Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2468480 C 2010/01/19

(11)(21) **2 468 480**

(12) **BREVET CANADIEN**
CANADIAN PATENT

(13) **C**

(22) Date de dépôt/Filing Date: 2004/05/26

(41) Mise à la disp. pub./Open to Public Insp.: 2004/11/26

(45) Date de délivrance/Issue Date: 2010/01/19

(30) Priorité/Priority: 2003/05/26 (FR0306308)

(51) Cl.Int./Int.Cl. *H04L 12/56* (2006.01), *H04L 12/66* (2006.01), *H04L 29/06* (2006.01)

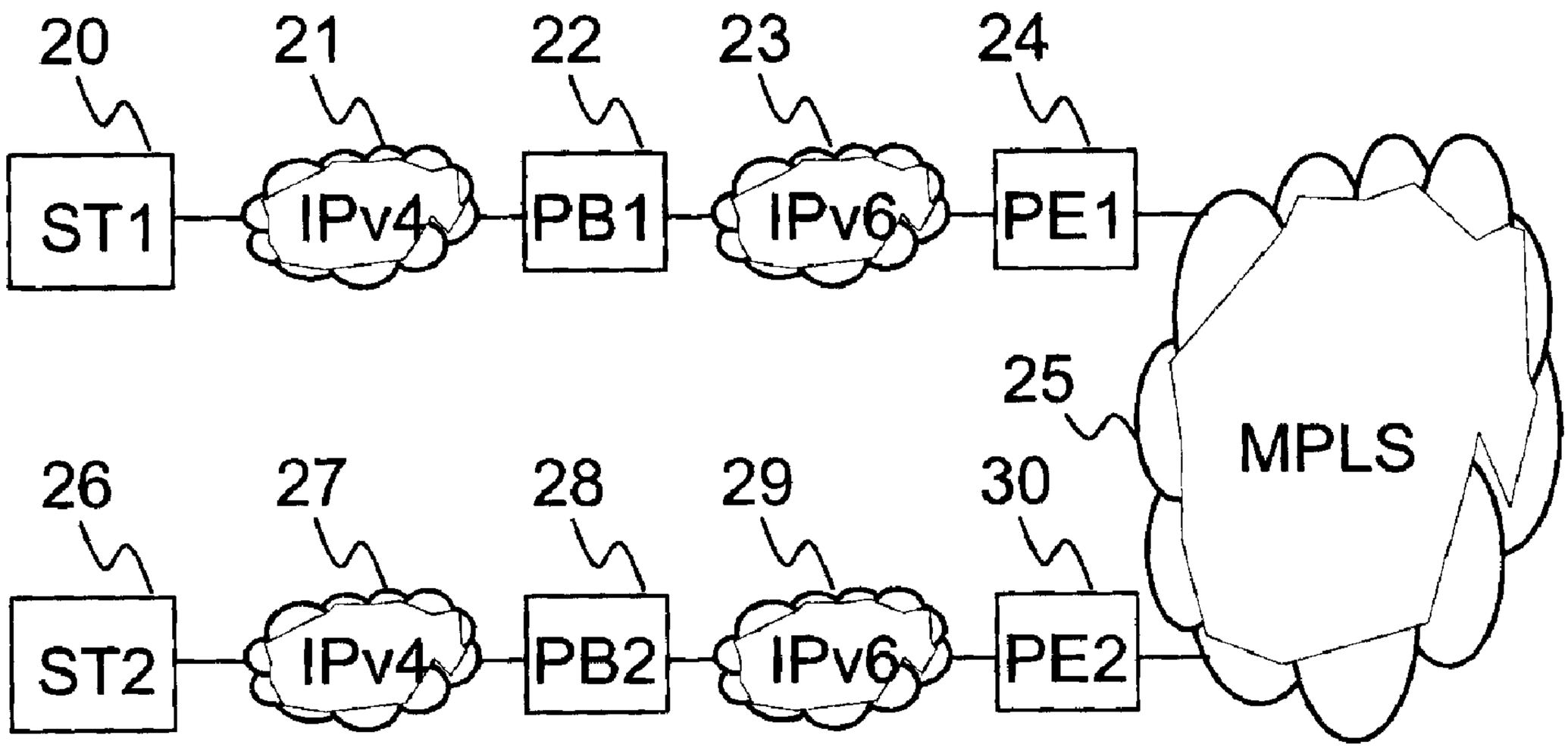
(72) Inventeurs/Inventors:
LE PENNEC, JEAN-FRANCOIS, FR;
BRUNO, AURELIEN, FR;
GALAND, CLAUDE, FR;
GIROIR, DIDIER, FR

(73) Propriétaire/Owner:
AT&T CORP., US

(74) Agent: KIRBY EADES GALE BAKER

(54) Titre : SYSTEME POUR CONVERTIR DES DONNEES IPv4 EN DONNEES IPv6 A TRANSMETTRE SUR UN RESEAU COMMUTE IP

(54) Title: SYSTEM FOR CONVERTING DATA BASED UPON IPv4 INTO DATA BASED UPON IPv6 TO BE TRANSMITTED OVER AN IP SWITCHED NETWORK



(57) Abrégé/Abstract:
System for converting data packets based upon the IPv4 protocol into data packets based upon the IPv6 protocol to be transmitted over a Virtual Private Network (VPN) in a data transmission system comprising an IP switched network (25) to which are connected a first workstation (20) via a first Provider Edge (PE) device (24) and a second workstation (26) via a second PE device (30), the workstations operating with the IPv4 protocol whereas the IP switched network operates with the IPv6 protocol. this system comprises a first packet builder (22) between the first workstation and the IP switched network for converting any data packet based upon the IPv4 protocol received from the first workstation and transmitted to the IP switched network into the IPv6 protocol and reciprocally, and a second packet builder (28) between the IP switched network and the second workstation for converting any data packet based upon the IPv6 protocol received from the network and transmitted to the second workstation into the IPv4 protocol and reciprocally.

SYSTEM FOR CONVERTING DATA BASED UPON IPv4 INTO DATA BASED UPON IPv6 TO BE TRANSMITTED OVER AN IP SWITCHED NETWORK

Abstract

System for converting data packets based upon the IPv4 protocol into data packets based upon the IPv6 protocol to be transmitted over a Virtual Private Network (VPN) in a data transmission system comprising an IP switched network (25) to which are connected a first workstation (20) via a first Provider Edge (PE) device (24) and a second workstation (26) via a second PE device (30), the workstations operating with the IPv4 protocol whereas the IP switched network operates with the IPv6 protocol. this system comprises a first packet builder (22) between the first workstation and the IP switched network for converting any data packet based upon the IPv4 protocol received from the first workstation and transmitted to the IP switched network into the IPv6 protocol and reciprocally, and a second packet builder (28) between the IP switched network and the second workstation for converting any data packet based upon the IPv6 protocol received from the network and transmitted to the second workstation into the IPv4 protocol and reciprocally.

SYSTEM FOR CONVERTING DATA BASED UPON IPv4 INTO DATA BASED UPON IPv6 TO BE TRANSMITTED OVER AN IP SWITCHED NETWORK

Technical field

The present invention generally relates to the transmission of data packets within an IP switched network such as a Multi-Protocol Label Switching (MPLS) network and relates in particular to a system for converting data based upon IPv4 into data based upon IPv6 to be transmitted over such a network.

Background

There are more and more Virtual Private Networks (VPNs) being implemented to enable private communications among devices, even if some or all these communications are transmitted over a public network. Most VPNs are built to support as layer 3 the IP protocol rules and mainly the IPv4 flavor of the IP protocol.

There is today a problem with the IPv4 protocol, which is raised by the well-known limitation of the address space within IPV4, but also a problem raised by the forwarding rules of IPv4 wherein the routing actions are based on the destination address without any influence due to the source address. Since any IPv4 network can be located anywhere in the world, the routers must maintain a

record for every active network. This is not only true for the Internet network, but also for VPNs, intranets and extranets.

The real, long term solution to these problems is a much larger address space, allowing for more addressing structure, less stringent allocation policies and more efficient routing. This is what IPv6 offers, which makes it attractive for backbone usage. The IP scalability and ease of management are some of the objectives of IPv6. But as all the devices are not v6, there is a need to find mechanisms to take advantage of v6 today. In addition, current proposed uses of IPv6 headers are not really associated with the label switching and VPN support capability of protocols like Multi-Protocol Label Switching (MPLS).

Native use of IPv6 based only on IPv6 address fields is even more complex than IPv4, if route filtering is used to provide connectivity and build VPNs due to the length of the address field on which a lookup should be performed. Such a "route filtering" can be implemented to control route propagation such that certain networks receive routes for other networks within their own community of interest (i.e., VPN).

Route filtering is based on the proposition that some network subset of an underlying IP network supporting the VPN (such as the Internet) actually forms the VPN. Routes associated with this network

subset are filtered such that they are not announced to any other network(s) connected to the network subset forming the VPN. Conversely, no other non-VPN route is announced to the network subset. Route filtering on IPv4 implies that VPNs use different address spaces. IPv6 simplifies this filtering since VPNs are identified in the IP address field but opens a security hole.

For privacy of services on a network-layer, route-filtering VPN is implemented by restricting any of the VPN hosts from responding to packets that contain source addresses from outside the VPN. Such restrictions are based on access control lists ("ACLs"), which are tables that tell a device which access rights each user has. That is, an ACL is a list of entries that grant or deny specific access rights to individuals or groups.

Route filtering VPNs, however, have various difficulties associated therewith. For example, such an arrangement can be wrongly configured such that it erroneously accepts packets, which it should not, and/or rejects packets that should be accepted. Another problem with route filtering is that, when implemented behind a gateway or firewall device, subscriber networks can define a router external to the gateway/firewall as the default router, so that users of the VPN behind the gateway/firewall can reach external networks as long as they know the default router address (even if it is not advertised). Additional shortcomings

of this technique include administrative mistakes, a static nature of the design and limitations on the security provided. In addition, the complexity for defining and maintaining all the rules is very high, so that the technique does not scale very well or very easily.

Various solutions have been put forward to achieve different levels of network privacy when building VPNs across a shared IP backbone. Many of these solutions require separate, per VPN forwarding capabilities, and make use of IP or MPLS-based tunnels across the backbone. Such tunnels add overhead and management complexity especially in tunnel peer devices. Also, within a VPN domain, an instance of routing is used to distribute VPN reachability information among routers. Any routing protocol can be used, and no VPN-related modifications or extensions are needed to the routing protocol for achieving VPN reachability.

Regardless of which of the above techniques (or other known techniques) is used to form a VPN, it should be understood that network security is a concern. In fact, network security is a concern in many contexts aside from VPNs, and, in general, increasing use of remote access over public networks and Internet access for inter-business communication are major driving forces behind the evolution of security technology.

Many techniques for network security revolve around the use of a firewall, which is generally known as a combination of hardware and software used to implement a security policy governing network traffic between two or more networks, some of which being under administrative control (e.g., organizational networks) and some of which being not under administrative control (e.g., the Internet).

Regardless of the routing technique used, the routing mechanism is usually not used to implement security policy. That is, a routing mechanism is often considered too dynamic and unreliable to perform security functions. Routing functions and supporting structures are primarily designed to route packets efficiently and reliably, not securely. Therefore, filtering techniques that can be implemented in connection with operation of a firewall (and/or router) for security purposes exist, and examples of these (as referred to above) are packet filtering, application proxies, and dynamic filtering (stateful inspection). The fields on which filtering is applied should be trusted fields, which cannot be easily the case with IPv4 header types. One of such necessary fields is the VPN ID type.

Although there are many techniques for implementing and securing individual VPNs, as discussed above, there is an additional need for VPNs which can cross-communicate without sacrificing service to

their users, e.g., without reducing the security of transmissions between the two (or more) VPNs or within a particular VPN.

Summary of the invention

Accordingly, the main object of the invention is to provide a system for converting the data based upon IPv4 into data based upon IPv6 and transmitting the converted data over an IP switched network, such a system avoiding the above-mentioned problems such as route filtering and improving the network security when building VPNs across a shared IP backbone.

The invention relates therefore to a system for converting data packets based upon the IPv4 protocol into data packets based upon the IPv6 protocol to be transmitted over a multiple Virtual Private Network (VPN) infrastructure in a data transmission system comprising an IP switched network to which are connected a first workstation via a first Provider Edge (PE) device and a second workstation via a second PE device, the workstations operating with the IPv4 protocol whereas the IP switched network operates with the IPv6 protocol. Such a system comprises a first packet builder between the first workstation and the IP switched network for converting any data packet based upon the IPv4 protocol received from the first workstation and transmitted to the IP switched network into the IPv6 protocol and

reciprocally, and a second packet builder between the IP switched network and the second workstation for converting any data packet based upon the IPv6 protocol received from the network and transmitted to the second workstation into the IPv4 protocol and reciprocally.

Brief description of the drawings

The above and other objects, features and advantages of the invention will be better understood by reading the following more particular description of the invention in conjunction with the accompanying drawings wherein:

- Fig. 1 represents a first embodiment of a data transmission system implementing the invention;
- Fig. 2A and 2B represent respectively the IPv6 aggregatable global unicast address and its modifications for implementing the invention;
- Fig. 3 shows the packet builders according to the invention connected to the information servers by the intermediary of a virtual network;
- Fig. 4 represents a second embodiment of the invention wherein the function of the packet builder is integrated in the PE device; and
- Fig. 5A and 5B represent respectively the change of the data packet structure as it flows from the source to the destination in the first and second embodiments of the invention.

Detailed description of the invention

A system according to the invention is illustrated in Fig. 1. In such a system, two workstations 20 and 26 exchange data through an IP switching network 25. Such a network is, in a preferred embodiment, based upon MPLS (Multi Protocol Label Switching) protocol which is an IP technology defining PE (Provider Edge) devices which define a physical path across the network for each customer flow, such a path being defined as a set of labels that are swapped along the path from an ingress PE such as PE1 24 to an egress PE such as PE2 30.

In IP switched network 25, which is preferably a MPLS network as already mentioned, several VPNs are implemented between the edge devices 24 and 30. The environment of the invention is related to how a specific VPN can be established and securely identified using layer 3 mechanisms like the one proposed in IPv6 protocol.

It is assumed that the data packets sent by each workstation 20 and 26 are based upon IPv4. Therefore, the workstations are each connected to an IPv4 network respectively network 21 for workstation 20 and network 27 for workstation 26. The IPv4 data flow sent from workstation 20 is then converted, as described hereunder, by a packet builder PB1 22 into an IPv6 data flow. The packets are forwarded to ingress PE1 24 through an IPv6 network 23, and then transmitted through network 25

using the label switching if this network is a MPLS network.

At the output of network 25, the data packets in IPv6 are received by PE2 30 and transferred to a packet builder PB2 28 through an IPv6 network 29. The packets which have been converted, as described hereunder, from IPv6 to IPv4 are forwarded to workstation ST2 28 through an IPv4 network 27. This data flow is of course reversed for the packets forwarded from workstation 26 to workstation 20.

The principle of the invention is to use an IP header which is derived from the IP header proposed by the IETF (Engineering Task Force) in RFCs 2373 and 2374 which is illustrated in Fig. 2A. The format of this global aggregatable unicast address is designed to support both the current provider based aggregation and a new type of aggregation used for the exchanges. The combination allows efficient routing aggregation for both sites, which connect directly to the providers. Sites have the choice to connect to either type of aggregation point.

Globally, in the address format illustrated in Fig. 2A, the first 48 bits are related to the public technology, the next 16 bits are related to the site topology and the remaining 64 bits are related to the interface identifier. The fields are the following:

FP Format Prefix (3 bit) for aggregatable

		global unicast addresses
	TLA ID	Top-Level Aggregation Identifier
	RES	Reserved for future use
	NLA ID	Next-Level Aggregation Identifier
	SLA ID	Site-Level Aggregation Identifier
	INTERFACE ID	Interface Identifier

This basic structure is re-used to provide a transparent use of IPv6 mechanisms such as the address lookup (longest prefix match). Unfortunately, the mechanism used to build the IPv6 format from the IPv4 format is static and manually set in the edge devices.

It is why the invention relates to a new Ipv6 format, illustrated in Fig. 2B, wherein the fields are the following:

- The provider ID (e.g., AT&T),
- The PE field including the number on the PE device attached to the IP switched network (e.g., 24 or 30). The way the PE field is defined is also part of the proposal and includes several methods: known active topology provided by network management tools, discovery at IP level: Traceroute + PE database with dynamic or regular update techniques, given by MPLS tools, forced or predefined,
- The VPN ID which may be subdivided into areas to provide a hierarchical VPN structure,

- Location ID which is the geographic location of the site,
- Host ID which refers to the device that uses the corresponding IPv4 address. The host may be a server, a client, a router, a firewall. Different rules may then be applied. In the preferred embodiment, it is the MAC address if this MAC address can be discovered and checked. It provides a unique identification which gives a supplementary privacy information. Host ID on the IPv6 address field may either contain the MAC address corresponding to the IPv4 address or just an equivalent ID and the correspondence between them is just maintained locally.
- IPv4 address (source or destination) which is present in the incoming packet.

The major advantage of the invention is to simplify the label distribution and management for VPN (MPLS VPN) since the destination PE is included in the destination address field of the new IPv6 header. A VPN Label per flow may be used instead of flow aggregation into labels insofar as there is no longer a need for a dual label implementation like what is done in a MPLS VPN. Using this method, a packet entering the MPLS network 25, for example on PE1, directly provides in its IPv6 header the destination PE such as PE2. No need for special

path identification is required if PE1 knows how to reach PE2.

Finally, for routing in the IP switched network, only the MSB bytes of the address field are used while on the access only the LSB bytes are used. This simplifies the address lookup mechanism even at PE level, which was one of the drawbacks of IPv6 since, normally, a 128 bit address lookup is required. Current PEs supporting IPv6 as input protocol may be used but may also be improved in identifying the sub-fields in the MSB part of the IPv6 address field in order to directly use the PE address for routing packets.

The MPLS network does not need to implement any additional VPN feature such as MPLS VPN as the privacy is insured by the IPv6 header which contains the VPN ID. As this field is not set by the user but by the Packet Builder device PB1 or PB2, there is no risk to have spoofing from user defined IPv6 flows. This is a security improvement that is worth using even for native IPv6 incoming flows. All secure fields should be validated, added or replaced at the ingress node of the network. Such fields include Provider field, PE field, VPN ID field, LOCATION ID field and HOST ID field if it is used also for filtering. When an IPv6 packet is received, it may be analyzed to verify that its header structure matches the local rules, which prevents such an IPv6 spoofing. If it doesn't match, depending on fields and rules, it may be

either modified or discarded. If modified, a swap table is built to rebuild the original address field for incoming packets from the network such as a stateful mechanism. It works, in that case, like an IPv6 NAT to convert IPv6 flows that do not match the proposed rules transparently. Such a conversion is performed in PB devices.

The invention improves the routing scalability. The network management of such an implementation is simpler since the IP header contains all the necessary fields to manage a flow, which is not the case today with IPv4 on one side and MPLS on the other side. It also provides an increased security.

As far as MPLS is concerned, current vendor and standard bodies are working on IPv6 through MPLS called 6PE, which reuses v4 MPLS VPN mechanisms. The invention consists in replacing v4 based MPLS and BGP mechanisms by native v6 mechanisms but it is still compatible with 6PE solution as an intermediate implementation. The advantage of introducing MPLS VPN knowledge in the IPv6 header is to converge into a single transmission element that has all the necessary information at any point in the network. MPLS to MPLS inter-working will also become easier to implement and manage in a more secure way insofar as the IPv6 header will contain the necessary crossover information and no need to bridge at the label level is required.

For implementing the invention, it is necessary to use servers that will give the value of each field for the source and destination addresses in a secure way. These servers include DNS, Whois, Route reflectors and also Certificate Authority since the IPv6 address has to be securely given. Fig. 3 represents the logical connections of packet builders 22 or 28 to such server devices 31 to 37 by means of a virtual network 39 which can be IPv4 network 21 or 27, IPv6 network 23 or 29, the MPLS network 25, or any other network connected to one of these networks or directly connected to the packet builder.

The IPv6 source or destination address reconstitution is a method for building IPv6 packets using IPv4 packets.

- Inverse DNS lookup on DNS server 31 and Whois lookup on server 32 are used to identify the location of the destination device and the owner (ISP / customer). Both are used to set the LOC ID field and the VPN ID field.
- Certificate request on Certificate Authority (CA) server 33 is used to validate source and / or destination IP addresses IPv4 and HOST ID for security purposes. It may be used in addition or as a replacement of DNS and Whois servers.
- A Directory server 34 contains the allowed subnets and authorized HOST IDs but not using certificates as an alternate method for

Certificate use. This server or the CA may also provide the VPN ID if more than one VPN is supported on an interface. Otherwise, the VPN ID may be defined just by the interface using a one to one mapping static table as it is done today with one physical or logical (generally a virtual circuit) corresponding to a single VPN.

- A Route Reflector RR 35 provides the PE field value for the destination address. The source address is filled with the locally attached PE identification field. RR solves the routing over the IP switching infrastructure using any kind of routing techniques which allows to provide routing for VPN groups or type of flows.
- A network management tool NM 36 provides filtering rules that can be dynamically applied if necessary. PB is the right place to put filtering rules at the network ingress point in order to avoid spoofing, for example. NM can easily identify attacks on the destination network and find from which ingress point these packets are sent and take appropriate measures.
- An ARP proxy AP 37 provides the MAC address of the sending device if this function is used for improved security and device identification. In some cases, the proxy can just be the default access router when no other routers are in between. In that case,

such a router should have been enabled to answer to ARP requests. The MAC address may also be pre-stored in DNS or Directory Server or in a Certificate. The advantage of ARP proxy or ARP request to default router is that the MAC address is dynamically validated against the source IP address of the flow. So, for example, a known MAC address may be granted even if it is not connected on the regular LAN network. An unknown MAC address from all servers may be forbidden to connect to some networks as another example of rule.

The IPv6 fields to be filled this way include the source address and the destination address. The figure shows an implementation where both PB devices have access to all servers in order to get information for both source and destination address fields. An alternate implementation is for the local Route Reflector to provide the IP address of the destination Packet builder (like PB2) when PE2 is given as the destination in order for PB1 to get destination information directly from PB2 and not from the servers. The advantage of such an implementation is the performance since each Packet builder device maintains a list of all locally accessible devices with associated parameters. Thus, in one call to the remote PB, all information may be provided. If the remote PB doesn't have this device in its cache, it will join the servers to get it. This analysis is only done once per flow (on the first packet) and the level of analysis

performed may depend on the network on which the packet should be sent to. In fact, it is less than once per flow insofar as only the source part (or the destination part if new) is processed and information is kept in a cache, which minimizes performance impact of such an implementation.

According to another embodiment, the packet builder is integrated in the PE device as illustrated in Fig. 4. In such a case, there is of course no IPv6 network as in the general embodiment illustrated in Fig. 1. Instead, the packet builder and the PE device constitute a single block, that is the block V6 PE1 41 replacing PB1 22 and PE1 24 and the block V6 PE2 replacing PB2 28 and PE2 30.

This proposed implementation is also used to perform the function on the PE itself, which is more efficient since the PE field of the address can directly be used by the PE for forwarding on the MPLS core without additional path discovery and additional level of Label like on MPLS VPN. In fact, a label may be used as a compressed header instead of transmitting the full IPv6 header, which means one label by transmission flow. The transmission of both (label followed by a full IPv6 header) would be redundant but possible. Several compatible implementations may be used in the core for packet transmission using IP switching labels. The need is to recover at the egress PE or PB the original IPv6 header on each packet.

The change of the data packet as it flows from the sending workstation to the receiving workstation is illustrated in Fig. 5A. In workstation 20, the data packet is a standard IPv4 packet format with an IPv4 header and a data payload. The packet is sent to PB1 device at step 61. The packet builder validates the IPv4 header fields and finds, thanks to the servers defined in Fig. 3, the value for the fields used to build the corresponding IPv6 header. It mainly includes the IPv6 destination address and the IPv6 source address. PB1 can perform additional filtering if defined.

The new packet is forwarded to PE1 at step 62. PE1 identifies the destination PE and forwards the packet to PE2 at step 63. PE1 can add a label such as a MPLS label for improving the packet switching in the core. PE2 receives the packet and removes the switching label if present. It then forwards the packet to PB2 at step 64 which first can again perform filtering and then remove the IPv6 header and easily rebuild the IPv4 packet insofar as the IPv6 packet header contains all the necessary fields. This IPv4 packet is then forwarded to the destination workstation 26 on step 65.

It should be noted that this process seems complex but in fact needs to be fully performed only once especially on PB1 and PB2. Further packets of the same flow may just need header fields swapping using the same field values. Therefore a swap table

can be implemented on Packet Builders to speed up the processing of swapping headers.

Rebuilding or adding packets checksums and rebuilding other IPv4 header fields are not detailed here insofar as these functions may be done using standard rules and translation mechanisms such as RFC2766 or other IETF RFCs or drafts.

If the second embodiment wherein the function of the packet builder is integrated in the PE device is used, the flow of the data packet (and the structure change) as illustrated in Fig. 5B is simplified but very similar to the data flow of the general embodiment.

CLAIMS

1. System for converting data packets based upon the IPv4 protocol into data packets based upon the IPv6 protocol to be transmitted over a multiple Virtual Private Network (VPN) infrastructure in a data transmission system comprising an IP switched network (25) to which are connected a first workstation (20) via a first Provider Edge (PE) device (24) and a second workstation (26) via a second PE device (30), said workstations operating with the IPv4 protocol whereas said IP switched network operates with the IPv6 protocol;

said system being characterized in that it comprises a first packet builder (22) between said first workstation and said IP switched network for converting any data packet based upon the IPv4 protocol received from said first workstation and transmitted to said IP switched network into the IPv6 protocol and reciprocally, and a second packet builder (28) between said IP switched network and said second workstation for converting any data packet based upon the IPv6 protocol received from said network and transmitted to said second workstation into the IPv4 protocol and reciprocally; and

said system further comprising at least a server (31 to 37) accessed by each of said first packet builder (22) or said second packet builder (28) for providing thereto the information used to convert a data packet based upon the IPv4 protocol into a data packet based upon the IPv6 protocol, wherein any data packet which has been converted into the IPv6 protocol by one of said first packet builder (22) or said second packet builder (28)

includes a packet header containing a PE field corresponding to the first PE device (24) or the second PE device (30) to be used either as source PE or destination PE and a VPN identification field further to an IPv4 address which was in said packet header before the conversion.

2. System according to claim 1, wherein any data packet which has been converted into the IPv6 protocol by one of said packet builders (22, 28) further includes a location (LOC) identification field referring to a geographic location of a site and a HOST identification field referring to the first workstation (20) or the second workstation (26) that has transmitted said data packet.

3. System according to claim 1, wherein the server (31 to 37) providing the information used to convert the data packet based upon the IPv4 protocol into the data packet based upon the IPv6 protocol can be accessed by each of said first and second packet builders through a virtual network (39).

4. System according to claim 3, wherein said server can be one of several servers among a Domain Name Server (31), a Whois server (32), a Certificate Authority server (33), a Directory server (34), a Route Reflector (35), a Network Management tool (36) and a Proxy (37).

5. System according to claim 1, wherein said first packet builder (22) or said second packet builder (28) is connected to said first PE device (24) or said second PE device (30) by a network based upon the IPv6 protocol (23 or 29).

6. System according to claim 1, wherein said first packet builder (22) or said second packet builder (28) is integrated into said first PE device (24) or said second PE device (30).

7. Method for converting data packets based upon the IPv4 protocol into data packets based upon the IPv6 protocol to be transmitted over a Virtual Private Network (VPN) in a data transmission system comprising an IP switched network (25) to which are connected a first workstation (20) via a first Provider Edge (PE) device (24) and a second workstation (26) via a second PE device (30), said workstations operating with the IPv4 protocol whereas said IP switched network operates with the IPv6 protocol,

said method being characterized by the steps comprising:

- converting any data packet based upon the IPv4 protocol into a data packet based upon the IPv6 protocol before transmitting it to said IP switched network using information provided by an external server, and

- converting any data packet based upon the IPv6 protocol provided by said IP switched network into a data packet based upon the IPv4 protocol before transmitting it to said first or second workstation, wherein any data packet which has been converted into the IPv6 protocol by one of said first packet builder (22) or said second packet builder (28) includes a packet header containing a PE field corresponding to the first PE device (24) or the second PE device (30) to be used

either as source PE or destination PE and a VPN identification field further to an IPv4 address which was in said packet header before the conversion.

8. Method according to claim 7, wherein any data packet which has been converted into the IPv6 protocol by one of said packet builders (22, 28) further includes a location (LOC) identification field referring to a geographic location of a site and a HOST identification field referring to the first workstation (20) or the second workstation (26) that has transmitted said data packet.

9. Method according to claim 8, wherein a server providing information for said conversion is one of several servers comprising a Domain Name Server (31), a Whois server (32), a Certificate Authority server (33), a Directory server (34), a Route Reflector (35), a Network Management tool (36) and a Proxy (37).

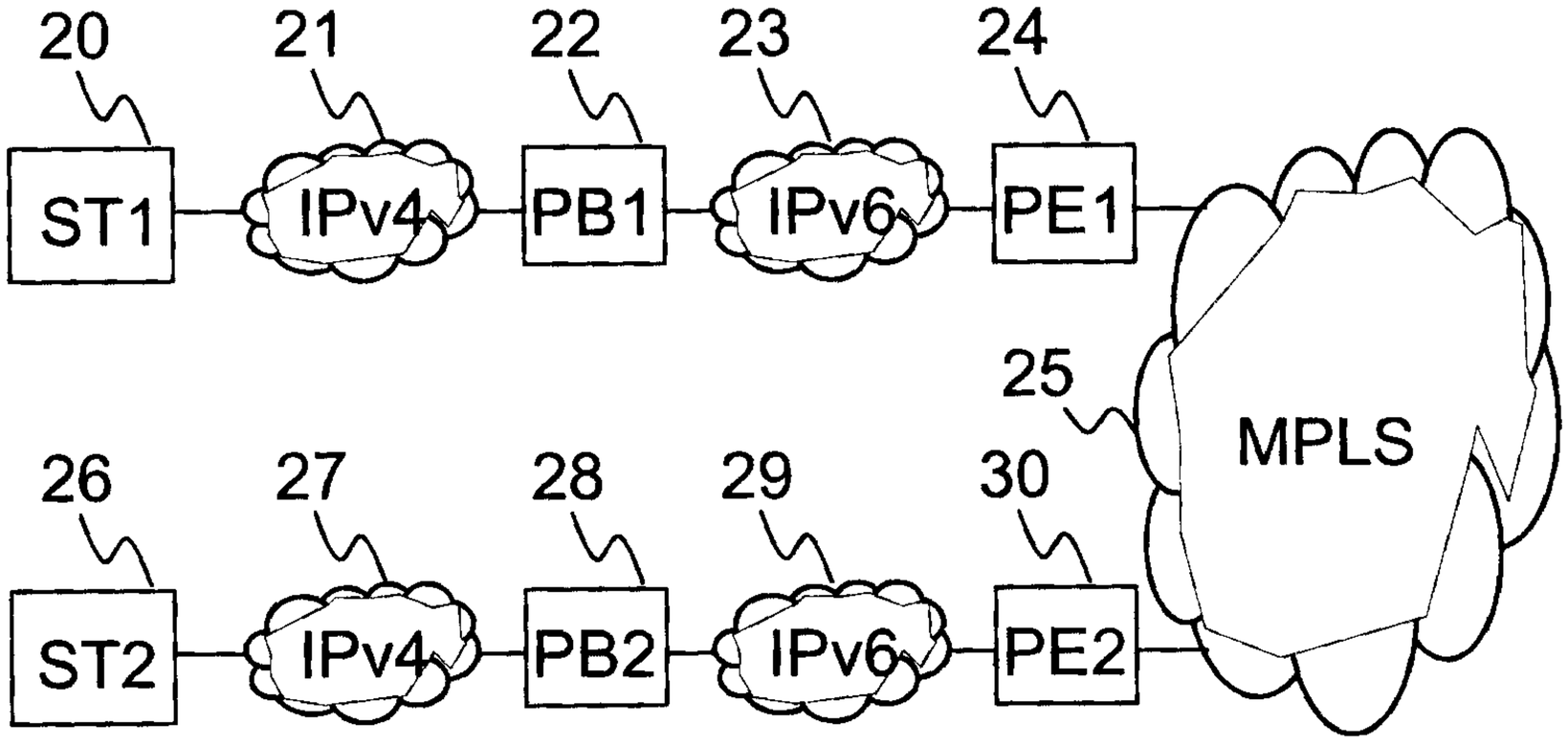


FIG. 1

3	13	8	24	16	64
FP	TLA ID	RES	NLA ID	SLA ID	INTERFACE ID

FIG. 2A

3	13	8	24	16	32	32
FP	PRO ID	PE	VPN ID	LOC ID	HOST ID	IPv4 ADD

FIG. 2B

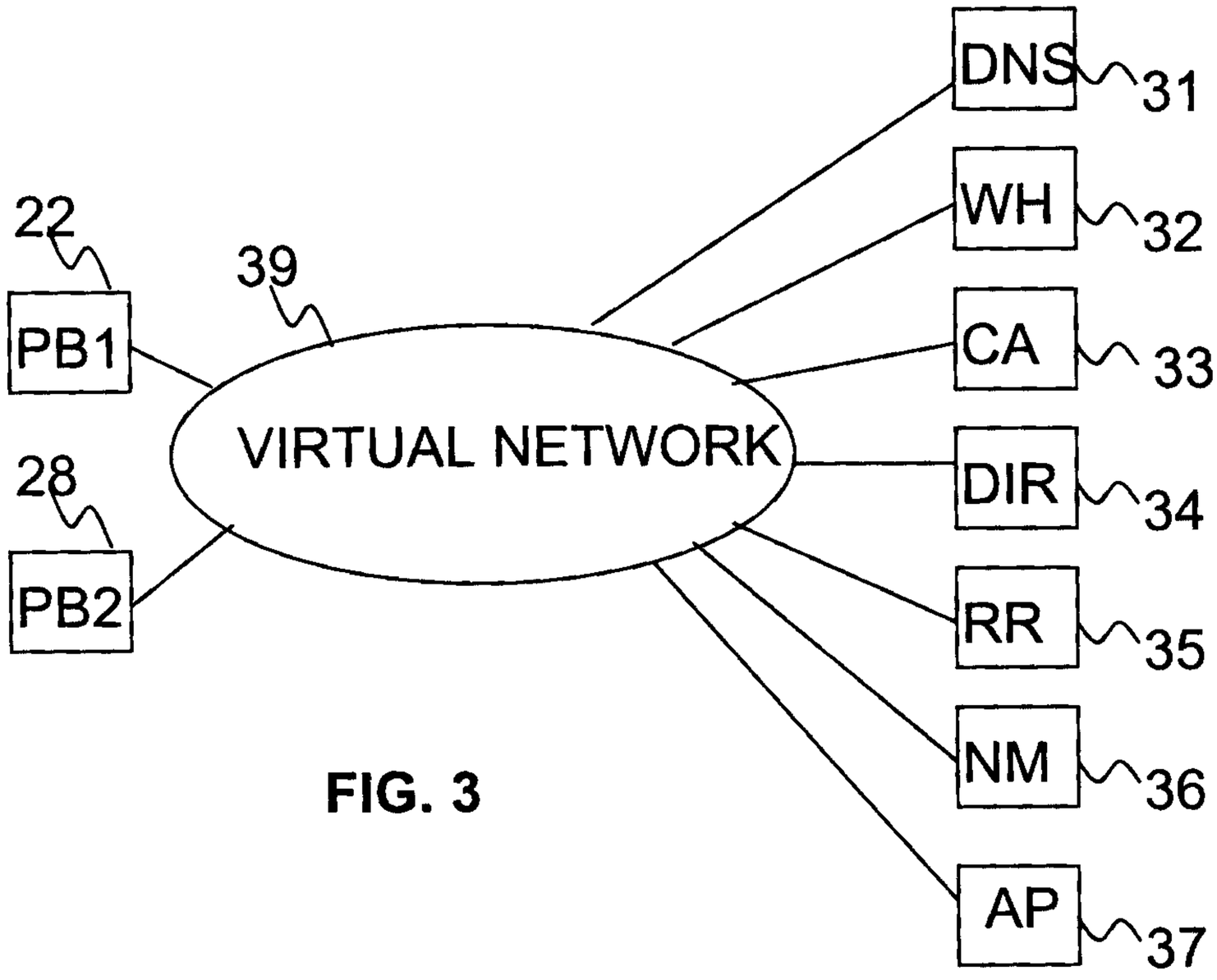

DNS
31
WH
32
CA
33
DIR
34
RR
35
NM
36
AP
37
22
PB1
39
VIRTUAL NETWORK
28
PB2

FIG. 3

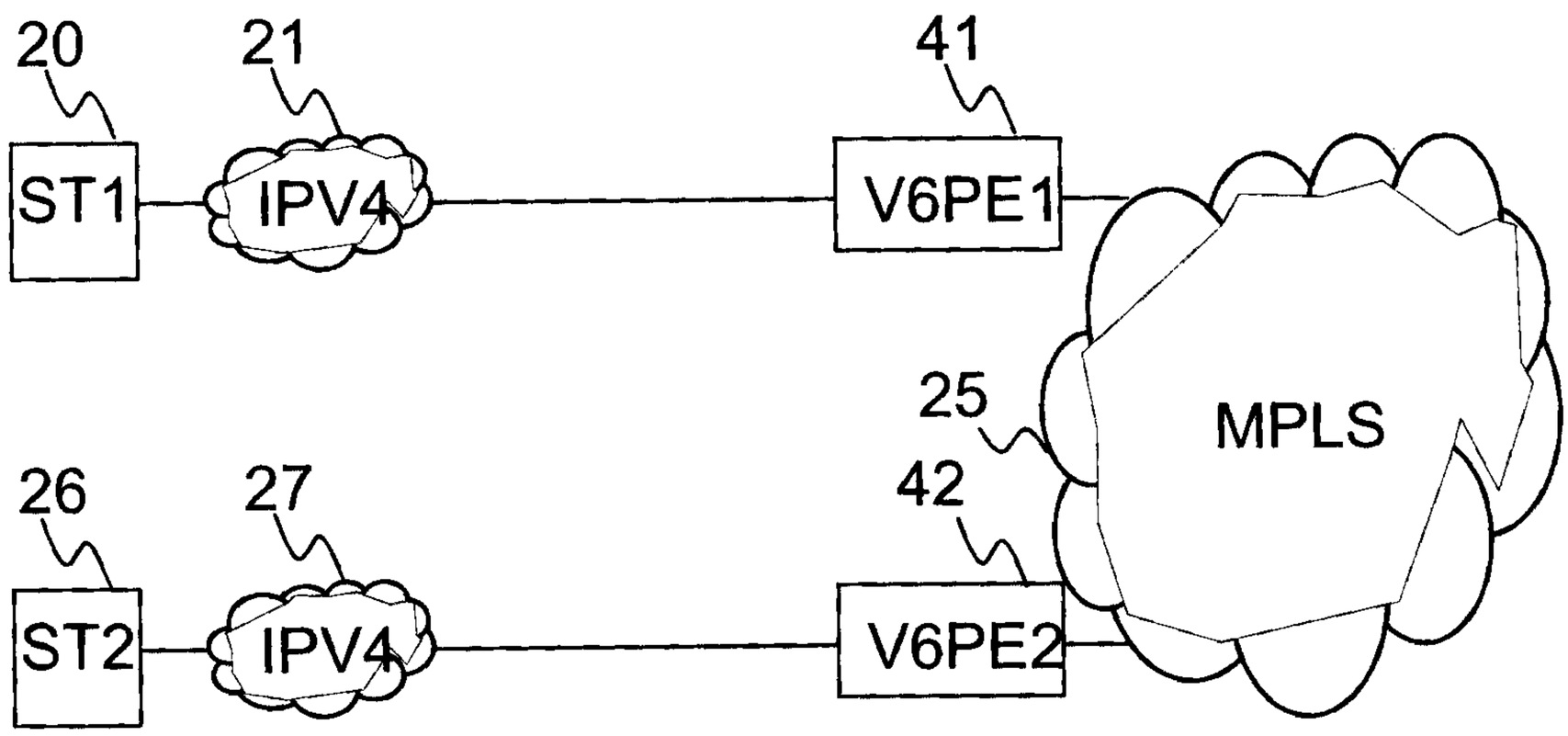

20
ST1
21
IPV4
41
V6PE1
MPLS
25
26
ST2
27
IPV4
42
V6PE2

FIG. 4

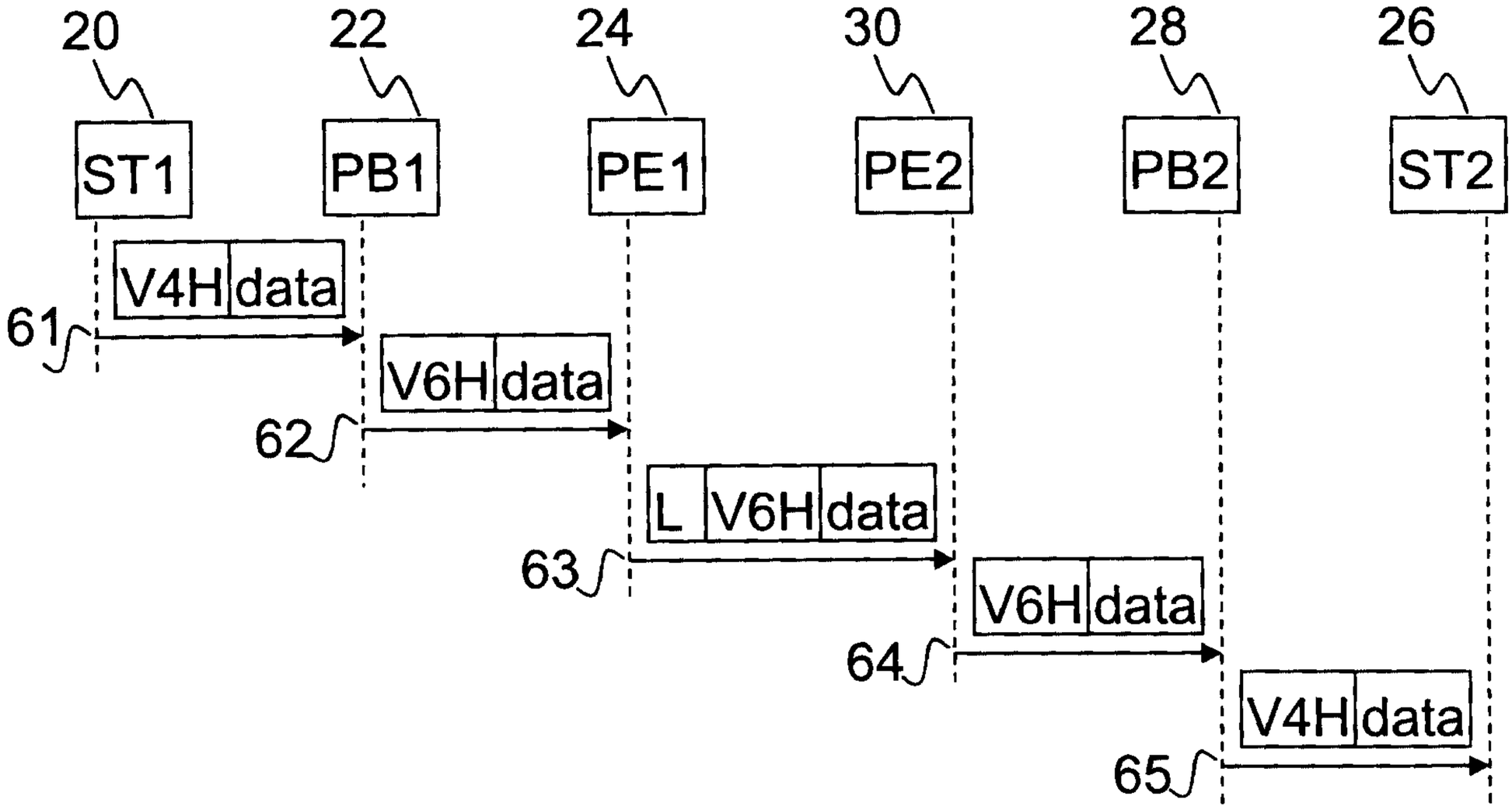

20
22
24
30
28
26
ST1
PB1
PE1
PE2
PB2
ST2
V4H data
61
V6H data
62
L V6H data
63
V6H data
64
V4H data
65

FIG. 5A

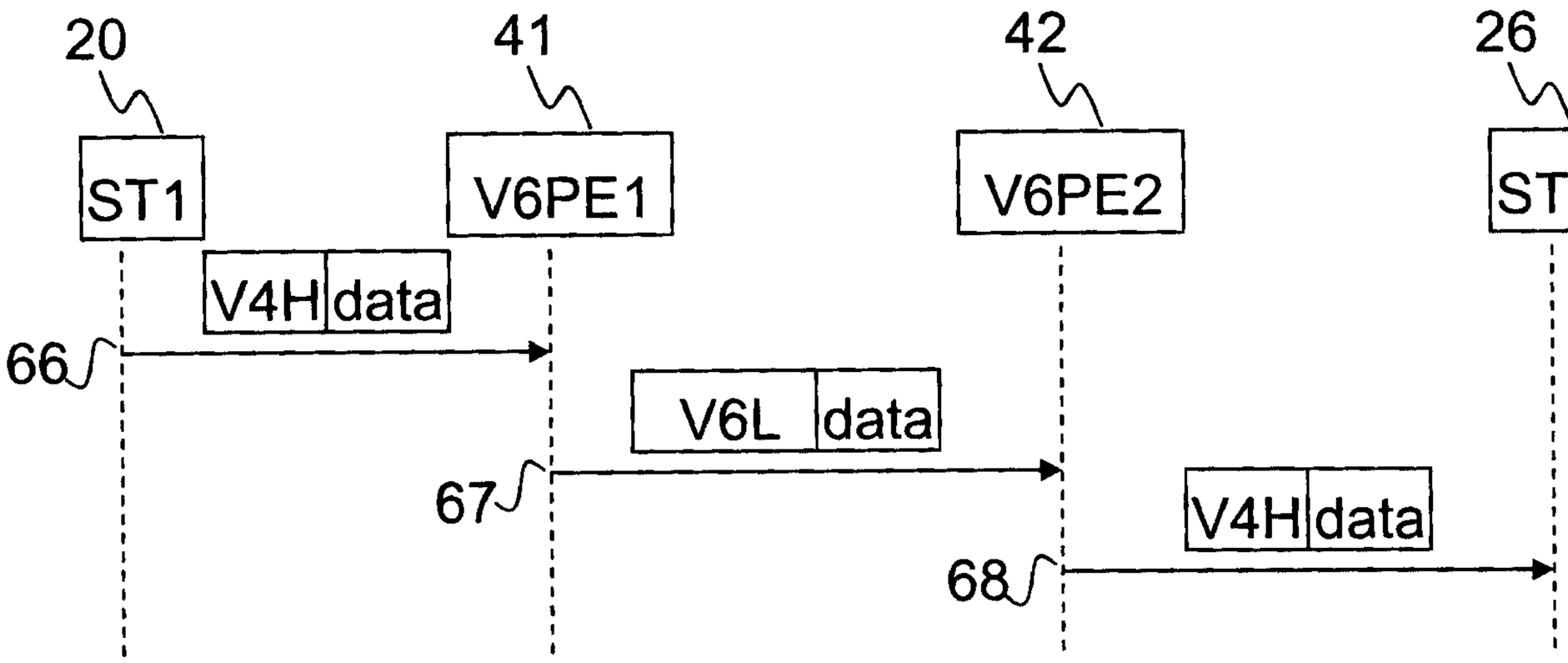
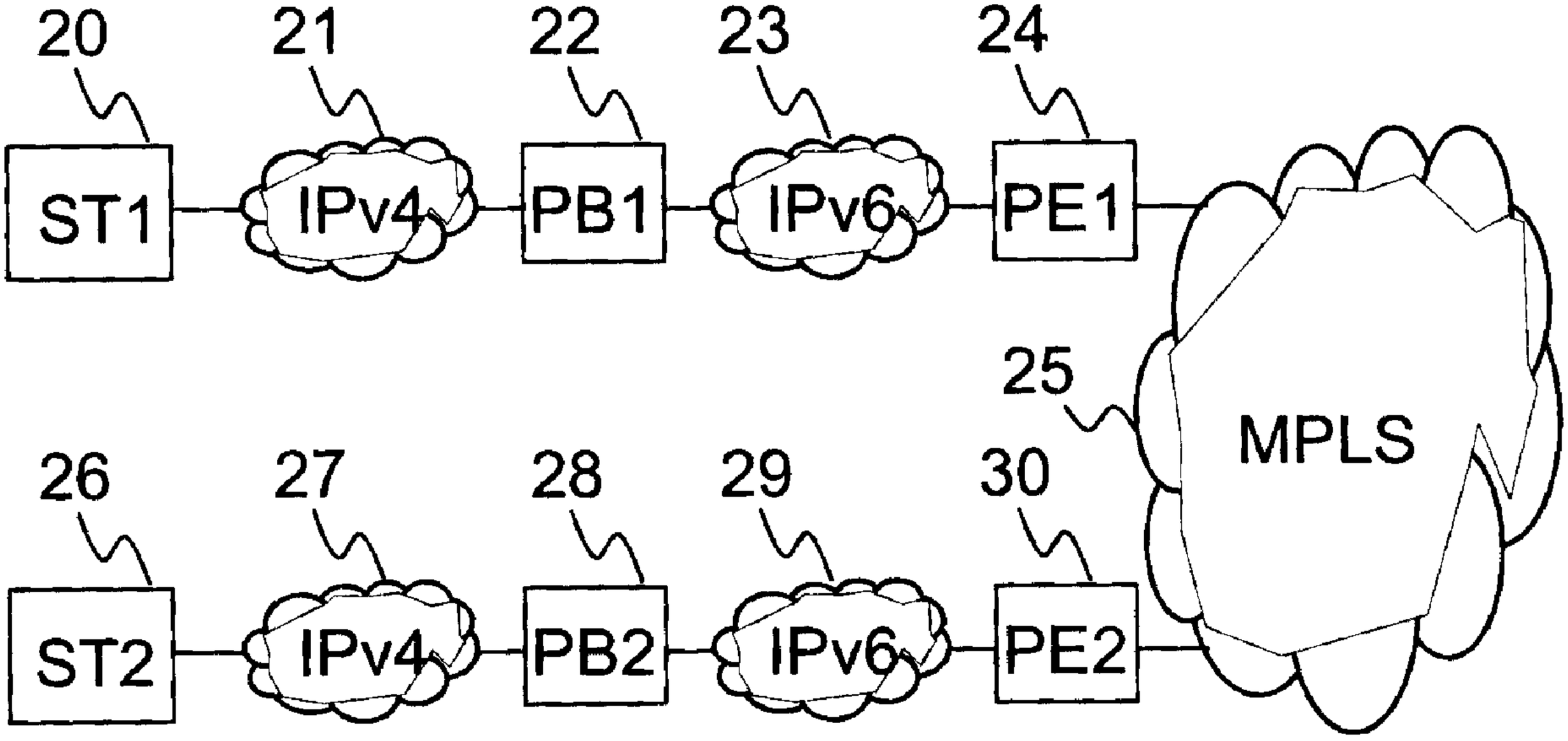

20
41
42
26
ST1
V6PE1
V6PE2
ST2
V4H data
66
V6L data
67
V4H data
68

FIG. 5B


20
21
22
23
24
ST1
IPv4
PB1
IPv6
PE1
25
MPLS
26
27
28
29
30
ST2
IPv4
PB2
IPv6
PE2