

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
22 February 2007 (22.02.2007)

PCT

(10) International Publication Number
WO 2007/021516 A1

- (51) International Patent Classification:
H04L 12/46 (2006.01)
- (21) International Application Number:
PCT/US2006/029659
- (22) International Filing Date: 31 July 2006 (31.07.2006)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
11/202,802 11 August 2005 (11.08.2005) US
- (71) Applicant (for all designated States except US): CISCO TECHNOLOGY, INC. [US/US]; 170 W. Tasman Drive, San Jose, CA 95134-1706 (US).
- (72) Inventor: PANG, Tak, Ming; 34227 Tony Terrace, Fremont, CA 94555 (US).
- (74) Agent: SPOLYAR, Mark, James; Law Office Of Mark J. Spolyar, 2200 Cesar Chavez Street, Suite 8, San Francisco, CA 94124 (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, 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, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, 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).

Declarations under Rule 4.17:

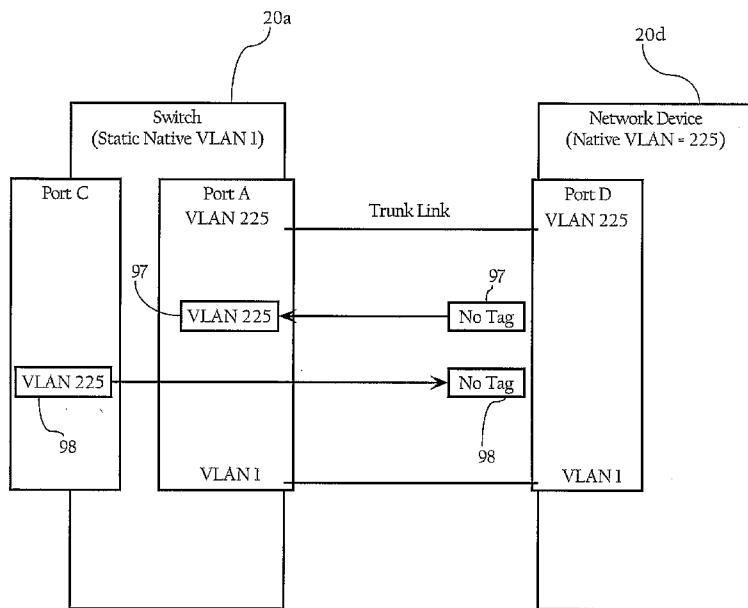
- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))
- as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))

Published:

- with international search report
- before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments

[Continued on next page]

(54) Title: SPANNING TREEBPOU PROCESSING METHOD AND APPARATUS FACILITATING INTEGRATION OF DIFFERENT NATIVE VLAN CONFIGURATIONS



(57) Abstract: Methods and apparatuses directed to enhancing the interoperability of network devices with static native virtual LAN (VLAN) configurations with other network devices where the native VLAN is a configurable parameter. In some implementations, the network devices having configurable native VLANs conditionally add VLAN tags to common Spanning Tree Protocol (STP) Bridge Protocol Data Units (BPDUs) transmitted to network devices where the native VLAN is static and strip any VLAN tags from the common STP BPDUs that are received.

WO 2007/021516 A1



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.

Spanning Tree BPDU Processing Method and System Facilitating
Integration of Different Native VLAN Configurations

FIELD OF THE INVENTION

[0001] The present invention relates to computer networks and, more particularly, to Spanning Tree Protocols in Link Layer networks.

BACKGROUND OF THE INVENTION

[0002] Spanning Tree Protocol is a link management protocol that provides path redundancy while preventing undesirable bridging loops in the network. For an Ethernet Layer-2 network to function properly, only one active path can exist between two stations. Multiple active paths between stations cause traffic to loop in the network. If a bridging loop exists in the network topology, it can cause broadcast and multicast frames to be duplicated, creating a traffic storm. When bridging loops occur, a bridge may see the same stations appearing on both of its interfaces. Additionally, switches may see the same stations appearing on different ports at different times. This condition confuses the frame forwarding logic. To provide path redundancy, Spanning Tree Protocol defines a tree that spans all devices in the Layer-2 network. Spanning-Tree Protocol forces all redundant data paths into a standby (blocked) state. If the network topology changes, or if the network Spanning Tree Protocol configuration changes, the spanning-tree algorithm reconfigures the spanning-tree topology and reestablishes the link by activating the standby path or putting active links into standby state. The IEEE 802.1D Standard, entitled "Media Access Control (MAC) Bridges," defines a Spanning Tree Protocol for use in local area networks (LANs).

[0003] Bridges in an extended LAN participating in Spanning Tree Protocol gather information on other bridges in the network through observation and forwarding of STP messages. These STP messages are so-called bridge protocol data units

(BPDUs). This results in selection of a unique root bridge for the stable spanning tree network topology and the removal of redundant path in the switched network by placing redundant switch ports in a blocked state. Spanning Tree Protocol operation is transparent to end stations, which are unaware of the network topology of the LAN segment to which they are being connected. Generally speaking, the root bridge generates configuration BPDUs, which other devices process and multicast out at STP-enabled ports.

[0004] A virtual LAN (VLAN) is a switched network that is logically segmented by one or more criteria without regard to the physical location of the end stations. For example, end stations might be grouped according to company departments, such as engineering or accounting rather than their physical locations. Multiple VLANs can be defined over the same network infrastructure to segment a LAN to different broadcast domains. The IEEE 802.1Q standard, entitled "Virtual Bridged Local Area Networks," sets forth a VLAN implementation in common use today.

[0005] To implement a VLAN, ports on the same or different bridges/switches are logically grouped so that traffic is confined only to members of that group. This feature restricts broadcast, multicast, and unicast flooding traffic only to ports included in a certain VLAN. For VLANs to span multiple switches, trunk ports have to be configured on the switches to establish a trunk link to connect the switches. A trunk link carries traffic for all VLANs by identifying the originating VLAN as the frame is carried between the switches. To this end, VLAN implementations employ frame tagging—that is, frames received from an end station connected to an access port associated with a given VLAN are tagged with an identifier (VLAN ID) corresponding to that VLAN prior to transmission across a trunk link. Frame tagging allows switches and network devices to determine the VLAN to which different frames belong when the frames are transmitted across trunk links. The frames can be tagged with VLAN information according to the IEEE 802.1Q standard, or other protocols (such as the Inter-Switch Link (ISL), a proprietary trunking mechanism developed by Cisco Systems, Inc.).

[0006] The IEEE 802.1Q standard also defines a native VLAN. According to this standard, a trunk port does not tag frames that are forwarded to any port that corresponds to the native VLAN. Some vendors allow the native VLAN to be a configurable parameter for their switching/bridging devices. For example, assume for didactic purposes that a network device supports configuration of up to 64 VLANs. A network administrator can configure multiple VLANs (e.g., 1, 2, 3, 4, etc.) and assign VLAN 2 as the native VLAN. According to this example, if the network device receives a frame on a trunk port without a tag, the network device assumes that the frame belongs to the native VLAN 2 and does not tag the frame. In some network switches (such as certain Ethernet Switches offered by Cisco Systems, Inc.(r) of San Jose, California), the native VLAN is a fixed parameter that cannot be changed. For example, the native VLAN in many Cisco switches is fixed to VLAN 1.

[0007] The fixed nature of the native VLAN on such switches can create certain interoperability problems with other network devices where the native VLAN is configurable and has not been set to VLAN 1. For didactic purposes, Figure 6 illustrates a trunk port connecting switch 20a (having a static native VLAN 1) and network device 20d (configured with a native VLAN 225). As discussed above, a pair of trunk ports forms a connection (called a trunk link) between two network devices over which traffic from a plurality of VLANs are transmitted. Given the native VLAN configurations discussed above, without some conversion mechanism, untagged frames have different significance between switches 20a and network device 20d, presenting certain interoperability issues. As Figure 6 illustrates, switch 20a, however, can be configured with knowledge that the native VLAN for network device 20d has been set to VLAN 225. With this knowledge, switch 20a can be configured to do the following to facilitate interoperation between the two network devices. For example, when switch 20a receives a frame 97 bearing no tag over the trunk port A from network device 20d, switch 20a tags the frame with a tag identifying VLAN 225 prior to processing. Still further, when switch 20a

receives a frame 98 destined for device 20d and tagged with VLAN 225 from port C, it removes the tag prior to forwarding the frame to port D of network device 20d. Similarly, switch 20a tags unmarked frames with a VLAN 1 tag at port A prior to transmission to device 20d. As discussed in more detail below, the fixed nature of the native VLAN setting on many switches can create certain interoperability problems with other network devices relative to operation of spanning tree protocols and the processing of Bridge Protocol Data Units (BPDUs).

[0008] As to the Spanning Tree Protocol, the IEEE 802.1D standard defines a unique spanning tree instance for the entire Layer 2 network. This single spanning tree instance runs on the native VLAN, which can be used to define the paths for all VLANs. This spanning tree instance is also required for Per-VLAN Spanning Tree Plus Protocol (PVST+) VLANs in the same network. The PVST+ Protocol, a proprietary spanning tree protocol developed by Cisco Systems, Inc., allows an instance of STP to run on each VLAN. With PVST+, a root bridge and a unique STP topology is selected and configured for each VLAN. For interoperability with 802.1Q bridges/switches, the PVST+ implementation, however, requires an IEEE 802.1D common spanning tree instance for all bridges/switches. This is accomplished with bridges/switches that forward and process PVST+ BPDUs on each VLAN, as well as BPDUs associated with the common spanning tree instance on the native VLAN, all originated from the STP root bridges. Typically, the common spanning tree instance operates according to the IEEE 802.1D standard. STP root bridges for all VLANs can be physically on the same or different bridges/switches. By configuring a different root priority or port cost on different devices on the VLANs, a network administrator can decide and control at which bridges/switches and ports redundant links are blocked.

[0009] Using PVST+, bridges/switches can send or forward PVST+ BPDUs as tagged frames using a pre-determined multicast address as the destination. The IEEE 802.1D BPDUs, according to the protocol, are sent without tags. In both switches 20a and device 20d, IEEE 802.1D BPDUs are either sent or received on

their respective native VLANs and do not include an IEEE 802.1Q tag. The VLAN tagging operations implemented by switch 20a on received frames, however, are also applied to the IEEE 802.1D BPDUs, causing them not to be processed. That is, as to BPDUs sourced from network device 20d, an IEEE 802.1D BPDU, which by protocol includes no tag, is tagged with VLAN 225, preventing the IEEE 802.1D BPDU from making it to higher layers of the protocol stack of switch 20a for processing. As a result, both IEEE 802.1D STP and PVST+ do not function properly to block redundant paths. This circumstance can cause flipping of entries in frame forwarding tables if there is any loop topology in the network infrastructure, and can cause affected parts of the network to become inaccessible. [0010] In light of the foregoing, a need exists in the art for methods, apparatuses and systems directed to facilitating the interoperability between network devices having static, unchangeable native VLANs with other network devices where the native VLANs are configurable. Embodiments of the present invention substantially fulfill this need.

DESCRIPTION OF THE DRAWINGS

[0011] Figure 1 is a functional block diagram illustrating a network environment in which embodiments of the present invention may operate.

[0012] Figure 2 is a functional block diagram illustrating another network environment in which embodiments of the present invention may operate.

[0013] Figure 3 is a flow chart diagram setting forth a method according to one embodiment of the invention.

[0014] Figure 4 is a flow chart diagram providing a method according to an embodiment of the present invention.

[0015] Figure 5 is a functional block diagram showing the components of a network device according to one implementation of the present invention.

[0016] Figure 6 is a functional block diagram illustrating the addition and removal of VLAN tags from frames transmitted between network devices.

DESCRIPTION OF PREFERRED EMBODIMENT(S)

[0017] Figure 1 illustrates a network environment in which embodiments of the present invention may operate. In a specific embodiment of the present invention, the network environment may include switches 20a, 20b, 20c, 20d (collectively referred to as switches 20) operably connected to each other as shown. As Figure 1 illustrates, end stations (such as servers 25 and client computers 26) are also connected to the switches 20. In one implementation, switches 20 are Ethernet switches implementing a Local Area Network (LAN) or LAN segment. Still further, router 45 and network 44, which may be a LAN, LAN segment, or a Wide Area Network (WAN), allow for the transmission of data between end stations connected to switches 20 and remote hosts reachable over network 44.

[0018] Figure 2 illustrates another network environment in which embodiments of the present invention can operate. The network illustrated in Figure 2 is similar to the network illustrated in Figure 1. However, the illustrated network environment includes wireless switch 22 operably connected to switch 20a and to wireless access points 50. The wireless access points 50 are enabled to wirelessly communicate with remote client devices or mobile stations (not shown). In one implementation, the wireless access points 50 implement the wireless network protocol specified in the IEEE 802.11 specification. The wireless access points 50 may be autonomous or so-called "fat" access points, or light-weight access points operating in connection with a wireless switch 22, as disclosed in U.S. patent application Ser. No. 10/407,584, now U.S. Patent No. _____. The wireless access points 50 are typically connected to the network via Ethernet links; however, other link layer connection protocols or communication means can be employed. In one implementation, a wireless access point 50 comprises a processor, a memory, a network interface (e.g., an Ethernet network interface) for communication with the LAN, a wireless network interface (e.g., an IEEE 802.11 WLAN interface) for communication with one or more mobile stations, a system bus interconnecting

these components, as well as software modules (including DHCP clients, CDP modules, access point modules, SNMP functionality, etc.) and device drivers (e.g., network and WLAN interface drivers) stored in persistent memory (e.g., a hard disk drive, flash memory, etc.). At start up, these software components are loaded into memory and then accessed and executed by processor.

[0019] Figure 5 illustrates the basic hardware components of switches 20 according to one implementation of the invention. As Figure 5 provides, switches 20 each comprise a processor 510, system memory 512, persistent memory 518 (e.g., flash memory or a hard disk drive), a switch fabric 504 connected to a plurality of ports 502, a system bus 508 interconnecting these components, as one more software modules (loadable into system memory 512) directed to network switching functions (e.g., switch fabric configuration, BPDU processing and the like). In one implementation, ports 502 are Ethernet interfaces. The switches 20 may optionally include a console port 516 allowing for administrative access for such purposes as configuration and diagnostics. In one implementation, switches 20 are operative to implement the spanning tree protocol defined in the IEEE 802.1D standard and the Per-VLAN Spanning Tree Plus Protocol (PVST+), described above. For example, a given switch 20 is operative to receive IEEE 802.1D and PVST+ BPDUs on STP-enabled ports, process them, and multicast the BPDUs to devices connected to other STP-enabled ports of the switch 20. In addition, wireless switch 22, in one implementation, includes the same or similar hardware components illustrated in Figure 5; however, it also includes one or more software modules directed to managing the access points 50.

[0020] The present invention provides, in one implementation, methods, apparatuses and systems directed to enhancing the interoperability of network devices with static native VLAN configurations with other network devices where the native VLAN is a configurable parameter. As described in more detail below, in some implementations, the network devices having configurable native VLANs conditionally add VLAN tags to common STP BPDUs transmitted to network

devices where the native VLAN is static and strip any VLAN tags from the common STP BPDUs that are received. For didactic purposes, the switches 20a, 20b and 20c are switches where the native VLAN is static, and set to a value of VLAN 1. Of course, the present invention can be applied in situations where the static native VLAN is set to any given VLAN identifier. Switches 20a, 20b, 20c, therefore, forward IEEE 802.1D BPDUs untagged for the common Spanning Tree instance on VLAN 1. PVST+ BPDUs are sent with tags for all VLANs other than VLAN 1. Furthermore, for didactic purposes, switch 20d (in Figure 1) and wireless switch 22 (in Figure 2) are network devices where the native VLAN is configurable, and set to VLAN 225. Furthermore, for didactic purposes, assume that switches 20a, 20b, and 20c have been configured with knowledge that switch 20d has the native VLAN set to 225, and to conditionally add or strip tags from the frames transmitted over the respective trunk links between them, as discussed above.

[0021] Although embodiments of the present invention are illustrated as operating in connection with switches, the present invention can be implemented in connection with other network devices, such as wireless switch 22, bridges and wireless access points, which implement/participate in spanning tree protocols. Furthermore, the invention can operate in a variety of network environments. For example, the wireless access points 50 can be connected directly to switch 20a. In other implementations, the network environment may also include Layer 2 bridges connecting different LAN segments of different media types or Layer 2 protocols together. Accordingly, the network environments illustrated above are for didactic purposes only. Still further, for didactic purposes, common STP BPDUs refer to bridge protocol data units transmitted as part of the Spanning Tree Protocol instance deployed across all network devices in the network domain. For example, common STP BPDUs may be the BPDUs transmitted according to the spanning tree protocol defined in the IEEE 802.1D standard. In addition, the term PVST BPDUs refer to messages transmitted as part of the per-VLAN spanning tree protocol implementation. In one implementation, the PVST BPDUs are the BPDUs

transmitted according to the Per-VLAN Spanning Tree Plus Protocol (PVST+) defined by Cisco Systems, Inc.

[0022] Figure 3 illustrates a process flow, according to one implementation of the invention, implemented by switch 20d. As Figure 3 illustrates, the native VLAN configured on switch 20d is not VLAN 1. In addition, switch 20d is configured to conditionally add an 802.1Q tag identifying VLAN 1 to IEEE 802.1D BPDUs prior to transmission to switch 20a or any other network device having a native VLAN set to VLAN 1. As Figure 3 illustrates, when switch 20d reads a frame from a transmit queue (102), it inspects the frame to determine whether it is a common STP BPDU (104). Identification of the common STP BPDU (such as an IEEE 802.1D BPDU) is based on inspection of one or more attributes of the frame. If the frame is a common STP BPDU, the switch 20d then tags the common STP BPDU with an 802.1Q tag identifying VLAN 1 (110), if the native VLAN configured on switch 20d is not VLAN 1 (106) and the receiving device is a network device including a static native VLAN 1 (108). The process flow shown above is implemented by switch 20d, in one implementation, on a per-port basis. For example, when switch 20d receives a common STP BPDU from switch 20c on a given port, it may process the common STP BPDU and multicast the common STP BPDU on all STP-enabled ports. Referring to Figure 1, switch 20d may then forward and transmit the common STP BPDU to switch 20a and 20b

[0023] . If switch 20a has the native VLAN set statically to VLAN 1 and switch 20d is configured to native VLAN 225, then the common STP BPDUs transmitted from each corresponding port 502 of switch 20d are tagged with an appropriate 802.1Q tag prior to transmission.

[0024] Figure 4 provides a process flow, according to one implementation of the present invention, directed to a process for conditionally removing 802.1Q tags added to common STP BPDUs received from another network device. Prior to the process illustrated in Figure 4, switch 20d has received a common STP BPDU on a given STP-enabled port connected to switch 20c (If switch 20d is the STP root

bridge of the common STP instance among all the devices on the native VLAN, it generates this BPDU itself) and has buffered the common STP BPDU on a queue for subsequent processing. As Figure 4 illustrates, when switch 20d reads a frame from the receive queue (202), it determines whether the frame is a common STP BPDU (204). If it is not, switch 20d applies normal processing to the frame. If the frame is a common STP BPDU, however, switch 20d removes any VLAN tags inserted into the frame (if any) (206, 208) prior to processing the common STP BPDU and forwarding the BPDU to the output queue(s) corresponding to the STP-enable ports of switch 20d (210). A separate process reads the frames from the output queue(s) and transmits them out the corresponding ports. In addition, the process flow illustrated in Figure 3 may be applied to conditionally tag the outgoing common STP BPDUs (for example, the common STP BPDU forwarded to switch 20a) depending on the configuration of the destination network device. Still further, the VLAN tagging and stripping processes set forth above can be applied regardless of whether switch 20d is the root STP device for the common spanning tree instance or for any of the per-VLAN STP instances.

[0025] The foregoing description of the embodiments of the invention has been presented for the purpose of illustration and description only. It is not intended to be exhaustive or to limit the invention to the specific forms disclosed. Many modifications and variations are possible in light of the above teaching. For example, the present invention may be employed with other frame marking schemes such as the use of encapsulating headers. Still further, although the embodiments described above involve Ethernet networks, the present invention can be used in connection with other link layer protocols. It is intended that the scope of the invention be limited by the claims appended hereto, and not by the detailed description.

CLAIMS

What is claimed is:

1. In a first network device operably connected via a trunk link to a second network device, wherein the first and second network devices are operative to implement a plurality of virtual LANs (VLANs) including a common spanning tree protocol instance and a plurality of per-VLAN spanning tree protocol instances; wherein the common spanning tree protocol instance is implemented across respective native VLANs on the first and second network devices, wherein the first network device supports a configurable native VLAN, wherein the second network device includes a fixed native VLAN, wherein the fixed native VLAN is different from the configurable native VLAN, and wherein the second network device is operative to receive untagged frames from the first network device and add tags to the frames corresponding to the configurable native VLAN prior to processing thereof, and to remove tags identifying the configurable native VLAN of the first device prior to transmission to the first network device, a method comprising
 - receiving, at the first network device, a common spanning tree bridge protocol data unit (BPDU) transmitted from a network device;
 - if the common spanning tree BPDU includes a VLAN identifier, removing the VLAN identifier from the common spanning tree BPDU prior to processing thereof; and
 - adding, prior to transmission of a common spanning tree BPDU to the second network device, a VLAN identifier corresponding to the fixed native VLAN of the second network device.
2. The method of claim 1 wherein the VLAN identifier is embedded in a tag added to the common spanning tree BPDU.

3. The method of claim 1 wherein the VLAN identifier is embedded in a header encapsulating the common spanning tree BPDU.
4. The method of claim 1 further comprising processing the common spanning tree BPDU and forwarding the common spanning tree BPDU to at least one network device.
5. The method of claim 1 wherein the common spanning tree BPDU is an IEEE 802.1D BPDU.
6. An apparatus operable in a network environment including a first network device, wherein the first network device is operative to implement a plurality of virtual LANs (VLANs) including a common spanning tree protocol instance and a plurality of per-VLAN spanning tree protocol instances; wherein the common spanning tree protocol instance is implemented on a native VLAN on the first network device, wherein the native VLAN on the first network device is a fixed parameter, wherein the first network device is operative to receive frames from a second network device having a native VLAN set to a value different than the fixed native VLAN, add tags to received frames corresponding to the native VLAN of the second network device prior to processing thereof, and to receive frames to be forwarded to the second network device and remove tags from the frames that correspond to the native VLAN of the second network device prior to transmission, the apparatus comprising
 - at least one port;
 - a processor;
 - a memory;
 - an application physically stored in the memory, comprising instructions operable to cause the processor and the apparatus to

implement, in connection with at least the first network device, a plurality of virtual LANs (VLANs) including a common spanning tree protocol instance and a plurality of per-VLAN spanning tree protocol instances, wherein the common spanning tree protocol is implemented on a native VLAN different from the native VLAN of the first network device;

receive a common spanning tree bridge protocol data unit (BPDU) transmitted from a network device;

remove, if the network bridge protocol data unit includes a VLAN identifier, the VLAN identifier from common spanning tree BPDU prior to processing thereof; and

add, prior to transmission of a common spanning tree BPDU to the first network device, a VLAN identifier corresponding to the fixed native VLAN of the first network device.

7. The apparatus of claim 6 further comprising a switch fabric operably connected to the at least one port.
8. The apparatus of claim 6 wherein the VLAN identifier is embedded in a tag added to the common spanning tree BPDU.
9. The apparatus of claim 6 wherein the VLAN identifier is embedded in a header encapsulating the common spanning tree BPDU.
10. The apparatus of claim 6 wherein the application further comprises instructions operative to cause the processor and the apparatus to process the common spanning tree BPDU and forward the common spanning tree BPDU to at least one network device.

11. The apparatus of claim 6 wherein the common spanning tree BPDU is an IEEE 802.1D BPDU.

12. An apparatus operable in a network environment including a first network device, wherein the first network device is operative to implement a plurality of virtual LANs (VLANs) including a common spanning tree protocol instance and a plurality of per-VLAN spanning tree protocol instances; wherein the common spanning tree protocol instance is implemented on a native VLAN on the first network device, wherein the native VLAN on the first network device is a fixed parameter, wherein the first network device is operative to receive frames from a second network device having a native VLAN set to a value different than the fixed native VLAN, add tags to received frames corresponding to the native VLAN of the second network device prior to processing thereof, and to receive frames to be forwarded to the second network device and remove tags from the frames that correspond to the native VLAN of the second network device prior to transmission, the apparatus comprising

means for enabling, in connection with at least the first network device, a plurality of virtual LANs (VLANs) including a common spanning tree protocol instance and a plurality of per-VLAN spanning tree protocol instances, wherein the common spanning tree protocol is implemented on a native VLAN different from the native VLAN of the first network device;

means for receiving a common spanning tree bridge protocol data unit (BPDU) transmitted from a network device;

means for removing, if the network bridge protocol data unit includes a VLAN identifier, the VLAN identifier from common spanning tree BPDU prior to processing thereof; and

means for adding, prior to transmission of a common spanning tree BPDU to the first network device, a VLAN identifier corresponding to the fixed native VLAN of the first network device.

13. The apparatus of claim 12 further comprising a switch fabric operably connected to the at least one port.
14. The apparatus of claim 12 wherein the VLAN identifier is embedded in a tag added to the common spanning tree BPDU.
15. The apparatus of claim 12 wherein the VLAN identifier is embedded in a header encapsulating the common spanning tree BPDU.
16. The apparatus of claim 12 wherein the common spanning tree BPDU is an IEEE 802.1D BPDU.
17. In a first network device operably connected via a trunk link to a second network device, wherein the second network devices is operative to implement a plurality of virtual LANs (VLANs) including a common spanning tree protocol instance and a plurality of per-VLAN spanning tree protocol instances; wherein the common spanning tree protocol instance is implemented across respective native VLANs on the first and second network devices, wherein the native VLAN on the first network device is different from the native VLAN on the second network device, and wherein the second network device is operative to receive untagged frames from the first network device and add tags to the frames corresponding to the native VLAN of the first network device prior to processing thereof, and to remove tags identifying the native VLAN of the first network device prior to transmission to the first network device, a method comprising
enabling, in connection with at least the second network device, a plurality of virtual LANs (VLANs) including a common spanning tree protocol instance and a plurality of per-VLAN spanning tree protocol instances, wherein the common

spanning tree protocol is implemented on a first native VLAN different from the native VLAN of the second network device;

conditionally adding VLAN tags to common Spanning Tree Protocol (STP) Bridge Protocol Data Units (BPDUs) transmitted to a network device where the native VLAN is different from the first native VLAN; and

stripping VLAN tags, prior to processing, from common STP BPDUs that are received.

18. The method of claim 17 further comprising
 - processing the common spanning tree BPDU; and
 - forwarding the common spanning tree BPDU to at least one network device.
19. The method of claim 17 wherein the common spanning tree BPDU is an IEEE 802.1D BPDU.
20. The method of claim 17 wherein the plurality of per-VLAN spanning tree protocol instances are PVST+ protocol instances.
21. An apparatus operable in a network environment including a first network device, wherein the first network device is operative to implement a plurality of virtual LANs (VLANs) including a common spanning tree protocol instance and a plurality of per-VLAN spanning tree protocol instances; wherein the common spanning tree protocol instance is implemented across a first native VLAN on the first network device, and wherein the first network device is operative to receive untagged frames from a network device and add tags to the frames corresponding to the native VLAN of the first network device prior to processing thereof, and to remove tags identifying the first native VLAN of the first network device prior to transmission to the network device, the apparatus comprising

at least one port;
a processor;
a memory;
an application physically stored in the memory, comprising instructions operable to cause the processor and the apparatus to
enable, in connection with at least the first network device, a plurality of virtual LANs (VLANs) including a common spanning tree protocol instance and a plurality of per-VLAN spanning tree protocol instances, wherein the common spanning tree protocol is implemented on a second native VLAN different from the first native VLAN of the first network device;
conditionally add VLAN tags to common Spanning Tree Protocol (STP) Bridge Protocol Data Units (BPDUs) transmitted to a network device where the native VLAN is different from the second native VLAN; and
strip VLAN tags, prior to processing, from common STP BPDUs that are received.

22. The apparatus of claim 21 wherein the application further comprises instructions operable to cause the processor and the apparatus to
process the common spanning tree BPDU; and
forward the common spanning tree BPDU to at least one network device.
23. The apparatus of claim 21 wherein the common spanning tree BPDU is an IEEE 802.1D BPDU.
24. The apparatus of claim 21 wherein the plurality of per-VLAN spanning tree protocol instances are PVST+ protocol instances.

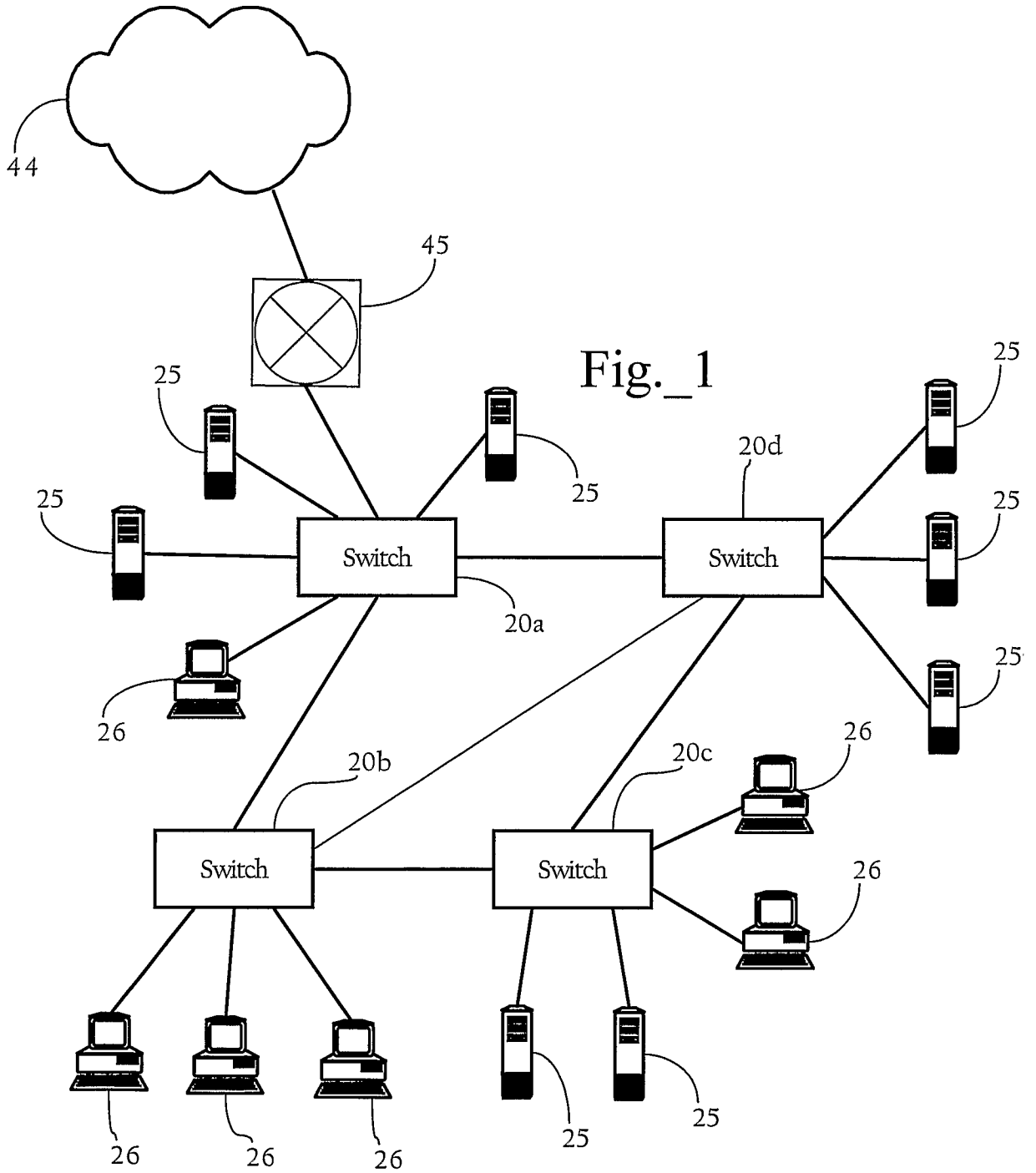
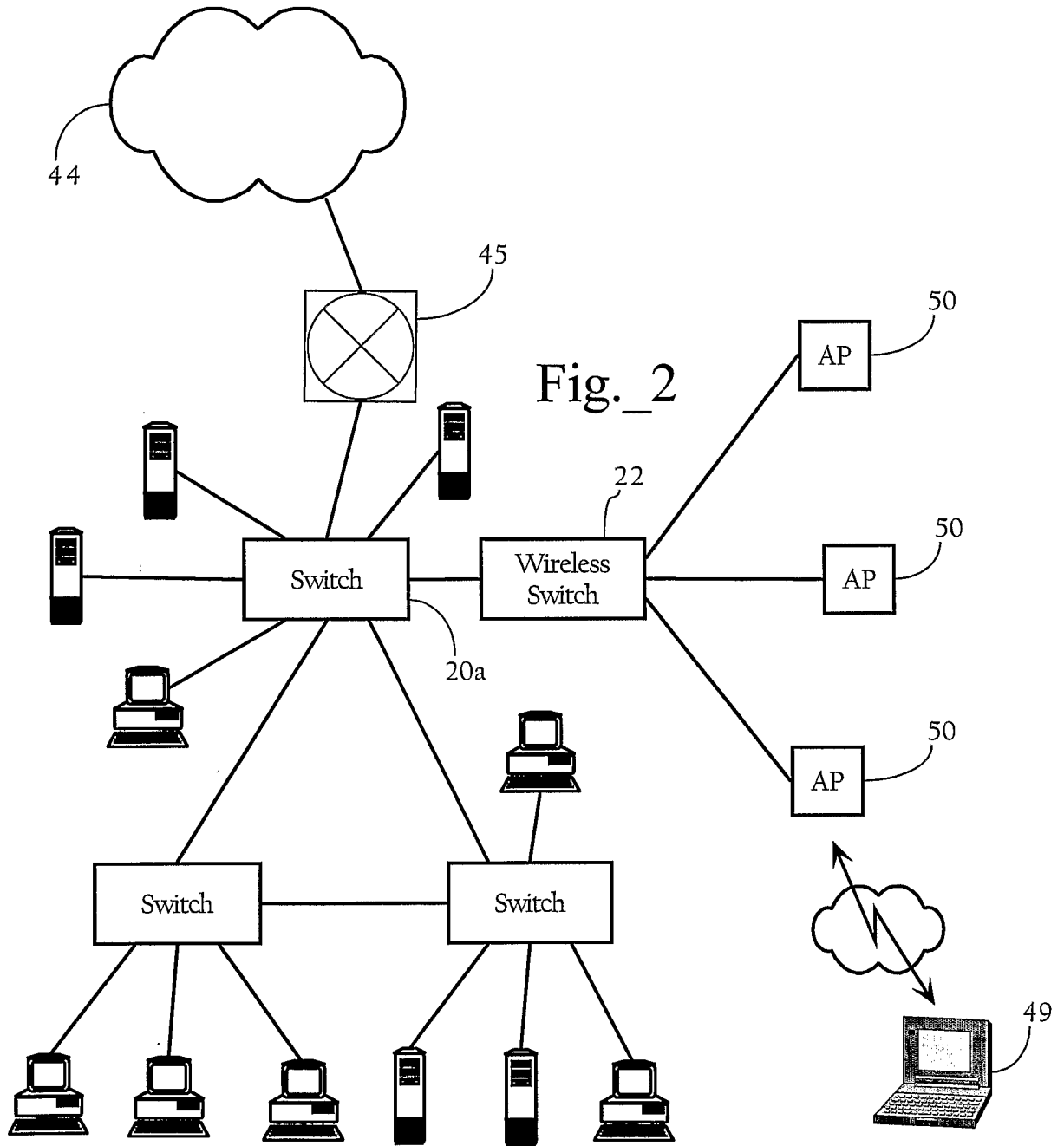


Fig. 1



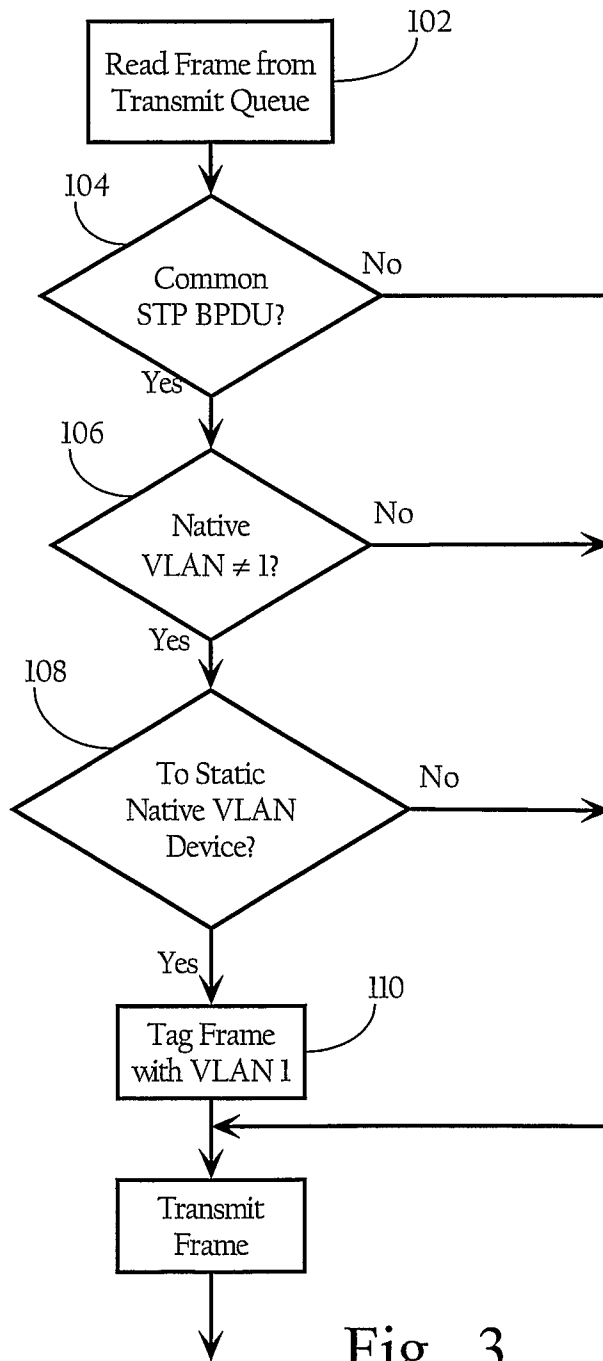


Fig._3

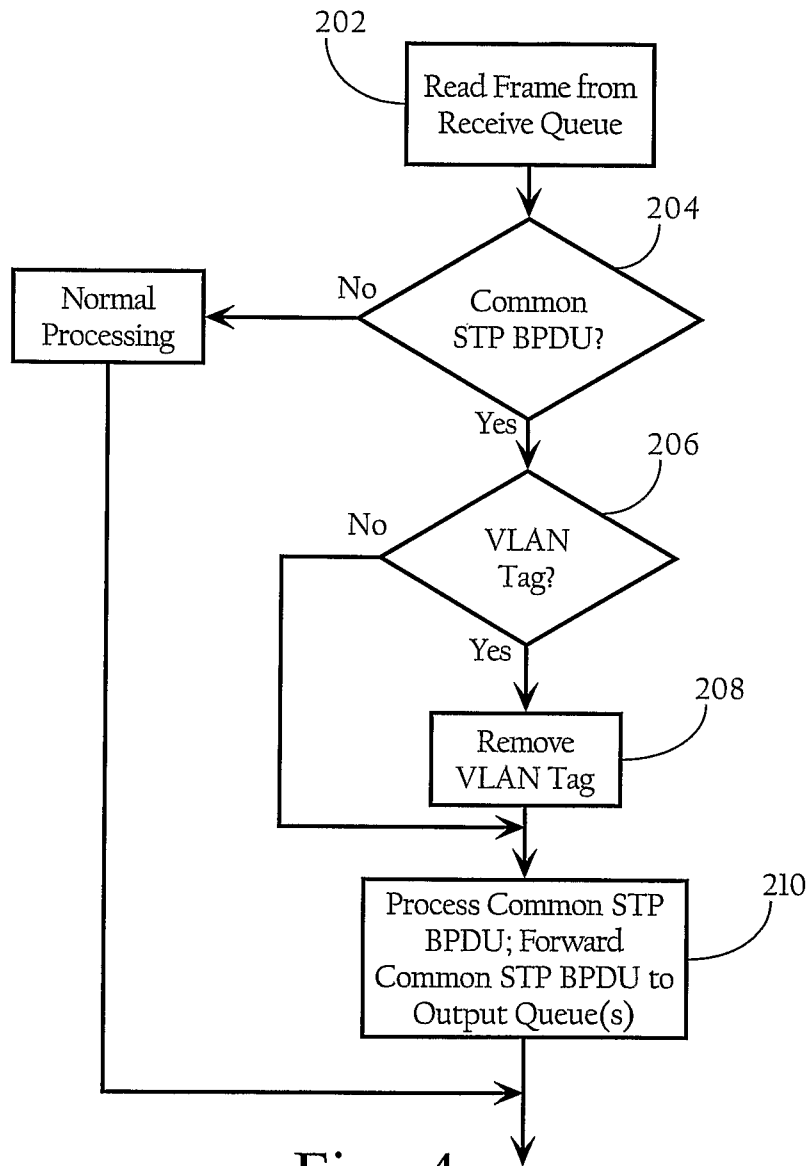


Fig._4

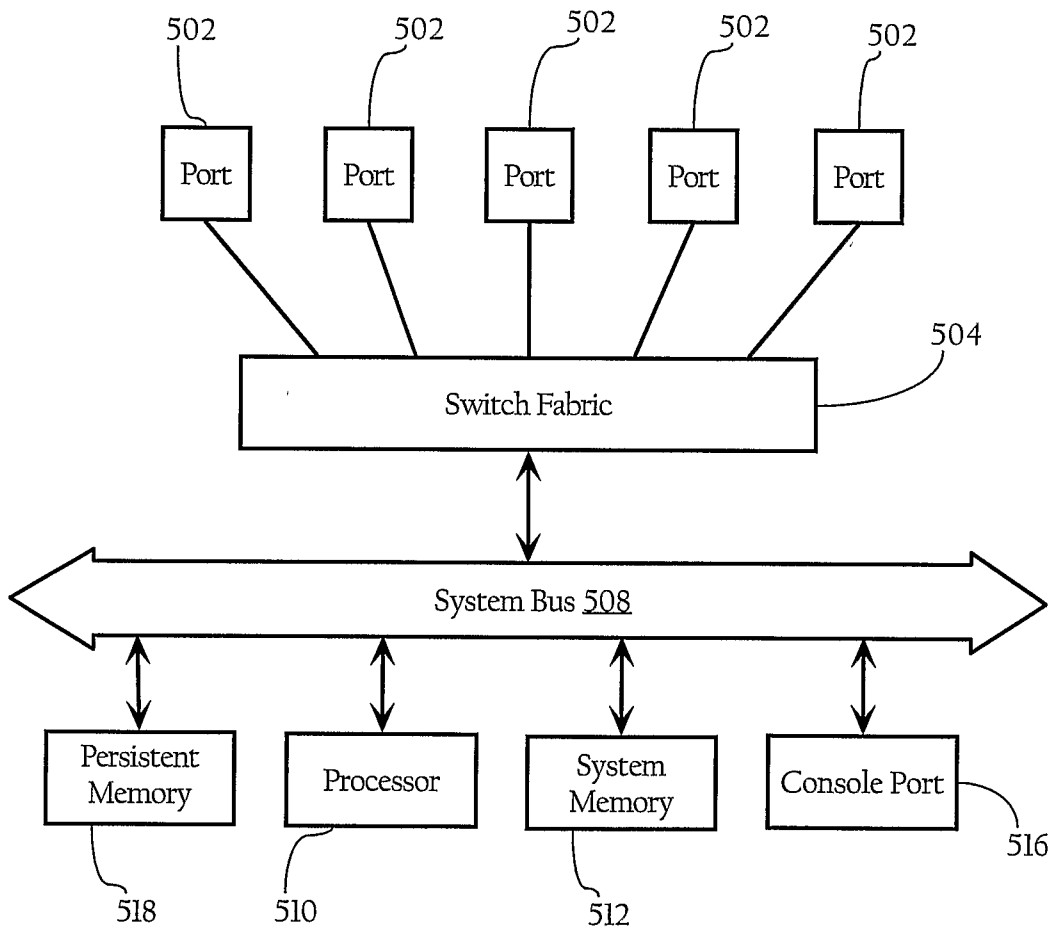


Fig._5

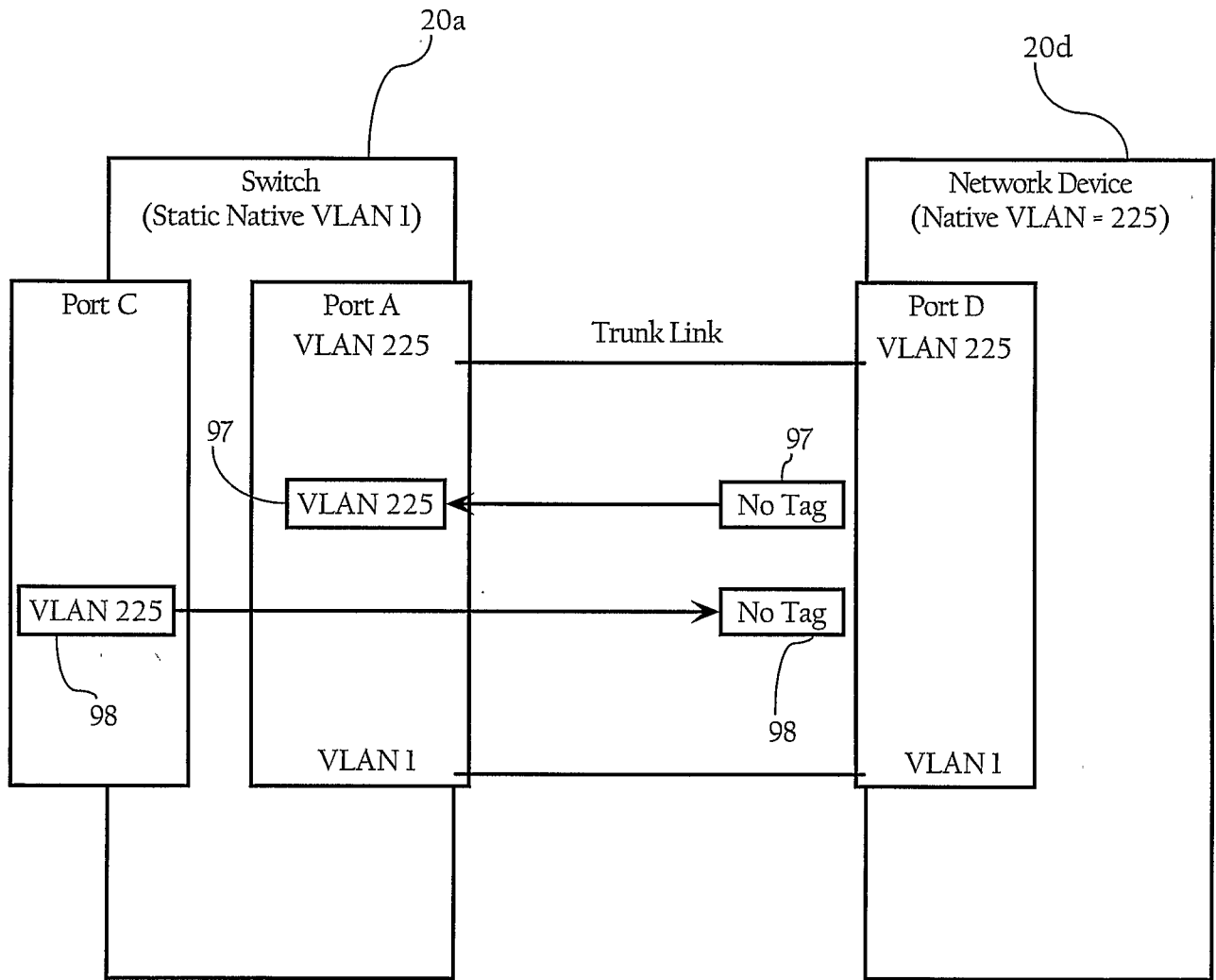


Fig._6

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2006/029659

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L12/46

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
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, PAJ, WPI Data, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	SZ: "Configuring the Avaya S8700 Media Server with Avaya G600 Media Gateway and Avaya IP Telephones in a Cisco ISL environment"[Online] 2002, pages 1-11, XP002410325 Internet Retrieved from the Internet: URL: http://www.avaya.com/master-usa/en-us/resource/assets/applicationnotes/isl_8021q.pdf [retrieved on 2006-12-04] page 4 page 10	1-24

Further documents are listed in the continuation of Box C.

See patent family annex.

* Special categories of cited documents :

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *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.
- *Z* document member of the same patent family

Date of the actual completion of the international search

5 December 2006

Date of mailing of the international search report

18/12/2006

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
 NL - 2280 HV Rijswijk
 Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
 Fax: (+31-70) 340-3016

Authorized officer

Siebel, Christian