

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
5 July 2001 (05.07.2001)

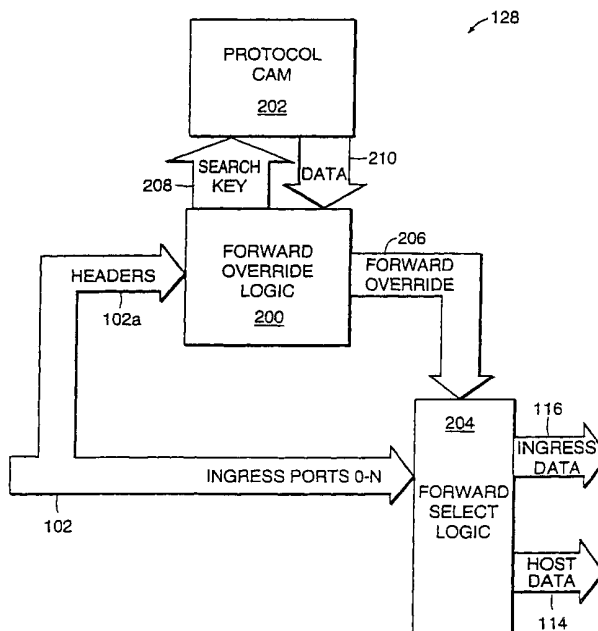
PCT

(10) International Publication Number  
**WO 01/48975 A2**

- (51) International Patent Classification<sup>7</sup>: **H04L 12/00**
- (21) International Application Number: PCT/CA00/01515
- (22) International Filing Date:  
22 December 2000 (22.12.2000)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:  
09/473,074 28 December 1999 (28.12.1999) US
- (71) Applicant (for all designated States except US): **MOSAID TECHNOLOGIES INCORPORATED** [CA/CA]; 11 Hines Road, Kanata, Ontario K2K 2X1 (CA).
- (72) Inventor; and
- (75) Inventor/Applicant (for US only): **BROWN, David, A.** [—/CA]; 110 Allissia Crescent, Carp, Ontario K0A 1L0 (CA).
- (74) Agent: **PILLAY, Kevin**; Fasken Martineau DuMoulin LLP, Suite 4200, Toronto Dominion Bank Tower, P.O. Box 20, Toronto-Dominion Centre, Toronto, Ontario M5K 1N6 (CA).
- (81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.
- (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- Published:**  
— Without international search report and to be republished upon receipt of that report.

[Continued on next page]

(54) Title: FORWARD OVERRIDES IN A PACKET SWITCH



(57) **Abstract:** A memory is successively searched using the contents of a header included in a data packet in order to select a forwarding entry for the data packet. The searches use a header field for a different protocol layer. A next field stored in a forwarding entry is used to determine if another search is required. The data packet is forwarded according to the forwarding override stored in the selected forwarding entry selected from the final successive search. The received data packet is forwarded to a host processor, through the switch, to both the host processor and through the switch, or the received data packet is not forwarded, dependent on the selected forward override.



WO 01/48975 A2



---

*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.*

## FORWARD OVERRIDES IN A PACKET SWITCH

## BACKGROUND OF THE INVENTION

In a computer network, a networking switch receives a data packet at an ingress port connected to the switch and forwards the data packet to an egress port connected to the switch. The switch determines the egress port to which the data packet is forwarded dependent on a destination address included in a header in the data packet received at the ingress port.

Typically, the switch includes a forwarding table typically implemented in forwarding logic in the ingress engine. The forwarding table is searched for one or more egress ports to which the data packet is to be forwarded dependent on the destination address included in the data packet. However, if the data packet is received for an unknown destination address the switch will not have a matching entry in the forwarding table. Thus, such data packets received at an ingress port are sent to a host processor in the switch; for example, data packets with no matching entry in the forwarding table are forwarded so that the host can update the forwarding table.

One known technique for determining which data packets to forward to the host processor is to make a forwarding decision based on destination address only; for example, to forward all broadcast data packets. If all broadcast data packets are forwarded to the host processor, the host processor may process data packets which may not include information that the host requires. Also, there may be non-broadcast data packets that the host processor requires that are not forwarded to the host. Thus, the forwarding of all broadcast data packets unnecessarily consumes host processor bandwidth and does not provide all the data packets that the host may require.

Another known technique for forwarding data packets to the host processor is to select the data packets to forward based on the data type included in the header of the data packet, such as the type stored in the data link header. The packet types to forward to the host are stored in a host forward list. A data packet is forwarded to the host if the packet type included in the header of the data packet is a member of the host forward list. The host forward list is hardcoded in the switch. In the open

systems interconnection, ("OSI") model, the L2 packet type is included in the data link layer header.

For example, the L2 packet type may be Address Resolution Type ("ARP") Bridge Protocol Data Unit, or Internet Protocol ("IP"). This technique reduces the number of data packets forwarded to the host processor by limiting the data packets forwarded to those of a particular type. For example, using the L2 data type stored in the data link header Bridge Protocol Data Unit data packets may be forwarded only to the host processor, ARP data packets may be forwarded to both the host processor and to the egress port and IP data packets may be forwarded to both the host and the egress port.

The disadvantage is that if a networking protocol with a new L2 data type is to be forwarded to the host, a new data packet type must be added to the host forwarding list, requiring a hardware update. This technique also may unnecessarily consume host processor bandwidth because not all data packets of a particular L2 data type, such as IP data packets, may be required for processing by the host.

## SUMMARY OF THE INVENTION

A memory in a network switch includes at least one forwarding entry having a forward override. Forward override logic in the switch successively searches the memory for forwarding entries matching portions of a received data packet. The forward override logic extracts the forward override from the matching entries. The matching portions of the data packet may be header fields and identify protocols at multiple layers.

The forwarding entry also includes a next field. The forward override logic determines whether to perform another search of the memory dependent on the state of the next field. The memory may be a content addressable memory.

After the search is complete, the forward override logic extracts a forward override for the received data packet from the forwarding entry. Forward select logic in the switch forwards the received data packet dependent on the extracted forward override.

The forwarding entry may include a priority field dependent on the contents of the received data packet and a traffic label field. The traffic label field and the priority field may be forwarded to a host for processing by the host.

## BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objects, features and advantages of the invention will be apparent from the following more particular description of preferred embodiments of the invention, as illustrated in the accompanying drawings in which  
5 like reference characters refer to the same parts throughout the different views. The drawings are not necessarily to scale, emphasis instead being placed upon illustrating the principles of the invention.

Fig. 1 illustrates a network switch including host forwarding logic for forwarding a data packet received at an ingress port connected to the switch  
10 according to the principles of the present invention;

Fig. 2 illustrates the host forwarding logic shown in Fig. 1;

Fig. 3 illustrates the format of an entry in the protocol Content Addressable Memory ("CAM") shown in Fig. 2;

Fig. 4A illustrates an empty entry in the protocol CAM shown in Fig. 2;

15 Fig. 4B illustrates a layer 2 type entry in the protocol CAM shown in Fig. 2;

Fig. 4C illustrates an IPv4-type entry in the protocol CAM shown in Fig. 2;

Fig. 4D illustrates an IP-Version entry in the protocol CAM shown in Fig. 2;

Fig. 4E illustrates a layer 4 entry in the protocol CAM shown in Fig. 2;

20 Fig. 5A illustrates a prior art data packet which may be received at an ingress port connected to the switch;

Fig. 5B illustrates a prior art Ethernet (data link layer (L2)) header which may be included in the data packet shown in Fig. 5A;

25 Fig. 5C illustrates a prior art Internet Protocol (network layer (L3)) header which may be included in the data packet shown in Fig. 5A;

Fig. 5D illustrates a prior art Transmission Control Protocol ("TCP") (transport layer (L4)) header which may be included in the data packet shown in Fig. 5A;

30 Fig. 5E illustrates a prior art User Datagram Protocol ("UDP") (transport layer (L4)) header which may be included in the data packet shown in Fig. 5A;

Fig. 6 is a flowchart of the steps performed in the forward override logic shown in Fig. 2 for selecting a forward override for the received data packet;

Fig. 7 illustrates entries in the protocol CAM.

## DETAILED DESCRIPTION OF THE INVENTION

Fig. 1 illustrates a network switch 100 including host forwarding logic 128 for forwarding a data packet received at an ingress port 102 connected to the switch 100. A data packet received at any one of the ingress ports 0-n 102 is processed in the ingress ports engine 104. Host forwarding logic 128 in the ingress ports engine 104 determines from a networking protocol encoded in the header of received data packet where to forward the received data packet. The received data packet may be forwarded on ingress data path 116 to segment buffer memory 108 and/or on host data path 114 to the host interface logic 106.

The host forwarding logic 128 may forward the data packet to both segment buffer memory 108 and the host interface 106 or to either segment buffer memory 108 or the host interface 106, or the host forwarding logic 128 may not forward the data packet; that is, it may drop the data packet. By assigning a forwarding decision for a data packet dependent on a networking protocol encoded in the header of a data packet, only the data packets required by the host are forwarded to the host. Also, if a new networking protocol is introduced, a new forwarding decision can be added to the host forwarding logic 128 by the host processor through the host interface logic 106 and host data path 114.

A data packet forwarded on ingress data path 116 is stored in segment buffer memory 108 and forwarded to the egress ports engine 110 on egress data path 118. The egress ports engine 110 forwards the data packet to one or more egress ports 112 dependent on the contents of the header included in the data packet. The storage of data packets in segment buffer memory 108 is described in co-pending U.S. Patent Application Serial Number 09/386,589, filed on August 31, 1999 entitled "Method and apparatus for an Interleaved Non-Blocking Packet Buffer," by David A. Brown, the entire teachings of which are incorporated herein by reference.

A data packet forwarded on host data path 114 to the host interface logic 106 is forwarded by the host interface logic 106 to a host or microprocessor (not shown) on host port data path 120. The host port interface logic 106 is a Peripheral Control Interface ("PCI") host port interface. The host port interface logic 106 is not limited to PCI; it may be other host interface logic known to one skilled in the art. The host interface logic 106 converts host data path 114 into host port data path 120 dependent on the host interface protocol implemented in the host interface logic 106.

Fig. 2 illustrates the host forwarding logic 128 shown in Fig. 1. The host forwarding logic 128 includes a protocol Content Addressable Memory ("CAM"), forward override logic 200 and forward select logic 204. As a data packet is received at an ingress port 102 (Fig. 1) in the ingress ports engine 104, (Fig. 1) the data packet's header 102a is forwarded to the forward override logic 200 in the host forwarding logic 128.

The forward override logic 200 successively searches the protocol CAM 202 for a forward override 206 stored in a forwarding entry 300 (Fig. 3), dependent on the contents of the received data packet's header 102a. The forwarding entry 300 (Fig. 3) is described in conjunction with Fig. 3. Multiple searches of the protocol CAM 200 may be performed using the contents of different fields in the data packet's header 102a. The data packet's header 102a includes at least one layer header. The layer header includes at least one field. The search for a forwarding entry 300 (Fig. 3) begins at the lowest layer header and subsequent searches search the next higher layer header. For example, a first search for a forwarding entry may be performed using the contents of the type field in the data link layer (L2) header. A second search for a forwarding entry 300 may be performed using the contents of the type field in the network link layer header (L3) dependent on the result of the L2 header search. A third search for a forwarding entry 300 may be performed using the contents of the source address field in the transport layer header (L4) dependent on the result of the L3 search. Whether the search continues to another layer is determined by the value of a next field in each forwarding entry.

At the start of each search, the forward override logic 200 generates a search key 208 by selecting from the contents of the one of the data packet's layer headers and forwards the search key 208 to a protocol Content Addressable Memory ("CAM") 202. If there is a forwarding entry 300 (Fig. 3) matching the search key 208 in the protocol CAM 202, the associated data 210 stored in the forwarding entry 300 (Fig. 3) is forwarded to the forward override logic 200 where a current copy of the forwarding entry is stored. The associated data 210 includes a forward override 206 which is implemented in this embodiment as a 2-bit binary code as shown in Table 1. The forward override 206 indicates where the data packet is to be forwarded. After the last search has been performed, the forward override 206 in the forwarding entry 300 (Fig. 3) selected in the last search is forwarded to the forward

select logic 204. The data packet is forwarded on ingress data path 116 and/or host data path 114 dependent on the state of forward override 206 as shown in Table 1.

5

Forward Override	Forwarding Decision
00	Ingress Data only
01	Ingress Data and Host Data
10	Host Data only
11	Do not forward (Drop)

TABLE 1

10

The protocol CAM 202 is a small table which may be implemented in registers using logic gates, with the host processor having read and write access to the registers. The forward select logic 204 forwards the data packet received at an ingress port 102 to ingress data 116 and/or host data 114 dependent on the state of the forward override 206 forwarded from the forward override logic 200. The forward select logic 204 includes a host buffer (not shown) and an ingress buffer (not shown). The input to the host buffer and the ingress buffer is the received buffer. The output of the host buffer is host data 114. The output of the ingress buffer is ingress data 116. The forward override 206 is encoded in buffer enable logic (not shown) to provide a host buffer enable signal and an ingress buffer enable signal. Thus, the forward override 206 determines whether the received data packet is forwarded through the host buffer to host data 114 and/or through the ingress buffer to ingress data 116.

15

20

25

Fig. 3 illustrates a forwarding entry 300 in the protocol CAM 202 (Fig. 2). The forwarding entry 300 includes a search key 208 and associated data 210. The associated data 210 includes a next field 302, an optional host traffic label 304, an optional Quality of Service ("QoS") or Etype IDentifier ("EID") field 306 and a forward override code 206. A copy of the forwarding entry 300 is stored in the forward override logic 200 (Fig. 1) and the forward override code 206 stored in the copy of the forwarding entry 300 is overridden with the forward override code 206

stored in another forwarding entry selected in a subsequent search of the protocol CAM 202 (Fig. 1).

The QoS field 306 provides a priority for the received data packet. The Host traffic label 304 is a tag which provides information on how the received data packet is classified. The host traffic label 306 and the QoS or EID field 306 can be forwarded to the host processor through the host interface logic 106 for processing by the host. The contents of the search key 208 and the associated data 210 included in the forwarding entry 300 are dependent on the contents of the header being processed by the forward override logic 200 (Fig. 2). A new search key 208 is generated for each search by selecting the contents of a different layer header. Forwarding entries 300 are described in conjunction with Figs. 4A-4E.

The forward override logic 200 (Fig. 2) determines from the state of the next field 302 whether to perform another search for a forwarding entry 300 using the contents of the next higher layer header included in the data packet. The forward override code 206 from the current search overrides the forward override code 206 from the previous search stored in the copy of the forwarding entry (not shown) in the forward override logic 200 (Fig. 2). Thus, the forward override code 206 selected in the last search is the forwarding decision for the data packet.

For example, if the type stored in the L2 header is ARP, no further search is required because all ARP data packets are forwarded on ingress data path 116 and on host data path 114. Thus, the next field 302 in the forwarding entry 300 is set to '0'. However, if the type stored in the L2 header is IP, a further search in the L3 layer header is required to determine the type of IP data packet. If the IP type stored in the L3 layer header's type field is Internet Group Management Protocol ("IGMP"), the next field 302 is set to '0' indicating that no further searches are required, and the data packet is forwarded on ingress data path 116 (Fig. 1) and on host data path 114 (Fig. 1).

The default forwarding decision for IP type data packets may be set to forward on ingress data path 116 (Fig. 1). The forwarding decision is overridden if a subsequent search using the contents of a of higher level layer header selects a forward override code 206 (Fig. 2) to forward the data packet on host data path 114 (Fig. 1). By storing the forwarding entries 300 in a CAM, the searches are

performed quickly and the forwarding decision is also determined quickly so that the data packet may be forwarded at wire-speed through the switch 100 (Fig. 1).

Fig. 4A illustrates an empty or invalid entry 300a. The three Most Significant Bits ("MSBs") 400 of the empty search key 208a are set to "000" and invalid associated data 210a is stored.

Fig. 4B illustrates an Etype IDentifier ("EID") entry 300b. The MSB 410 of the EID search key 208b is set to "1". The EID search key 208b includes an L2 payload type 414 and a Custom OUI Indicator 412. The EID associated data 210b includes a next field 402b, a Host Traffic Label ("HTL") 404b, an EID 406 and a Forward Override ("FO") 408b.

Fig. 4C illustrates an IPv4-Type Entry 300c. The three MSBs 426 of the IP type search key 208c are set to "001". The IP type search key 208c includes an IPv4-Type 428. The IP type associated data 210c includes a next field 402c, a HTL 404c, an LN QoS valid bit 420, a Layer 3 QoS 422 and a FO 408c.

Fig. 4D illustrates an IP-version entry 300d. The three MSBs 416 of the IP version search key 208d are set to "001". The IP version search key 208d includes an IP-Version 418. The IP version associated data 210d includes a next field 402d, a HTL 404d, an LN QoS valid bit 420, a layer 3 QoS 422 and a FO 408d.

Fig. 4E illustrates a L4 port entry 300e. The two MSBs 430 of the L4 search key 208e are set to "01". The L4 port search key 208e includes an L4 port number 432. The L4 port associated data 210e includes a HTL 404e, LN QoS valid field 420, an L4 QoS 424 and a FO 408e.

Fig. 5A illustrates a prior art data packet which may be received at an ingress port 102 (Fig. 1) connected to the switch 100 (Fig. 1). Figs. 5B-5E illustrate the contents of prior art data packet headers which may be included in the prior art data packet shown in Fig. 5A.

Fig. 5A shows a prior art data packet 500 including a data field 510 and headers for networking layers 502, 504, 506, 508. Headers for four of the layers in the Open Systems Interconnection ("OSI") Reference Model are shown. They include: the physical layer (L1) header 502, the data link layer (L2) header 504, the networking layer (L3) header 506 and the transport layer (L4) header 508. For example, the data link (L2) layer 504 may be Ethernet, the networking layer (L3) header 506 may be Internet Protocol ("IP") and the transport layer (L4) header 508

may be Transmission Control Protocol ("TCP") or Unicast Datagram Protocol ("UDP").

Fig. 5B illustrates the format of an Ethernet data link (L2) header 504. The Ethernet data link (L2) header 504 includes a destination address 514, a source address 516, an optional Virtual Local Area Network Identification ("VLAN ID") field 518 and a length/type field 520.

Fig. 5C illustrates the format of an IP network layer (L3) header 506. The IP network layer (L3) header 506 includes a source IP address 544, a destination IP address 546, an IP Version field 522, an IP length field 524, Type of Service ("TOS") 526, Total Length 528, identification 530, Flags 532, Fragment Offset 534, Time to Live ("TTL") 536, Protocol Type 540, Header Checksum 542, Options 542 and Pad 550.

Fig. 5D illustrates the format of a TCP transport layer (L4) header 508a. The TCP transport layer (L4) header 508a includes the following fields: a TCP source port 552a, a TCP destination port 554a, a sequence number 556, an acknowledgment number 558, TCP offset 560, a reserved field 562, TCP flags 564, Window 566, TCP header Checksum 568, Urgent Pointer 570, Options 572 and TCP pad 574.

Fig. 5E illustrates the format of a UDP transport layer (L4) header 508b. The UDP Transport Layer (L4) header 508b includes the following fields: an UDP Source Port 552b, a UDP Destination Port 554b, a UDP Message Length 582, and a UDP Header Checksum 584.

Fig. 6 is a flowchart of the steps performed in the forward override logic 200 (Fig. 2) to select a forward override 206 (Fig. 2) for the received data packet dependent on the network protocol encoded in the received data packet's headers. Fig. 6 is described in conjunction with Figs. 2, 4A-4E and 5A-5E.

At step 600, the forward override logic 200 (Fig. 2) receives a data packet header from an ingress port 102 (Fig. 1). The forward override logic 200 (Fig. 2) initializes the forward override 206 (Fig. 2) and initializes an EID register (not shown) in the forward override logic 200. Processing continues with step 602.

At step 602, the forward override logic 200 (Fig. 2) forwards an EID search key 208b (Fig. 4B) to the protocol CAM 202 (Fig. 1). The MSB of the EID search key 208b (Fig. 4B) is set to "1" and includes the L2 payload type 414 (Fig. 4B) from the length/type field 520 (Fig. 5B) in the Ethernet data link (L2) header 504 (Fig.

5B) included in the received data packet. If there is a matching EID entry 300b (Fig. 4B) processing continues with step 604. If not, processing of the received data packet is complete and the forward override 206 (Fig. 2) is set to a default value stored in the Unknown\_EID\_Default\_Override register(not shown) in the forward  
5 override logic 202 (Fig. 2). The default value for the forward override may be selected by the host processor to forward the data packet to segment buffer memory 108 (Fig. 1).

At step 604, forward override 206 (Fig. 2) is set to the L2 forward override value stored in the forward override field 408b (Fig. 4B) in the EID associated data  
10 210b (Fig. 4B). Processing continues with step 606.

At step 606, the forward override logic 200 (Fig. 2) examines the next field 402b (Fig. 4B) in the EID entry 300b (Fig. 4B). If the next field 402b (Fig. 4B) indicates that a further search of the protocol CAM 202 (Fig. 2) is required, processing continues with step 608. If not, processing of the received data packet is  
15 complete.

At step 608, the forward override logic 200 (Fig. 2) forwards an IP version search key 208d (Fig. 4D) to the protocol CAM 202 (Fig. 2). The IP version search key's MSBs 416 are set to "001" and the IP version search key 208d (Fig. 4D) includes the IP version 432 (Fig. 4D) extracted from the IP version field 522 (Fig. 5C) in the IP networking layer (L3) header 506 (Fig. 5C) of the ingress data packet  
20 (Fig. 5A). If a match is found in the protocol CAM 202 (Fig. 2), the IP version associated data 210d is forwarded to the forward override logic 200 and processing continues with step 610. If not, processing of the received data packet is complete.

At step 610, the forward override logic 200 (Fig. 2) sets forward override  
25 206 (Fig. 2) to the forward override value 408d (Fig. 4D) stored in the IP version associated data 210d (Fig. 4D) in the IP version type entry 300d (Fig. 4D). Processing continues with step 612.

At step 612, the forward override logic 200 (Fig. 2) examines the next field 402c (Fig. 4C) in the IP version entry 300c. If the next field 402d (Fig. 4D) indicates that a further search of the protocol CAM 202 (Fig. 2) is required  
30 processing continues with step 614. If not, processing of the received data packet is complete.

At step 614, the forward override logic 200 (Fig. 2) forwards an IP Type search key 208c (Fig. 4C), to the protocol CAM 202 (Fig. 2). MSBs 426 (Fig. 4C) of the IP type search key 208c (Fig. 4C) are set to "001" and the IP type search key 208c includes the IPv4 type 428 (Fig. 4C) extracted from the protocol type field 540 (Fig. 5C) in the networking layer (L3) header 506 (Fig. 5C) of the ingress data packet (Fig. 5A). If a match is found for the IP type search key 208c in the protocol CAM 202 (Fig. 2), the IP type associated data 210c (Fig. 4C) is forwarded to the forward override logic 200 (Fig. 2) and processing continues with step 610. If not, processing of the received data packet is complete.

At step 616, the forward override logic 200 (Fig. 2) sets forward override 206 (Fig. 2) to the forward override value 408c stored in the IP type entry 300c. Processing continues with step 618.

At step 618, the forward override logic 200 (Fig. 2) examines the next field 402c (Fig. 4C) in the IP type entry 300c. If the next field 402c (Fig. 4C) indicates that a further search of the protocol CAM 202 (Fig. 2) is required processing continues with step 620. If not, processing of the received data packet is complete.

At step 620, the forward override logic 200 (Fig. 2) forwards an L4 port search key 208e (Fig. 4E) including MSBs 430 (Fig. 4E) set to "01" and L4 source port number extracted from the source port field 552 (Figs. 5D-5E) in the L4 header 508 (Figs. 5D, 5E) to the protocol CAM 202 (Fig. 2). If a source port match is found in the protocol CAM 202 (Fig. 2), a copy of the L4 port associated data 210e (Fig. 4E) is forwarded to the forward override logic 200 (Fig. 2) and processing continues with step 624. If not, processing continues with step 622.

At step 624, the forward override logic 200 (Fig. 2) sets forward override 206 (Fig. 2) to the forward override value 408e (Fig. 4E) stored in the L4 port associated data 210e (Fig. 4E) in the L4 port entry 300e (Fig. 4E). Processing of the received data packet is then complete.

At step 622, the forward override logic 200 (Fig. 2) forwards an L4 port search key 208e (Fig. 4E) including MSBs 430 (Fig. 4E) set to "01" and L4 destination port number from the destination port 554 (Figs. 5D, 5E) stored in the L4 header 508 (Figs. 5D, 5E) of the ingress data packet (Fig. 5A) to the protocol CAM 202 (Fig. 2). If a destination port match is found in the protocol CAM 202 (Fig. 2),

a copy of the L4 port associated data 210e (Fig. 4E) is forwarded to the forward override logic 200 (Figs. 2) and processing continues with step 626.

At step 626, the forward override logic 200 (Fig. 2) sets forward override 206 (Fig. 2) to the forward override value 408e (Fig. 4E) stored in the L4 port associated data 200e (Fig. 4E) in the L4 port entry 300e (Fig. 4E). Processing of the received data packet is then complete.

Thus, as many as five successive searches may be performed using the contents of the received data packet's header by the forward override logic 200 (Fig. 2) in the protocol CAM 202 (Fig. 2) in order to select the forward override 206 dependent on the network protocol encoded in the received data packet's header.

Fig. 7 illustrates the four protocol CAM 300 entries searched in the protocol CAM 202 to select a forward override for a Routing Information Protocol ("RIP") data packet 206 (Fig. 2). The entries include an EID entry 300ba, an IPv4 Type entry 300ca an IP version entry 300da and an L4 port entry 300ea. The searching of the protocol CAM 202 (Fig. 2) to select a forward override for a RIP data packet are described in conjunction with Fig. 6.

At step 602 (Fig. 6) a search is performed using the contents of the Ethernet data link (L2) header 504 (Fig. 5B) included in the RIP data packet. A search key 208ba with MSB set to "1" and L2 payload type 414 set to IP; that is, the contents of the length/type field 520 (Fig. 5B) in the Ethernet data link (L2) header 504 (Fig. 5B) included in the RIP data packet, is forwarded to the protocol CAM from the forward override logic 200 (Fig. 2). The matching forwarding entry 300ba in the protocol CAM 202 includes the associated data 210ba. The associated data 210ba includes a next field 402ba set to "NEXT" and a forward override field 408ba set to "Ingress data only".

At step 606 (Fig. 6), another search of the protocol CAM 202 (Fig. 2) is performed because the next field 402ba is set to "NEXT". The search is performed using the contents of the IP network layer (L3) header 506 (Fig. 5C) included in the RIP data packet. A search key 208da with MSB set to "001" and IP-version 418a set to the contents of the IP version field 522 in the IP network layer (L3) header 506 (Fig. 5C) included in the RIP data packet is forwarded to the protocol CAM 202 (Fig. 2) from the forward override logic 200 (Fig. 2). The matching forwarding entry 300da in the protocol CAM 202 includes the IP version associated data 210da.

The IP version associated data 210da includes a next field 402da set to "NEXT" and a forward override field 408da set to "ingress data only".

At step 612 (Fig. 6) another search of the protocol CAM 202 (Fig. 2) is performed because the next field 402da is set to "NEXT". The search is performed using the contents of the IP network layer (L3) header 506 (Fig. 5C) included in the RIP data packet. An IP type search key 208ca with MSBs set to "001" and IP type 428 (Fig. 4C) set to UDP; that is, the contents of the IP Protocol Type field 540 (Fig. 5C) in the IP network layer (L3) header 506 (Fig. 5C) included in the RIP data packet, is forwarded to the protocol CAM 202 from the forward override logic 200 (Fig. 2). The matching forwarding entry 300ca in the protocol CAM 202 includes the associated data 210ca. The associated data 210ca includes a next field 402ca set to "NEXT" and a forward override field 408ca set to "ingress data only".

At step 618 (Fig. 6) another search of the protocol CAM 202 is performed because the next field 402ca is set to "NEXT". The search is performed using the contents of the UDP transport layer (L4) header 508b (Fig. 5E) included in the received RIP data packet. A search key 208ea with MSBs set to "001" and L4 port number set to the contents of the UDP source port field 552b (Fig. 5E) in the UDP transport layer (L4) header 508b (Fig. 5E), is forwarded to the protocol CAM 202 (Fig. 2) from the forward override logic 200 (Fig. 2). The matching forwarding entry 300ea in the protocol CAM 202 (Fig. 2) includes the L4 port associated data 210ea. The L4 port associated data 210ea includes a next field 402ea set to "STOP" and forward override field 408ea set to "ingress data and host data". The forward override 206 (Fig. 2) is set according to the contents of the forward override 408ea. The RIP data packet is forwarded to both the host interface logic 106 (Fig. 1) and the segment buffer memory 108 (Fig. 1).

The search of the protocol CAM 202 (Fig. 2) is complete because the next field 408ea is set to "STOP". Thus, four searches of the protocol CAM 202 (Fig. 2) are performed to select the forward override 206 (Fig. 2) for the RIP data packet.

As shown in Figs. 4B-4E a Quality of Service (QoS) field 422, 424 may also be stored in the associated data 210 in the protocol CAM 202 (Fig. 2). The QoS field 306 (Fig. 3) may be used to assign a priority to the received data packet. The assigned priority is used by the egress ports engine 110 (Fig. 1) to select the order of packets to be forwarded on an egress port 112 (Fig. 1). For example, the contents of

the QoS field 306 (Fig. 3) may be used to assign different priorities such as, to assign highest priority to a Telnet data packet and lowest priority to a HyperText Markup language ("HTTP") data packet. The switch 100 may include a separate port queues for each QoS value, with a pointer to the location in memory of the data packet stored in one or more of the port queues dependent on the contents of the QoS field 306 (Fig. 3) in the forwarding entry 300 (Fig. 3) for the data packet type.

The invention has been described for a Local Area Network ("LAN"), but it is not limited to LANs it may also be implemented to select a forward override for data packets in a Wide Area Network.

While this invention has been particularly shown and described with references to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the scope of the invention encompassed by the appended claims.

## CLAIMS

What is claimed is:

1. A switch comprising:  
5 a memory comprising at least one forwarding entry, the forwarding entry including a forward override; and  
forward override logic which, for a data packet received by the switch, successively searches the memory for forwarding entries matching portions of the received data packet and extracts the forward override from a  
10 matching forwarding entry.
2. A switch as claimed in Claim 1 wherein the forwarding entry further comprises a next field and the forward override logic determines whether to perform another search of the memory dependent on the state of the next field in a current forwarding entry.
- 15 3. A switch as claimed in Claim 1 wherein the memory is a content addressable memory.
4. A switch as claimed in Claim 1 further comprising:  
forward select logic which forwards the received data packet  
dependent on the extracted forward override corresponding to a final search.
- 20 5. A switch as claimed in Claim 1 wherein the forwarding entry further comprises a priority field, the priority field including an assigned priority for the received data packet.
6. A switch as claimed in Claim 1 wherein the forwarding entry further  
comprises a host traffic label field, the host traffic label field including  
25 classification data for the received data packet.

7. A switch as claimed in Claim 1 wherein the portions of the received data packets identify protocols.
8. A method for selecting a forward override for a received data packet in a switch comprising the steps of:
  - 5 providing a memory, the memory comprising at least one forwarding entry, the forwarding entry including a forward override;  
successively searching the memory for forwarding entries matching portions of a received data packet; and  
extracting the forward override from a matching protocol entry.
- 10 9. A method as claimed in Claim 8 further comprising the step of:  
determining whether to perform another search of the memory dependent on the state of a next field in a current forwarding entry.
10. A method as claimed in Claim wherein the memory is content addressable memory.
- 15 11. A method as claimed in Claim 10 wherein the content addressable memory comprises registers.
12. A method as claimed in Claim 8 further comprising the step of:  
forwarding the received data packet dependent on the extracted forward override corresponding to a final search.
- 20 13. A method as claimed in Claim 8 wherein the forwarding entry further comprises a priority field, the priority field including an assigned priority for the received data packet.
- 25 14. A method as claimed in Claim 8 wherein the forwarding entry further comprises a host traffic label field, the host traffic label field including classification data for the received data packet.

15. A method as claimed in Claim 8 wherein the portions of the received data packets identify protocols.
16. A switch comprising:  
a memory comprising at least one forwarding entry, the forwarding entry including a forward override;  
5 means for successively searching the memory for forwarding entries matching portions of a received data packet; and  
means for extracting the forward override from a matching forwarding entry.
- 10 17. A switch as claimed in Claim 16 further comprising:  
means for determining whether to perform another search of the memory dependent on the state of a next field included in the forwarding entry.
- 15 18. A switch as claimed in Claim 16 wherein the memory is a content addressable memory.
19. A switch as claimed in Claim 16 further comprising:  
means for forwarding the received data packet dependent on the extracted forward override corresponding to a final search.
- 20 20. A switch as claimed in Claim 16 wherein the forwarding entry further comprises a priority field the priority field including an assigned priority for the received data packet.
21. A switch as claimed in Claim 16 wherein the forwarding entry further includes a host traffic label field, the host traffic label field including classification data for the received data packet.
- 25 22. A switch as claimed in Claim 16 wherein the portions of the received data packets identify protocols.

23. A switch comprising:  
a protocol CAM comprising at least one forwarding entry, the forwarding entry including a forward override; and  
5 forward override logic which, for a received data packet, successively searches the protocol CAM for forwarding entries matching portions of the received data packet and extracts the forward override from a matching forwarding entry.
24. A switch as claimed in Claim 23 wherein the forwarding entry further  
10 comprises a next field and the forward override logic determines whether to perform another search of the protocol CAM dependent on the state of the next field in a current forwarding entry.
25. A switch as claimed in Claim 23 further comprising:  
forward select logic which forwards the received data packet  
15 dependent on the extracted forward override corresponding to a final search.
26. A switch as claimed in Claim 23 wherein the forwarding entry further comprises a priority field, the priority field including an assigned priority for the received data packet.
27. A switch as claimed in Claim 23 wherein the forwarding entry further  
20 comprises a host traffic label field, the host traffic label field including classification data for the received data packet.
28. A method for selecting a forward override for a received data packet in a switch comprising the steps of:  
providing a protocol CAM, the protocol CAM comprising at least  
25 one forwarding entry, the forwarding entry including a forward override;  
successively searching the protocol CAM for forwarding entries matching portions of a received data packet; and  
extracting the forward override from a matching protocol entry.

29. A method as claimed in Claim 28 further comprising the step of:  
determining whether to perform another search of the protocol CAM  
dependent on the state of a next field in a current forwarding entry.
- 5 30. A method as claimed in Claim 28 further comprising the step of:  
forwarding the received data packet dependent on the extracted  
forward override.
31. A method as claimed in Claim 28 wherein the forwarding entry further  
comprises a priority field, the priority field including an assigned priority for  
the received data packet.
- 10 32. A method as claimed in Claim 28 wherein the forwarding entry further  
comprises a host traffic label field, the host traffic label field including  
classification data for the received data packet.
- 15 33. A switch comprising:  
a protocol CAM comprising at least one forwarding entry, the  
forwarding entry including a forward override;  
means for successively searching the protocol CAM for forwarding  
entries matching portions of a received data packet; and  
means for extracting the forward override from a matching  
forwarding entry.
- 20 34. A switch as claimed in Claim 33 further comprising:  
means for determining whether to perform another search of the  
protocol CAM dependent on the state of a next field included in a current  
forwarding entry.
- 25 35. A switch as claimed in Claim 33 further comprising:  
means for forwarding the received data packet dependent on the  
extracted forward override resulting from a final search.

36. A switch as claimed in Claim 33 wherein the forwarding entry further comprises a priority field the priority field including an assigned priority for the received data packet.
- 5 37. A switch as claimed in Claim 33 wherein the forwarding entry further includes a host traffic label field, the host traffic label field including classification data for the received data packet.

100

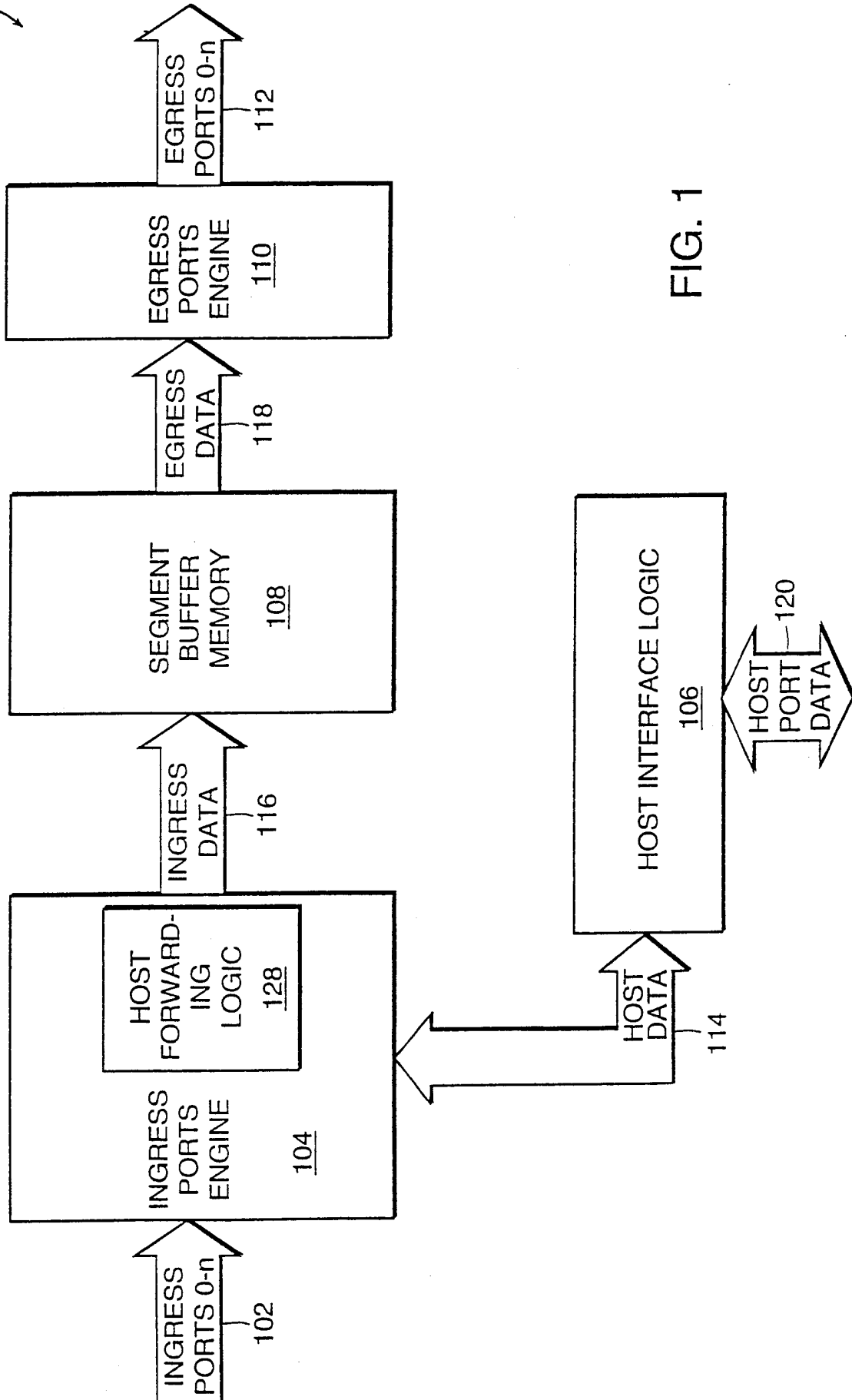


FIG. 1

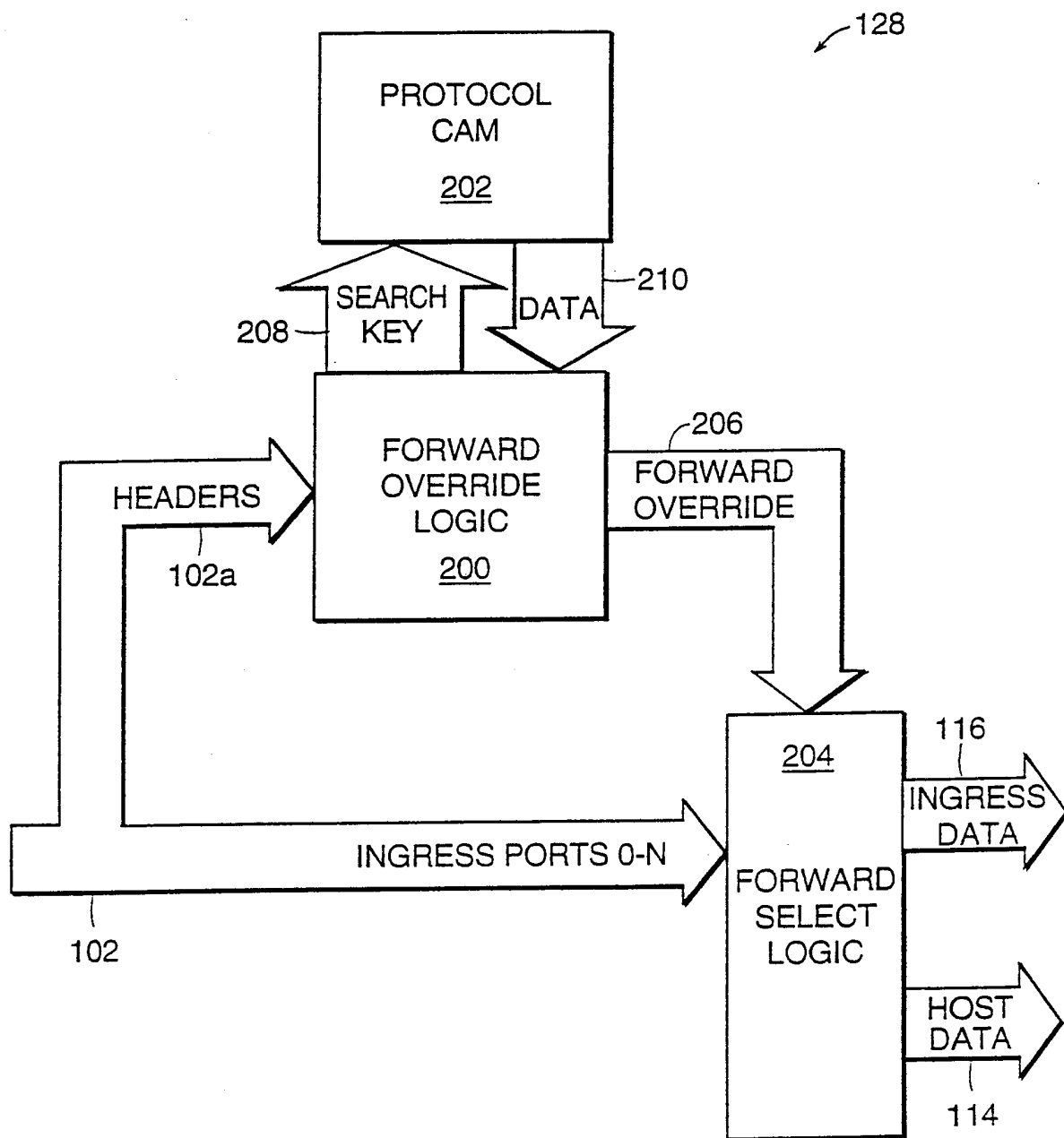


FIG. 2

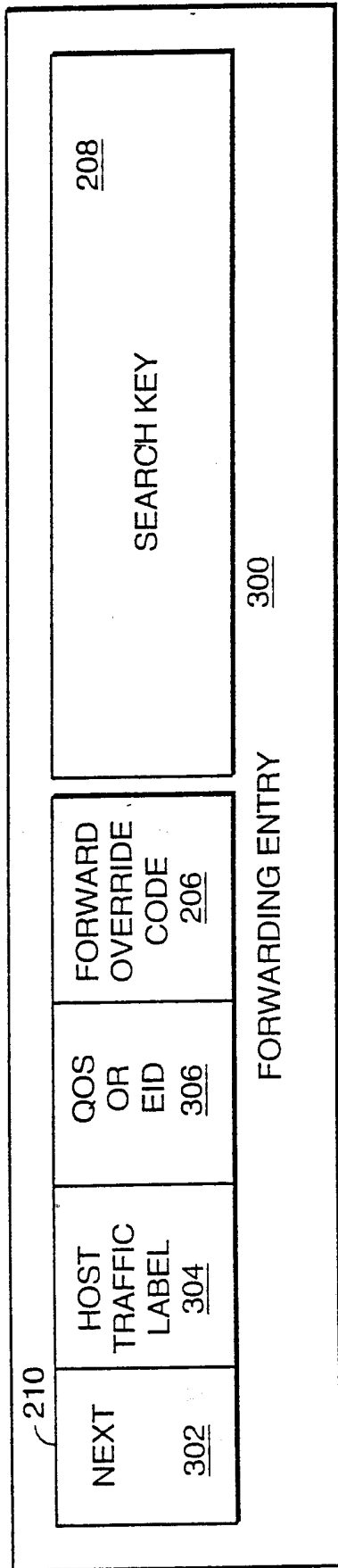


FIG. 3

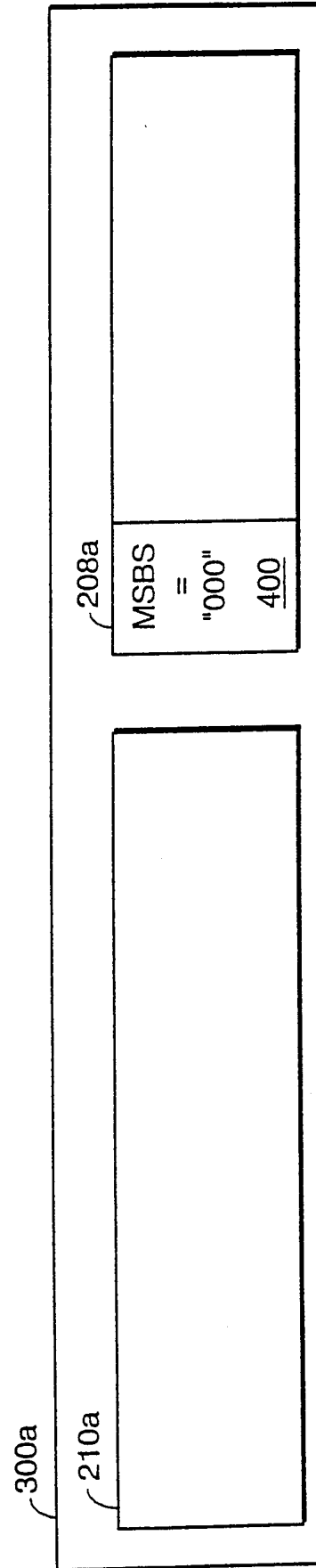


FIG. 4A

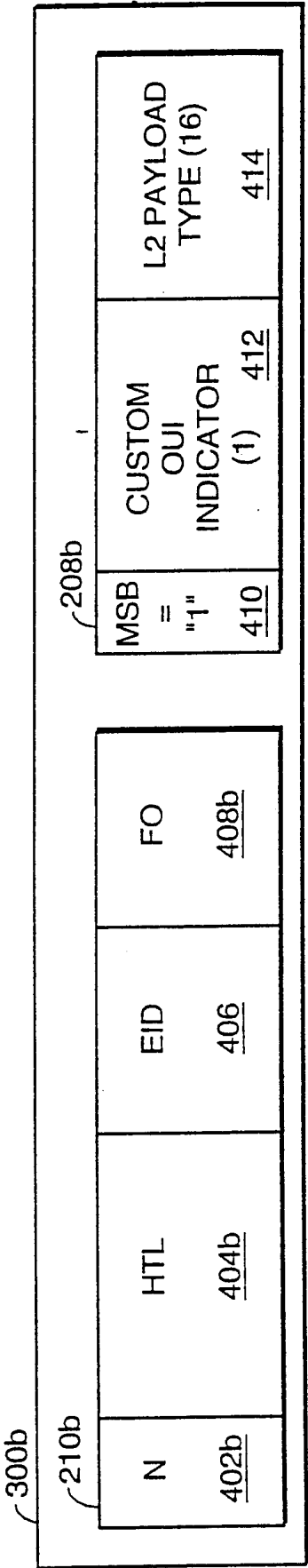


FIG. 4B

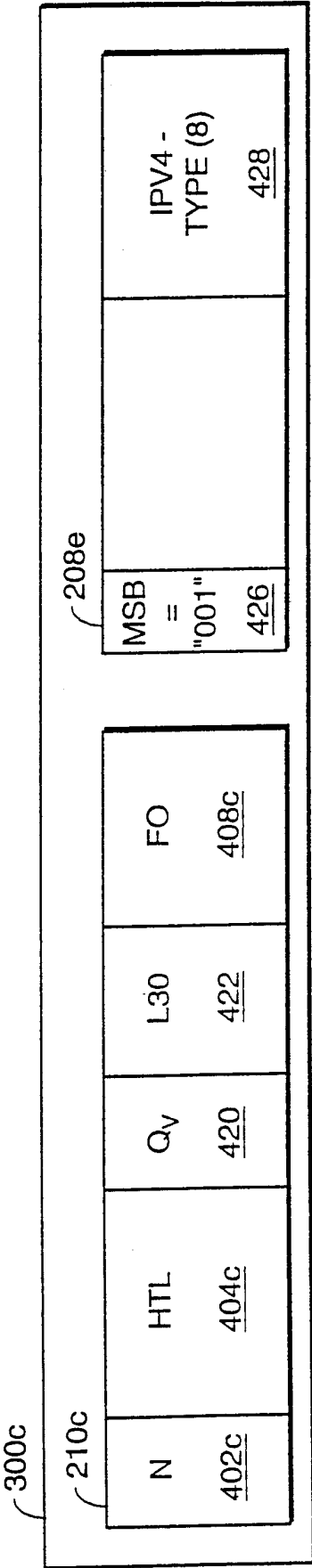


FIG. 4C

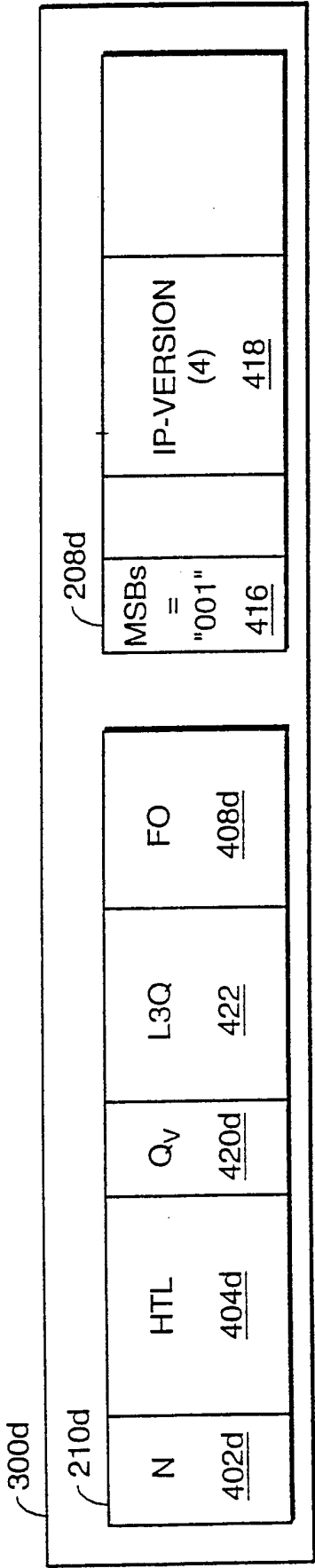


FIG. 4D

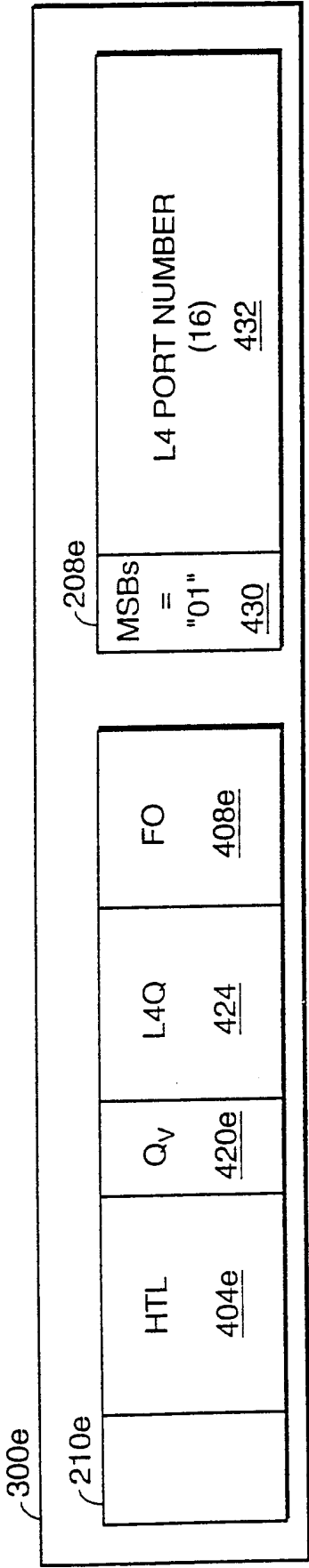
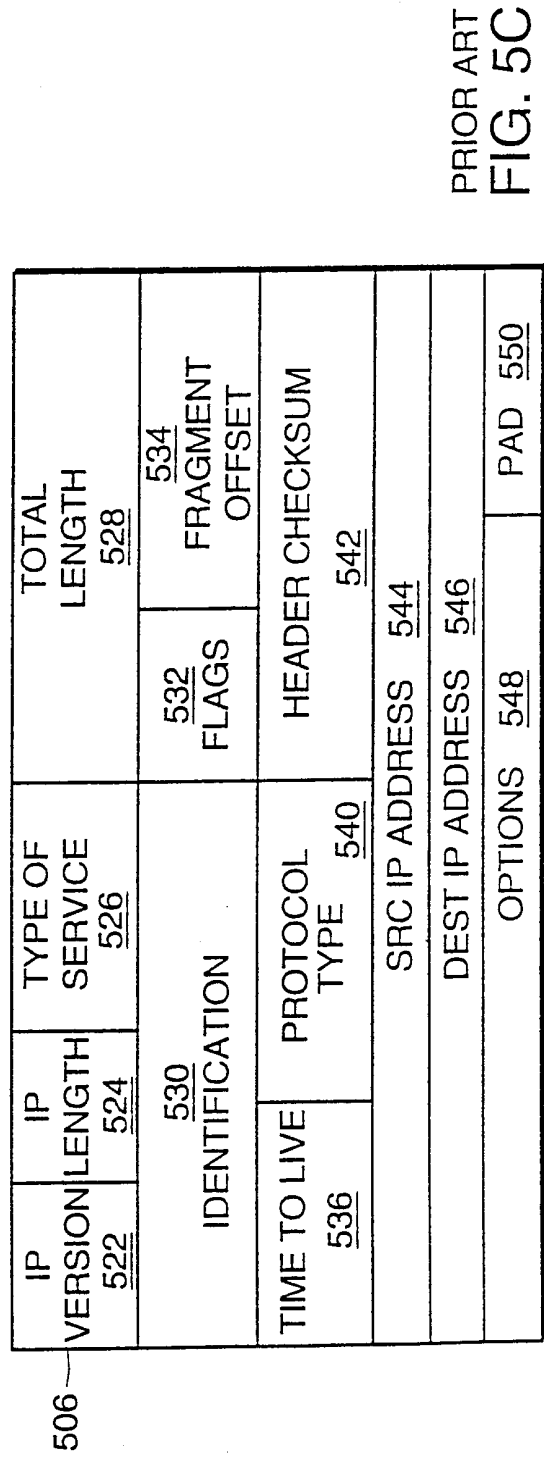
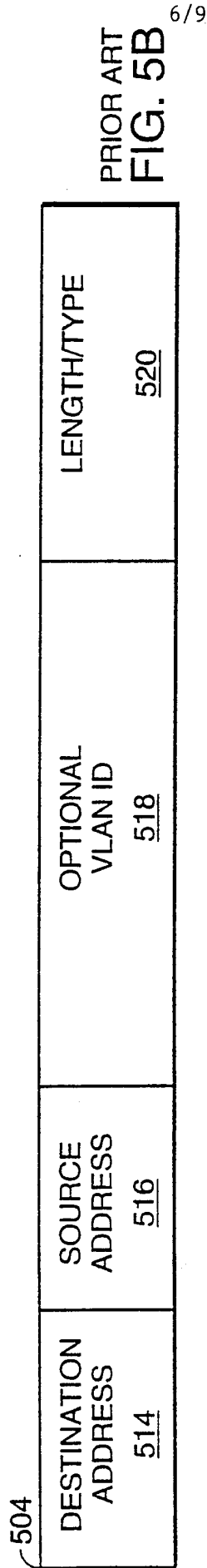
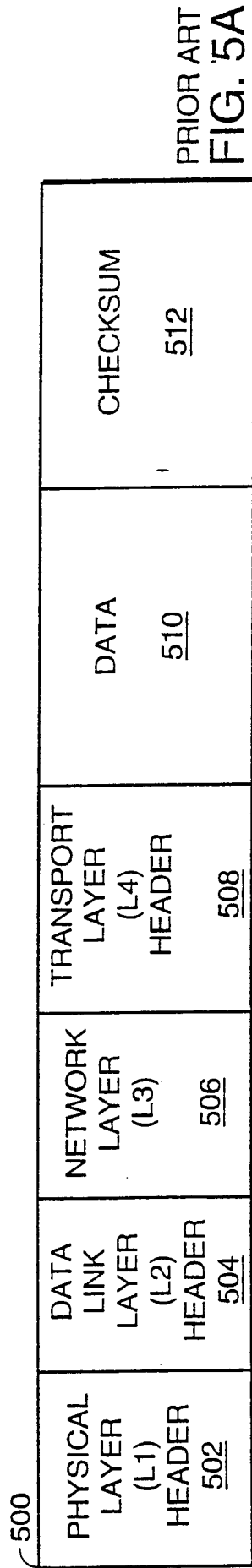


FIG. 4E



508a

SOURCE PORT (TCP) <u>552a</u>		DESTINATION PORT (TCP) <u>554a</u>	
SEQUENCE NUMBER <u>556</u>			
ACKNOWLEDGEMENT NUMBER <u>558</u>			
TCP OFFSET <u>560</u>	RESERVED <u>562</u>	TCP FLAGS <u>564</u>	WINDOW <u>566</u>
TCP HEADER CHECKSUM <u>568</u>			URGENT POINTER <u>570</u>
OPTIONS <u>572</u>			TCP PAD <u>574</u>

FIG. 5D

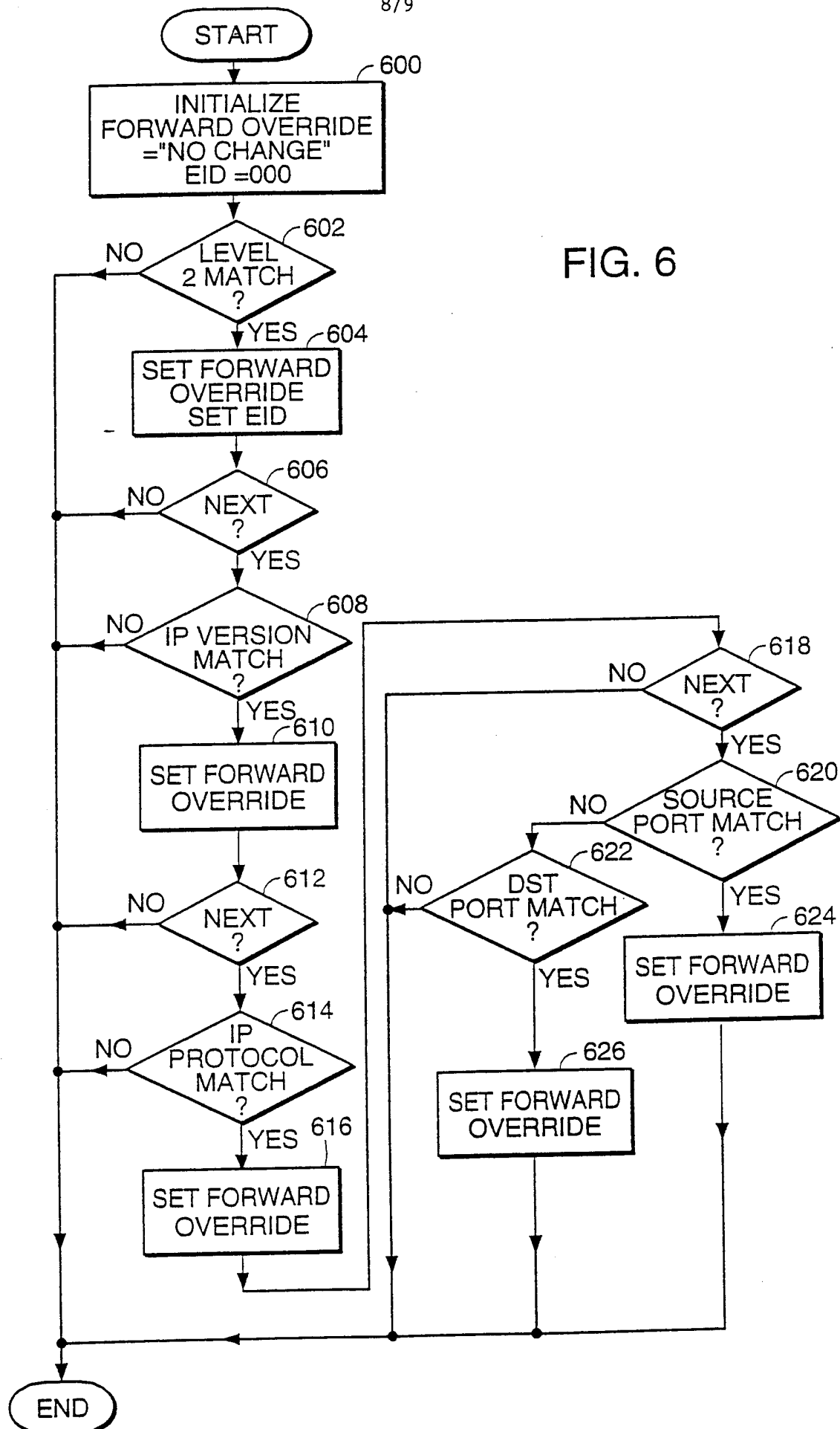
PRIOR ART

508b

(UDP) SOURCE PORT <u>552b</u>	(UDP) DEST. PORT <u>554b</u>
UDP MESG LENGTH <u>582</u>	UDP HEADER CHECKSUM <u>584</u>

FIG. 5E

PRIOR ART



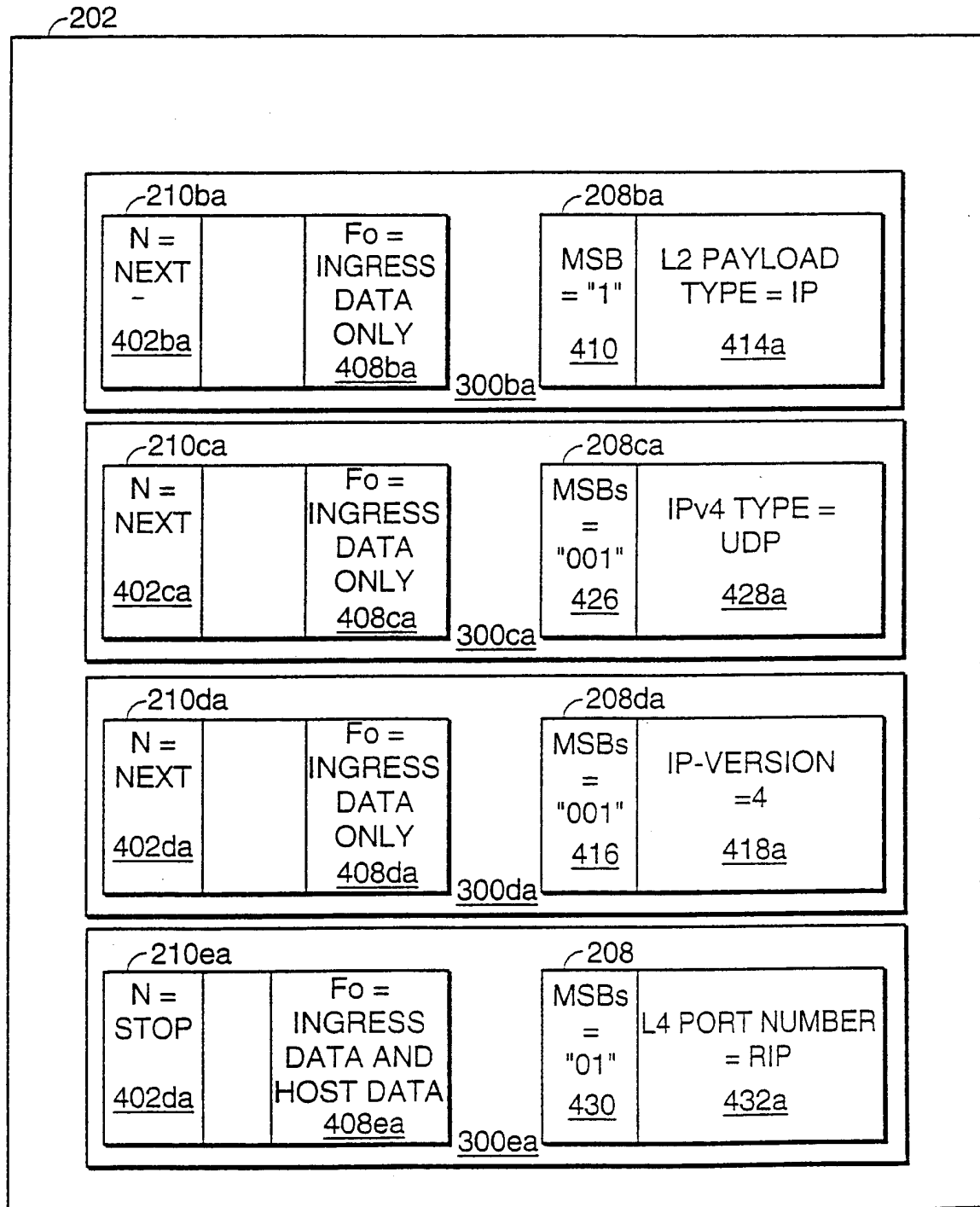


FIG. 7