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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
17 April 2008 (17.04.2008)

PCT

(10) International Publication Number
WO 2008/045581 A1

(51) International Patent Classification:
G06F 15/16 (2006.01)

(21) International Application Number:
PCT/US2007/060770

(22) International Filing Date: 19 January 2007 (19.01.2007)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/549,452 13 October 2006 (13.10.2006) US

(71) Applicant *(for all designated States except US)*: **CISCO TECHNOLOGY, INC.** [US/US]; 170 West Tasman Drive, San Jose, CA 95134-1706 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **WING, Daniel, G.** [US/US]; 222 Coffeeberry Drive, San Jose, CA 95123 (US). **JENNINGS, Cullen** [US/US]; 620 Highland Drive, Santa Cruz, CA 95060 (US). **ROSENBERG, Jonathan, D.** [US/US]; 197 Pin Oak Road, Freehold, NJ 07728 (US).

(74) Agents: **FORD, Stephen, S.** et al.; MARGER JOHNSON & MCCOLLOM, P.C., 210 SW MORRISON STREET, Suite 400, Portland, OR 97204 (US).

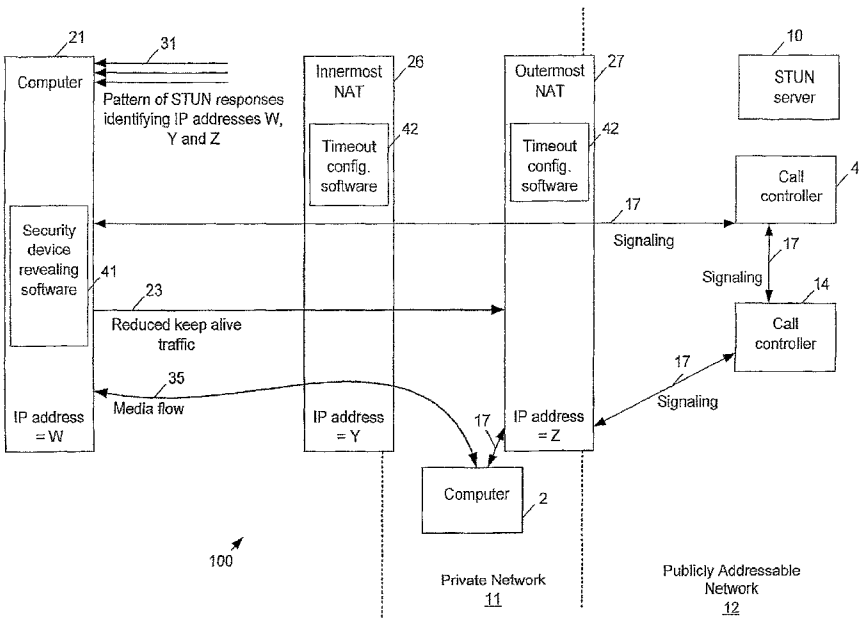
(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR,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

[Continued on next page]

(54) **Title:** DISCOVERING SECURITY DEVICES LOCATED ON A CALL PATH AND EXTENDING BINDINGS AT THOSE DISCOVERED SECURITY DEVICES



(57) **Abstract:** In one embodiment, an endpoint elicits a pattern of STUN responses to identify security devices located on a call path. The endpoint then uses address information from the identified security devices to establish an efficient media flow with a remote endpoint. The endpoint can optimize the number of network devices and network paths that process the endpoint's keepalive message. Additionally, the endpoint may request custom inactivity timeouts with each of the identified security devices for reducing bandwidth consumed by keepalive traffic.

WO 2008/045581 A1

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

DISCOVERING SECURITY DEVICES LOCATED ON A CALL PATH AND EXTENDING BINDINGS AT THOSE DISCOVERED SECURITY DEVICES

TECHNICAL FIELD

The present disclosure relates generally to the field of networking.

BACKGROUND

Endpoints such as Internet Protocol (IP) phones can make multimedia communications such as Voice over IP (VoIP) calls over a packet switched network using multimedia session signaling protocols such as Session Initial Protocol (SIP). Security devices such as firewalls and Network Address Translators (NATs) located between two endpoints can prevent the flow of multimedia messages between the two endpoints. Simple Traversal of User Datagram Protocol (UDP) Through NATs (STUN) was developed to allow multimedia communications to operate through NATs. STUN is described in Request for Comment (RFC) 3489.

STUN is used by a calling device to determine a NAT public address and NAT port number associated with the calling device. The calling device provides the NAT public address and NAT port number to the target endpoint during a call establishment process. The use of STUN as described in RFC 3489 when more than one NAT or any amount of firewalls are interposed between a calling device and the Internet produces certain inefficiencies and problems. The disclosure that follows solves these and other problems.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 illustrates Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs) (STUN) operation with NATs and firewalls according to Request For Comment (RFC) 3489.

FIG. 2 illustrates an example system for finding NAT devices located on a call path and extending their bindings.

FIG. 3 illustrates an example of the computer illustrated in FIG. 2 for discovering a firewall.

FIG. 4 illustrates an example of the computer and the NATs illustrated in FIGS. 2.

FIG. 5 illustrates an example of the STUN server from FIG. 4 providing an internally reachable address.

FIG. 6 illustrates an example of the firewall illustrated in FIG. 3.

FIG. 7 illustrates an example method for using the computer illustrated in FIG. 2

FIG. 8 illustrates an example method for using the NATs illustrated in FIG. 2.

FIG. 9 illustrates an example method for using the firewall illustrated in FIG. 3.

DESCRIPTION OF EXAMPLE EMBODIMENTS

Overview

In one embodiment, an endpoint elicits a pattern of STUN responses to identify security devices located on a call path. The endpoint then uses address information from the identified security devices to establish an efficient media flow with a remote endpoint. The endpoint can optimize the number of network devices and network paths that process the endpoint's keepalive message. Additionally, the endpoint may request custom inactivity

timeouts with each of the identified security devices for reducing bandwidth consumed by keepalive traffic.

Description

Several preferred examples of the present application will now be described with reference to the accompanying drawings. Various other examples of the invention are also possible and practical. This application may be exemplified in many different forms and should not be construed as being limited to the examples set forth herein.

The figures listed above illustrate preferred examples of the application and the operation of such examples. In the figures, the size of the boxes is not intended to represent the size of the various physical components. Where the same element appears in multiple figures, the same reference numeral is used to denote the element in all of the figures where it appears. When two elements operate differently, different reference numerals are used regardless of whether the two elements are the same class of network device.

Only those parts of the various units are shown and described which are necessary to convey an understanding of the examples to those skilled in the art. Those parts and elements not shown are conventional and known in the art.

FIG. 1 illustrates STUN operation with NATs and firewalls according to RFC 3489. Referring to FIG. 1, computer 1 has received a user request to establish multimedia communications with computer 2. As computer 1 is unaware of how many, if any, NATs are interposed between itself and the publicly addressable network 12, computer 1 needs to send, for this call and any other new calls, a STUN request 8A to public STUN server 10. As part of its basic address translation function, inner nested NAT 6 modifies the STUN request 8A by replacing the source address indicating IP address W with an IP address Y. Outer nested

NAT 7 functions in a similar manner so that a STUN request 8B identifies the source as IP address Z for NAT 7.

As part of STUN operation defined by RFC 3489, STUN server 10 inserts the current source address information, such as IP address Z and a port number on NAT 7, into the payload of a STUN response 9. The STUN response 9 is then sent back to the computer 1.

The computer 1 extracts the address information and observes that the address information, address Z, is different than a local IP address W. This difference indicates the presence of only the outermost NAT 7. The firewall 5 and the NAT 6 remain undiscovered, however.

Now that its publicly-reachable address has been learned, the computer 1 continues the process of establishing a media path with computer 2 by exchanging signaling messages 17. The signaling messages 17 are exchanged using a call controller 4 for computer 1 and a call controller 14 for computer 2. After the signaling messages 17 are received by computer 2, the IP address Z and the port number are used to establish media flow 15 through NAT 7 having IP address Z.

As demonstrated by the above example, the signaling messages 17 necessitate that computer 1 and call controller 4 are constantly able to contact each other to establish the media flow 15 as well as other media flows with other endpoints. To remain contactable, computer 1 sends frequent signaling "keepalive" traffic 13 all the way to call controller 4 to keep any on path security devices, discovered or undiscovered, from timing out bindings or closing pinholes. The traffic 13 is sent according to a predetermined default, such as once every thirty seconds, to accommodate any low timeout values that could be included on undiscovered security devices (such as firewall 5 and NAT 6).

The keepalive traffic 13, as well as the per-call STUN requests, consume bandwidth on the publicly addressable network 12. When network 12 is a wireless network, the bandwidth consumed by these types of communications is particularly problematic. In such a case, each of these keepalive messages 13 and the per-call STUN request consume airtime, which is expensive.

FIG. 2 illustrates an example system 100 for finding NAT devices located on a call path and extending their bindings.

Several differences are immediately apparent when comparing FIGS. 1 and 2. First, computer 21 includes security device revealing software 41 that elicits a pattern of STUN responses 31 for discovering on path security devices between computer 21 and STUN server 10, call controller 4, and remote endpoint 2, so that computer 21 can optimize the keepalive traffic with each of those connections. Second, the software 41 leverages information about the on path security devices to establish a media flow 35 that is more efficient than the media flow 15 shown in FIG. 1. Third, the NATs 26 and 27 include timeout configuration software 42 for adjusting local timeouts according to communications sent by the software 41 located on computer 21. Accordingly, extended timeout settings can be set, which allows for reduced keepalive traffic 23 as compared to the frequent keepalive traffic 13 in FIG. 1. Fourth, the keepalive traffic 23 is directed to the NAT 27 rather than call controller 4, which reduces bandwidth consumption on publicly addressable network 12.

FIG. 3 illustrates an example of the computer illustrated in FIG. 2 for discovering a firewall.

Other differences are immediately apparent when comparing FIGS. 1 and 3. First, the security device revealing software 41 elicits a STUN response 59 that identifies an IP address X for firewall 25. Second, the firewall 25 includes STUN recognition software 43 that both

facilitates identification of the firewall 25 to computer 21 and facilitates timeout negotiation with firewall 25 for reducing keepalive traffic.

FIG. 4 illustrates an example of the computer and the NATs illustrated in FIGS. 2.

Referring to FIG. 4, to identify security devices located on a call path, computer 1 sends an initial STUN request 28A. The STUN request 28A is addressed to the default STUN port 3489 on STUN server 10. After STUN requests 28B and 28C are forwarded, STUN server 10 responds with a STUN response 29 that identifies a public IP address Z and port number for outermost NAT 27.

After receiving the STUN response 29, computer 21 compares the IP address Z included in the STUN response 29 with a local IP address W. The computer 21 identifies NAT 27 according to the comparison.

In response to the computer 21 identifying the NAT 27, even though the computer 21 may not have determined whether the NAT 27 locally includes a STUN server, the software 41 sends a STUN request 38A to STUN port number 3489 on IP address Z. STUN port number 3489 is a designated port number that STUN servers, such as STUN servers 10, 30 and 32, use for STUN communications. In other words, the STUN request 38A is sent to an undiscovered STUN server using the same STUN port that was used to send STUN request 28A. In other embodiments, the STUN request 38A may be addressed to any other port number associated with STUN communications. NAT 26 receives the STUN request 38A and communicates STUN request 38B to NAT 27.

Next, the STUN request 38B is received at NAT 27, which is illustrated in greater detail in FIG. 5. The NAT 27 processes the received STUN request 38B using the address translation table 45 or any other binding data structure so that the STUN request 38B is received at the STUN server 32.

According to software 42, the STUN server 32 sends a request 40A for an internally reachable address for the source address included on the STUN request 38B when received by NAT 27. The NAT 27 accesses the address translation table 45 or other binding data structure to identify an internally reachable address Y, belonging to NAT 26. A response 40B that includes the internally reachable address Y is sent back to the STUN server 32.

Next, the STUN server 32 generates a STUN response 39 that identifies IP address Z because the address translation function of NAT 27 operated before the STUN server 32 processed the request. Additionally, according to software 42, the STUN server 32 includes the internally reachable address Y within the STUN response 39. The STUN response 39 is then sent back to computer 21.

Referring back to FIG. 4, computer 21 compares the IP address Y identified in the STUN response 39 to the local IP address W and accordingly identifies NAT 26. In response to the computer 21 not receiving the local address W, the software 41 continues the pattern by sending another STUN request to the newly identified address Y. This pattern of sending a new STUN request to any newly identified addresses continues until the software 41 receives a STUN response including the local IP address W. In other words, the computer 21 "walks backwards" through the NATs until every NAT located between itself and the publicly addressable network 12 has been identified. Therefore, to complete the example, computer 21 sends a STUN request 48 to STUN port number 3489 on STUN server 30 to elicit STUN response 49 including binding information identifying IP address W and IP address Y. The receipt of the STUN response 49 identifying IP address W signals computer 21 that all NATs have been discovered.

The software 41 can leverage the binding information for each NAT to establish an efficient media flow 35 with an endpoint, such as computer 2, located behind one or more of

the same NATs as computer 1. In other words, computer 21 provides the discovered IP address Y as a reachable address to computer 2. Computer 2 directs multimedia communications at IP address Y. This multimedia packet flow avoids NAT 27, which is an improvement over the flow shown in FIG. 1.

Software 41 may also leverage the binding information for each NAT to establish longer bindings, which can reduce the frequency required for keepalive traffic. For example, STUN request 38A may include an attribute requesting a custom timeout value for NAT 27. Likewise, STUN request 48 may include an attribute requesting the same or a different custom timeout value for NAT 26. Accordingly, a longer lifetime can be requested for the session, e.g. five minutes, so that NATs 26 and 27 continue to accept incoming messages from computer 2 despite prolonged inactivity. Such a change can, for example, reduce keepalive messages from being sent once every thirty seconds to only once every five minutes. In other examples, timeout negotiation uses messages separate from the STUN requests 38A and 48.

Not only are bandwidth savings realized by sending less frequent keepalive traffic, but bandwidth savings are realized on the network 12 because keepalive traffic 23 does not need to be sent all the way to call controller 4. For example, since all on path security devices have been discovered using the pattern of STUN responses, computer 21 no longer needs to send keepalive traffic 23 all the way to call controller 4. Accordingly, the keepalive traffic 23 may instead be addressed to STUN port 3489 on the public side of the outermost security device, e.g. NAT 27, which ensures that all other security devices do not timeout the session due to inactivity.

Endpoints such as computer 21 are preconfigured to access a public STUN server at a default address to determine an associated reachable address. Whenever the endpoints need

to send a STUN request, for example to determine updated binding information, a STUN request is sent to this public STUN server according to these local configuration settings.

However, computer 21 discovered that NAT 27 includes a STUN server 32 due to the STUN response 39 elicited by STUN request 38A, which was address to an undiscovered STUN server. The discovery of the STUN server 32 can be leveraged to reduce traffic over the publicly accessible network 12 to the STUN server 10. For example, the discovered STUN server 32 located on NAT 27 can be used for future STUN requests instead of the STUN server 10. The software 41 reconfigures computer 21 to communicate with NAT 27 for future binding information inquires. Such reconfiguring reduces the amount of traffic over publicly addressable network 12 and reduces the amount of processing required by STUN server 10.

FIG. 6 illustrates an example of the firewall illustrated in FIG. 3.

Computer 21 sends STUN request 58A containing a STUN magic cookie or other identifier to public STUN server 10. The STUN magic cookie is a fixed 32 bit value that identifies STUN traffic and is described in more detail in the document draft-ietf-behave-rfc-3489bis-04.txt, which is located on the Internet Engineering Task Force (IETF) website. In other examples, the STUN request 58A includes an attribute explicitly requesting any on path firewalls to include a local IP address. In yet other examples, any other method may be used by the firewall 25 to distinguish STUN requests from other traffic.

Also, the STUN request 58A may include an attribute requesting a custom timeout value. When the custom timeout is requested, the firewall 25 will set a session accordingly, otherwise, a default timeout will be used.

The firewall 25 receives an incoming message and identifies it as a STUN message by observing the STUN magic cookie. In the present example, the software 43 then stores a

STUN transaction identifier included in the STUN request 58A in a local memory. Storing by the software 43 may be responsive to identifying the STUN magic cookie or by identifying the attribute explicitly requesting inclusion of a local IP address. In other examples, no STUN transaction identifiers are stored.

Next, the firewall 25 intercepts a STUN response 59A identifying IP address W. The firewall 25 compares a STUN transaction identifier included in the STUN response 59A to the value stored in memory. When there is a match, the firewall 25 attaches a local IP address X for the firewall 25 to the STUN response 59A. In the present example, the local IP address X is attached after a message integrity attribute included in the STUN response 59A to avoid interference with the STUN message integrity scheme used by computer 21. In other words, the local IP address X is added outside a digital signature for the STUN response 59A in contrast to the IP address W that is located within the digital signature. The firewall 25 then forwards STUN response 59B identifying the IP addresses W and X to computer 21.

Computer 21 receives the STUN response 59B and accesses the address information contained within. In the present example, address information included before the message integrity attribute corresponds to the associated public address to be provided to a called endpoint, while the address information (including IP address X) after the message integrity attribute corresponds to an on-path firewall. In other examples, the address information after the message integrity attribute may contain addresses for multiple on-path firewalls.

Next, the firewall 25 opens a pinhole to receive incoming messages from computer 2. During inactivity, the pinhole persists for the duration of any custom timeout value included in the STUN request 58A, otherwise, the pinhole persists for a default duration during inactivity.

FIG. 7 illustrates an example method for using the computer illustrated in FIG. 2

In block 701, the computer 21 sends a STUN request or other address request including a STUN magic cookie or other identifier to a default STUN port on a public STUN server to identify a reachable address. The computer 21 receives back a STUN response that identifies the reachable address in block 702.

In block 703, the computer 21 determines whether the STUN response includes any other addresses for other on-path security devices. When no additional addresses are included in block 704, in block 705A the computer observes that there are no STUN-aware non-address-translating devices such as firewalls protecting the computer 21. When additional addresses are included, in block 705B the computer 21 observes that each additional address identifies a STUN-aware non-address-translating device.

The computer 21 also compares the reachable address to a local address in block 706. When the addresses do not match in block 707, the computer 21 observes that the reachable address corresponds to an address translating security device located between the computer 21 and the Internet in block 708. In block 709, the computer 21 sends a STUN request to a default STUN port on the reachable address regardless of whether the address translating security device actually includes a STUN server. Optionally, the computer 21 may include an attribute requesting a custom inactivity timeout value in the STUN request sent in block 709 for reducing a required frequency for sending keepalive messages.

Next, the computer 21 receives back a STUN response that identifies another address in block 710 when the address translating security device includes a STUN server. The computer 21 then returns to block 706.

When the compared address finally matches the local address in block 707, the computer 21 has finished walking backwards through all on path network address translating

security devices. In block 711, the computer 21 establishes an efficient media path with a remote endpoint using one or more of the addresses for the discovered security devices.

In block 712, the computer 21 updates a local configuration file so that future binding verifications are addressed to the observed STUN server located on one of the identified security devices instead of the public STUN server. Future STUN requests are sent to the observed STUN server rather than the public STUN server according to the configuration file.

FIG. 8 illustrates an example method for using the NATs illustrated in FIG. 2.

In block 801 the NAT 26 or 27 receives a STUN request generated by an associated endpoint and addressed to a default STUN port on a local STUN server. In block 802, the NAT 26 or 27 determines whether the STUN request includes an attribute requesting a custom inactivity timeout. When an attribute requesting a custom inactivity timeout is not included, in block 803A the NAT 26 or 27 locally sets an inactivity timeout according to a default value. When an attribute requesting a custom inactivity timeout is included, the NAT 26 or 27 locally sets an inactivity timeout according to the indicated value.

In block 804, the NAT 26 or 27 generates a STUN response containing a payload having a source network address for the STUN request. The NAT 26 or 27 may also include the internal mapping from their respect address translation table in block 805. Next, in block 806 the NAT 26 or 27 generates a binding for forwarding incoming messages from a particular remote endpoint for the duration of the configured inactivity timeout.

FIG. 9 illustrates an example method for using the firewall illustrated in FIG. 3.

In block 901 the firewall 25 stores a STUN transaction identifier for a received STUN request responsive to observing a STUN magic cookie or other identifier included in the received STUN request. In block 902 the firewall 25 determines whether the STUN request

includes an attribute requesting a custom inactivity timeout. When an attribute requesting a custom inactivity timeout is not included, in block 903A the firewall 25 locally sets an inactivity timeout according to a default value. When an attribute requesting a custom inactivity timeout is included, the firewall 25 locally sets an inactivity timeout according to the indicated value.

In block 904, the firewall 25 receives a STUN response and attaches a local address outside of a digital signature included in the STUN response when the STUN response includes a value corresponding to the stored STUN transaction identifier. In block 905 the firewall 25 forwards the STUN response including the attached address to an endpoint.

In block 906 the firewall 25 opens a pinhole that allows incoming messages from a particular remote endpoint to reach the associated endpoint. The pinhole is maintained for the duration of the configured inactivity timeout.

The above examples describe the identification of firewalls and NATs in different figures only for ease of explanation. It will be understood by one of ordinary skill in the art that an endpoint may be located behind any combination of firewalls and NATs, which can all be identified using a single pattern of STUN requests and responses. In other words, a single STUN request sent to an undiscovered STUN server located on a NAT may elicit a STUN response that identifies another NAT as well as one or more firewalls. Also, certain benefits that are explained with reference to the example including NATs also apply to the example including the firewall, e.g. the bandwidth savings realized by eliminating the requirement for keepalive traffic all the way to the call controller are also realized in a firewall environment.

The above examples describe a computer identifying on path security devices. In other examples, other endpoints such as a personal computer, an IP phone, a Personal Digital

Assistant (PDA), a cell phone, a smart phone, a PSTN gateway, etc., may identify security devices using the methods described above.

Several preferred examples have been described above with reference to the accompanying drawings. Various other examples of the invention are also possible and practical. The system may be exemplified in many different forms and should not be construed as being limited to the examples set forth above.

The figures listed above illustrate preferred examples of the application and the operation of such examples. In the figures, the size of the boxes is not intended to represent the size of the various physical components. Where the same element appears in multiple figures, the same reference numeral is used to denote the element in all of the figures where it appears.

Only those parts of the various units are shown and described which are necessary to convey an understanding of the examples to those skilled in the art. Those parts and elements not shown are conventional and known in the art.

The system described above can use dedicated processor systems, micro controllers, programmable logic devices, or microprocessors that perform some or all of the operations. Some of the operations described above may be implemented in software and other operations may be implemented in hardware.

For the sake of convenience, the operations are described as various interconnected functional blocks or distinct software modules. This is not necessary, however, and there may be cases where these functional blocks or modules are equivalently aggregated into a single logic device, program or operation with unclear boundaries. In any event, the functional blocks and software modules or features of the flexible interface can be implemented by themselves, or in combination with other operations in either hardware or software.

Having described and illustrated the principles of the invention in a preferred embodiment thereof, it should be apparent that the invention may be modified in arrangement and detail without departing from such principles. I claim all modifications and variation coming within the spirit and scope of the following claims.

CLAIMS

1. A method comprising:

sending a first address request to a publicly accessible server that identifies publicly addressable network addresses to endpoints;

receiving back a first response that identifies a publicly addressable network address associated with an originating source of the first address request;

comparing the publicly addressable network address to a local network address; and

sending a second address request to a network address translating security device using the publicly addressable network address when the publicly addressable network address is different than the local network address.

2. The method of claim 1, further comprising:

receiving back a Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs) (STUN) second response from the network address translating security device, the STUN second response identifying a different network address;

comparing the different network address to the local address; and

sending a STUN request to a STUN server located on a different network address translating security device when the different network address does not match the local address.

3. The method of claim 2 further comprising providing the different network address to a remote network device that is reachable using the publicly addressable network address to establish a media path that circumvents the network address translating security device.

4. The method of claim 1 wherein the first response includes a payload containing an address for a firewall that does not perform network address translation for outgoing communications sent from the originating source.

5. The method of claim 1 wherein the first response includes a NAT address for the network address translating security device and a source address contained in the first address request when received by the network address translating security device.

6. The method of claim 1 wherein the first address request includes an attribute requesting a custom timeout for a corresponding session.

7. The method of claim 1 further comprising updating a local configuration so that periodic binding verifications are addressed to the network address translating security device instead of the publicly accessible server.

8. An apparatus, comprising:

one or more processors; and

a memory coupled to the one or more processors comprising instructions executable by the processors, the processors operable when executing the instructions to:

receive and forward an outgoing address request at a security device;

attach a local address for the security device to an incoming response elicited by the outgoing address request;

forward the incoming response including the attached local address; and

set a timeout duration of a local pinhole according to a setting provided by a generation source of the outgoing address request.

9. The apparatus of claim 8 wherein the processors are operable when executing the instructions to:

locally store a Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs) (STUN) transaction identifier included in the outgoing address request; and

compare a value included in the incoming response to the locally stored STUN transaction identifier before attaching the local address to the incoming response.

10. The apparatus of claim 9 wherein the local address is attached to the incoming response only when the value included in the incoming response corresponds to the locally stored STUN transaction identifier.

11. The apparatus of claim 8 wherein the forwarded incoming response includes the local address contained outside a digital signature and a remote network address that corresponds to either the generation source or a network address translator and is contained inside the digital signature.

12. The apparatus of claim 8, wherein the security device is a firewall that does not modify source network addresses for outgoing messages.

13. The apparatus of claim 8 wherein the outgoing address request includes a STUN identifier or an explicit request for a firewall address.

14. An apparatus, comprising:

one or more processors; and

a memory coupled to the one or more processors comprising instructions executable by the processors, the processors operable when executing the instructions to:

forward a first address request generated by an endpoint and received at a first security device to a second security device that provides network address translation;

forward a first response generated by the second security device, the first response including a payload containing a local network address for the first security device;

receive a second address request generated by the endpoint;

generate a second response that includes a second payload containing a source network address that corresponds to the second address request in response to receiving the second address request; and

send the second response to the endpoint.

15. The apparatus of claim 14 wherein the processors are operable when executing the instructions to:

configure the first security device to forward incoming messages received from a remote endpoint during a session; and

set a local inactivity timeout for the session according to an attribute included in the second address request.

16. The apparatus of claim 15 wherein the processors are operable when executing the instructions to process periodic keepalive messages that are addressed to the second security device and sent according to a period corresponding to a value specified by the attribute.

17. The apparatus of claim 14 wherein the processors are operable when executing the instructions to process a media flow that circumvents the second security device.

18. The apparatus of claim 14 wherein the processors are operable when executing the instructions to forward Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs) (STUN) requests addressed to a STUN server located on the second security device.

19. The apparatus of claim 14 wherein the processors are further operable to include a local address in the second response.

20. The apparatus of claim 16 wherein one of the responses includes a firewall address for a firewall that intercepted the first address request.

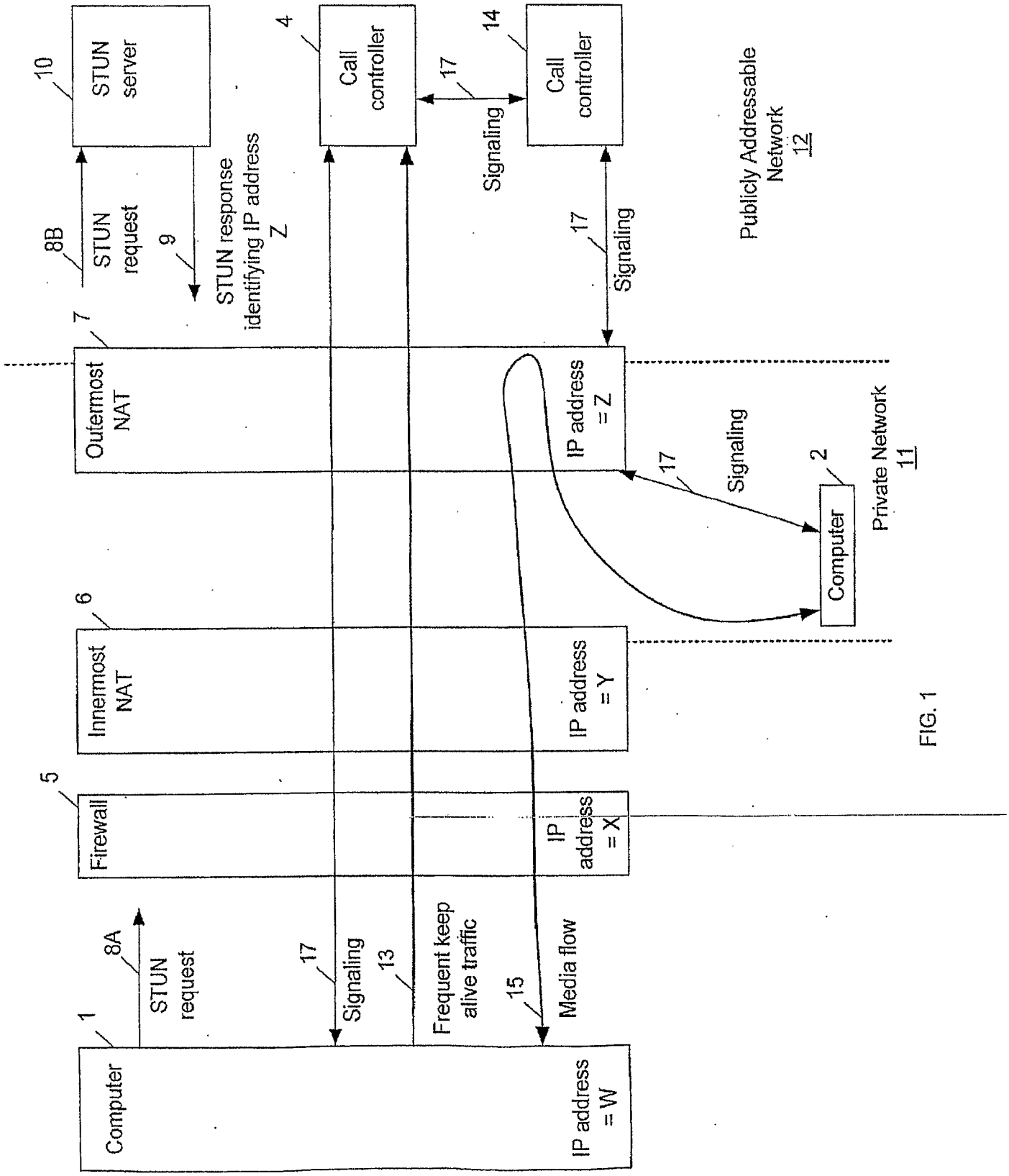

Computer
1
IP address = W
8A
STUN request
Firewall
5
IP address = X
Innermost NAT
6
IP address = Y
Outermost NAT
7
IP address = Z
8B
STUN request
9
STUN response identifying IP address Z
STUN server
10
Call controller
4
Call controller
14
17
Signaling
13
Frequent keep alive traffic
15
Media flow
17
Signaling
17
Signaling
17
Signaling
Computer
2
Publicly Addressable Network
12
Private Network
11

FIG. 1

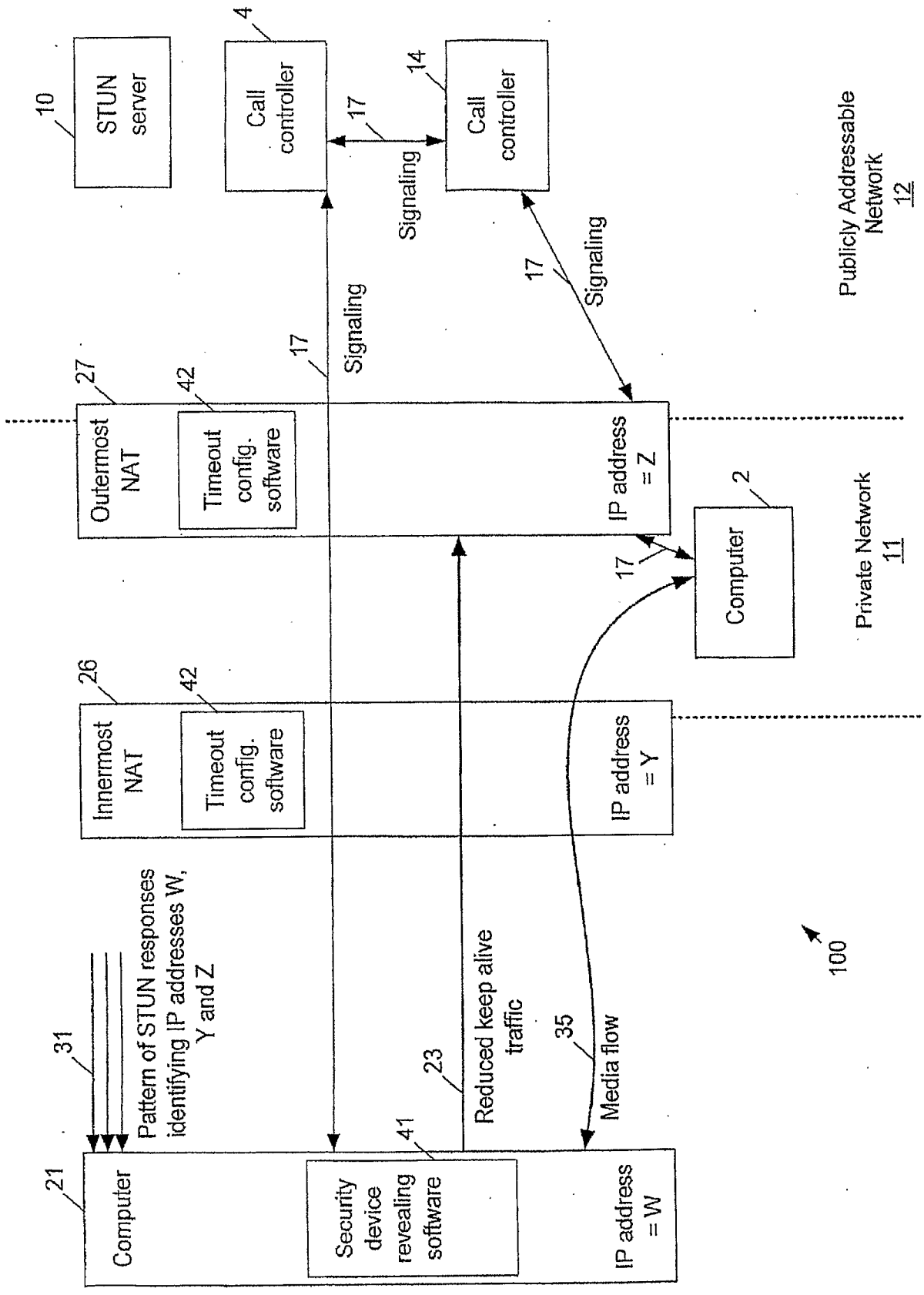

10
STUN server
4
Call controller
14
Call controller
17
Signaling
Signaling
17
Signaling
17
Publicly Addressable Network 12
27
Outermost NAT
42
Timeout config. software
IP address = Z
2
Computer
17
Private Network 11
26
Innermost NAT
42
Timeout config. software
IP address = Y
31
Pattern of STUN responses identifying IP addresses W, Y and Z
23
Reduced keep alive traffic
35
Media flow
41
Security device revealing software
21
Computer
IP address = W
100

FIG. 2

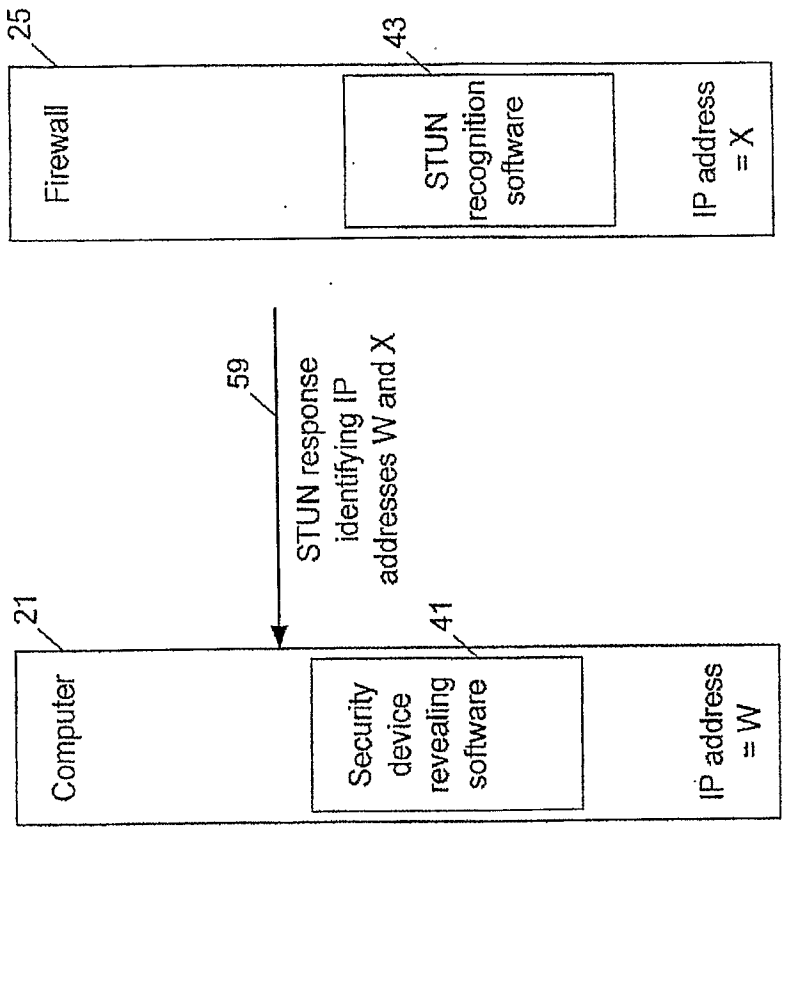

25
Firewall
43
STUN recognition software
IP address = X
59
STUN response identifying IP addresses W and X
21
Computer
41
Security device revealing software
IP address = W
100

FIG. 3

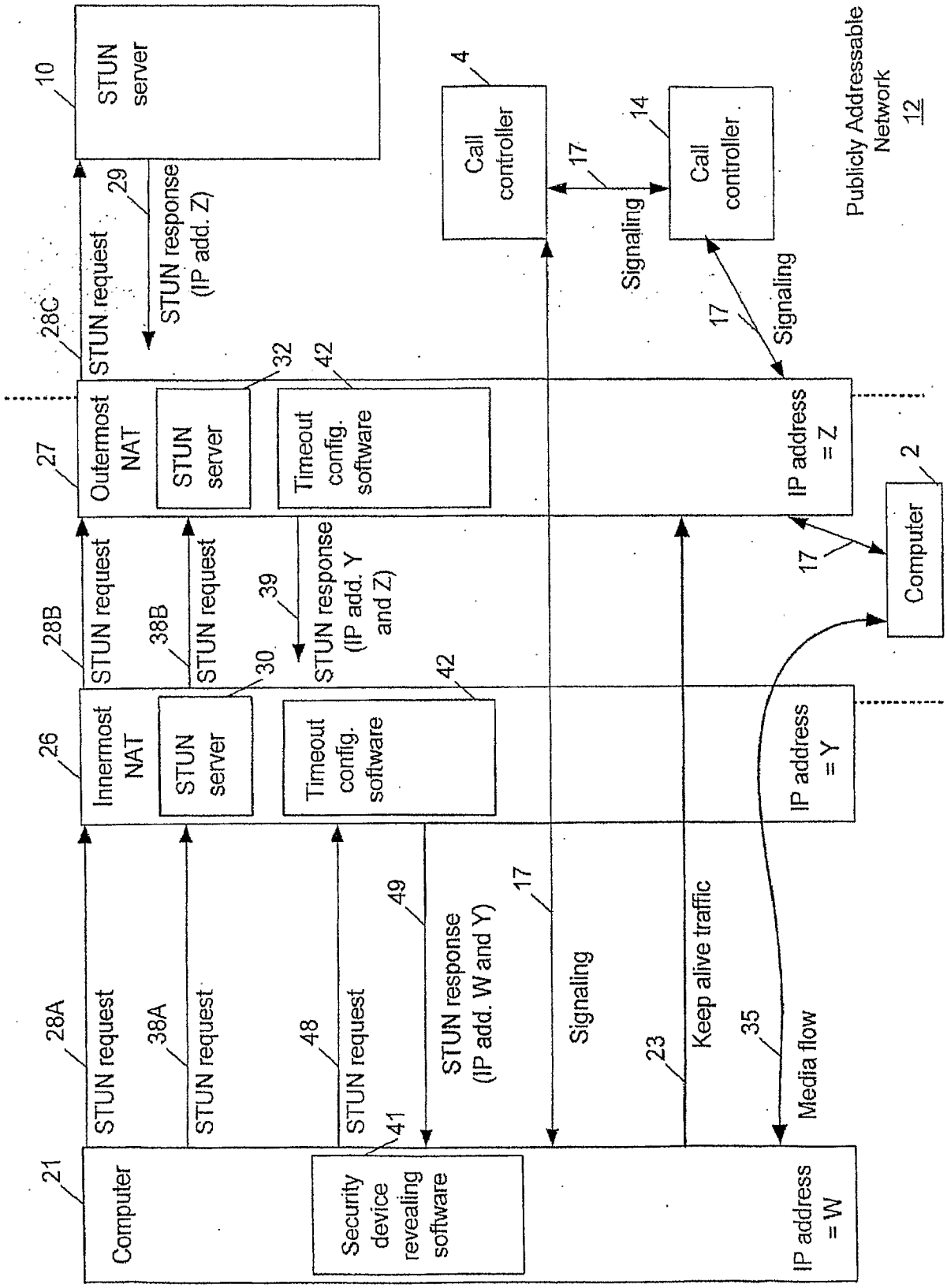

10
STUN server
29
STUN response (IP add. Z)
28C
STUN request
4
Call controller
17
Signaling
14
Call controller
Publicly Addressable Network 12
27
Outermost NAT
STUN server
32
Timeout config. software
42
IP address = Z
17
Signaling
2
Computer
Private Network 11
28B
STUN request
38B
STUN request
39
STUN response (IP add. Y and Z)
26
Innermost NAT
STUN server
30
Timeout config. software
42
IP address = Y
28A
STUN request
38A
STUN request
48
STUN request
49
STUN response (IP add. W and Y)
17
Signaling
23
Keep alive traffic
35
Media flow
21
Computer
Security device revealing software
41
IP address = W

FIG. 4

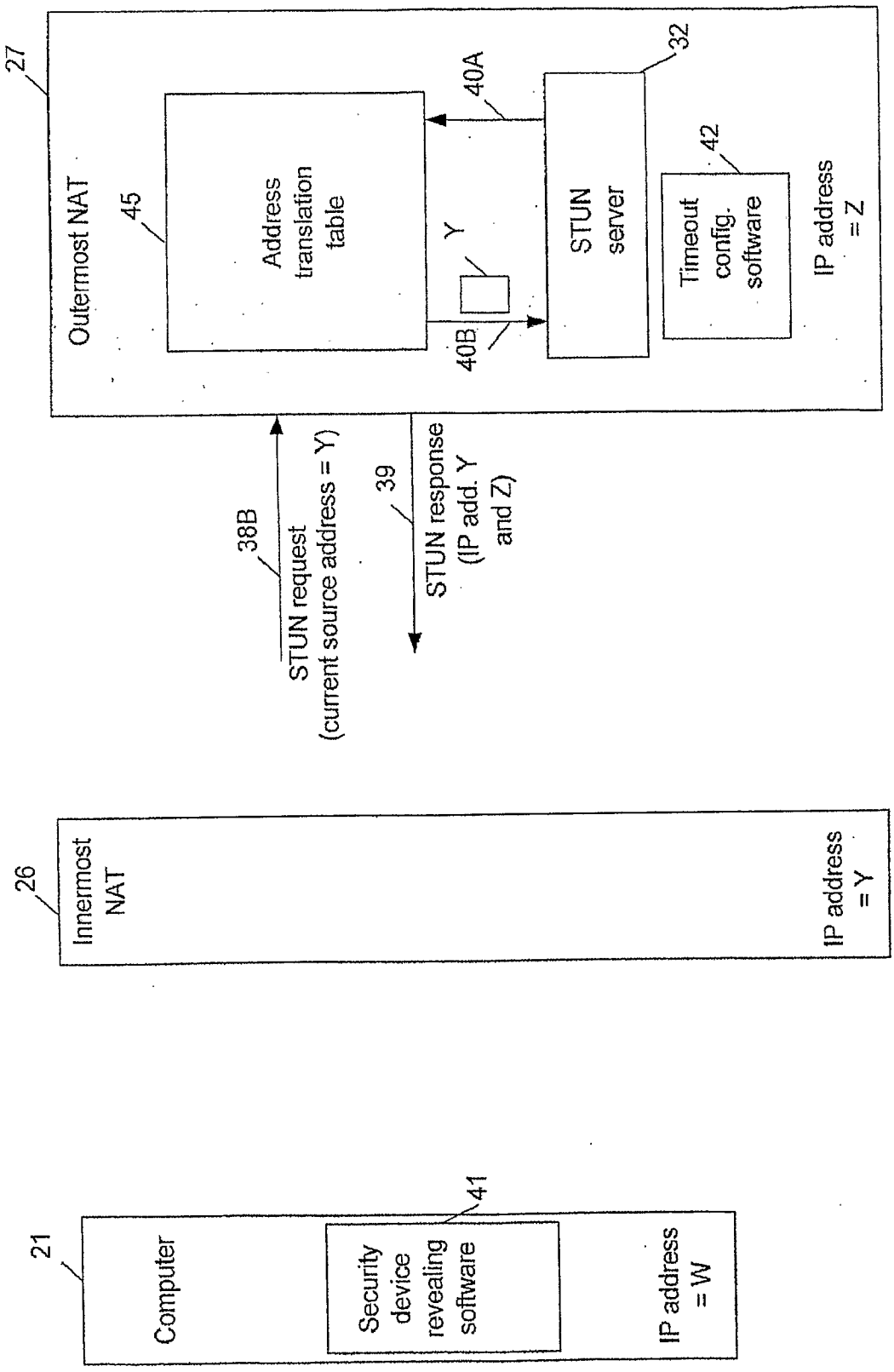

27
Outermost NAT
45
Address translation table
40A
32
STUN server
42
Timeout config. software
IP address = Z
Y
40B
38B
STUN request (current source address = Y)
39
STUN response (IP add. Y and Z)
26
Innermost NAT
IP address = Y
21
Computer
41
Security device revealing software
IP address = W

FIG. 5

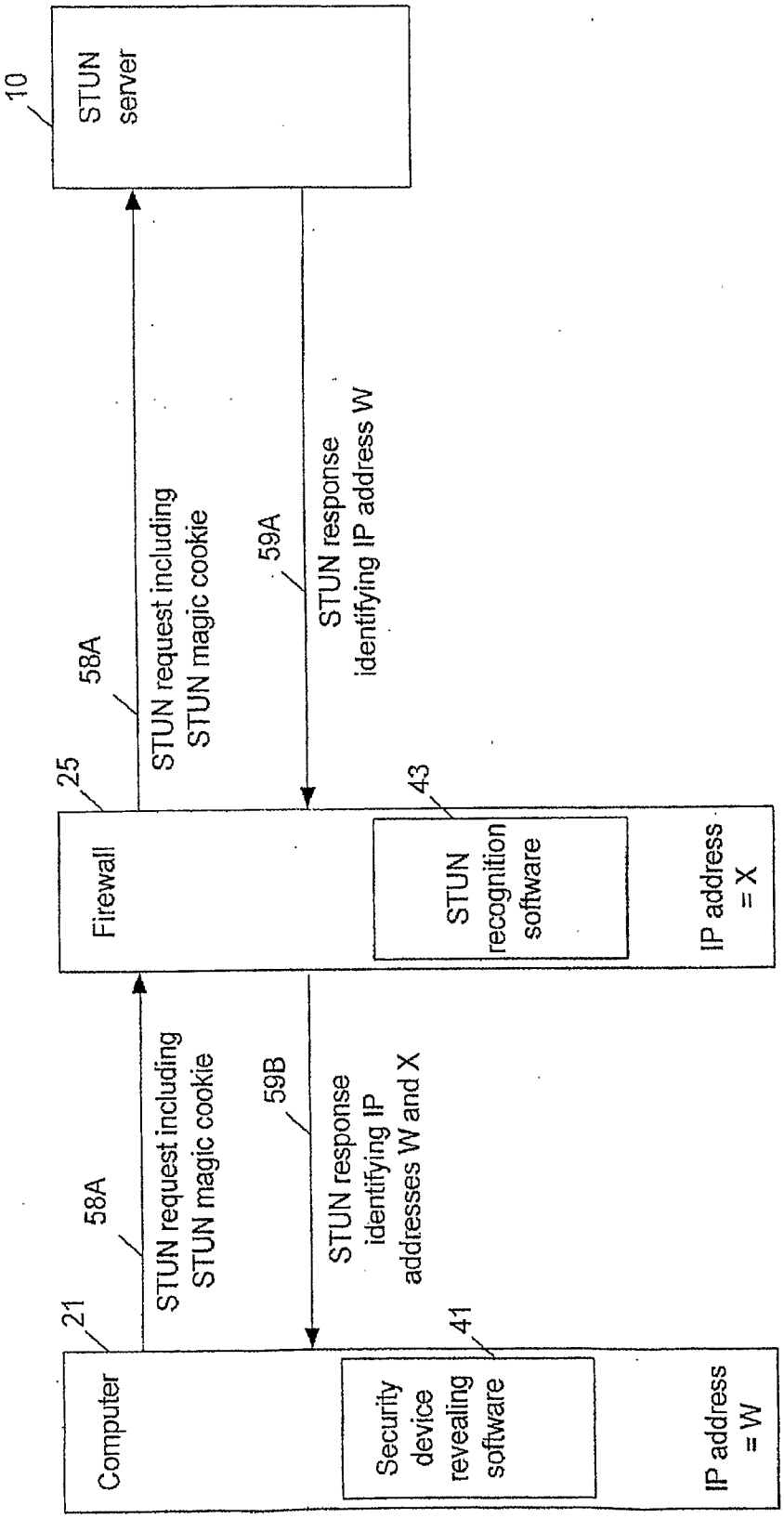

10
STUN server
58A
STUN request including STUN magic cookie
59A
STUN response identifying IP address W
25
Firewall
43
STUN recognition software
IP address = X
58A
STUN request including STUN magic cookie
59B
STUN response identifying IP addresses W and X
21
Computer
41
Security device revealing software
IP address = W

FIG. 6

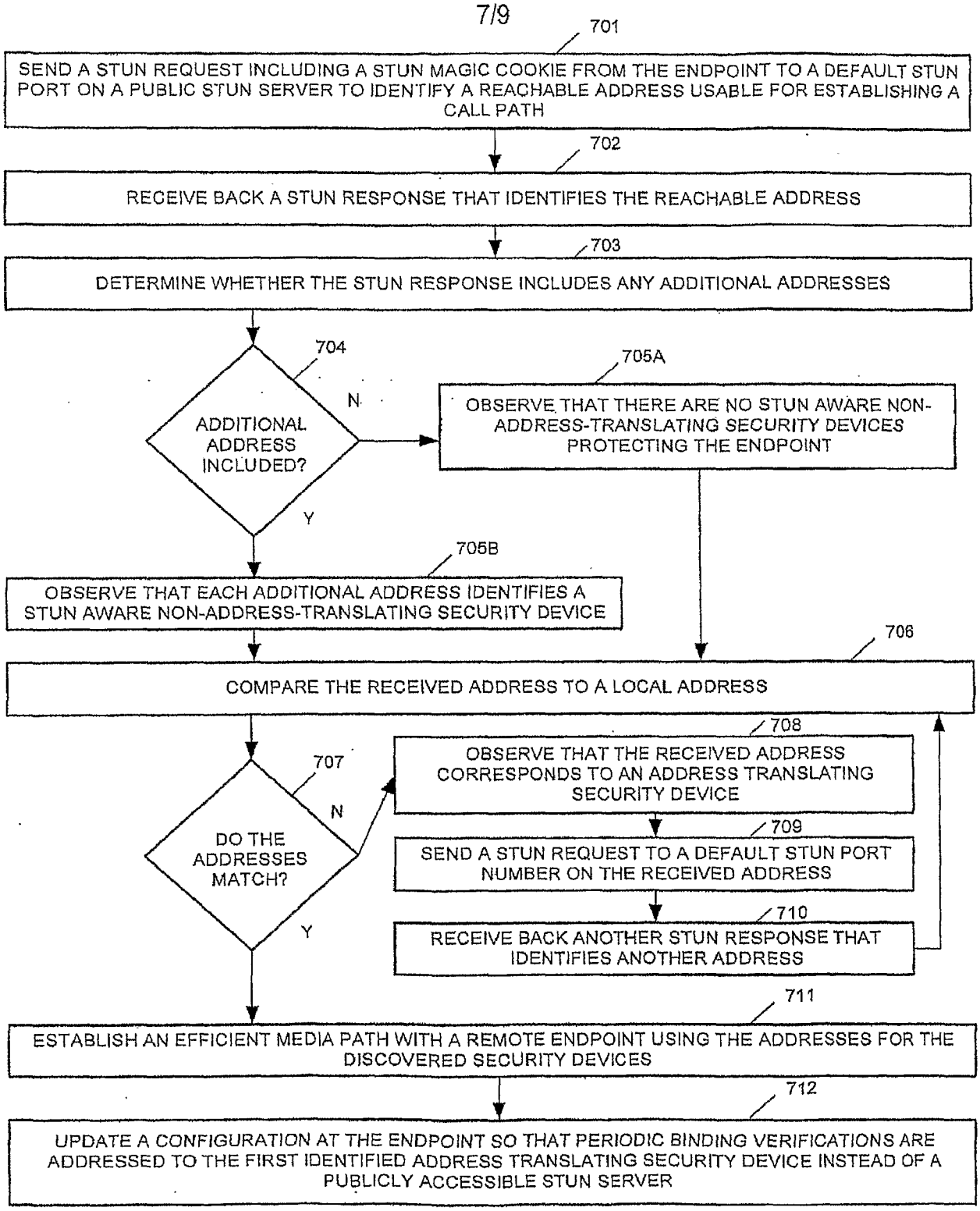

701
SEND A STUN REQUEST INCLUDING A STUN MAGIC COOKIE FROM THE ENDPOINT TO A DEFAULT STUN PORT ON A PUBLIC STUN SERVER TO IDENTIFY A REACHABLE ADDRESS USABLE FOR ESTABLISHING A CALL PATH
702
RECEIVE BACK A STUN RESPONSE THAT IDENTIFIES THE REACHABLE ADDRESS
703
DETERMINE WHETHER THE STUN RESPONSE INCLUDES ANY ADDITIONAL ADDRESSES
704
ADDITIONAL ADDRESS INCLUDED?
N
Y
705A
OBSERVE THAT THERE ARE NO STUN AWARE NON-ADDRESS-TRANSLATING SECURITY DEVICES PROTECTING THE ENDPOINT
705B
OBSERVE THAT EACH ADDITIONAL ADDRESS IDENTIFIES A STUN AWARE NON-ADDRESS-TRANSLATING SECURITY DEVICE
706
COMPARE THE RECEIVED ADDRESS TO A LOCAL ADDRESS
707
DO THE ADDRESSES MATCH?
N
Y
708
OBSERVE THAT THE RECEIVED ADDRESS CORRESPONDS TO AN ADDRESS TRANSLATING SECURITY DEVICE
709
SEND A STUN REQUEST TO A DEFAULT STUN PORT NUMBER ON THE RECEIVED ADDRESS
710
RECEIVE BACK ANOTHER STUN RESPONSE THAT IDENTIFIES ANOTHER ADDRESS
711
ESTABLISH AN EFFICIENT MEDIA PATH WITH A REMOTE ENDPOINT USING THE ADDRESSES FOR THE DISCOVERED SECURITY DEVICES
712
UPDATE A CONFIGURATION AT THE ENDPOINT SO THAT PERIODIC BINDING VERIFICATIONS ARE ADDRESSED TO THE FIRST IDENTIFIED ADDRESS TRANSLATING SECURITY DEVICE INSTEAD OF A PUBLICLY ACCESSIBLE STUN SERVER

FIG. 7

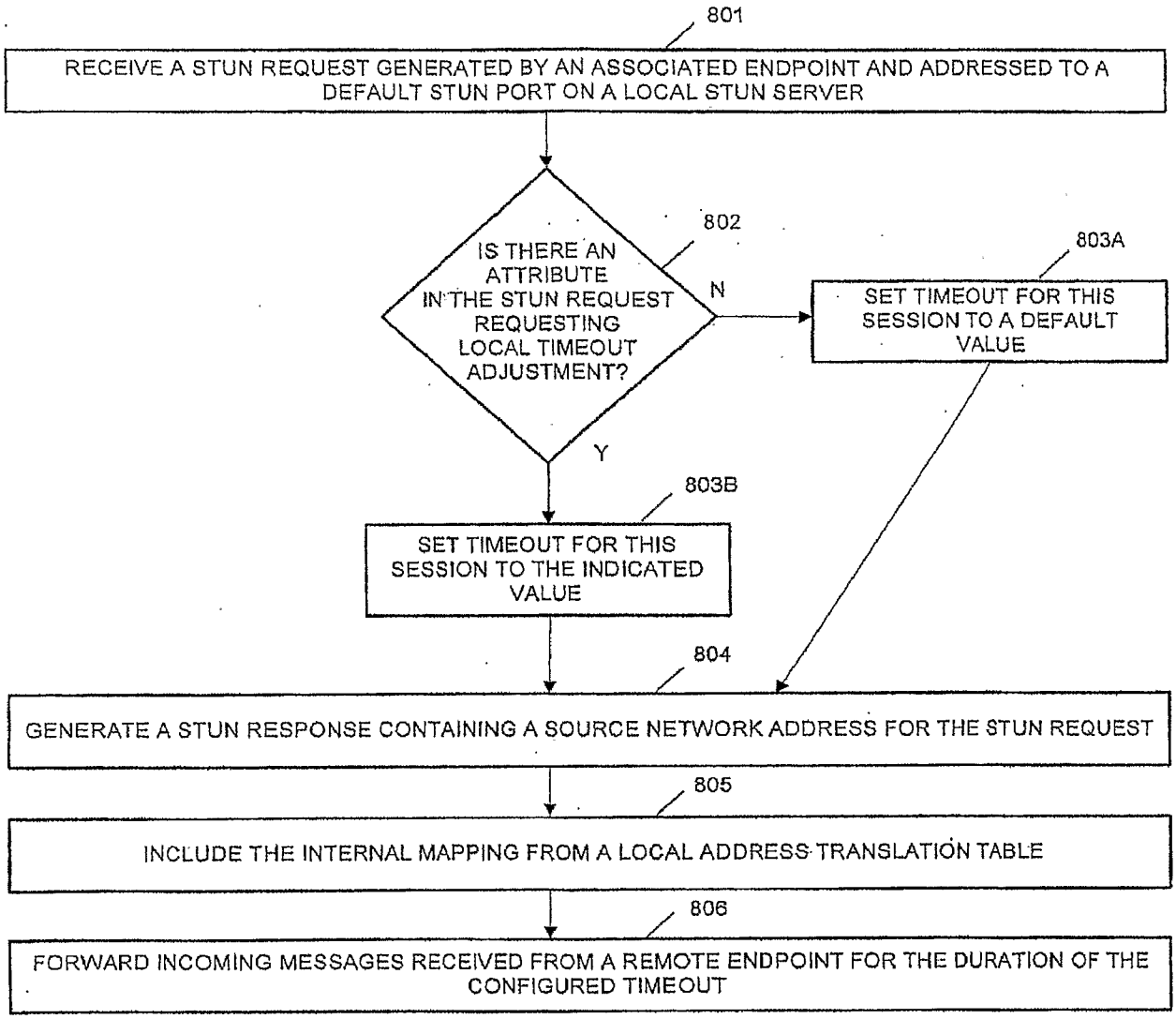

801
RECEIVE A STUN REQUEST GENERATED BY AN ASSOCIATED ENDPOINT AND ADDRESSED TO A DEFAULT STUN PORT ON A LOCAL STUN SERVER
802
IS THERE AN ATTRIBUTE IN THE STUN REQUEST REQUESTING LOCAL TIMEOUT ADJUSTMENT?
N
803A
SET TIMEOUT FOR THIS SESSION TO A DEFAULT VALUE
Y
803B
SET TIMEOUT FOR THIS SESSION TO THE INDICATED VALUE
804
GENERATE A STUN RESPONSE CONTAINING A SOURCE NETWORK ADDRESS FOR THE STUN REQUEST
805
INCLUDE THE INTERNAL MAPPING FROM A LOCAL ADDRESS TRANSLATION TABLE
806
FORWARD INCOMING MESSAGES RECEIVED FROM A REMOTE ENDPOINT FOR THE DURATION OF THE CONFIGURED TIMEOUT

FIG. 8

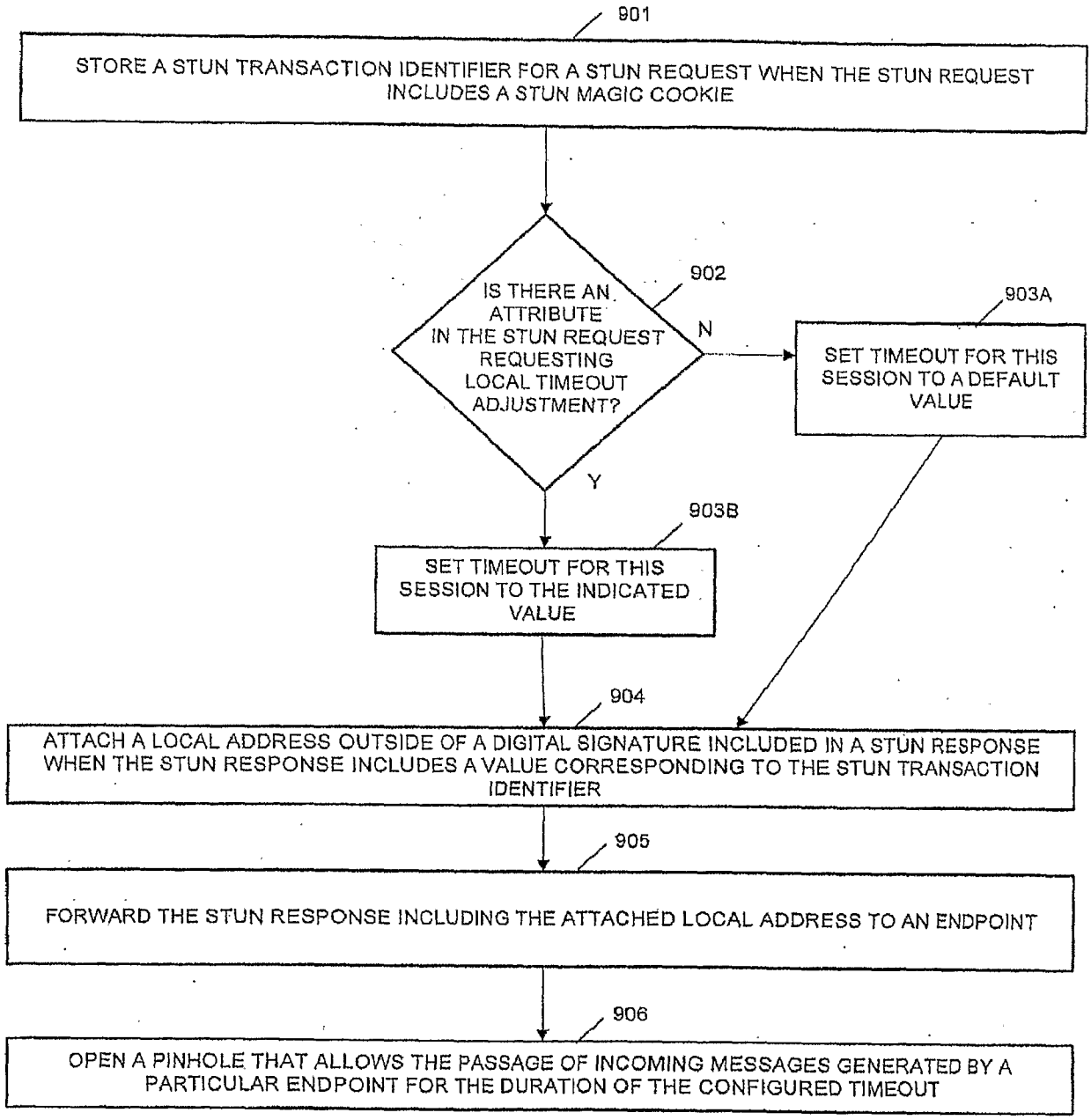

901
STORE A STUN TRANSACTION IDENTIFIER FOR A STUN REQUEST WHEN THE STUN REQUEST INCLUDES A STUN MAGIC COOKIE
902
IS THERE AN ATTRIBUTE IN THE STUN REQUEST REQUESTING LOCAL TIMEOUT ADJUSTMENT?
N
903A
SET TIMEOUT FOR THIS SESSION TO A DEFAULT VALUE
Y
903B
SET TIMEOUT FOR THIS SESSION TO THE INDICATED VALUE
904
ATTACH A LOCAL ADDRESS OUTSIDE OF A DIGITAL SIGNATURE INCLUDED IN A STUN RESPONSE WHEN THE STUN RESPONSE INCLUDES A VALUE CORRESPONDING TO THE STUN TRANSACTION IDENTIFIER
905
FORWARD THE STUN RESPONSE INCLUDING THE ATTACHED LOCAL ADDRESS TO AN ENDPOINT
906
OPEN A PINHOLE THAT ALLOWS THE PASSAGE OF INCOMING MESSAGES GENERATED BY A PARTICULAR ENDPOINT FOR THE DURATION OF THE CONFIGURED TIMEOUT

FIG. 9

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 07/60770

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G06F 15/16 (2007.01)
USPC - 726/3
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(8): G06F 15/16 (2007.01)
USPC: 726/3

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
709/201, 217-220, 223-225, 227-229, 249; 726/3-6;

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
USPTO WEST (PGPB, USPT, EPAB, JPAB); DIALOG PRO; GOOGLE
Search Terms Used: STUN, compare, match, local, private, public, bind, security, path, pinhole, media, path, network, translate, security, device, request, address, identify$, timeout, firewall, NAT, digital signature

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X ---- Y	US 2006/0215684 A1 (CAPONE) 28 September 2006 (28.09.2006), entire document, especially para [0010] -[0015], [0038],[0057]-[0058],[0066]-[0069], [0073], [0094]-[0096] [0112] and Figs 3-4 and 8	1, 4-6, 8-9, 12, 14-17, 19-20 ---------- 2-3,7, 10-11, 13, 18
Y	US 2006/0075127 A1 (JUNCKER et al.) 06 April 2006 (06.04.2006), entire document, especially para [0009]-[0014] and Figs 1-2	2-3, 7, 10, 13 and 18
Y	US 2004/0024882 A1 (AUSTIN et al.) 05 February 2004 (05.02.2004), entire document, para [0057]-[0058] and Figs 3-4	11
A	US 2004/0057385 A1 (ROSHKO) 25 March 2004 (25.03.2004)	1-20
A	US 2006/0212702 A1 (FIRESTONE et al.) 21 September 2006 (21.09.2006)	1-20

☐ Further documents are listed in the continuation of Box C. ☐

* 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 application or patent 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
26 September 2007 (26.09.2007)	05 FEB 2008
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Authorized officer: Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774