(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
1 December 2005 (01.12.2005)

PCT

(10) International Publication Number
WO 2005/114356 A1

(51) International Patent Classification[7]: G06F 1/00, H04L 29/06, 12/26

(21) International Application Number: PCT/US2005/017442

(22) International Filing Date: 18 May 2005 (18.05.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/572,656 19 May 2004 (19.05.2004) US

(71) Applicant *(for all designated States except US)*: **COMPUTER ASSOCIATES THINK, INC.** [US/US]; One Computer Associates Plaza, Islandia, NY 11749 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **GASSOWAY, Paul** [US/US]; 219 Normandy Drive, Norwood, MA 02062 (US).

(74) Agent: **BHAVSAR, Samir, A.**; Baker Botts LLP, 2001 Ross Avenue, Dallas, TX 75201 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, 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 SYSTEMS FOR COMPUTER SECURITY

(57) Abstract: A method for maintaining computer security includes detecting a connection failure, storing information relating to the connection failure, determining a number of connection failures and determining whether a machine is infected with malicious code based on the determined number of connection failures.

WO 2005/114356 A1

METHOD AND SYSTEMS FOR COMPUTER SECURITY

Reference to Related Application

This application is based on and claims the benefit of Provisional Application Serial No. 60/572,656 filed May 19, 2004, the entire contents of which are herein incorporated by reference.

BACKGROUND

1. TECHNICAL FIELD

The present disclosure relates generally to security and, more particularly, to a method and system for computer security.

2. DESCRIPTION OF THE RELATED ART

With the growth of the Internet, the increased use of computers and the exchange of information between individual users has posed a threat to the security of computers. Computer security attempts to ensure the reliable operation of networking and computing resources and attempts to protect information on the computer or network from unauthorized access or disclosure. Computer system(s) as referred to herein include(s) individual computers, servers, computing resources, networks, etc. Among the various security threats that present increasingly difficult challenges to the secure operation of computer systems are computer viruses, Trojan horses, worms, etc. Computer viruses are programs that can infect other programs by modifying them in such a way as to include a copy of themselves. A Trojan horse is an executable program that appears to be desirable but is merely disguised as

"friendly" and actually contains harmful code, allowing an attacker to come in through the "back door" and perform malicious actions on the computer system.

Worms are independent programs that are capable of reproducing themselves, spreading from machine to machine, usually across network connections. Unlike computer viruses, worms do not need to infect other programs and do not need any user action to spread. Worms exist as a separate code in memory and spread silently on their own from computer system to computer system (instead of just spreading within one system). Worms are thus self propagating pieces of code designed to copy themselves from one computer system to another, trying to infect as many machines as possible, clogging up communications between, for example, computer systems and the Internet.

Email viruses use email to propagate from computer system to computer system. Infection of an email virus can occur when an attachment is opened that contains the virus. The attachment may appear to be harmless to the user because it might be sent from a contact that the user knows. However, the contact's email client, infected by the virus, may have sent copies of itself to the email addresses listed in the contact's address book. Thus, when a user opens the attachment that contains the virus, its own email client becomes infected and also starts sending the virus to the contacts in that user's address book. This process continues, causing the virus to spread throughout the Internet at a high rate, slowing down local area networks and the Internet.

System administrators responsible for the efficient operation of computer networks may utilize scanning programs in order to protect their computer systems from security threats such as worms, viruses, etc. Scanning programs operate to protect from the spread of these threats by detecting the threat and isolating and/or removing the malicious code.

Scanning programs may contain a list of previously defined threat signatures or definitions, containing the binary patterns or signatures of the threat. The scanning programs scan the various files of a system looking for a match to a particular threat's signature. If a threat is detected, the user may be notified and further steps may be taken to remove or isolate the malicious code. However, conventional scanning products may not provide comprehensive protection against these threats. Scanning software may detect the threats present in the system, but it does nothing to prevent them from infiltrating the system in the first place. The scanning software should be continuously updated in order to be effective in detecting new and modified threats. This not only proves to be a very tedious and time consuming task for computer users, but also may not happen often enough to provide adequate safeguards against new threats. Even if a user is diligent about updating the signature database (e.g., weekly), this still leaves a large window of opportunity for new worms to propagate and infect a large number of machines. Malicious code can therefore spread very quickly and infect many computer systems before being detected.

Accordingly, it would be beneficial to have methods and systems for providing a quick and effective way of ensuring that exploitation of system vulnerabilities will not come to light.

SUMMARY

A method for maintaining computer security comprises detecting a connection failure, storing information relating to the connection failure, determining a number of connection failures and determining whether a machine is infected with malicious code based on the determined number of connection failures.

A system for maintaining computer security comprises a system for detecting a connection failure, a system for storing information relating to the connection failure, a system for determining a number of connection failures and a system for determining whether a machine is infected with malicious code based on the determined number of connection failures.

A computer recording medium including computer executable code for maintaining computer security comprises code for detecting a connection failure, code for storing information relating to the connection failure in a database, code for determining a number of connection failures and code for determining whether a machine is infected with malicious code based on the determined number of connection failures.

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete appreciation of the present disclosure and many of the attendant advantages thereof will be readily obtained as the same becomes better understood by reference to the following detailed description when considered in connection with the accompanying drawings, wherein:

Figure 1 shows a block diagram of an exemplary computer system capable of implementing the method and system of the present disclosure;

Figure 2 is a block diagram for describing communication between hosts;

Figure 3 is a block diagram illustrating a system for maintaining computer security, according to an embodiment of the present disclosure;

Figure 4 shows a block diagram illustrating a system for maintaining computer security, according to an embodiment of the present disclosure;

Figure 5 shows a block diagram illustrating the storage of destination and source addresses in a database, according to an embodiment of the present disclosure; and

Figure 6 is a flow chart for describing a system for maintaining computer security according to an embodiment of the present disclosure.

DETAILED DESCRIPTION

The present disclosure provides tools (in the form of methodologies, apparatuses, and systems) for maintaining computer security. The tools may be embodied in one or more computer programs stored on a computer readable medium or program storage device and/or transmitted via a computer network or other transmission medium.

The following exemplary embodiments are set forth to aid in an understanding of the subject matter of this disclosure, but are not intended, and should not be construed, to limit in any way the claims which follow thereafter. Therefore, while specific terminology is employed for the sake of clarity in describing some exemplary embodiments, the present disclosure is not intended to be limited to the specific terminology so selected, and it is to be understood that each specific element includes all technical equivalents which operate in a similar manner.

The specific embodiments described herein are illustrative, and many variations can be introduced on these embodiments without departing from the spirit of the disclosure or from the scope of the appended claims. Elements and/or features of different illustrative embodiments may be combined with each other and/or substituted for each other within the scope of this disclosure and appended claims.

Figure 1 shows an example of a computer system **100** which may implement the method and system of the present disclosure. The system and method of the present disclosure may be implemented in the form of a software application running on a computer system, for example, a mainframe, personal computer (PC), handheld computer, server, etc. The software application may be stored on a recording media locally accessible by the computer system, for example, floppy disk, compact disk, hard disk, etc., or may be remote from the computer system and accessible via a hard wired or wireless connection to a network, for example, a local area network, or the Internet.

The computer system **100** can include a central processing unit (CPU) **102**, program and data storage devices **104**, a printer interface **106**, a display unit **108**, a (LAN) local area network data transmission controller **110**, a LAN interface **112**, a network controller **114**, an internal bus **116**, and one or more input devices **118** (for example, a keyboard, mouse etc.). As shown, the system **100** may be connected to a database **120**, via a link **122**.

Figure 2 shows an example of a system in which host A 22 is sending data to a host B 24 on network 20. Upon receipt of a frame from host A 22, router 26 begins processing it, discarding frame encapsulation, making routing decisions and preparing to transmit it to the destination host (host B 24). To prepare an appropriate frame header, router 26 queries the network 20 by broadcasting an Address Resolution Protocol (ARP) address request for host B 24. If host B 24 is available, it will respond with an ARP reply which includes its IP address. Using this information, router 26 then prepares the frame header and delivers the packet to host B 24. If for some reason host B 24 is not available (e.g., it does not exist or is not listening), router 26 will not receive an ARP reply. In this case, router 26 responds to host A 22 by encapsulating a portion of the original datagram in an ICMP destination unreachable

message (e.g., an ICMP Type 3 message) indicating that host B 24 is not available and returns it to host A 22. The encapsulated information may include, for example, the source IP address and the intended destination IP address as well as other information.

One common characteristic among various types of worms is that they often attempt to replicate by sending copies of themselves to random IP addresses sometimes using a sequential or pseudo-random selection process. Since these IP addresses are random, messages will be sent to many IP addresses not associated with an active machine. For example, there may be no machine at the IP address or the machine at the IP address may not be listening. Accordingly, many ICMP destination unreachable messages will be returned to the host from which the messages originated (e.g., the machine infected with a worm.)

According to an embodiment of the present disclosure, the connection failures identified by the ICMP destination unreachable messages are stored and used to determine when a worm has entered a machine and is attempting to propagate. The system may passively monitor for ICMP destination unreachable messages and record the source and destination addresses of the ICMP destination unreachable messages. If a single destination address has received destination unreachable messages from a large number of devices, it is likely that the machine identified by the destination address is infected with a worm.

An example of a worm infected machine (host A 22) attempting to spread the worm is shown in Figure 3. Host A 32 is infected with a particular type of worm, which attempts to connect to random IP addresses (e.g., host B 34, host C 36 , host D 38, etc. as well as hosts that do not exist on network 30.) For ease of description, only one router is shown. Of course, in reality, many routers and hosts may be involved when the worm is attempting to propagate. Upon receipt of the frames from host A 32, router 40 processes them, discarding

frame encapsulation, making routing decisions and preparing to transmit them to the destination hosts. To prepare an appropriate frame header, router 40 queries the network 30 by broadcasting Address Resolution Protocol (ARP) address requests for hosts B, C, D, etc. as well as for the non-existent hosts. If a host is available, it will respond with an ARP reply which includes its IP address. In this example, hosts B-D, etc. will return ARP replies. However, the non-existent or non-listening machines will not return replies. Router 40 then prepares the frame header and delivers the packets to hosts B, C, D, etc. However, for those hosts not available, router 40 (source) will prepare an ICMP destination unreachable message informing host A 32 (destination) that the host at the source address is unreachable. According to an embodiment of the present disclosure, the number of ICMP destination unreachable messages can be monitored and used to determine whether a worm is present on host A 32. The monitoring may be performed by a machine (not shown) passively listening to traffic to/from host A, or may be performed by host A itself. For example, Host A, may store information indicating the number of ICMP destination unreachable messages it has received. When the number reaches a defined level, host A can take one or more actions including, but not limited to, notifying a user that a worm may be present, automatically isolating the computer and/or automatically shutting the computer down, etc.

Figure 4 is a block diagram illustrating a system for maintaining computer security, according to another embodiment of the present disclosure. In this example, hosts 304-312 on network 302 are attempting to send messages to one or more hosts reachable by routers 318-328. Passive listening device 314 monitors for destination unreachable messages being returned to machines on network 302. Figure 5 depicts examples of information stored in a database (not shown) by passive listening device 314, for explaining various aspects of the

present disclosure.

Figure 6 is a flow chart for describing various steps performed by passive listening device 314. In Step S600, the system detects a destination unreachable message. The system then determines the destination and source addresses of the message (Step S602). The system then determines whether the destination address is present in the database. If the destination address is not in the database (No, Step S604), the destination and source addresses are then added to the database (Step S606). A counter is associated with each destination address in the database. Each time a source is added to a corresponding destination address, the counter is incremented. In this case, the counter is incremented (Step S612) and the process returns to Step S600. If the destination address of the message is in the database (Yes, Step S604), a determination is made whether the source address of the message is in database and associated with the corresponding destination address (Step S608). If the associated source address is not in the database, the source is added to the database and associated with the corresponding destination address (Step S610), and the counter C is incremented (Step S612) and the process returns to Step S600. If the corresponding source address is in the database (Yes, Step S608), the process returns to Step S600.

An example of the operation of the system shown in Figures 4-6 will now be described. Host 304 (address v.v.v.v.) has attempted to send a message by presenting a datagram to router 318 (address a.a.a.a.). The machine for which the message is destined is not there or is otherwise not available. Accordingly, router 318 has returned an ICMP destination unreachable message to host 304, the message having a source address of a.a.a.a. and a destination address of v.v.v.v. Passive listening device 314, listening on network 302,

detects the message and stores the destination address v.v.v.v. along with corresponding source address a.a.a.a. and increments counter 413 to C=1.

Host 306 (address w.w.w.w.) has attempted to send a message by presenting a datagram to router 320 (address b.b.b.b.). The machine for which the message is destined is not there or is otherwise not available. Accordingly, router 320 has returned an ICMP destination unreachable message to host 306, the message having a source address of b.b.b.b. and a destination address of w.w.w.w. Passive listening device 314, listening on network 302, detects the message and stores the destination address w.w.w.w. along with corresponding source address b.b.b.b. and increments counter 414 to C=1.

Host 306 (address w.w.w.w.) has also attempted to present another datagram to router 322 (address c.c.c.c.). The machine for which the message is destined is not there or is otherwise not available. Accordingly, router 322 has returned an ICMP destination unreachable message to host 306, the message having a source address of c.c.c.c. and a destination address of w.w.w.w. Passive listening device 314, listening on network 302, detects the message and, after determining that destination address w.w.w.w. is already in the database, stores the corresponding source address c.c.c.c. and increments counter 414 to C=2.

Host 306 (address w.w.w.w.) has also attempted to send a message by presenting another datagram router 324 (address d.d.d.d.). The machine for which the message is destined is not there or is otherwise not available. Accordingly, router 324 has returned an ICMP destination unreachable message to host 306, the message having a source address of d.d.d.d. and a destination address of w.w.w.w. Passive listening device 314, listening on network 302, detects the message and, after determining that destination address w.w.w.w. is

already in the database, stores the corresponding source address d.d.d.d. and increments counter 414 to C=3.

Host 308 (address x.x.x.x.) has attempted to send a message by presenting a datagram to router 318 (address a.a.a.a.). The machine for which the message is destined is not there or is otherwise not available. Accordingly, router 318 has returned an ICMP destination unreachable message to host 308, the message having a source address of a.a.a.a. and a destination address of x.x.x.x. Passive listening device 314, listening on network 302, detects the message and stores the destination address x.x.x.x. along with corresponding source address a.a.a.a. and increments counter 415 to C=1.

Host 308 (address x.x.x.x.) has also attempted to send a message by presenting another datagram to router 326 (address e.e.e.e.). The machine for which the message is destined is not there or is otherwise not available. Accordingly, router 326 has returned an ICMP destination unreachable message to host 308, the message having a source address of e.e.e.e. and a destination address of x.x.x.x. Passive listening device 314, listening on network 302, detects the message and, after determining that destination address x.x.x.x. is already in the database, stores the corresponding source address e.e.e.e. and increments counter 415 to C=2.

Host 310 (address y.y.y.y.) has attempted to send a message by presenting a datagram to router 328 (address f.f.f.f.). The machine for which the message is destined is not there or is otherwise not available. Accordingly, router 328 has returned an ICMP destination unreachable message to host 310, the message having a source address of f.f.f.f. and a destination address of y.y.y.y. Passive listening device 314, listening on network 302, detects the message and stores the destination address y.y.y.y. along with corresponding source

address f.f.f.f. and increments counter 416 to C=1.

Host 312 (address z.z.z.z.) has attempted to send a message by presenting a datagram to an unreachable network. In this case, router 316 (address g.g.g.g.) returns a network unreachable message to host 312, the message having a source address g.g.g.g. and a destination address z.z.z.z. Passive listening device 314, listening on network 302, detects the message and stores the destination address z.z.z.z. along with corresponding source address g.g.g.g. and increments counter 417 to C=1.

This process continues, populating the database with the source and destination address information of the destination unreachable messages received across network 302.

As noted above, if the destination address is already in the database, the system checks to see whether the corresponding source address is already associated with the destination address. If the destination address is not already in the database, the corresponding source address is added to the database for that destination address. On the other hand, if the corresponding source address is already associated with the destination address, another entry can be added for it and/or the corresponding counter can be incremented.

Periodically, passive listening device 314 can review the database. If there are a large number of source addresses associated with a single destination address, as indicated by the counter, then it is likely that the machine identified by the destination address is infected with a worm. This machine can then be disconnected until signatures that can isolate and remove the worm become available.

According to various embodiments of the present disclosure, corresponding information, such as timestamp information, destination port for a connection failure

information, etc. may also be provided for each source address stored in the database. For example, a time (actual time and/or date) indicating when the entry (destination address and/or source address) was made in the database can be included in the database. The system can be arranged so that it will periodically review the database and remove entries that have been there for a long period of time. If after removal a destination address still has too many corresponding source addresses, then the destination address can be reported as a possibly infected machine. The corresponding information can also help to identify possible infections and the services that the infections exploit. For example, based on the destination port of the source address, it can be determined which service an alleged worm seems to be targeting.

Numerous additional modifications and variations of the present disclosure are possible in view of the above-teachings. It is therefore to be understood that within the scope of the appended claims, the present disclosure may be practiced other than as specifically described herein.

What is claimed is:

1. A method for maintaining computer security, comprising:

detecting a connection failure;

storing information relating to the connection failure;

determining a number of connection failures; and

determining whether a machine is infected with malicious code based on the determined number of connection failures.

2. The method of claim 1, wherein a connection failure is identified by a ICMP destination unreachable message.

3. The method of claim 2, wherein the ICMP destination unreachable message comprises a destination address and a corresponding source address, the source address identifying a machine that sent the ICMP destination unreachable message.

4. The method of claim 3, wherein the corresponding source address identifies a machine that attempted a connection for which the connection failure occurred.

5. The method of claim 3, wherein storing the information relating to the connection failure comprises determining whether the destination address is already stored.

6. The method of claim 5, further comprising determining whether the corresponding source address is already stored if the destination address is already stored.

7. The method of claim 6, wherein the corresponding source address is stored only if it is determined that the corresponding source address is not already stored.

8. The method of claim 7, further comprising determining how long a corresponding source address has been stored and removing the corresponding source address if it is determined that the corresponding source address has been stored for a predetermined period of time.

9. The method of claim 8, wherein a counter is incremented for each corresponding source address that is stored and decremented for each source address that is removed.

10. A system for maintaining computer security, comprising:

a system for detecting a connection failure;

a system for storing information relating to the connection failure;

a system for determining a number of connection failures; and

a system for determining whether a machine is infected with malicious code based on the determined number of connection failures.

11. The system of claim 10, wherein a connection failure is identified by a ICMP destination unreachable message.

12. The system of claim 11, wherein the ICMP destination unreachable message comprises a destination address and a corresponding source address, the source address identifying a machine that sent the ICMP destination unreachable message.

13. The system of claim 12, wherein the corresponding source address identifies a machine that attempted a connection for which the connection failure occurred.

14. The system of claim 12, wherein storing the information relating to the connection failure comprises determining whether the destination address is already stored.

15. The system of claim 14, further comprising a system for determining whether the corresponding source address is already stored if the destination address is already stored.

16. The system of claim 15, wherein the corresponding source address is stored only if it is determined that the corresponding source address is not already stored.

17. The system of claim 16, further comprising a system for determining how long a corresponding source address has been stored and removing the corresponding source address if it is determined that the corresponding source address has been stored for a predetermined period of time.

18. The system of claim 17, wherein a counter is incremented for each corresponding source address that is stored and decremented for each source address that is removed.

19. A computer recording medium including computer executable code for maintaining computer security, comprising:

code for detecting a connection failure;

code for storing information relating to the connection failure in a database;

code for determining a number of connection failures; and

code for determining whether a machine is infected with malicious code based on the determined number of connection failures.

20. The computer recording medium of claim 19, wherein a connection failure is identified by a ICMP destination unreachable message.

21. The computer recording medium of claim 20, wherein the ICMP destination unreachable message comprises a destination address and a corresponding source address, the source address identifying a machine that sent the ICMP destination unreachable message.

22. The computer recording medium of claim 21, wherein the corresponding source address identifies a machine that attempted a connection for which the connection failure occurred.

23. The computer recording medium of claim 21, wherein the code for storing the information relating to the connection failure comprises code for determining whether the destination address is already stored.

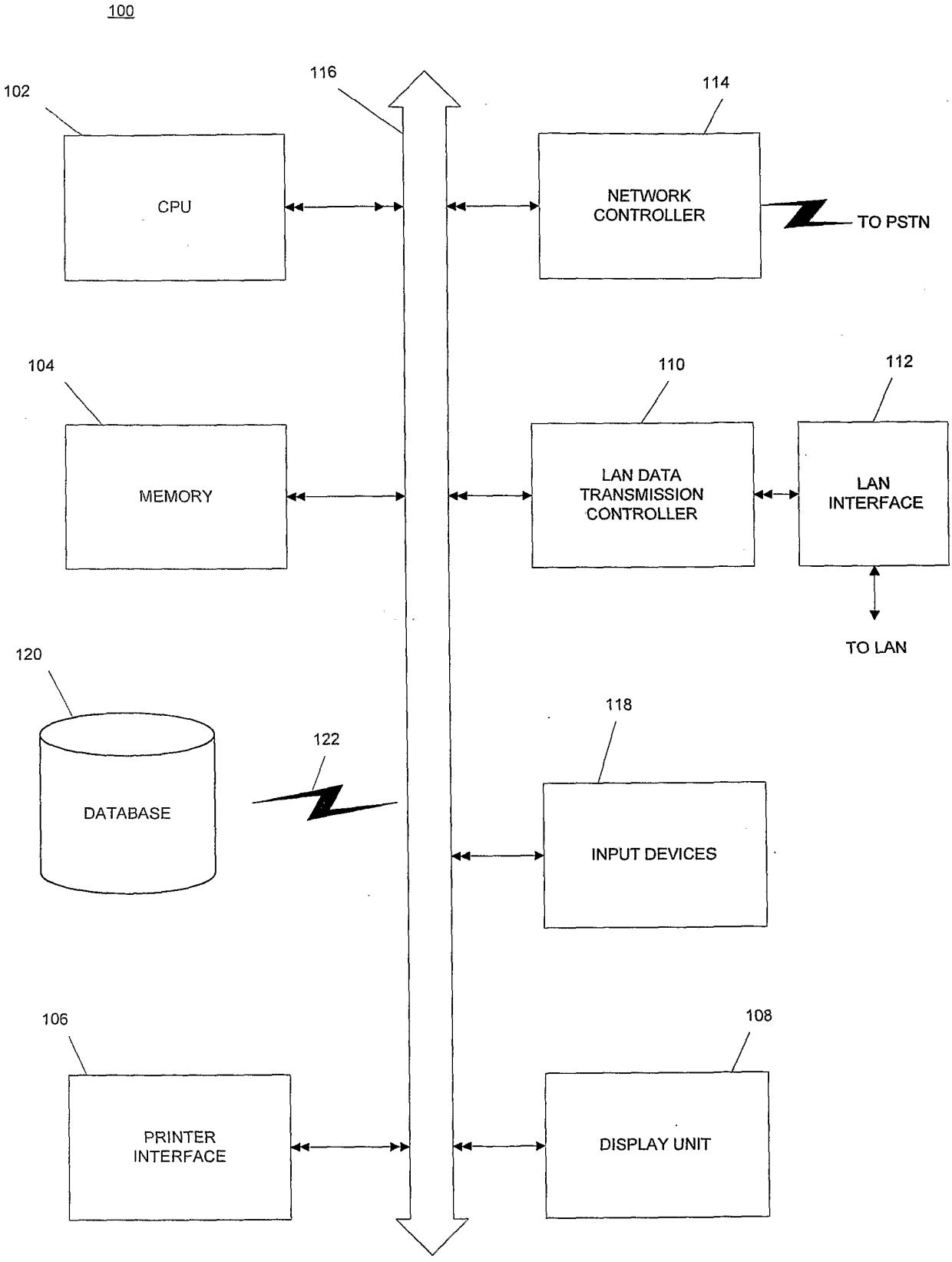
24. The computer recording medium of claim 23, further comprising code for determining whether the corresponding source address is already stored if the destination address is already stored.

25. The computer recording medium of claim 24, wherein the corresponding source address is stored only if it is determined that the corresponding source address is not already stored.

26. The computer recording medium of claim 25, further comprising code for determining how long a corresponding source address has been stored and removing the corresponding source address if it is determined that the corresponding source address has been stored for a predetermined period of time.

27. The computer recording medium of claim 26, wherein a counter is incremented for each corresponding source address that is stored and decremented for each source address that is removed.

FIGURE 1


100
102
CPU
116
114
NETWORK CONTROLLER
TO PSTN
104
MEMORY
110
LAN DATA TRANSMISSION CONTROLLER
112
LAN INTERFACE
TO LAN
120
DATABASE
122
118
INPUT DEVICES
106
PRINTER INTERFACE
108
DISPLAY UNIT

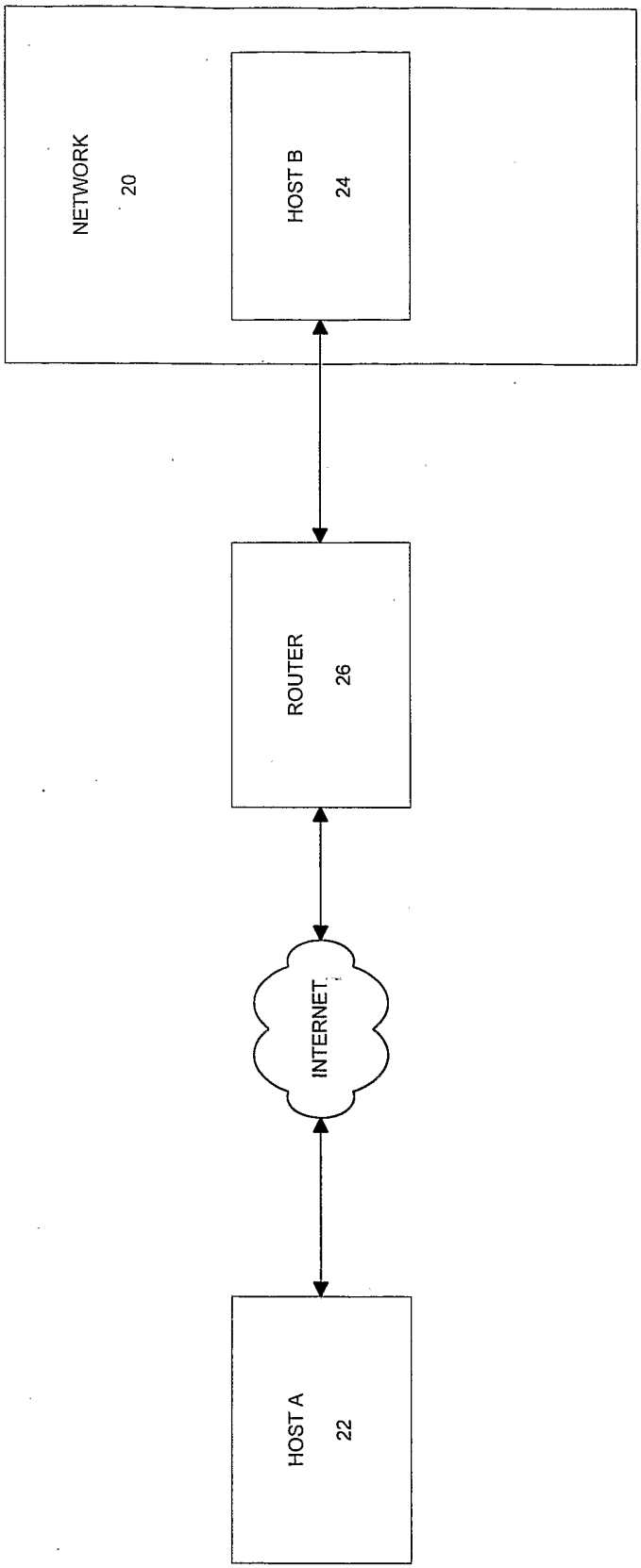


FIGURE 2

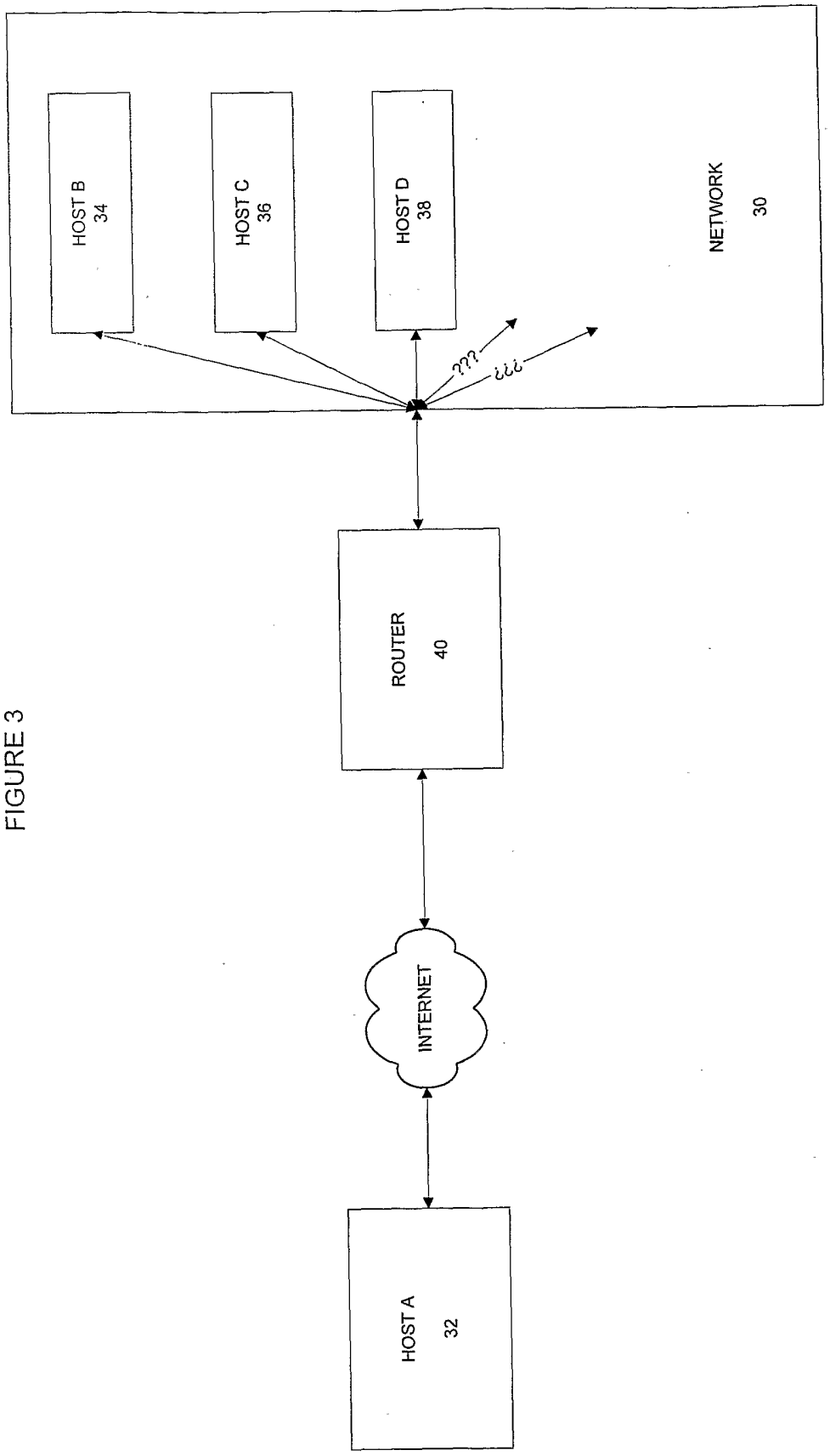

HOST B
34
HOST C
36
HOST D
38
NETWORK
30
???
¿¿¿
ROUTER
40
INTERNET
HOST A
32

FIGURE 3

FIGURE 4

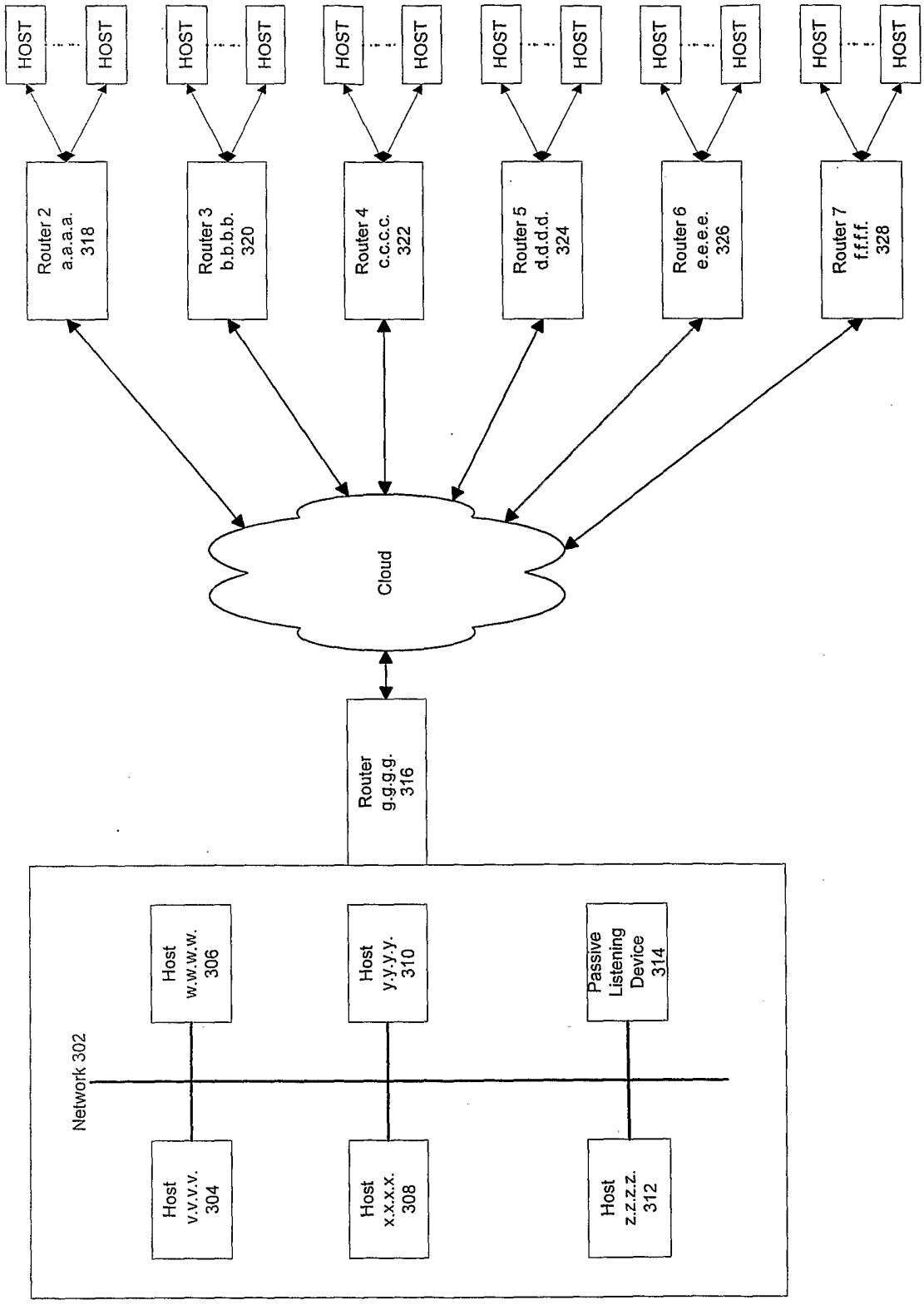

HOST
HOST
Router 2
a.a.a.a.
318
Router 3
b.b.b.b.
320
Router 4
c.c.c.c.
322
Router 5
d.d.d.d.
324
Router 6
e.e.e.e.
326
Router 7
f.f.f.f.
328
Cloud
Router
g.g.g.g.
316
Network 302
Host
w.w.w.w.
306
Host
y.y.y.y.
310
Passive
Listening
Device
314
Host
v.v.v.v.
304
Host
x.x.x.x.
308
Host
z.z.z.z.
312

FIGURE 5

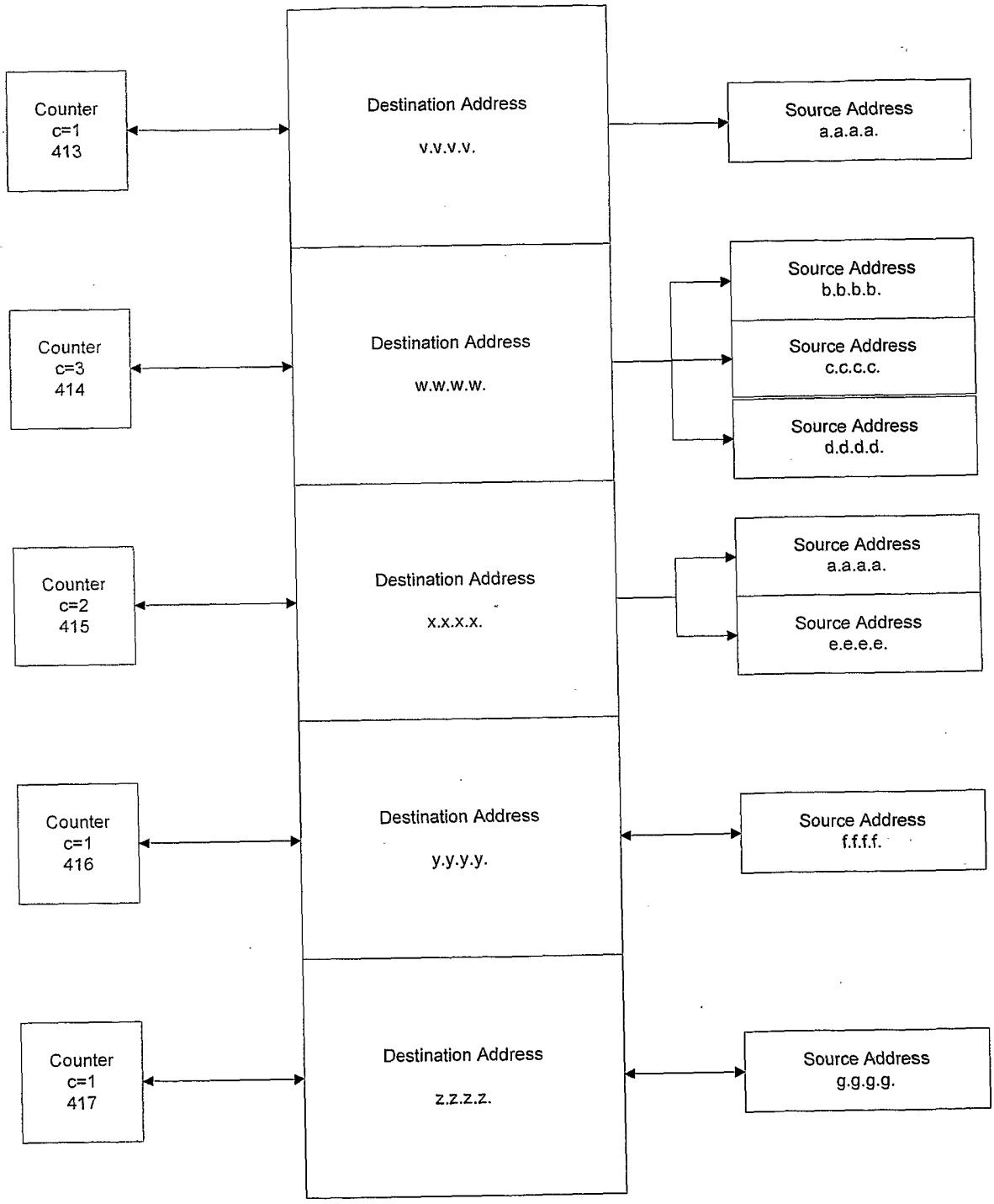
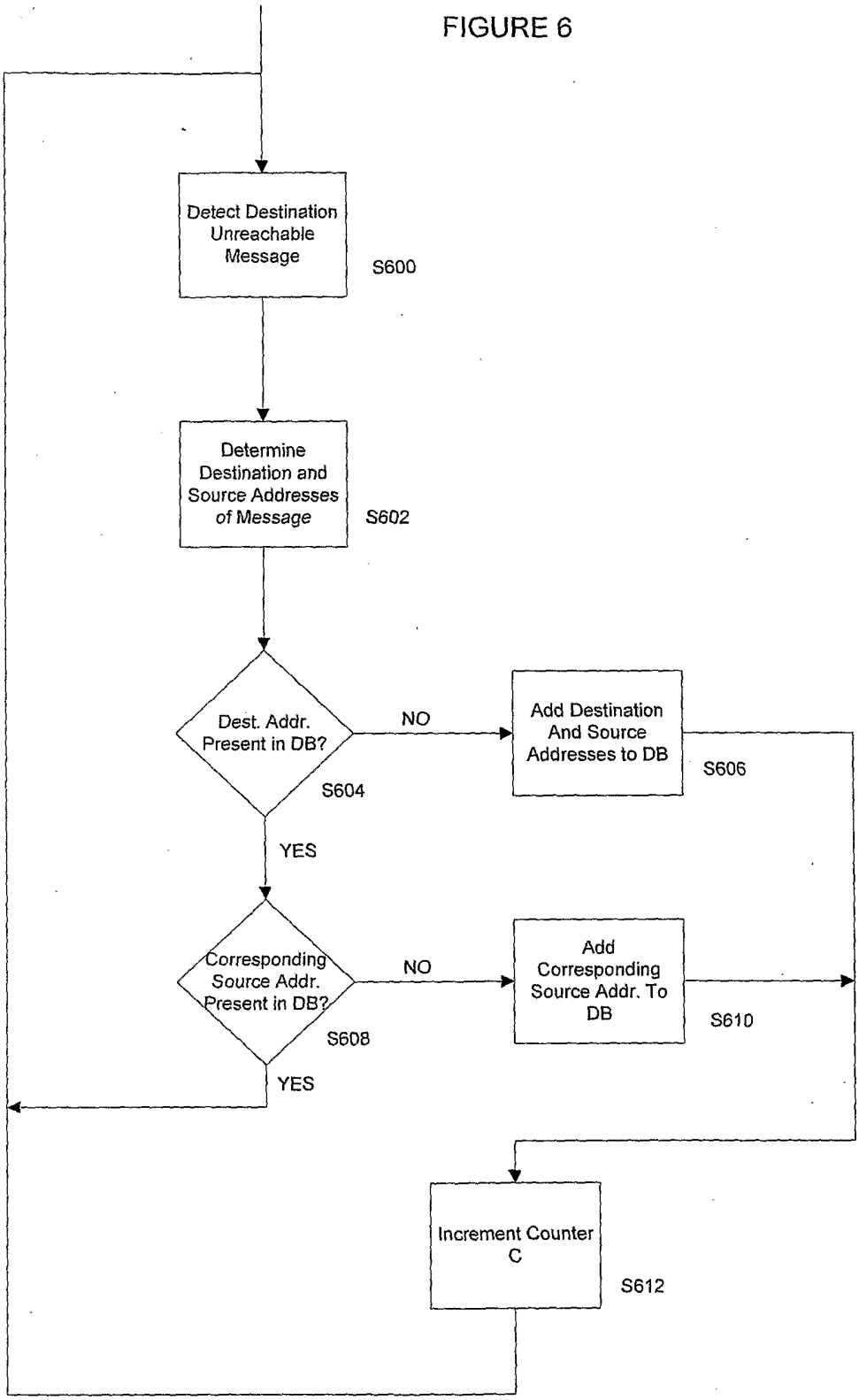

Counter
c=1
413
Destination Address
v.v.v.v.
Source Address
a.a.a.a.
Counter
c=3
414
Destination Address
w.w.w.w.
Source Address
b.b.b.b.
Source Address
c.c.c.c.
Source Address
d.d.d.d.
Counter
c=2
415
Destination Address
x.x.x.x.
Source Address
a.a.a.a.
Source Address
e.e.e.e.
Counter
c=1
416
Destination Address
y.y.y.y.
Source Address
f.f.f.f.
Counter
c=1
417
Destination Address
z.z.z.z.
Source Address
g.g.g.g.

FIGURE 6


Detect Destination Unreachable Message
S600
Determine Destination and Source Addresses of Message
S602
Dest. Addr. Present in DB?
S604
NO
Add Destination And Source Addresses to DB
S606
YES
Corresponding Source Addr. Present in DB?
S608
NO
Add Corresponding Source Addr. To DB
S610
YES
Increment Counter C
S612

INTERNATIONAL SEARCH REPORT

Inte nal Application No
PCT/US2005/017442

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F1/00 H04L29/06 H04L12/26

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 G06F 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, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	BAKOS G ET AL: "Early detection of Internet worm activity by metering ICMP destination unreachable messages" PROCEEDINGS OF THE SPIE - THE INTERNATIONAL SOCIETY FOR OPTICAL ENGINEERING SPIE-INT. SOC. OPT. ENG USA, vol. 4708, 2002, pages 33-42, XP002342234 ISSN: 0277-786X the whole document paragraphs '03.3!, '04.1! figure 4 ----- -/--	1-27

[X] 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	Date of mailing of the international search report
25 August 2005	05/09/2005

Name and mailing address of the ISA	Authorized officer
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	Mäenpää, J

4

Form PCT/ISA/210 (second sheet) (January 2004)

INTERNATIONAL SEARCH REPORT

Inter ıl Application No
PCT/US2005/017442

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	GRAY R S ET AL: "Rapid detection of worms using ICMP-T3 analysis" PROCEEDINGS OF THE SPIE - THE INTERNATIONAL SOCIETY FOR OPTICAL ENGINEERING SPIE-INT. SOC. OPT. ENG USA, vol. 5403, no. 1, 12 April 2004 (2004-04-12), pages 89-101, XP002342235 ISSN: 0277-786X the whole document	1-27
A	LUCENT A CONTA ET AL: "Internet Control Message Protocol (ICMPv6) Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification Specification" NETWORK WORKING GROUP REQUEST FOR COMMENTS, no. 2463, 31 December 1998 (1998-12-31), pages 1-18, XP002285507 the whole document	1-27

4