United States Patent [19]
W00tt0n et al.

US006128298A

6,128,298
Oct. 3, 2000

[11] Patent Number:

[45] Date of Patent:

[54] INTERNET PROTOCOL FILTER

[75] Inventors: Bruce Anthony Wootton, Raleigh, NC; William G. Colvin, Milton,
Canada

[73] Assignee: Nortel Networks Corporation, Montreal, Canada

[21] Appl. No.2 08/842,328

[22] Filed: Apr. 24, 1997

Related US. Application Data
[60] Provisional application No. 60/015,945, Apr. 24, 1996.

[51] Int. C1.7 .... H04L 12/56 [52] US. Cl. 370/392, 370/390, 370/401, 713/201
[58] Field Of Search 370/351, 352, 370/355, 389, 390, 392, 393, 400, 401,
402, 409, 395/2006, 200.62, 200.68, 20072,
713/201

[56] References Cited

U.S. PATENT DOCUMENTS

5,309,437 5/1994 Perlman et al. .... 370/401 5,383,179 1/1995 Saini et al. .... 370/393 5,400,334 3/1995 Hayssen .... 370/245 5,606,668 2/1997 Shwed.
5,623,601 4/1997 Vu.
5,778,174 7/1998 Cain.
5,781,550 7/1998 Templin et al. .... 370/401 5,793,763 8/1998 Mayes et al. .... 370/389 5,826,014 10/1998 Coley et al. .
5,835,726 11/1998 Shwed et al. .

FOREIGN PATENT DOCUMENTS

0 465 201 1/1992 European Pat. Off. .

OTHER PUBLICATIONS

Axner, “Differing Approaches to Virtual LANs”, Business Communications Review, Dec. 1993, pp. 42—45.
Bryan, “Build a Firewall”, Byte, Apr. 1995, pp. 91—96. Bryan, “Firewalls for Sale”, Byte, Apr. 1995, pp. 99—104.

PRIVATE
N ETWORK

Carl—Mitchell, et al., “Building Internet Firewalls”, Unix Worla', Feb. 1992, pp. 93—103.
Chapman, “Network (In)Security Through IP Packet Filter ing”, UNIX Security Symposium III Proceedings, Balti more, MD, Sep. 14—16, 1992, pp. 63—76.
Cheswick, “The Design of a Secure Internet Gateway”, USENIX Summer Conference, Anaheim, CA, Jul. 11—15, 1990, pp. 233—237.
Ho, “Implementation of a Secure Gateway on Hughes Aircraft’s Engineering Design Network”, 15”” Conference on Local Computer Networks, IEEE, Minneapolis, MN., Sep. 30—Oct. 3, 1990, pp. 180—182.
Hoover, “Securing the Enterprise, Firewalls Can Keep You from Getting Burned”,Internet World, Feb. 1995 , pp. 39—47. Koblas, et al., “SOCKS”, UNIX Security Symposium III Proceedings, Baltimore, MD, Sep. 14—16, 1992, pp. 77—83. Lottor, “TCP Port Service Multiplexer (TCPMUX)”, Inter net rfc 1078 (1988), pp. 1,2.
Luotonen, et al., “World—Wide Web Proxies”, Computer Networks and ISDN Systems 27 (1994), pp. 147—154.

(List continued on neXt page.)

Primary Examiner—Ajit Patel
Assistant Examiner—Bob A. Phunkulh

Attorney, Agent, or Firm—Foley & Lardner

[57] ABSTRACT

The IP ?lter, embodying the present invention, is a commu nications device designed to provide public network or Internet access to nodes of private networks, advantageously without requiring the private nodes on such networks to register public Internet addresses. The IP ?lter presents a single IP address to the Internet and uses a plurality of IP ports to solve the problem of IP address conservation. It initiates sessions by assigning private side IP sessions to a unique port of the IP ?lter’s public address. The IP ?lter effects a translation between a source port number for the private network and a destination port number for the public network for communication therebetween. Bene?ts of the IP

?lter include private node security and conservation of Internet-registered addresses.

32 Claims, 2 Drawing Sheets

/6

INTERNET

PUBLIC
NETWORK

6,128,298
Page 2

OTHER PUBLICATIONS

Marotta, et a1., “Internetworking Data Services”, 16th Con
ference on Local Computer Networks, IEEE, Minneapolis,
MN, Oct. 14—17, 1991, pp. 223—229.
PanZieri, et a1., “Interfacing UNIX to Data Communications Networks”, IEEE Transactions on Software Engineering, vol. SE—11, Oct. 1985, pp. 1016—1032.
Schauer, et al., “An Internet Gatekeeper”, UNIX Security Symposium III Proceedings, Baltimore, MD, Sep. 14—16, 1992, pp. 49—61.
Schroeder, et al. “Autonet: A High Speed, Self—Con?guring Local Area Network Using Point—to—Point Links”, IEEE Journal on Selected Areas in Communications, vol. 9, No. 8, Oct. 1991, pp. 1318—1334.
Shapiro, “Structure and Encapsulation in Distribution Sys tems: The Proxy Principle”, The 6th International Confer ence on Distributed Computing Systems, IEEE, Cambridge, MA, May 19—23, 1986, pp. 198—204.
Snyder, “Choosing the Right Firewall to Defend Your Net work” Network World, vol. 12, No. 10, Mar. 5, 1995, p. 1. Stephensen, “A Blueprint for Firewalls”, LAN Magazine, Feb. 1995, pp. 63—70.
Tam, et al. “CAPNET—An Approach to Ultra High Speed Networ ”, IEEE International Conference on Communica tions, 1990, pp. 323.1.1—323.1.7.

Tolly, “Evaluating Port Switching Hubs—A reality check for virtual workgroups”, Data Communications, Jun. 1993, pp. 52—62.

Treese, et al., “X Through the Firewall, and Other Applica tion Relays”, USENIX Summer 1993 Technical Conference, Cincinnati, OH, Jun. 21—25, 1993, pp. 87—98.

Cheswick and Bellovin, “Firewalls and Internet Security: Repelling the Wily Hacker”, Addison—Wesley, 1994, pp. 34—36, 54—75.

Comer, “Internetworking with TCP/IP”, Prentice—Hall, Inc., 1988, pp. 120—127, 137—141, 194, 195, 208—214, 346, 347.

McClimans, “Workarounds Ease the IP Address Shortage”, Data Communications, section Software Views, vol 24, No. 2, Feb. 23, 1995, (p. 33), pp. 3—5.

Kostick, “Building a Linux Firewall”, Linux Journal, Apr. 1996, pp. 49, 52, 53, 55, 57, 58, 61.

Egevang et al., “Internet Engineering Task Force, USA” XP2040992 pp. 1—8 (1994).

Stallings, “Internet Security Handbook” XP2040993 pp. 27—37 (1995).

U.S. Patent 0a. 3, 2000 Sheet 1 of2 6,128,298

INTERNET

PR IVATE

NETWORK NETWORK

FIG. I

U.S. Patent 0a. 3, 2000 Sheet 2 of2 6,128,298

/2 \

40 \ ADDRESS usER A 42
TRANSLATION INTERFACE

33 '— lP HANDLER

34 '- ARP ETHERNET TABLE ‘\36

30 '- PACKET DRIVER PACKET DRIVER A 32

/0 /4 H.W. H.W.

PRIVATE

NETWORK PUBLIC

NETWORK

FIG. 2

6,128,298
1
INTERNET PROTOCOL FILTER

This application is based on provisional application 60/015,945 ?led Apr. 26, 1996.

BACKGROUND OF THE INVENTION

The present invention generally relates to internetWork ?reWalls and, in particular, to an internet protocol (IP) ?lter Whereby a private IP netWork domain is mapped to a single IP address on the public Internet.
FireWalls are generally knoWn and characteriZed by com puter servers Which function to couple nodes Within the domain of the private netWork to nodes in a public netWork domain, such as the Internet. A de?ciency of the knoWn ?reWall products is the need for a unique public IP address for each concurrent session or interaction betWeen public and private nodes.
A ?reWall providing conservation of public IP addresses Would be desirable.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide a neW and improved apparatus for communicatively coupling tWo netWorks.

The invention, therefore, according to a ?rst exemplary aspect provides a method of interfacing private and public data communications netWorks, through a ?lter node in communication With both netWorks, the ?lter node having an address knoWn in the public netWork, comprising the steps of: routing from nodes in the private netWork, to the ?lter node, data packets having destination information, Which includes a destination address and a destination port, corresponding to nodes in the public netWork and having source information, Which includes a source address and a source port, of the respective private netWork nodes; for each data packet received from the private netWork, at the ?lter node, maintaining the source information taken from the data packet in correlation With a unique value represent ing a port of the ?lter node, and replacing in the data packet
the source address With the ?lter node address and the source

port With the ?lter node port value; and routing from the ?lter node, in the public netWork, the data packets having the replaced source information, according to the destination information in each, to the corresponding public netWork nodes.

According to a second exemplary aspect, the invention provides a method of interfacing private and public data communications netWorks, through a ?lter node in commu nication With both netWorks, comprising the steps of: (a) receiving at the ?lter node, from the private netWork, a data packet having an a destination address corresponding to a node in the public netWork and a source address correspond ing to a node in the private netWork; (b) maintaining, by the ?lter node, the source address taken from the data packet; (c) replacing, in the data packet, the source address With an address of the ?lter node; (d) routing from the ?lter node, in the public netWork, the data packet having the replaced source address, according to the destination address, to the corresponding public netWork node; (e) Waiting for a return packet from the public netWork, responsive to the data packet having the replaced source information; replacing, in the return packet, the destination address With the main tained source address; and (g)routing from the ?lter node, in the private netWork, the return packet having the replaced destination address to the corresponding private netWork node.

10

15

25

35

45

55

65

2

According to a third eXemplary aspect, the invention provides a method of operating a ?lter node for interfacing ?rst and second data communications netWorks, comprising the steps of: receiving from the ?rst netWork, a data packet having destination information, Which includes a destination address and a destination port, corresponding to a node in the second netWork and having source information, Which includes a source address and a source port, corresponding to a node in the ?rst netWork; maintaining the source information taken from the data packet in correlation With a unique value representing a port of the ?lter node; replacing in the data packet the source address With an address of the ?lter node and the source port With the ?lter node port value; and sending to the second netWork the data packet having the replaced source information, Whereby that packet is routed according to its destination information to the corre sponding second netWork node.
According to a fourth exemplary aspect, the invention provides a ?lter node for interfacing ?rst and second data communications netWorks, comprising: means for receiving from the ?rst netWork, a data packet having destination information, Which includes a destination address and a destination port, corresponding to a node in the public netWork and having source information, Which includes a source address and a source port, corresponding to a node in the ?rst netWork; means for maintaining the source infor mation taken from the data packet in correlation With a unique value representing a port of the ?lter node; means for replacing in the data packet the source address With an address of the ?lter node and the source port With the ?lter node port value; and means for sending to the second netWork, the data packet having the replaced source information, Whereby that packet is routed according to its destination information to the corresponding second net Work node.

An IP ?lter, embodying the present invention, is a com munications device designed to provide public netWork or Internet access to nodes of private netWorks, advantageously Without requiring the private nodes on such netWorks to register public Internet addresses. The IP ?lter presents a single IP address to the Internet and uses a plurality of IP ports to solve the problem of IP address conservation. It initiates sessions by assigning private side IP sessions to a unique port of the IP ?lter’s public address Whereby up to 64,512 (=65,536 total —1,024 Well knoWn ports) concurrent sessions may be supported through the single IP address. The IP ?lter effects a translation betWeen a source port number for the private netWork and a destination port number for the public netWork for communication therebe tWeen. Bene?ts of the IP ?lter include private node security and conservation of Internet-registered addresses.
In a particular embodiment, the IP ?lter may support three data transport protocols over the internet protocol: transmis sion control protocol (TCP), user datagram protocol (UDP) and Internet control message protocol (ICMP). Packets of other protocols may be ignored.
The TCP protocol prepends a TCP header to a data packet. The source port and destination port numbers are contained in this header. The Internet addresses of the source and destination nodes are contained in the IP header. The IP

address and port information extracted from each packet Will be used to determine Where the IP ?lter should route this

packet.
The IP ?lter maintains a lookup table of information on each TCP connection. This information includes the port from the private node, the private IP address, the assigned

6,128,298
3

port number of the destination node, and the port number of
the IP ?lter in the form of an index. When a packet is received from the private network, the private address and
port number are added to the table as a neW entry, if an entry corresponding to this packet is not found in the table and if the TCP header indicates that this is a neW connection

request. Then the source address and port number in the packet header are replaced With the IP ?lter’s IP address and port number, and the packet is transmitted to the Internet. When the IP ?lter receives a packet from the Internet, the destination port number is used to index the lookup table. When the corresponding table entry is found, the destination address and port number are replaced With the private networks IP address and port number, and the packet is transmitted to the private netWork. If the received packet’s source port is different from the port recorded in the table, and if the packet header information indicates that this packet is the ?rst response on the connection, then the lookup table is updated With the port number assigned by the Internet node, if needed. When the IP ?lter detects an end of transmission code in the packet, the lookup table entry is Zeroed. If the IP ?lter receives packets from the Internet that do not have entries in the lookup table corresponding to the IP ?lter port, it ignores the packets.
The UDP protocol is connectionless, as opposed to TCP, a connection-oriented protocol. The UDP header contains no codes governing initial connection or end of transmission. The data of interest in the UDP header are the source port and destination port. This information, along With the Inter net addresses contained in the IP header, are used to deter mine Where the IP ?lter should route this packet.
The IP ?lter maintains a lookup table of information on each UDP session. When the IP ?lter receives a UDP packet from the private netWork, it records the source address, the source port number, the destination port number, and the assigned IP ?lter port number as the index to the table. Then the private node address and port number in the packet header are replaced With the address and assigned port number of the IP ?lter. Then the packet is transmitted to the
Internet.

When the IP ?lter receives a UDP packet from the Internet, it indexes the UDP lookup table and replaces the packet’s destination information, namely the IP ?lter address and assigned port number, With the private address and port number from the lookup table. The lookup table also main tains an interval indication for an expiration timer on data gram packets received as per standard UDP implementa tions. If the IP ?lter receives packets from the Internet that do not have entries in the lookup table corresponding to the IP ?lter port, it ignores the packets.
As ICMP packets do not contain port numbers of either source or destination, any ICMP packets received from the private netWork are processed one at a time, With buffering of additional ICMP packets. The IP ?lter reads the private address from the packet header and replaces it With the address of the IP ?lter. The packet is transmitted to the Internet, and the IP ?lter Waits for the response. When it receives the responding packet, the destination address in the packet header is changed from that of the IP ?lter to that of the node on the private netWork. Then the IP ?lter transmits the packet to the private netWork.
To successfully deliver packets over an IP protocol netWork, each node must maintain a table of other hosts’ IP addresses and their corresponding Ethernet addresses in an Ethernet based data communications netWork. The nodes actually use the IP addresses and the Ethernet addresses to

10

15

25

35

45

55

65

4

address packets. The relationship betWeen the tWo addresses is dynamic; that is, a node With an IP address may change its Ethernet address. The information in the address table is obtained from the replies to the node’s broadcast of ARP packets. The source node broadcasts ARP packets to request the Ethernet address of the destination node, given the destination node’s IP address. If the destination node

receives the packet, it sends a reply packet With the requested information.
Though it does not maintain a true ARP table, the IP ?lter passes ARP packets in a manner similar to TCP and UDP packet passing. When the IP ?lter receives an ARP packet from a node on the private netWork destined for the public netWork, it replaces the source address information With the ?lter’s address information. The private node’s IP address and the target IP address are placed in a lookup table. When the target node replies With its oWn Ethernet address, the destination address information is changed from that of the IP ?lter to that of the private node before transmitting the packet to the private node. The private node address infor mation is obtained from the table. When an ARP packet is destined for the ?reWall, the ARP packet does not pass through the IP ?lter but is restricted to communications betWeen the ?lter and the one side of the netWork.

Events and errors encountered by the IP ?lter may be logged, for example, by Writing them into a text ?le. The IP ?lter ideally Will process packets as fast as the netWorks present them but When netWork traf?c is too heavy, the IP ?lter Will then buffer the packets in tWo queues, one for the private netWork and one for the Internet.
TWo source and destination lookup tables may be utiliZed, one for TCP packets and the other for UDP packets. Each table is directly indexed by the IP ?lter port number assigned to the communication session. The table entries contain the

IP address of the private node, the source port of the private node, and the destination port of the Internet node. If there is no connection on a certain IP ?lter port, then the corre sponding entry in the table may be Zeroed. Packets arriving from both the private netWork and the Internet are processed using the same lookup table. This arrangement assumes that of the available IP ?lter communications ports some are designated for UDP communication and some for TCP communication.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention Will be better understood from the folloW

ing description together With reference to the accompanying draWings, in Which:
FIG. 1 is a schematic representing an internet protocol ?lter coupling a private netWork and a public netWork; and FIG. 2 is a block diagram representing internal compo nents of the ?lter.

DETAILED DESCRIPTION

Referring to FIG. 1, shoWn for illustration of the present invention is a private netWork 10 communicatively coupled through an internet protocol (IP) ?lter 12 to a public netWork 14 Which may form part of a global data netWork, otherWise referred to as the Internet 16. The private netWork 10 represents a conventional data communications netWork, such as a local area netWork (LAN), having a plurality of nodes 18 each being identi?ed by a unique IP address Within the domain of the private netWork 10. The public netWork 14 and Internet 16 are representative of public domain data communications netWorks also having a plurality of nodes 20 With corresponding IP addresses.

6,128,298
5

The IP ?lter 12 acts as a gateway through which data packets are exchanged between the private network 10 and
the public network 14, thereby providing Internet access to
the nodes 18 of the private network 10. The IP ?lter 12 constitutes one of the private network nodes 18 and is the only such node to have a public IP address that is Internet registered, whereby the IP ?lter 12 essentially also consti tutes one of the public nodes 20 and its IP address is known in the public domain. The IP addresses of the other private network nodes 18 are reserved for the private network 10, and not known or registered in the public Internet address domain. As is conventional, associated with the IP address of the IP ?lter 12 are a plurality of IP ports, speci?cally 65,536 in total of which 64,512 are not reserved for pre de?ned protocols and can be used for address translations. Communications between nodes 18 on the private net work 10 are unaffected by the presence of the IP ?lter 12, but to access the public network 14 and particularly the nodes 20 therein, the private nodes 18 route all communications requests through the IP ?lter 12. The IP ?lter 12 manages the communications between private nodes 18 and the Internet nodes 20 by modifying header information of data packets received from the private network 10 before transmitting each to the public network 14. The modi?cations cause the communications between the private nodes 18 and the public Internet nodes 20 to actually be between the IP ?lter 12 and the Internet nodes 20, which route all return com munications to the IP ?lter 12 which subsequently routes the return data packets to the private nodes 18.
The IP ?lter 12 accepts no connection requests from the public network 14. All communications between private nodes 18 and public nodes 20 are initiated by the private nodes 18. The IP ?lter 12 is designed to support three data transport protocols over the internet protocol: TCP, UDP and ICMP messages; packets of other protocols are rejected or ignored.
Atranslation table is maintained by the IP ?lter 12 to map address and ports for packets received from the private network 10 destined to the public network 14 and vise versa. The translation table contains the following for each entry:

private IP address
private port
internet (public) IP address
internet (public) Port
timer

session type/state
Ethernet address

(PIP) (pPort)
(iIP)
(iPort)

The basic translation substitutes IP addresses and ports from the private network side to the IP ?lter’s IP address and ports, thereby hiding all nodes 18 on the private network 10 from the public network 14.
Apacket originating on the private network side speci?es
a source—destination of

(pIP, pPort—iIP, iPort)

This de?nes a “socket” in which the endpoints of the connection (source and destination) are de?ned by the IP addresses in the IP header and the ports in the TCP or UDP header.
The IP ?lter 12 will translate the above to

(frIP, frPort—iIP, iport)

15

25

35

45

55

65

6

where frIP is the IP address of the IP ?lter 12 on the public network 14, and frPort is the index into the translation table plus an offset value, for example, of 1024 to skip using well known ports. The frPort represents an arbitrary port.
The internet node 20 will reply with a packet

(iIP, iPort—frIP, frPort)

which will be received by the IP ?lter 12 and translated thereby to

(iIP, iport—pIP, pPort)

In general, to translate from the private side, the values (protocol type, pIP, pPort, iIP, iport) must be located in the
translation table. This should be done with a hash table

lookup.
Translating from the public side can be a direct table lookup since frPort minus 1024 is the index into the table. If (iIP, iport) in the packet does not match the corresponding entries in the table, then an unauthoriZed access is logged and the packet dropped.
In translating packets, when a port is substituted in the TCP or UDP header, the checksum in both the TCP/UCP and IP header must be recalculated. When an IP address is substituted in the IP header, the IP header checksum must be recalculated.

Following are special considerations for different proto cols supported by the IP ?lter 12.
In respect of TCP, when a SYN packet is received from the private network 10, the IP ?lter 12 locates an unused entry in the table and ?lls it in, setting the type to TCP and state to SYN. Then the packet is forwarded by the general scheme above. If no free entries exist in the table, then the packet is dropped and the event is logged.
If a SYN packet is received from the public network 14 interface, it is treated as unauthoriZed and logged (except for FTP special case described below). However, a SYN+ACK packet is forwarded if the state of the translation table entry is SYN. After forwarding such a packet the state set to OPEN.

If a FIN packet is received by the IP ?lter 12 and if the state in the translation table is not FIN, the state is set to FIN and the packet forwarded. If the state is FIN, then the packet is forwarded and the translation table entry is deleted by setting it to 0. AFIN must be sent by each side to close a TCP connection.

If a RST packet is received, then the translation table entry is deleted.

Having regard now to the UDP protocol, when any UDP packet is received from the private network 10 side, the IP ?lter 12 ?rst tries its standard lookup. If a translation table entry is not found, an unused entry is set up and the state set to OPEN. If a free entry is not found in the table, then rather than dropping the packet, a random UDP in the table is overwritten. Since UDP is connectionless and consequently an unreliable transport, if a packet is received from the public network 14 that would have needed the entry that was overwritten, that packet will be dropped and the node 18 on the private side will need to retry.
With regard to FTP, an FTP client establishes a TCP “control” connection with an FTP server on a particular port, for example, port 21. However, when data is to be transmitted, the FTP server will open a TCP connection from its “data” port, for example, which is default 20, to a destination port speci?ed by the client.

6,128,298
7

To support this, packets sent by the private network 10 to
port 21 need to be analyzed for an FTP “port” command at
the IP ?lter 12. If detected, then a neW entry in the table must be set up With pPort set to the value in the FTP port command. The IP address and port number in the FTP command must be changed to the IP ?lter’s address and port before forwarding the packet. The state is set to FTPDATA. When a SYN packet is received from the public netWork 14, if a table entry exists and is in FTPDATA state, then the packet is forWarded and the state set to OPEN.
For the ICMP protocol, if an ICMP packet is received from the private netWork 10 and if that packet is an echo request (ping), then the IP ?lter 12 locates a neW entry in the translation table. The sequence ?eld of the packet is stored in pPort in the table and the table indeX is put in the sequence ?eld of the packet. The ICMP checksum is recalculated and the standard IP header substitution is done. The type is set to ICMP and state to PING and the timer set to 1 minute.

If an echo reply (ping) is received from the public netWork 14 interface, then the sequence ?eld is used as the indeX into the table. If the state is PING, then pPort in the table is substituted into the sequence ?eld of the packet, the ICMP checksum recalculated and the standard IP header substitu tion is done. The table entry is then deleted.
If an echo request (ping) is received from the public netWork 14, then the IP ?lter 12 Will reply. This alloWs internet access to con?rm that the IP ?lter 12 is reachable

and running.
If a Destination Unreachable packet is received from the public netWork 14, then the header information contained is eXtracted. If the protocol Was TCP or UDP, the (frIP, frPort—iIP, iport) of the originating packet can be deter mined and the translation table entry located.
If the IP address eXtracted from the ICMP matches the

address in the table, the IP ?lter 12 forWards the packet to the private netWork 10 using the standard scheme.
All other ICMP packets received from either side are dropped and logged.
Since most data communications protocols are based on either the UDP or TCP protocols, these other protocols are compatible With the IP ?lter 12 as long as they do not initiate negotiations like FTP to have the server open a connection back to the client. Examples of other compatible protocols include: Telnet; TFTP (Trivial File Transfer Protocol); DNS (Domain Name Services); and Web broWsers.
Whenever a packet is transmitted in either direction, the timer ?eld of the translation table entry is set to the con?g ured timeout value (except ping). Each minute, the timer
?eld of all active entries in the tables are decremented and

if they become 0, then the translation table entry is deleted.
This Will clear out UDP and PING entries Which are no

longer in use and also TCP entries Which have had an abnormal termination and did not send FIN from each side. It could be a security hole to leave an unused entry in the table for too long. A good timeout value to be con?gured Would be just longer than the typical TCP keep alive. According to a particular embodiment, the private net Work 10 and the public netWork 14 are Ethernet based LANs. The IP ?lter 12 may be implemented by a data processing platform Which is equipped With tWo conven tional Ethernet hardWare interfaces connected to netWorks

10 and 14, respectively, and Which is provisioned With appropriate softWare to implement the functionality of the IP ?lter 12.
Internal components of the IP ?lter 12 in terms of soft Ware eXecutable by the data processing platform are shoWn in FIG. 2. The internal components include tWo packet

10

15

25

35

45

55

65

8

drivers 30 and 32, an address resolution protocol (ARP) table 34, an Ethernet address table 36, an IP handler 38, an address translation 40 and a user interface 42. The packet drivers 30 and 32 control the Ethernet hardWare interfaces in

order to communicate With, respectively, the private netWork 10 and the public netWork 14. The IP handler 38 provides a router functionality for receiving and forWarding messages, and maintains the ARP table 34 and the Ethernet table 36. The address translation 40 effects translation betWeen source

port numbers from the private netWork 10 and the destina tion port numbers on the public netWork side 14. The user interface 42 enables an operator, via a keyboard and display terminal attached to the processing platform, to interface With the IP ?lter 12. Functions keys are provided to con?g ure the IP ?lter, vieW or copy log ?les, display status, etc. The log ?le Will contain the connect time of TCP or UDP sessions, inbound and outbound traf?c statistics, and invalid access to the IP ?lter 12. To prevent the log ?le from groWing too large, this information Will be logged to a neW ?le When the date changes.
Routing of packets to and from the IP ?lter 12 is described in the folloWing in terms of a public interface, from the vieW of the public netWork 14, and of a private interface, from the vieW of the private netWork 10.
The public interface behaves as a host on the LAN segment. To forWard a packet, it checks to see if the destination IP is on the local LAN segment. If it is, it looks up the IP address in its ARP table to ?nd the Ethernet address. If there is no entry in the ARP table, it must put the packet on a queue and send out an ARP request to get the Ethernet address. Standard aging out of ARP table entries needs to be done. If the IP destination is not on the LAN

segment, it Will forWard the packet to the con?gured default router. ICMP Redirect messages sent by the default router Will be ignored.
The private interface effects the functionality of a router, as it needs to be able to forWard packets to one or more routers to communicate With the remote client stations. A

large remote client netWork may access multiple router machines. Conventional routing can result in large routing tables because the routing entries become host addresses instead of subnet addresses. That is, if the netWork is set up so that a client may come in through either Router1 or Router2, then no single router can be the router for the subnet that that client station is on. A conventional router

that Would get routing tables via RIP from all routers on the private netWork Would end up With a large table of host addresses for each remote client connected. This can affect

performance in the search time necessary to ?nd the route, the memory required for large tables and the amount of RIP traffic on the LAN segment betWeen all these routers.
To handle routing in this environment, the IP ?lter Will maintain an Ethernet table. For every packet that is for Warded from the private to public side, if a translation entry eXists, use its Ethernet indeX to compare With the Ethernet source address of the incoming packet. If they match, nothing more needs to be done. OtherWise, the Ethernet table is searched for the source Ethernet address, adding a neW Ethernet table entry if not found. The indeX to the Ethernet table is then saved in the translation table entry. Then When a packet is being translated from the public to private side, the Ethernet address can be retrieved directly from the indeX in the translation table. Thus packets Will be routed to the router Which forWarded the packet to the IP ?lter.
Those skilled in the art Will recogniZe that various modi ?cations and changes could be made to the invention With

6,128,298
9

out departing from the spirit and scope thereof. It should
therefore be understood that the claims are not to be con

sidered as being limited to the precise embodiments set forth
above, in the absence of speci?c limitations directed to each embodiment.
What is claimed is:

1. A method of interfacing private and public data com munications networks, through a ?lter node in communica tion With both networks, the ?lter node having an address knoWn in the public netWork, comprising the steps of: routing from nodes in the private netWork, to the ?lter node, outgoing data packets having destination information, Which includes a destination address and
a destination port, corresponding to nodes in the public netWork and having source information, Which includes
a source address and a source port, of the respective private netWork nodes;
for each outgoing data packet received from the private netWork, at the ?lter node, maintaining the source information taken from the outgoing data packet in correlation With a unique value representing a port of the ?lter node, and replacing in the outgoing data packet the source address With the ?lter node address and the source port With the ?lter node port value; and routing from the ?lter node, to nodes in the public netWork, the outgoing data packets having the replaced source information, according to the destination infor mation in each, to the corresponding public netWork nodes.

2. A method as claimed in claim 1, comprising the steps of:

routing from nodes in the public netWork, to the ?lter node, incoming data packets each having the address of the ?lter node as the destination address;
for each incoming data packet received from the public netWork, at the ?lter node, correlating the destination port of the destination information in the incoming data packet to particular source information being main tained and replacing, in the incoming data packet, the destination information With the particular source infor mation;
routing from the ?lter node, in the private netWork, the incoming data packets having the replaced destination information to the corresponding private netWork nodes.

3. Amethod as claimed in claim 2, comprising ignoring by the ?lter node any incoming data packet received from the public netWork, if the destination port of the destination information in that incoming data packet can not be corre lated to the maintained source information.

4. A method as claimed in claim 3, Wherein maintaining the source information includes storing the source informa tion from each outgoing data packet as an entry in a lookup table, and the ?lter node port value correlating to the source information constitutes an indeX into the table for that entry.
5. Amethod as claimed in claim 4, Wherein the incoming and outgoing data packets include packets in accordance With a transmission control protocol (TCP) over an internet

protocol (IP).
6. A method of interfacing private and public data com munications netWorks, through a ?lter node in communica tion With both netWorks, the ?lter node having an address knoWn in the public netWork, comprising the steps of: routine from nodes in the private netWork, to the ?lter node, outgoing data packets having destination information, Which includes a destination address and

10

15

25

35

45

55

65

10

a destination port, corresponding to nodes in the public netWork and having source information, Which includes
a source address and a source port, of the respective

private netWork nodes;
for each outgoing data packet received from the private netWork, at the ?lter node, maintaining the source information taken from the outgoing data packet in correlation With a unique value representing a port of the ?lter node, and replacing in the outgoing data packet the source address With the ?lter node address and the source port With the ?lter node port value;

routing from the ?lter node, to nodes in the public netWork, the outgoing data packets having the replaced

source information, according to the destination infor mation in each, to the corresponding public netWork nodes;
routing from nodes in the public netWork, to the ?lter node, incoming data packets each having the address of the ?lter node as the destination address;
for each incoming data packet received from the public netWork, at the ?lter node, correlating the destination port of the destination information in the incoming data packet to particular source information being main tained and replacing, in the incoming data packet, the destination information With the particular source infor mation;
routing from the ?lter node, in the private netWork, the incoming data packets having the replaced destination information to the corresponding private netWork nodes;
ignoring by the ?lter node any incoming data packet received from the public netWork, if the destination port of the destination information in that incoming data packet can not be correlated to the maintained source information,
Wherein maintaining the source information includes stor ing the source information from each outgoing data packet as an entry in a lookup table, and the ?lter node port value correlating to the source information con stitutes an indeX into the table for that entry;
Wherein the incoming and outgoing data packets include packets in accordance With a transmission control pro tocol (TCP) over an internet protocol (IP); and receiving at the ?lter node an outgoing TCP packet from the private netWork; and if an entry corresponding to the outgoing TCP packet is not found in the lookup table and the outgoing TCP packet indicates that this is
a connection request, storing the source information together With the destination information from the outgoing TCP packet as a neW entry in the lookup table.
7. A method as claimed in claim 6, comprising receiving at the ?lter node an incoming TCP packet from the public netWork; and if the source port in the received incoming TCP packet is different from the destination port in a source information entry of the lookup table, indeXed by the des tination port in the outgoing TCP packet, and if the incoming TCP packet indicates that this packet is a ?rst response to the connection request, then updating by the ?lter node the destination port in the table entry With the source port from the received incoming TCP packet.
8. A method as claimed in claim 7, comprising receiving at the ?lter node any incoming TCP packet having an end of transmission code in the packet and Zeroing an entry in the lookup table corresponding to that received incomingTCP packet.
9. A method as claimed in claim 4, Wherein the data packets include packets in accordance With a user datagram protocol (UDP) over an internet protocol (IP).

6,128,298
11

10. A method of interfacing private and public data communications networks, through a ?lter node in commu nication with both networks, the ?lter node having an address known in the public network, comprising the steps
of:

routing from nodes in the private network, to the ?lter node, outgoing data packets having destination information, which includes a destination address and
a destination port, corresponding to nodes in the public network and having source information, which includes
a source address and a source port, of the respective

private network nodes;
for each outgoing data packet received from the private network, at the ?lter node, maintaining the source information taken from the outgoing data packet in correlation with a unique value representing a port of the ?lter node, and replacing in the outgoing data packet the source address with the ?lter node address and the source port with the ?lter node port value; routing from the ?lter node, to nodes in the public network, the outgoing data packets having the replaced source information, according to the destination infor mation in each, to the corresponding public network nodes;
routing from nodes in the public network, to the ?lter node, incoming data packets each having the address of the ?lter node as the destination address;
for each incoming data packet received from the public network, at the ?lter node, correlating the destination port of the destination information in the incoming data packet to particular source information being main tained and replacing, in the incoming data packet, the destination information with the particular source infor mation;
routing from the ?lter node, in the private network, the incoming data packets having the replaced destination information to the corresponding private network nodes:

ignoring by the ?lter node any incoming data packet received from the public network, if the destination port of the destination information in that incoming data packet can not be correlated to the maintained source information,
wherein maintaining the source information includes stor ing the source information from each outgoing data packet as an entry in a lookup table, and the ?lter node port value correlating to the source information con stitutes an indeX into the table for that entry;
wherein the data packets include packets in accordance with a user datagram protocol (UDP) over an internet protocol (IP); and
receiving at the ?lter node a UDP data packet from the private network, and adding the source information and the destination information from the UDP packet together with an interval indication for an expiration timer as a new entry in the lookup table.
11. A method of interfacing private and public data communications networks, through a ?lter node in commu nication with both networks, comprising the steps of:
(a) receiving at the ?lter node, from the private network,
a data packet having a destination address correspond
ing to a node in the public network and a source address corresponding to a node in the private network;
(b) maintaining, by the ?lter node, the source address taken from the data packet;

10

15

25

35

45

55

65

12

(c) replacing, in the data packet, the source address with
an address of the ?lter node, wherein the source address includes a port number of the node in the private network and the address of the ?lter node includes a

port number of the ?lter node;
(d) routing from the ?lter node, in the public network, the data packet having the replaced source address, accord ing to the destination address, to the corresponding public node network;
(e) waiting for a return packet from the public network, responsive to the data packet having the replaced source information;
(f) replacing, in the return packet, the destination address with the maintained source address; and
(g) routing from the ?lter node, in the private network, the return packet having the replaced destination address to the corresponding private network node.
12. A method as claimed in claim 11, comprising buffering, at the ?lter node, further data packets received from the private network while waiting for the return packet, and repeating steps (b) through (g) on an individual basis for the further packets, if any, that were buffered.
13. A method as claimed in claim 12, wherein the data packets include packets in accordance with an internet control message protocol (ICMP).
14. A method of operating a ?lter node for interfacing ?rst and second data communications networks, comprising the steps of:
receiving from the ?rst network, an outgoing data packet having destination information, which includes a des tination address and a destination port, corresponding
to a node in the second network and having source information, which includes a source address and a source port, corresponding to a node in the ?rst net work;
maintaining the source information taken from the out going data packet in correlation with a unique value representing a port of the ?lter node;
replacing in the outgoing data packet the source address with an address of the ?lter node and the source port with the ?lter node port value; and
sending to the second network the outgoing data packet having the replaced source information, whereby the packet is routed according to its destination information
to the corresponding second network node.
15. A method as claimed in claim 14, further comprising the steps of:
receiving from the second network, an incoming data packet having the address of the ?lter node as the destination address;
correlating the destination port of the destination infor mation in the incoming data packet to particular source information being maintained;
replacing, in the incoming data packet, the destination information with the particular source information; sending to the ?rst network the incoming data packet having the replaced destination information, whereby that packet is routed according to its destination infor mation to the corresponding ?rst network node.
16. A method as claimed in claim 15, comprising ignoring the incoming data packet received from the second network, if the destination port of the destination information in that data packet can not be correlated to the maintained source information.
17. Amethod as claimed in claim 16, wherein maintaining the source information includes storing the source informa

6,128,298
13

tion from the outgoing data packet as an entry in a lookup
table, and the ?lter node port value correlating to the source information constitutes an index into the table for that entry.
18. A method as claimed in claim 17, Wherein the incom ing and outgoing data packets include packets in accordance With a transmission control protocol (TCP) over an internet

protocol (IP).
19. A method of operating a ?lter node for interfacing ?rst and second data communications netWorks comprising the steps of:
receiving from the ?rst netWork, an outgoing data packet having destination information, Which includes a des tination address and a destination port, corresponding
to a node in the second netWork and having source information, Which includes a source address and a source port, corresponding to a node in the ?rst net Work:

maintaining the source information taken from the out going data packet in correlation With a unique value representing a port of the ?lter node;
replacing in the outgoing data packet the source address With an address of the ?lter node and the source port With the ?lter node port value;
sending to the second netWork the outgoing data packet having the replaced source information, Whereby that packet is routed according to its destination information to the corresponding second netWork node,
receiving from the second network, an incoming data packet having the address of the ?lter node as the destination address:

correlating the destination port of the destination infor mation in the incoming data packet to particular source information being maintained;
replacing, in the incoming data packet, the destination information With the particular source information; sending to the ?rst netWork the incoming data packet having the replaced destination information Whereby that packet is routed according to its destination infor mation to the corresponding ?rst netWork node; ignoring the incoming data packet received from the second netWork, if the destination port of the destina tion information in that data packet can not be corre lated to the maintained source information,
Wherein maintaining the source information includes stor ing the source information from the outgoing data packet as an entry in a lookup table, and the ?lter node port value correlating to the source information con stitutes an index into the table for that entry
Wherein the incoming and outgoing data packets include packets in accordance With a transmission control pro tocol (TCP) over an internet protocol (IP): and receiving an outgoing TCP packet from the ?rst netWork; and if an entry corresponding to the outgoing TCP packet is not found in the lookup table and the outgoing TCP packet indicates that this is a connection request, storing the source information together With the desti nation information from the TCP packet as a neW entry
in the lookup table.
20. A method as claimed in claim 19, comprising receiv ing any incoming TCP packet from the second netWork; and if the source port in that received incoming TCP packet is different from the destination port in a source information entry of the lookup table, indexed by the destination port in the outgoing TCP packet, and if that incoming TCP packet indicates that it is a ?rst response to the connection request,

10

15

25

35

45

55

a O

14

then updating the destination port in the table entry With the source port from that received incoming TCP packet.
21. A method as claimed in claim 20, comprising receiv
ing any incoming TCP packet having an end of transmission code in the packet, and Zeroing an entry in the lookup table corresponding to that received incoming TCP packet.
22. A method as claimed in claim 17, Wherein the out going and incoming data packets include packets in accor dance With a user datagram protocol (UDP) over an internet

protocol (IP).
23. A method of operating a ?lter node for interfacing ?rst and second data communications netWorks, comprising the steps of:
receiving from the ?rst netWork, an outgoing data packet having destination information, Which includes a des tination address and a destination port, corresponding
to a node in the second netWork and having source information, Which includes a source address and a source port, corresponding to a node in the ?rst net Work:

maintaining the source information taken from the out going data packet in correlation With a unique value representing a port of the ?lter node;
replacing in the outgoing data packet the source address With an address of the ?lter node and the source port With the ?lter node port value;
sending to the second netWork the outgoing data packet having the replaced source information, Whereby that packet is routed according to its destination information to the corresponding second netWork node;
receiving from the second netWork, an incoming data packet having the address of the ?lter node as the destination address;
correlating the destination port of the destination infor mation in the incoming data packet to particular source information being maintained;
replacing, in the incoming data packet, the destination information With the particular source information; sending to the ?rst netWork the incoming data packet having the replaced destination information, Whereby that packet is routed according to its destination infor mation to the corresponding ?rst netWork node; and ignoring the incoming data packet received from the second netWork, if the destination port of the destina tion information in that data packet can not be corre lated to the maintained source information,
Wherein maintaining the source information includes stor ing the source information from the outgoing data packet as an entry in a lookup table, and the ?lter node port value correlating to the source information con stitutes an index into the table for that entry;
Wherein the outgoing and incoming data packets include packets in accordance With a user datagram protocol (UDP) over an internet protocol (IP); and
receiving a UDP data packet from the ?rst netWork, and adding the source information and the destination infor mation from the UDP packet together With an interval indication for an expiration timer as a neW entry in the lookup table.
24. A method of operating a ?lter node for interfacing ?rst and second data communications netWorks, comprising the steps of:
(a) receiving from the ?rst netWork, a data packet having a destination address corresponding to a node in the second netWork and a source address corresponding to
a node in the ?rst netWork;

6,128,298
15

(b) maintaining the source address taken from the data packet;
(c) replacing, in the data packet, the source address With
an address of the ?lter node, Wherein the source address includes a source port number and the address of the ?lter node includes a port number of the ?lter node; (d) sending to the second netWork the data packet having the replaced source address, Whereby that packet is routed to the corresponding second netWork node;
(e) receiving a return packet from the second network, responsive to the data packet having the replaced source information;
(f) replacing, in the return packet, the destination address With the maintained source address; and
(g) sending to the ?rst netWork the return packet having the replaced destination address, Whereby that packet is routed to the corresponding ?rst netWork node.
25. A method as claimed in claim 24, comprising buffer ing further data packets received from the ?rst netWork While Waiting for the return packet, and repeating steps (b) through (g) on an individual basis for the further packets, if any, that Were buffered.
26. A method as claimed in claim 25, Wherein the data packets include packets in accordance With an internet control message protocol (ICMP).
27. A ?lter node for interfacing ?rst and second data communications netWorks, comprising:
means for receiving from the ?rst netWork, a data packet having destination information, Which includes a des tination address and a destination port, corresponding
to a node in the second netWork and having source information, Which includes a source address and a source port, corresponding to a node in the ?rst net Work;
means for maintaining the source information taken from the data packet in correlation With a unique value representing a port of the ?lter node;
means for replacing in the data packet the source address With an address of the ?lter node and the source port With the ?lter node port value; and
means for sending to the second netWork, the data packet having the replaced source information, Whereby that packet is routed according to its destination information to the corresponding second netWork node.
28. A ?lter node as claimed in claim 27, comprising: means for receiving from the second netWork, a data packet having the address of the ?lter node as the destination address;
means for correlating the destination port of the destina tion information in the data packet to particular source information being maintained;

15

25

35

45

16

means for replacing, in the data packet, the destination information With the particular source information; and means for sending to the ?rst netWork the data packet having the replaced destination information, Whereby
that packet is routed according to its destination infor mation to the corresponding ?rst netWork node.
29. A?lter node as claimed in claim 28, comprising means for ignoring a data packet received from the second netWork, if the destination port of the destination information in that data packet can not be correlated to the maintained source information.
30. A ?lter node as claimed in claim 29, Wherein the means for maintaining the source information includes means for storing the source information from the data packet as an entry in a lookup table, and Wherein the ?lter node port value correlating to the source information con stitutes an indeX into the table for that entry.
31. A ?lter node for interfacing ?rst and second data communications netWorks, comprising:
(a) means for receiving from the ?rst netWork, a data packet having a destination address corresponding to a node in the second netWork;
(b) means for maintaining the source address taken from the data packet;
(c) means for replacing, in the data packet, the source address With an address of the ?lter node, Wherein the source address includes a source port number and the address of the ?lter node includes a port number of the ?lter node;
(d) means for sending to the second netWork the data packet having the replaced source address, Whereby that packet is routed to the corresponding second netWork node;
(e) means for receiving a return packet from the second netWork, responsive to the data packet having the replaced source information;
(f) means for replacing, in the return packet, the destina tion address With the maintained source address; and
(g) means for sending to the ?rst netWork the return packet having the replaced destination address, Whereby that packet is routed to the corresponding the ?rst netWork node.
32. A?lter node as claimed in claim 31, comprising means for buffering further data packets received from the ?rst netWork While Waiting for the return packet, and means for controlling means (b) through (g) on an individual basis for processing the further packets, if any, that Were buffered.