUNGER *et al.* ) Filed: February 15, 2000  
)  
) Issue Date: December 31, 2002 ) Attorney Docket No. 000479.85672

For: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

Certificate  
APR 16 2003  
of Correction

REQUEST FOR CERTIFICATE OF CORRECTION

Honorable Commissioner of Patents and Trademarks  
Washington, D.C. 20231

Sir:

Pursuant to 35 U.S.C. § 254 and 37 C.F.R. § 1.322, this is a request for the issuance of a Certificate of Correction in the above-identified patent. Two (2) copies of PTO Form 1050 are appended. The complete Certificate of Correction involves one (1) page.

The mistakes identified in the appended Form occurred through no fault of the Applicants, as clearly disclosed by the records of the application, which matured into this patent. Enclosed for your convenience are the relevant portions of the Information Disclosure Statements dated considered October 7, 2002, and November 7, 2002.

Issuance of the Certificate of Correction containing the corrections is respectfully requested. Since these changes are necessitated through no fault of the Applicants, no fee is believed to be associated with this request. Nonetheless, should the Patent and Trademark Office determine that a fee is required, please charge our Deposit Account No. 19-0733.

Respectfully submitted,

Date: 4/10/03

By: Ross A. Dannenberg  
Ross A. Dannenberg  
Reg. No. 49,024

Banner & Witcoff  
1001 G Street, N.W., 11<sup>th</sup> Floor  
Washington, D.C. 20001  
(202) 508-9100

APR 16 2003

UNITED STATES PATENT AND TRADEMARK OFFICE  
**CERTIFICATE OF CORRECTION**

PATENT NO. : 6,502,135 *B1*  
DATED : December 31, 2002  
INVENTOR(S) : Edmund Colby Munger *et al.*

Page 1 of 1

It is certified that errors appear in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Title page,

Item [56] References Cited, Other Publications, *insert the following:*

--Search Report (dated 8/20/02), International Application No. PCT/US01/04340

Search Report (dated 8/23/02), International Application No. PCT/US01/13260

James E. Bellaire, "New Statement of Rules - Naming Internet Domains",  
Internet Newsgroup, July 30, 1995, 1 page.

D. Clark, "US Calls for Private Domain-Name System", Computer, IEEE Computer Society, August 1, 1998, pages 22-25.

August Bequai, "Balancing Legal Concerns Over Crime and Security in Cyberspace",  
Computer & Security, Vol. 17, No. 4, 1998, pages 293-298.

Rich Winkel, "CAQ: Networking With Spooks: The NET & The Control Of  
Information", Internet Newsgroup, June 21, 1997, 4 pages.-- ~~has been inserted~~

Column 48,

Line 2, "VPN target computer" has been replaced with --VPN with the target computer--.

Mailing Address of Sender:

Banner & Witcoff, Ltd.  
1001 G Street, N.W., 11<sup>th</sup> Floor  
Washington, DC 20001-4597

FORM PTO 1050 (Rev.2-93)

U.S. PAT. NO 6,502,135

No. of add'l copies

@ 50¢ per page

ψ

Petitioner RPX Corporation - Ex. 1002, p.705

UNITED STATES PATENT AND TRADEMARK OFFICE  
**CERTIFICATE OF CORRECTION**

PATENT NO. : 6,502,135  
DATED : December 31, 2002  
INVENTOR(S) : Edmund Colby Munger *et al.*

Page 1 of 1

It is certified that errors appear in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Title page,  
Item [56] References Cited, Other Publications,

--Search Report (dated 8/20/02), International Application No. PCT/US01/04340

Search Report (dated 8/23/02), International Application No. PCT/US01/13260

James E. Bellaire, "New Statement of Rules – Naming Internet Domains",  
Internet Newsgroup, July 30, 1995, 1 page.

D. Clark, "US Calls for Private Domain-Name System", Computer, IEEE Computer Society, August 1, 1998, pages 22-25.

August Bequai, "Balancing Legal Concerns Over Crime and Security in Cyberspace",  
Computer & Security, Vol. 17, No. 4, 1998, pages 293-298.

Rich Winkel, "CAQ: Networking With Spooks: The NET & The Control Of  
Information", Internet Newsgroup, June 21, 1997, 4 pages.-- has been inserted.

Mailing Address of Sender:

Banner & Witcoff, Ltd.  
1001 G Street, N.W., 11<sup>th</sup> Floor  
Washington, DC 20001-4597

FORM PTO 1050 (Rev.2-93)

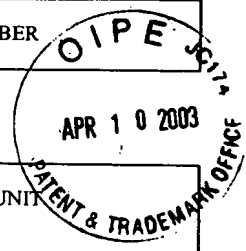
U.S. PAT. NO 6,502,135

No. of add'l copies

@ 50¢ per page

Ψ

PTO-1449 (Modified)  U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE  INFORMATION DISCLOSURE STATEMENT BY APPLICANT	ATTY. DOCKET NO. 000479.85672	SERIAL NUMBER 09/504,783
	APPLICANT Edmond Colby Munger et al.	
	FILING DATE February 15, 2000	GROUP ART UNIT 2153



**U.S. PATENT DOCUMENTS**

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE

**FOREIGN PATENT DOCUMENTS**

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO
<i>K</i>	0 858 189	8/12/98	EPO	<del>RECEIVED SEP 19 2002 Technology Center 2100</del>		
<i>K</i>	WO 01 50688	7/12/01	PCT			
<i>K</i>	WO 98 59470	12/30/98	PCT			
<i>K</i>	WO 99 48303	9/23/99	PCT			
<i>K</i>	WO 99 38081	7/29/99	PCT			

**OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)**

<i>K</i>	Search Report (dated 8/20/02), International Application No. PCT/US01/04340
<i>K</i>	Shree Murthy et al., "Congestion-Oriented Shortest Multipath Routing", Proceedings of IEEE INFOCOM, 1996, pages 1028-1036
<i>K</i>	Jim Jones et al., "Distributed Denial of Service Attacks: Defenses", Global Integrity Corporation, 2000, pages 1-14

EXAMINER <i>KRISNA Lim</i>	DATE CONSIDERED <i>11/7/02</i>
EXAMINER: Initial citation if reference was considered. Draw line through citation if not in conformance to MPEP 609 and not considered. Include copy of this form with next communication to applicant.	

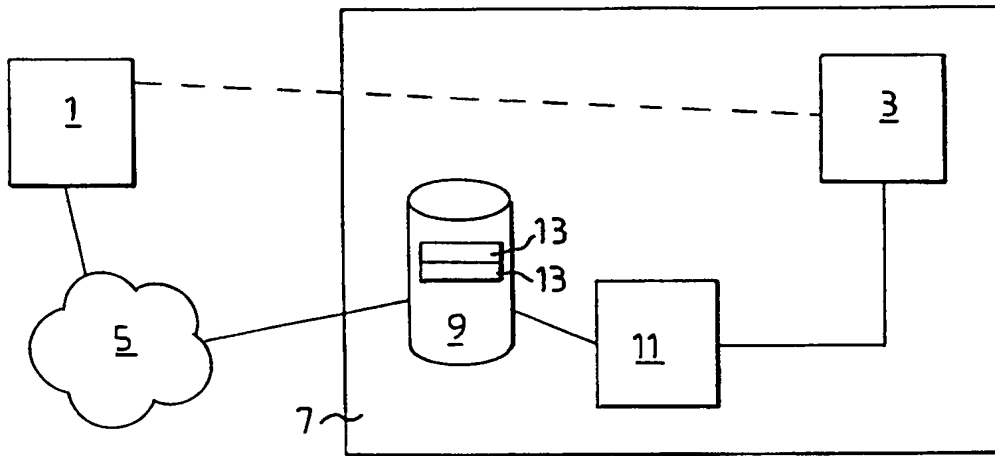
IDS w/1449 form filed: September 12, 2002



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(51) International Patent Classification <sup>6</sup> : <b>H04L 12/56, 29/02</b></p>	<p><b>A3</b></p>	<p>(11) International Publication Number: <b>WO 98/59470</b> (43) International Publication Date: 30 December 1998 (30.12.98)</p>
<p>(21) International Application Number: PCT/SE98/01217 (22) International Filing Date: 23 June 1998 (23.06.98) (30) Priority Data: 9702385-7 23 June 1997 (23.06.97) SE (71) Applicants (for all designated States except US): TELEFON-AKTIEBOLAGET LM ERICSSON (publ) [SE/SE]; S-126 25 Stockholm (SE). TELIA AB [SE/SE]; S-123 86 Farsta (SE). (72) Inventors; and (75) Inventors/Applicants (for US only): KANTER, Theo [NL/SE]; Rönninge skolväg 35E, S-144 62 Rönninge (SE). FOGELHOLM, Rabbe [SE/SE]; Turevågen 54 B, S-191 47 Sollentuna (SE). (74) Agents: HERBJØRNSSEN, Ru: et al.; Albinus Patentbyrå Stockholm AB, P.O. Box 3137, S-103 62 Stockholm (SE).</p>		<p>(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, GM, GW, HU, ID, 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, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, 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, CY, 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).  <b>Published</b> <i>With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>  (88) Date of publication of the international search report: 18 March 1999 (18.03.99)</p>

(54) Title: METHOD AND APPARATUS TO ENABLE A FIRST SUBSCRIBER IN A LARGER NETWORK TO RETRIEVE THE ADDRESS OF A SECOND SUBSCRIBER IN A VIRTUAL PRIVATE NETWORK



(57) Abstract

The present invention relates to an apparatus and a method for use in a virtual private network, VPN, (7, 7'), or a network domain forming part of a larger network, such as the Internet, to enable a first subscriber (1; 1') in the larger network to retrieve the address of a second subscriber (3; 3') in the VPN. The address may be returned to the first subscriber (1; 1') or a connection means (11) may set up the connection between the subscribers (1, 3; 1', 3') automatically.



**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		
CN	Cameroon			PL	Poland		
CM	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		

## INTERNATIONAL SEARCH REPORT

International application No.  
PCT/SE 98/01217

A. CLASSIFICATION OF SUBJECT MATTER		
IPC6: H04L 12/56, H04L 29/02 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)		
IPC6: H04L		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
SE,DK,FI,NO classes as above		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)		
WPIL, EDOC, JAPIO		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	ITU-T Recommendation H. 323, 1996, "Visual telephone systems and equipment for local area networks which provide a non- guaranteed quality of service" Paragraph 6.4, 3.41, 3.43	4-6
Y	--	1-3,7-12
Y	IETF RFC 883, Volume, November 1983, P. Mockapetris, "DOMAIN NAMES - IMPLEMENTATION and SPECIFICATION" page 23	1-3,7-12
A	IETF RFC 1383, Volume, December 1992, C. Huitema, "An Experiment in DNS Based IP Routing", paragraph 2	1-12
	--	
<input checked="" type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> 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 but published on or after the international filing date "I" 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		Date of mailing of the international search report
12 January 1999		22 -01- 1999
Name and mailing address of the ISA/ Swedish Patent Office Box 5055, S-102 42 STOCKHOLM Facsimile No. +46 8 666 02 86		Authorized officer  Christina Halldin Telephone No. +46 8 782 25 00

Form PCT/ISA/210 (second sheet) (July 1992)

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/SE 98/01217

## C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	IETF RFC 2052, Volume, October 1996, A. Gulbrandsen et al, "A DNS RR for specifying the location of services (DNS SRV)", see the whole document  --	1-12
A	EP 0752674 A1 (SUN MICROSYSEMS, INC.), 8 January 1997 (08.01.97), abstract  -- -----	1-12

Form PCT/ISA:210 (continuation of second sheet) (July 1992)

# INTERNATIONAL SEARCH REPORT

Information on patent family members

01/12/98

International application No.

PCT/SE 98/01217

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0752674 A1	08/01/97	JP 9171465 A US 5745683 A	30/06/97 28/04/98
-----			

Form PCT/ISA/210 (patent family annex) (July 1992)

INTL

From the INTERNATIONAL SEARCHING AUTHORITY

**PCT**

To:  
**BANNER & WITCOFF, LTD.**  
 Attn. Wright, Bradley C.  
 1001 G Street, N.W.  
 Eleventh Floor  
 Washington, DC 20001-4597  
 UNITED STATES OF AMERICA

NOTIFICATION OF TRANSMITTAL OF  
 THE INTERNATIONAL SEARCH REPORT  
 OR THE DECLARATION

(PCT Rule 44.1)

00479.0029  
**RECEIVED**

AUG 27 2002

BANNER WITCOFF

Date of mailing  
 (day/month/year)

20/08/2002

Applicant's or agent's file reference

00479.00029

**FOR FURTHER ACTION**

See paragraphs 1 and 4 below

International application No.

PCT/US 01/ 04340

International filing date

(day/month/year)

12/02/2001

Applicant

**SCIENCE APPLICATIONS INTERNATIONAL CORPORATION**

1.  The applicant is hereby notified that the International Search Report has been established and is transmitted herewith.

**Filing of amendments and statement under Article 19:**

The applicant is entitled, if he so wishes, to amend the claims of the International Application (see Rule 46):

**When?** The time limit for filing such amendments is normally 2 months from the date of transmittal of the International Search Report; however, for more details, see the notes on the accompanying sheet.

**Where?** Directly to the International Bureau of WIPO  
 34, chemin des Colombettes  
 1211 Geneva 20, Switzerland  
 Facsimile No.: (41-22) 740.14.35

**For more detailed instructions,** see the notes on the accompanying sheet.

2.  The applicant is hereby notified that no International Search Report will be established and that the declaration under Article 17(2)(a) to that effect is transmitted herewith.

3.  **With regard to the protest** against payment of (an) additional fee(s) under Rule 40.2, the applicant is notified that:

the protest together with the decision thereon has been transmitted to the International Bureau together with the applicant's request to forward the texts of both the protest and the decision thereon to the designated Offices.

no decision has been made yet on the protest; the applicant will be notified as soon as a decision is made.

4. **Further action(s):** The applicant is reminded of the following:

Shortly after **18 months** from the priority date, the international application will be published by the International Bureau. If the applicant wishes to avoid or postpone publication, a notice of withdrawal of the international application, or of the priority claim, must reach the International Bureau as provided in Rules 90bis.1 and 90bis.3, respectively, before the completion of the technical preparations for international publication.

Within **19 months** from the priority date, a demand for international preliminary examination must be filed if the applicant wishes to postpone the entry into the national phase until 30 months from the priority date (in some Offices even later).

Within **20 months** from the priority date, the applicant must perform the prescribed acts for entry into the national phase before all designated Offices which have not been elected in the demand or in a later election within 19 months from the priority date or could not be elected because they are not bound by Chapter II.

Name and mailing address of the International Searching Authority



European Patent Office, P.B. 5818 Patentlaan 2  
 NL-2280 HV Rijswijk  
 Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
 Fax: (+31-70) 340-3016

Authorized officer

Claude Berthon

These Notes are intended to give the basic instructions concerning the filing of amendments under article 19. The Notes are based on the requirements of the Patent Cooperation Treaty, the Regulations and the Administrative Instructions under that Treaty. In case of discrepancy between these Notes and those requirements, the latter are applicable. For more detailed information, see also the PCT Applicant's Guide, a publication of WIPO.

In these Notes, "Article", "Rule", and "Section" refer to the provisions of the PCT, the PCT Regulations and the PCT Administrative Instructions, respectively.

## INSTRUCTIONS CONCERNING AMENDMENTS UNDER ARTICLE 19

The applicant has, after having received the international search report, one opportunity to amend the claims of the international application. It should however be emphasized that, since all parts of the international application (claims, description and drawings) may be amended during the international preliminary examination procedure, there is usually no need to file amendments of the claims under Article 19 except where, e.g. the applicant wants the latter to be published for the purposes of provisional protection or has another reason for amending the claims before international publication. Furthermore, it should be emphasized that provisional protection is available in some States only.

### What parts of the international application may be amended?

Under Article 19, only the claims may be amended.

During the international phase, the claims may also be amended (or further amended) under Article 34 before the International Preliminary Examining Authority. The description and drawings may only be amended under Article 34 before the International Examining Authority.

Upon entry into the national phase, all parts of the international application may be amended under Article 28 or, where applicable, Article 41.

### When?

Within 2 months from the date of transmittal of the international search report or 16 months from the priority date, whichever time limit expires later. It should be noted, however, that the amendments will be considered as having been received on time if they are received by the International Bureau after the expiration of the applicable time limit but before the completion of the technical preparations for international publication (Rule 46.1).

### Where not to file the amendments?

The amendments may only be filed with the International Bureau and not with the receiving Office or the International Searching Authority (Rule 46.2).

Where a demand for international preliminary examination has been/is filed, see below.

### How?

Either by cancelling one or more entire claims, by adding one or more new claims or by amending the text of one or more of the claims as filed.

A replacement sheet must be submitted for each sheet of the claims which, on account of an amendment or amendments, differs from the sheet originally filed.

All the claims appearing on a replacement sheet must be numbered in Arabic numerals. Where a claim is cancelled, no renumbering of the other claims is required. In all cases where claims are renumbered, they must be renumbered consecutively (Administrative Instructions, Section 205(b)).

**The amendments must be made in the language in which the international application is to be published.**

### What documents must/may accompany the amendments?

#### Letter (Section 205(b)):

The amendments must be submitted with a letter.

The letter will not be published with the international application and the amended claims. It should not be confused with the "Statement under Article 19(1)" (see below, under "Statement under Article 19(1)").

**The letter must be in English or French, at the choice of the applicant. However, if the language of the international application is English, the letter must be in English; if the language of the international application is French, the letter must be in French.**

The letter must indicate the differences between the claims as filed and the claims as amended. It must, in particular, indicate, in connection with each claim appearing in the international application (it being understood that identical indications concerning several claims may be grouped), whether

- (i) the claim is unchanged;
- (ii) the claim is cancelled;
- (iii) the claim is new;
- (iv) the claim replaces one or more claims as filed;
- (v) the claim is the result of the division of a claim as filed.

**The following examples illustrate the manner in which amendments must be explained in the accompanying letter:**

1. [Where originally there were 48 claims and after amendment of some claims there are 51]:  
"Claims 1 to 29, 31, 32, 34, 35, 37 to 48 replaced by amended claims bearing the same numbers; claims 30, 33 and 36 unchanged; new claims 49 to 51 added."
2. [Where originally there were 15 claims and after amendment of all claims there are 11]:  
"Claims 1 to 15 replaced by amended claims 1 to 11."
3. [Where originally there were 14 claims and the amendments consist in cancelling some claims and in adding new claims]:  
"Claims 1 to 6 and 14 unchanged; claims 7 to 13 cancelled; new claims 15, 16 and 17 added." or  
"Claims 7 to 13 cancelled; new claims 15, 16 and 17 added; all other claims unchanged."
4. [Where various kinds of amendments are made]:  
"Claims 1-10 unchanged; claims 11 to 13, 18 and 19 cancelled; claims 14, 15 and 16 replaced by amended claim 14; claim 17 subdivided into amended claims 15, 16 and 17; new claims 20 and 21 added."

**"Statement under article 19(1)" (Rule 46.4)**

The amendments may be accompanied by a statement explaining the amendments and indicating any impact that such amendments might have on the description and the drawings (which cannot be amended under Article 19(1)).

The statement will be published with the international application and the amended claims.

**It must be in the language in which the international application is to be published.**

It must be brief, not exceeding 500 words if in English or if translated into English.

It should not be confused with and does not replace the letter indicating the differences between the claims as filed and as amended. It must be filed on a separate sheet and must be identified as such by a heading, preferably by using the words "Statement under Article 19(1)."

It may not contain any disparaging comments on the international search report or the relevance of citations contained in that report. Reference to citations, relevant to a given claim, contained in the international search report may be made only in connection with an amendment of that claim.

**Consequence if a demand for international preliminary examination has already been filed**

If, at the time of filing any amendments and any accompanying statement, under Article 19, a demand for international preliminary examination has already been submitted, the applicant must preferably, at the time of filing the amendments (and any statement) with the International Bureau, also file with the International Preliminary Examining Authority a copy of such amendments (and of any statement) and, where required, a translation of such amendments for the procedure before that Authority (see Rules 55.3(a) and 62.2, first sentence). For further information, see the Notes to the demand form (PCT/IPEA/401).

**Consequence with regard to translation of the international application for entry into the national phase**

The applicant's attention is drawn to the fact that, upon entry into the national phase, a translation of the claims as amended under Article 19 may have to be furnished to the designated/elected Offices, instead of, or in addition to, the translation of the claims as filed.

For further details on the requirements of each designated/elected Office, see Volume II of the PCT Applicant's Guide.



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

R COFC PATENT #2002012002 Certificate of Correction MAY 07 2003

U.S. Patent No. 6,502,135 Serial No. 09/504,783
Inventors: Edmund Colby MUNGER et al. Filed: February 15, 2000
Issue Date: December 31, 2002 Attorney Docket No. 000479.85672

For: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

REQUEST FOR CERTIFICATE OF CORRECTION

Honorable Commissioner of Patents and Trademarks Washington, D.C. 20231

Sir:

Pursuant to 35 U.S.C. § 254 and 37 C.F.R. § 1.322, this is a request for the issuance of a Certificate of Correction in the above-identified patent. Two (2) copies of PTO Form 1050 are appended. The complete Certificate of Correction involves one (1) page.

The mistake identified in claim 9, column 48, line 2, occurred as a result of an error by the Applicants, as disclosed by the records of the application which matured into this patent. The requested correction simply changes "VPN target computer" to "VPN with the target computer" for clarification and does not introduce new matter.

Issuance of the Certificate of Correction containing the correction is respectfully requested. Since this change is necessitated by an error on the part of Applicants, the Patent and Trademark Office is authorized to charge the requisite fee of \$100.00 pursuant to 37 C.F.R. § 1.20(a) to our Deposit Account No. 19-0733.

05/01/2003 BNGUYEN2 00000026 190733 6502135
01 FC:1811 100.00 CH

Respectfully submitted,

Date: April 28, 2003

By: Ross A. Dannenberg Reg. No. 49,024

Banner & Witcoff 1001 G Street, N.W., 11th Floor Washington, D.C. 20001 (202) 824-3000



UNITED STATES PATENT AND TRADEMARK OFFICE  
**CERTIFICATE OF CORRECTION**

PATENT NO. : 6,502,135

Page 1 of 1

DATED : December 31, 2002

INVENTOR(S) : Edmund Colby Munger *et al.*

It is certified that errors appear in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 48,

Line 2, "VPN target computer" has been replaced with --VPN with the target computer--.

Mailing Address of Sender:

Banner & Witcoff, Ltd.  
1001 G Street, N.W., 11<sup>th</sup> Floor  
Washington, DC 20001-4597

FORM PTO 1050 (Rev.2-93)

U.S. PAT. NO 6,502,135

No. of add'l copies

@ 50¢ per page

Ψ

UNITED STATES PATENT AND TRADEMARK OFFICE  
**CERTIFICATE OF CORRECTION**

PATENT NO. : 6,502,135 B1  
DATED : December 31, 2002  
INVENTOR(S) : Edmund Colby Munger et al.

Page 1 of 1

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Title page.

Item [56], **References Cited**, OTHER PUBLICATIONS, insert the following:

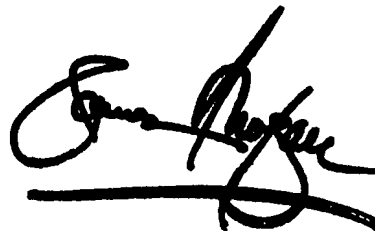
-- Search Report (dated 8/20/02), International Application No. PCT/US01/04340  
Search Report (dated 8/23/02), International Application No. PCT/US01/13260  
James E. Bellaire, "New Statement of Rules - Naming Internet Domains", Internet Newsgroup, July 30, 1995, 1 page.  
D. Clark, "US Calls for Private Domain-Name System", Computer, IEEE Computer Society, August 1, 1998, pages 22-25.  
August Bequai, "Balancing Legal Concerns Over Crime and Security in Cyberspace", Computer & Security, Vol. 17, No. 4, 1998, pages 293-298.  
Rich Winkel, "CAQ: Networking With Spooks: The NET & The Control Of Information", Internet Newsgroup, June 21, 1997, 4 pages. --

Column 48.

Line 2, "VPN target computer" has been replaced with -- VPN with the target computer --.

Signed and Sealed this

Ninth Day of September, 2003



JAMES E. ROGAN  
*Director of the United States Patent and Trademark Office*

NOTICE RE: CERTIFICATES OF CORRECTION

DATE : July 24, 2003

Paper No. 21

TO : Supervisor, Art Unit 2153

SUBJECT : Certificate of Correction Request in Patent No.: 6,502,135

A response to the following question is requested with respect to the accompanying request for a certificate of correction.

With respect to the change(s) requested, correcting Office and/or Applicant's errors, should the patent read as shown in the certificate of correction? No new matter should be introduced, nor should the scope or meaning of the claims be changed.

PLEASE COMPLETE THIS FORM AND RETURN WITH FILE, WITHIN 7 DAYS,

TO CERTIFICATES OF CORRECTION BRANCH - PK 3-915/922

PALM LOCATION 7580 - TEL. NO. 305-8309

Ernest C. White, LIE  
(703) 305- 8339

THANK YOU FOR YOUR ASSISTANCE!

Note your decision by placing a check mark in the appropriate box below, indicating whether all changes requested in the Request for Certificate of Correction should be applied. Please specify which changes should not be applied and indicate the reason(s) for denial, in the "Comments" section below.

YES       NO

Comments: N/A



Supervisor

2123

Art Unit

<b>POWER OF ATTORNEY OR REVOCATION OF POWER OF ATTORNEY WITH A NEW POWER OF ATTORNEY AND CHANGE OF CORRESPONDENCE ADDRESS</b>	Application Number	09/504,783
	Filing Date	02/15/2010
	First Named Inventor	Murger, Edmund
	Title	Agile Network Protocol for Secure Communica
	Art Unit	2153
	Examiner Name	Linn, Kristina
Attorney Docket Number	77560-017 (VRNK-1CP)	

I hereby revoke all previous powers of attorney given in the above-identified application.

A Power of Attorney is submitted herewith.

OR

I hereby appoint Practitioner(s) associated with the following Customer Number as my/our attorney(s) or agent(s) to prosecute the application identified above, and to transact all business in the United States Patent and Trademark Office connected therewith:

23630

OR

I hereby appoint Practitioner(s) named below as my/our attorney(s) or agent(s) to prosecute the application identified above, and to transact all business in the United States Patent and Trademark Office connected therewith.

Practitioner(s) Name	Registration Number

Please recognize or change the correspondence address for the above-identified application to:

The address associated with the above-mentioned Customer Number.

OR

The address associated with Customer Number:

OR

Firm or Individual Name:

Address:

City

State

Zip

Country

Telephone

Email

I am the:

Applicant/Inventor

OR

Assignee of record of the entire interest. See 37 CFR 3.71. Statement under 37 CFR 3.73(b) (Form PTO/SB/96) submitted herewith or filed on \_\_\_\_\_

SIGNATURE of Applicant or Assignee of Record

Signature

Date

Name

Telephone

Title and Company

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.

Total of \_\_\_\_\_ forms are submitted.

This collection of information is required by 37 CFR 1.31, 1.32 and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1480, Alexandria, VA 22313-1480. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1480, Alexandria, VA 22313-1480.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

**STATEMENT UNDER 37 CFR 3.73(b)**

Applicant/Patent Owner: VirnetX Inc.

Application No./Patent No.: 6,502,135 Filed/Issue Date: 12/31/2002

Titled: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

VirnetX Inc, a corporation  
(Name of Assignee) (Type of Assignee, e.g., corporation, partnership, university, government agency, etc.)

states that it is:

- 1.  the assignee of the entire right, title, and interest in;
- 2.  an assignee of less than the entire right, title, and interest in (The extent (by percentage) of its ownership interest is \_\_\_\_\_ %); or
- 3.  the assignee of an undivided interest in the entirety of (a complete assignment from one of the joint inventors was made)

the patent application/patent identified above, by virtue of either:

A.  An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel \_\_\_\_\_, Frame \_\_\_\_\_, or for which a copy therefore is attached.

OR

B.  A chain of title from the inventor(s), of the patent application/patent identified above, to the current assignee as follows:

1. From: Munger et al. To: Science Applications International Corp.

The document was recorded in the United States Patent and Trademark Office at Reel: 010564 Frame: 0243 or for which a copy thereof is attached.

2. From: Science Applications International Corp. To: VirnetX Inc.

The document was recorded in the United States Patent and Trademark Office at Reel: 018757 Frame: 0326 or for which a copy thereof is attached

3. From: \_\_\_\_\_ To: \_\_\_\_\_

The document was recorded in the United States Patent and Trademark Office at Reel: \_\_\_\_\_ Frame: \_\_\_\_\_ or for which a copy thereof is attached.

Additional documents in the chain of title are listed on a supplemental sheet(s).

As required by 37 CFR 3.73(b)(1)(i), the documentary evidence of the chain of title from the original owner to the assignee was, or concurrently is being, submitted for recordation pursuant to 37 CFR 3.11.

[NOTE: A separate copy (i.e., a true copy of the original assignment document(s)) must be submitted to Assignment Division in accordance with 37 CFR Part 3, to record the assignment in the records of the USPTO. See MPEP 202.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.

[Signature]  
\_\_\_\_\_  
Signature  
Randall Larsen  
\_\_\_\_\_  
Printed or Typed Name

12/15/09  
\_\_\_\_\_  
Date  
President  
\_\_\_\_\_  
Title

This collection of information is required by 37 CFR 3.73(b). The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-0198 and select option 2.

## Electronic Acknowledgement Receipt

<b>EFS ID:</b>	6639599
<b>Application Number:</b>	09504783
<b>International Application Number:</b>	
<b>Confirmation Number:</b>	8308
<b>Title of Invention:</b>	AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILTY
<b>First Named Inventor/Applicant Name:</b>	Edmund Colby Munger
<b>Customer Number:</b>	22907
<b>Filer:</b>	Toby H. Kusmer./Kelly Ciarmataro
<b>Filer Authorized By:</b>	Toby H. Kusmer.
<b>Attorney Docket Number:</b>	00479.85672
<b>Receipt Date:</b>	15-DEC-2009
<b>Filing Date:</b>	15-FEB-2000
<b>Time Stamp:</b>	15:15:02
<b>Application Type:</b>	Utility under 35 USC 111(a)

### Payment information:

Submitted with Payment	no
------------------------	----

### File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Power of Attorney	Munger_POA.pdf	766996 <small>5514324ed7e6dd934d53387e2fa26dc2bf3d0c5e</small>	no	1

### Warnings:

### Information:

2	Assignee showing of ownership per 37 CFR 3.73(b).	Munger_Statement.pdf	743279 90cd8c671ccaca8ac532e9410923f9013764f325	no	1
---	---	----------------------	--	----	---

**Warnings:**

**Information:**

<b>Total Files Size (in bytes):</b>	1510275
-------------------------------------	---------

**This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.**

**New Applications Under 35 U.S.C. 111**

**If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.**

**National Stage of an International Application under 35 U.S.C. 371**

**If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.**

**New International Application Filed with the USPTO as a Receiving Office**

**If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.**



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	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
09/504,783	02/15/2000	Edmund Colby Munger	00479.85672

**CONFIRMATION NO. 8308**

**POWER OF ATTORNEY NOTICE**



22907  
BANNER & WITCOFF, LTD.  
1100 13th STREET, N.W.  
SUITE 1200  
WASHINGTON, DC 20005-4051

Date Mailed: 12/30/2009

**NOTICE REGARDING CHANGE OF POWER OF ATTORNEY**

This is in response to the Power of Attorney filed 12/15/2009.

- The Power of Attorney to you in this application has been revoked by the assignee who has intervened as provided by 37 CFR 3.71. Future correspondence will be mailed to the new address of record(37 CFR 1.33).

/mnguyen/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101





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	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
09/504,783	02/15/2000	Edmund Colby Munger	77580-017 (VRNK-1CP)

**CONFIRMATION NO. 8308**

**POA ACCEPTANCE LETTER**

23630  
MCDERMOTT WILL & EMERY LLP  
28 STATE STREET  
BOSTON, MA 02109-1775



Date Mailed: 12/30/2009

**NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY**

This is in response to the Power of Attorney filed 12/15/2009.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

/mnguyen/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

<b>TO:</b> Mail Stop 8 Director of the U.S. Patent and Trademark Office P.O. Box 1450 Alexandria, VA 22313-1450	<b>REPORT ON THE                  FILING OR DETERMINATION OF AN                  ACTION REGARDING A PATENT OR                  TRADEMARK</b>
--	--

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 Eastern District of Texas on the following  Patents or  Trademarks:

DOCKET NO. 6:10-cv-417	DATE FILED 8/11/2010	U.S. DISTRICT COURT Eastern District of Texas
PLAINTIFF VirnetX Inc.,		DEFENDANT Aastra USA, Inc., Aastra Technologies Ltd., Apple, Inc., Cisco Systems, Inc., NEC Corporation, and NEC Corporation of America
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 6,502,135	12/31/2002	VirnetX Inc.
2 6,839,759	1/4/2005	VirnetX Inc.
3 7,188,180	3/6/2007	VirnetX Inc.
4 7,418,504	8/26/2008	VirnetX Inc.
5 7,490,151	2/10/2009	VirnetX Inc.

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY <input type="checkbox"/> Amendment <input type="checkbox"/> Answer <input type="checkbox"/> Cross Bill <input type="checkbox"/> Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
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



#09504783

IRJ-

IN THE UNITED STATES  
PATENT AND TRADEMARK OFFICE

In re patent of  
VirnetX Inc.  
Patent No. 6,502,135  
Issued: December 31, 2002  
For: Agile network protocol for secure communications with assured system availability

Submission of Prior Art Under 37 CFR 1.501

Hon. Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

The undersigned herewith submits in the above-identified patent the following prior art which is pertinent and applicable to the patent and is believed to have a bearing on the patentability of at least claim 1 thereof:

Gooderum et al U.S. 5,918,018 June 29, 1999

The reference discloses a system and method of a computing system having a plurality of network interfaces between a client/server network strikingly similar to the device of VirnetX inc. It is believed that the reference has a bearing on the patentability of at least claim 1 of the VirnetX Inc. patent.

Insofar as claim 1 is concerned, the reference clearly anticipates the claimed subject matter under 35 U.S.C. 102.

Below is a list of other references which affect one or more of the claims in the patent.

Patent No.:

US5918018	US7020700	US6751738	US6571290	US6501767	US6487598	US6473406	US6345303
US6266704	US6263444	US6226748	US6173399	US6144934	US6072942	US6061721	US6044402
US5999629	US5991881	US5961644	US5935245	US5918019	US5864683	US5848233	US5835727
US5835726	US5835718	US5822531	US5805820	US5793763	US5790548	US5781534	US5768271
US5673322	US5623601	US5606668	US5566170	US5559883	US5557742	US5550984	US5504921
US5491752	US5440723	US5341426	US5337309	US5311593	US5278901	US5276735	US5268962
US5241599	US5220655	US5177788	WO9501023	US7752649	US7702540	US7647243	US7613633
US7583665	US7580919	US7546251	US7475156	US7272625	US7165174	US7149208	US7145898
US7143438	US7143290	US7133940	US7133846	US7133845	US7124302	US7120802	US7120800
US7117165	US7113508	US7095854	US7076652	US7073056	US7069451	US7062500	US7010702
US6950436	US6944657	US6912222	US6909708	US6901509	US6892354	US6874090	US6849045
US6832256	US6816966	US6798776	US6785288	US6775692	US6772332	US6754212	US6754181
US6742040	US6741909	US6731625	US6721286	US6711171	US6704866	US6701377	US6697836
US6687551	US6681213	US6678822	US6625166	US6621505	US6598081	US6598075	US6595417

US6591291	US6590894	US6584480	US6584098	US6581090	US6577734	US6571338	US6571296
US6567471	US6560581	US6553002	US6546011	US6532540	US6519365	US6515968	US6510154
US6505241	US6505232	US6493347	US6482156	US6470383	US6466780	US6463443	US6463057
US6457039	US6453345	US6453343	US6453327	US6446164	US6445703	US6442689	US6438127
US6421714	US6412035	US6411806	US6408367	US6408341	US6397330	US6396831	US6385647
US6378028	US6377691	US6375780	US6373950	US6363363	US6359887	US6359885	US6359882
US6359855	US6357046	US6356948	US6353856	US6351775	US6351772	US6351467	US6345288
US6343298	US6343072	US6341310	US6339830	US6339596	US6336141	US6335927	US6332195
US6330608	US6330240	US6324582	US6324525	US6324161	US6321268	US6321201	US6317729
US6314520	US6314406	US6311218	US6311207	US6308213	US6307837	US6304915	US6304908
US6301223	US6298041	US6295285	US6292834	US6288739	US6285679	US6285675	US6282172
US6275941	US6272341	US6272148	US6272110	US6269394	US6263394	US6256715	US6253027
US6249820	US6247129	US6247128	US6247059	US6243360	US6240513	US6240084	US6237006
US6230173	US6226642	US6222842	US6219803	US6219707	US6216212	US6216163	US6215514
US6212636	US6209041	US6208652	US6206829	US6205483	US6202096	US6202060	US6201819
US6199082	US6195692	US6195366	US6192408	US6189032	US6185680	US6185619	US6182116
US6181711	US6181698	US6178409	US6170012	US6167438	US6163844	US6163843	US6163772
US6161145	US6158011	US6157935	US6157829	US6157644	US6154775	US6154745	US6151679
US6147976	US6145004	US6144962	US6144638	US6141749	US6141423	US6137884	US6137796
US6134591	US6134217	US6130889	US6128316	US6128298	US6125366	US6125186	US6122255
US6119234	US6119230	US6119105	US6118817	US6115780	US6115378	US6112085	US6111893
US6111883	US6111567	US6108786	US6108782	US6108692	US6105134	US6105027	US6105012
US6104716	US6101606	US6101549	US6101531	US6101182	US6098172	US6098108	US6098096
US6094715	US6094485	US6092200	US6092191	US6092116	US6092101	US6091951	US6091733
US6088796	US6088361	US6085238	US6084878	US6081598	US6078733	US6078582	US6076113
US6075783	US6073202	US6073185	US6073176	US6073168	US6072870	US6072800	US6072781
US6070243	US6070191	US6069894	US6067572	US6067569	US6063129	US6061798	US6061796
US6061448	US6058117	US6055575	US6055562	US6052718	US6052629	US6049872	US6049835
US6047338	US6047054	US6046988	US6046980	US6044224	US6041408	US6041379	US6041355
US6041345	US6041342	US6041043	US6041041	US6038677	US6038219	US6035360	US6034680
US6032190	US6029245	US6029175	US6026379	US6023724	US6022315	US6021198	US6018766
US6016319	US6016317	US6014706	US6014694	US6014686	US6014660	US6012066	US6011797
US6011782	US6009474	US6009467	US6009173	US6006328	US6006264	US6005926	US6005843
US6005621	US6003084	US6002767	US5999967	US5999940	US5999525	US5996077	US5996076
US5996016	US5996015	US5996011	US5995705	US5995503	US5991810	US5987611	US5987572
US5987480	US5987132	US5983350	US5983327	US5983270	US5983233	US5983208	US5978951
US5978880	US5978840	US5978594	US5978567	US5974453	US5974056	US5968176	US5968158
US5968133	US5968116	US5966705	US5966528	US5966509	US5963981	US5963746	US5963745
US5963642	US5963202	US5961593	US5960179	US5960170	US5959989	US5959974	US5958052
US5958015	US5958012	US5958008	US5953503	US5951694	US5950195	US5949975	US5949883
US5946674	US5946464	US5944783	US5943424	US5941988	US5940591	US5938737	US5937163
US5937162	US5936940	US5935212	US5931961	US5931946	US5931917	US5930479	US5930359
US5926463	US5926458	US5925126	US5923849	US5923756	US5923646	US5919257	US5918016
US5917824	US5917822	US5917820	US5915087	US5913041	US5913038	US5913024	US5907704
US5907680	US5907677	US5905872	US5905730	US5903651	US5898780	US5896499	US5890010
US5890005	US5889953	US5889943	US5889863	US5884272	US5884246	US5884033	US5884027
US5881131	US5878241	US5878212	US5872920	US5872847	US5872783	US5870610	US5870559
US5870550	US5870545	US5867704	US5867667	US5867660	US5867650	US5867494	US5867484
US5864678	US5862339	US5860073	US5859979	US5859835	US5856974	US5855020	US5854901
US5852607	US5850516	US5850449	US5850446	US5848258	US5848159	US5845091	US5845070
US5844888	US5842043	US5842031	US5841775	US5838683	US5835725	US5835720	US5835714
US5835495	US5832222	US5832216	US5832092	US5831609	US5828894	US5828876	US5828833
US5828832	US5826014	US5825917	US5825884	US5825774	US5822608	US5822537	US5822524
US5822523	US5822434	US5818842	US5815723	US5815501	US5812819	US5812775	US5812771
US5812670	US5812668	US5812552	US5812533	US5809292	US5809147	US5805915	US5805818
US5805803	US5805785	US5805700	US5805595	US5805572	US5802554	US5802320	US5802291
US5802286	US5802278	US5799016	US5798706	US5796951	US5796944	US5796727	US5794059
US5793965	US5793768	US5787472	US5787253	US5787172	US5787080	US5784582	US5781743
US5777989	US5774689	US5774668	US5774660	US5771459	US5771353	US5765015	US5765012
US5764935	US5764909	US5764235	US5761523	US5758085	US5758083	US5758076	US5757924
US5754871	US5754656	US5754651	US5752260	US5752241	US5752067	US5752003	US5748871
US5748736	US5748633	US5745728	US5742762	US5740402	US5740375	US5740362	US5737525
US5734921	US5734865	US5734853	US5734744	US5734656	US5734654	US5732406	US5732078

USS727147	US5727146	US5717944	US5717943	US5717686	US5713037	US5712981	US5710970
US5710935	US5708836	US5708659	US5708655	US5706425	US5699532	US5699528	US5699521
US5699513	US5699500	US5699403	US5699361	US5692124	US5689508	US5687235	US5685004
US5684800	US5682480	US5678045	US5678006	US5673319	US5671279	US5671225	US5668876
US5668857	US5668809	US5666486	US5666484	US5664199	US5659542	US5657452	US5657390
US5654695	US5651002	US5648965	US5646996	US5644751	US5644720	US5640399	US5638448
US5636371	US5636216	US5634099	US5633916	US5633371	US5632029	US5632011	US5630162
US5625836	US5625626	US5625622	US5623605	US5623492	US5621889	US5621796	US5621727
US5621201	US5619648	US5619621	US5617577	US5617547	US5617540	US5615340	US5614891
US5612865	US5611075	US5610981	US5610903	US5608908	US5606493	US5602918	US5600644
US5596686	US5594918	US5594200	US5592470	US5590299	US5590285	US5590199	US5590122
US5588152	US5587726	US5586263	US5586046	US5583940	US5577209	US5574475	US5570360
US5568181	US5566297	US5563875	US5561669	US5559986	US5557747	US5557712	US5553239
US5550816	US5548721	US5548648	US5548646	US5544340	US5541927	US5541921	US5537535
US5537099	US5535365	US5533033	US5533029	US5530809	US5528503	US5526489	US5524238
US5517620	US5515508	US5513346	US5511122	US5509006	US5506973	US5500513	US5495580
US5495533	US5491808	US5491693	US5490729	US5490212	US5488715	US5485579	US5483661
US5477547	US5477531	US5477038	US5475687	US5474430	US5473603	US5473599	US5465206
US5463755	US5463752	US5463702	US5457797	US5457786	US5455865	US5452420	US5446880
US5444491	US5442633	US5442624	US5440719	US5440547	US5440334	US5438571	US5437024
US5432783	US5432775	US5430715	US5423020	US5423002	US5421024	US5421019	US5418922
US5418854	US5416842	US5416781	US5414833	US5408465	US5406628	US5404562	US5404402
US5398245	US5394408	US5392357	US5392223	US5390336	US5388211	US5386548	US5384722
US5381541	US5379289	US5375219	US5375207	US5373559	US5373504	US5371877	US5371852
US5371794	US5369707	US5369640	US5369570	US5367643	US5367517	US5365585	US5361256
US5359717	US5353283	US5349693	US5349686	US5347642	US5347450	US5341477	US5339356
US5337313	US5335366	US5329521	US5325504	US5325433	US5323146	US5321815	US5321695
US5315586	US5309562	US5305385	US5301337	US5301247	US5297242	US5291609	US5291442
US5289585	US5288942	US5285477	US5280480	US5280475	US5276789	US5276678	US5271000
US5263165	US5262875	US5261070	US5261064	US5245606	US5241625	US5241594	US5239648
US5237611	US5227778	US5226172	US5224099	US5222133	US5220516	US5214767	US5212806
US5204966	US5201000	US5195181	US5195089	US5193151	US5191611	US5185860	US5181200
US5179704	US5179556	US5164988	US5164986	US5164839	US5163049	US5159592	US5146581
US5146574	US5142622	US5142272	US5142167	US5136716	US5136642	US5136523	US5133068
US5132964	US5115495	US5111504	US5101402	US5101374	US5091851	US5090583	US5090563
US5086467	US5083265	US5073852	US5067074	US5062060	US5060263	US5057932	US5051991
US5051982	US5038345	US5032979	US5029164	US5025256	US5018137	US5014265	US5010549
US5008815	US5001755	US5001752	US4988990	US4967345	US4964164	US4963995	US4941089
US4935870	US4933937	US4930159	US4924500	US4922486	US4916704	US4905176	US4860201
US4823338	US4821259	US4817091	US4799156	US4799153	US4769811	US4769810	US4745593
US4722502	US4713753	US4692918	US4680753	US4672535	US4667337	US4652993	US4633434
US4613935	US4592049	US4577313	US4574350	US4470114	US4449181	US4438493	US4413315
US4403282	US4400770	US4386416	US4371929	US4325116	US4282572	US4262359	US4227253
US4223380	US4164787	US4034347	US3697768	US20020012473	US20020010793	US20010039579	USRE39360
USRE36751	USH1641	EP0865180	EP0827307	EP0822513	EP0814589	EP0812086	EP0801481
EP0762704	EP0752674	EP0743777	EP0729256	EP0680187	EP0465016	EP0324277	WO9923538
WO9909725	WO9907116	WO9905584	WO9857465	WO9855930	WO9826555	WO9817039	WO9817006
WO9726735	WO9726734	WO9726731	WO9723972	WO9720419	WO9713340	WO9700471	WO9613774
WO9605549	WO9316538	WO9315581	WO9313481	WO9303562	GB2334181	GB2322994	GB2277181
GB2248535	DE4202852	JP11184780	JP09261265	AU0692872			

Respectfully submitted,



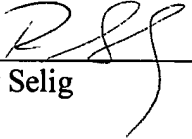
Ray Selig, Esq.  
M-CAM, Inc.

210 Ridge-McIntire Road, Suite 300  
Charlottesville, VA 22903

Certificate of Service

I hereby certify on this 14th day of January 2011, that a true and correct copy of the forgoing "Submission of Prior Art" was mailed by first-class mail, postage paid, to:

VirnetX Inc..  
c/o McDermott Will & Emery  
600 13th Street, NW  
Washington DC 20005-3096

  
\_\_\_\_\_  
Ray Selig

TO: <b>Mail Stop 8</b> <b>Director of the U.S. Patent and Trademark Office</b> <b>P.O. Box 1450</b> <b>Alexandria, VA 22313-1450</b>	<b>REPORT ON THE</b> <b>FILING OR DETERMINATION OF AN</b> <b>ACTION REGARDING A PATENT OR</b> <b>TRADEMARK</b>
---	---

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 Eastern District of Texas - Tyler Division on the following

Trademarks or  Patents. (  the patent action involves 35 U.S.C. § 292.):

DOCKET NO. 6:11-cv-18	DATE FILED 1/12/2011	U.S. DISTRICT COURT Eastern District of Texas - Tyler Division
PLAINTIFF VirnetX, Inc.		DEFENDANT Mitel Networks Corp., et al.
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 6,502,135	12/31/2002	VirnetX, Inc.
2 7,418,504	8/26/2008	VirnetX, Inc.
3		
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY <input type="checkbox"/> Amendment <input type="checkbox"/> Answer <input type="checkbox"/> Cross Bill <input type="checkbox"/> Other Pleading		
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK	
1			
2			
3			
4			
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



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	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
09/504,783	02/15/2000	Edmund Colby Munger	77580-017 (VRNK-1CP)

**CONFIRMATION NO. 8308**

**POA ACCEPTANCE LETTER**

22852  
FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER  
LLP  
901 NEW YORK AVENUE, NW  
WASHINGTON, DC 20001-4413



Date Mailed: 11/02/2011

**NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY**

This is in response to the Power of Attorney filed 11/01/2011.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

/sdstevenson/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101





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	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
09/504,783	02/15/2000	Edmund Colby Munger	77580-017 (VRNK-1CP)

**CONFIRMATION NO. 8308**

**POWER OF ATTORNEY NOTICE**

23630  
McDermott Will & Emery  
600 13th Street, NW  
Washington, DC 20005-3096



Date Mailed: 11/02/2011

**NOTICE REGARDING CHANGE OF POWER OF ATTORNEY**

This is in response to the Power of Attorney filed 11/01/2011.

- The Power of Attorney to you in this application has been revoked by the assignee who has intervened as provided by 37 CFR 3.71. Future correspondence will be mailed to the new address of record(37 CFR 1.33).

/sdstevenson/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re *Inter Partes* Reexamination of: )  
)  
Edmund MUNGER et al. ) Control Nos.: 95/001,679; 95/001,682  
)  
U. S. Patent No. 6,502,135 ) Group Art Unit: 3992  
)  
Issued: December 31, 2002 ) Examiner: Behzad Peikari  
)  
For: AGILE NETWORK PROTOCOL FOR ) Confirmation Nos. 9786; 1074  
SECURE COMMUNICATIONS WITH )  
ASSURED SYSTEM AVAILABILITY )

Mail Stop *Inter Partes* Reexam  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Dear Commissioner:

**REVOCATION OF POWER OF ATTORNEY,  
STATEMENT UNDER 37 C.F.R. § 3.73(b),  
AND GRANT OF NEW POWER OF ATTORNEY**

The undersigned, a representative authorized to sign on behalf of the assignee owning all of the interest in U.S. Patent No. 6,502,135 (“the ’135 patent”), hereby revokes all previous powers of attorney or authorization of agent granted in the ’135 patent before the date of execution hereof.

In compliance with 37 C.F.R. § 3.73(b), the undersigned verifies that VirnetX Inc. is the assignee of the entire right, title, and interest in the ’135 patent by virtue of an assignment recorded in the U.S. Patent and Trademark Office at Reel 018757, Frame 0326 on January 10, 2007.

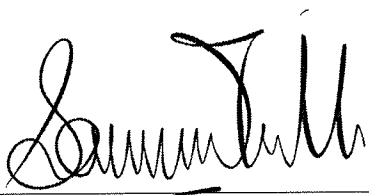
The undersigned representative of the assignee hereby grants its power of attorney to the patent practitioners associated with **Finnegan, Henderson, Farabow, Garrett & Dunner,**

Attorney Docket Nos. 11798.0001; 11798.0009  
Control Nos. 95/001,679; 95/001,682

**L.L.P., Customer Number 22,852**, to transact all business in the Patent and Trademark Office connected with the '135 patent, including the reexamination proceedings assigned control nos. 95/001,679 and 95/001,682, and in any other proceedings involving the '135 patent.

Please also send all future correspondence concerning the '135 patent to the address associated with **Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P., Customer Number 22,852**.

Dated: 11/30/12

By:   
\_\_\_\_\_  
Sameer Mathur  
Vice President, Corporate Development and Product  
Marketing  
VirnetX Inc.

## Electronic Acknowledgement Receipt

<b>EFS ID:</b>	14367135
<b>Application Number:</b>	09504783
<b>International Application Number:</b>	
<b>Confirmation Number:</b>	8308
<b>Title of Invention:</b>	AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILTY
<b>First Named Inventor/Applicant Name:</b>	Edmund Colby Munger
<b>Customer Number:</b>	23630
<b>Filer:</b>	Joseph Edwin Palys./connie sisk
<b>Filer Authorized By:</b>	Joseph Edwin Palys.
<b>Attorney Docket Number:</b>	77580-017 (VRNK-1CP)
<b>Receipt Date:</b>	03-DEC-2012
<b>Filing Date:</b>	15-FEB-2000
<b>Time Stamp:</b>	14:59:01
<b>Application Type:</b>	Utility under 35 USC 111(a)

### Payment information:

Submitted with Payment	no
------------------------	----

### File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Power of Attorney	POA_135.pdf	56554 <small>58f785d74ed85192982446e052d13aae168c6b6b</small>	no	2

### Warnings:

### Information:

**This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.**

**New Applications Under 35 U.S.C. 111**

**If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.**

**National Stage of an International Application under 35 U.S.C. 371**

**If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.**

**New International Application Filed with the USPTO as a Receiving Office**

**If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.**



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	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
09/504,783	02/15/2000	Edmund Colby Munger	11798.0001;11798.0009

**CONFIRMATION NO. 8308**

**POA ACCEPTANCE LETTER**

22852  
FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER  
LLP  
901 NEW YORK AVENUE, NW  
WASHINGTON, DC 20001-4413



Date Mailed: 12/20/2012

**NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY**

This is in response to the Power of Attorney filed 12/03/2012.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

/sharris/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101



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	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
09/504,783	02/15/2000	Edmund Colby Munger	77580-017 (VRNK-1CP)

**CONFIRMATION NO. 8308**

**POWER OF ATTORNEY NOTICE**



23630  
McDermott Will & Emery  
The McDermott Building  
500 North Capitol Street, N.W.  
Washington, DC 20001

Date Mailed: 12/20/2012

**NOTICE REGARDING CHANGE OF POWER OF ATTORNEY**

This is in response to the Power of Attorney filed 12/03/2012.

- The Power of Attorney to you in this application has been revoked by the assignee who has intervened as provided by 37 CFR 3.71. Future correspondence will be mailed to the new address of record(37 CFR 1.33).

/sharris/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101