EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent: 13.04.2005 Bulletin 2005/15

(21) Application number: 94921361.5

(22) Date of filing: 29.06.1994

(54) COMMUNICATION APPARATUS AND METHODS

ÜBERTRAGUNGSVORRICHTUNG UND VERFAHREN
APPAREIL ET PROCEDE DE COMMUNICATION

(84) Designated Contracting States: AT BE CH DE DK ES FR GB GR IE LI LU MC NL PT SE


(43) Date of publication of application: 08.05.1996 Bulletin 1996/19

(73) Proprietor: Marconi Intellectual Property (Ringfence) Inc.
Warrendale, Pennsylvania 15086 (US)

(72) Inventors:
• MURTHY, Manohar
Milpitas, CA 95035 (US)

• WAKERLY, John, F.
Mountain View, CA 94041 (US)

• LAURSEN, Arthur, I.
Mountain View, CA 94041 (US)

(74) Representative: O’Connell, David Christopher
HASELTINE LAKE,
Redcliff Quay
120 Redcliff Street
Bristol BS1 6HU (GB)

(56) References cited:
US-A- 4 817 080
US-A- 4 920 529


Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention.)
FIELD OF INVENTION

[0001] This invention relates to packet oriented multi-port bridges and routers and, in particular, to the monitoring of packet traffic arriving at the bridges and routers or generated internally.

DESCRIPTION OF RELATED ART

[0002] Multi-port bridges and routers allow the connection of two or more packet-based networks of possibly different types. Information in such networks is transmitted by means of packets, each containing data and appropriate addressing information. The purpose of the bridge or router is to relay packets between network segments (a process called forwarding) so that stations connected to different network segments may communicate. An example of a packet-based network protocol is that implemented by the IEEE 802.3 Ethernet standard.

[0003] Larger networks can be built by using multiple bridges, routers, or combinations thereof, and the extent and topology of a multi-bridge or multi-router network can be quite complex. Even small single-bridge networks can exhibit complex behavior which may affect performance, security or other aspects of network operations. Analysis of such issues and their correction is usually the responsibility of a network manager, who must examine transmissions on the network and make adjustments to network parameters.

[0004] Monitoring of packet networks can be carried out with monitoring devices such as Sniffer™ from Network General of Menlo Park, California or LANalyzer™ from Novell, Inc. of Provo, Utah. These devices are connected to the network medium, such as coaxial cable, and examine each network transmission regardless of the actual destination of the packets. Typically, network monitors provide the capability of filtering the examined transmission so that only packets with properties of interest to the network manager are captured or displayed. Facilities are usually provided to gather statistics, such as error rates, traffic between stations or groups of stations and so forth, as well as the packets themselves. Because of the need to capture and analyze large amounts of data, and the potential complexity of filtering, network monitors are expensive relative to other network components such as stations or bridges.

[0005] A serious limitation of prior-art network monitors is that the monitor must be connected physically to the network segment to be monitored. In a multi-port bridge where several network segments are connected by a bridge, it is only possible to examine one of the attached network segments at a time since the bridge isolates the physical media of the network segments. A further limitation is that the network monitor is not able to easily differentiate packets originating on the attached network segment and those originating on other network segments attached to the bridge and forwarded to the monitored network segment, especially if the packets have wrong addresses due to malfunction or sabotage. A router, moreover, replaces the source address of the packet by the router address, which makes it even more difficult for the network monitor to determine where the packet originated. In particular, it may be difficult or impossible for the monitor to isolate, for example, all the packets originating on a selected network segment.

[0006] One prior art approach to overcoming the limitation of connecting the monitor to only one network segment is the Distributed Sniffer™ from Network General. Each Sniffer is a network monitor coupled to a processing element that can be controlled over the network. If several network segments attached to a bridge are to be monitored, then one Distributed Sniffer must be attached to each physical network segment. Operation of each Distributed Sniffer can be controlled over the network from a network-attached station using an upper level protocol such as TELNET. With this approach, one station located on any attached network segment can view results obtained from each Distributed Sniffer. The clear disadvantage of this approach is the cost of multiple Sniffers. A further shortcoming is a limited ability to correlate information gathered on different Sniffers. In particular, a Sniffer detecting a packet may be unable to determine the network segment on which the packet originated even if that network segment is connected to another Sniffer which has detected the packet, because the two Sniffers may be unable to determine whether the packet they have detected is the same packet or two different packets.

[0007] Additionally, each Distributed Sniffer must use some portion of the bandwidth of the monitored network to send information to the monitoring station, and thus the performance of the monitored network is affected.

SUMMARY OF THE INVENTION

[0008] According to a first aspect of the invention, there is provided an apparatus for allowing communication between units connected to different network segments, the apparatus comprising:

- a plurality of ports for connection to the networks segments and to one or more monitoring systems; and
- first means for transmitting packets of information to one or more of the ports, wherein each packet of information comprises forwarding information to be used in determining the packet destination;
wherein the first means comprises means for transmitting each of one or more packets: (a) to one or more ports
determined from the packet destination if the destination includes a unit other than the apparatus; and, in addition, (b)
to one or more monitoring ports.

According to another aspect of the invention, there is provided a method for monitoring a network comprising
an apparatus interconnecting a plurality of network segments at least one of which comprises a network monitor, the
method comprising:

(a) obtaining, from each packet received or generated by the apparatus, forwarding information to be used in
determining the packet destination;
(b) if a packet destination includes a station other than the apparatus, then transmitting the packet, by the apparatus,
to one or more of the network segments in order to deliver the packet to the packet destination; and
(c) if a packet is to be delivered to a network monitor, then transmitting the packet, by the apparatus, to a network
segment comprising the network monitor.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objects, aspects and advantages of the invention will be better understood from the
following detailed description of the preferred embodiment of the invention with reference to the accompanying draw-
ings, in which:

Figure 1 illustrates an example multi-port bridge with six attached network segments;
Figure 2 depicts the format of a packet in conformance with the Ethernet standard;
Figure 3 sets out two formats of packet destination address;
Figure 4 exhibits a Bridging Table related to the example system;
Figure 5 shows the evaluation of a Custom Filtering Rule;
Figure 6 is a block diagram of the example bridge;
Figure 7 depicts shared memory data structures related to packet reception and transmission;
Figure 8 illustrates the format of a Packet Descriptor;
Figure 9 illustrates the format of the XMASK;
Figures 10A and 10B illustrate the reception of a packet and the transmission of a packet, respectively;
Figure 11 is a state diagram illustrating the sequencing of Packet Descriptor state;
Figure 12 illustrates the Forwarding Table for the example bridge;
Figure 13 illustrates the Broadcast/Multicast Table for the example bridge;
Figure 14 illustrates the Management Table for the example bridge;
Figure 15 depicts a Bridging Cache;
Figure 16 is a flowchart of the forwarding algorithm;
Figures 17A and 17B depict the Forwarding Table and Broadcast/Multicast Table, respectively, after modification
to support monitoring of incoming packets;
Figures 18A and 18B depict the Forwarding Table and Broadcast/Multicast Table, respectively, after modification
to support monitoring of forwarded packets;
Figure 19 illustrates the Management Table after modification to support monitoring of generated packets.
Figures 20A and 20B depict the Forwarding Table and Broadcast/Multicast Table, respectively, after modification
to support port-pair monitoring.

DESCRIPTION OF PREFERRED EMBODIMENT

The purpose of the bridge to be described below is to connect together multiple packet-based segments of
a network, allowing efficient communications between stations on each network segment and also between stations
located on different network segments connected to the bridge. It is also possible for stations on network segments
not connected to a common bridge to communicate, provided that there is at least one segment-to-segment path
between the stations.

The example provided here is of a bridge, however, the operation is similar for routers, and the extension to
routers will be clear to those skilled in the art.

In some embodiments, network segments attached to the bridge will employ a packet-based communication
protocol based on either Ethernet or FDDI. Other packet-based protocols are possible. The details of Ethernet and
FDDI protocols are well known and are documented in standards, particularly in IEEE Standard 802.3 for Ethernet and
ANSI Standard X3T9.5 for FDDI. The following review of packet communications is intended to establish a terminology
for further exposition of the preferred embodiment. The Ethernet scheme will be used as an example.
Figure 1 illustrates an example of a bridge with the port monitoring feature. In this example, the bridge provides bridging services to six attached network segments 2.0 through 2.5 via ports 3.0 through 3.5 numbered 0 through 5. Item 2.0 illustrates a typical Ethernet configuration based on "10Base5" technology or "10Base2" technology in which the attached stations are connected via a coaxial cable of the appropriate type. Such a cable would be terminated electrically via terminator 6. An alternative arrangement making use of "10BaseT" technology is shown for Port 5. In this instance, each station is connected via a twisted pair of wires to a unique connection on the port.

Each station illustrated has been given a unique name consisting of a letter followed by a port number. This naming is arbitrary and is used only to simplify discussion in order to illustrate the operation of the invention.

Figure 1 also shows the attachment of a monitoring device 9 to the monitoring port 10. In the example system and in the discussion to follow, the monitoring port will be Port 4. In some embodiments, the monitoring device 9 will be the only station on the network segment attached to the monitoring port 10. A supervisory terminal 12 may also be attached to the bridge to provide control of the bridge in general and of the port-monitoring feature in particular. In the example system, this attachment is made via a supervisory port 11, which is independent of the other ports illustrated and is only used to provide access to the bridge. It is possible, through the appropriate protocol, to provide access to supervisory terminal services at any of the connected stations 4. In the example system, any or all of the ports 3 may be monitored ports.

In order to simplify discussion, it will be assumed that all ports (excepting the supervisory port 11) in the example bridge 1 employ the Ethernet protocol. Under this protocol, stations communicate by sending and receiving packets of information. Figure 2 illustrates the logical composition of a single packet 13. The packet itself consists of a variable number of octets, or 8 bit data units, and is divided into fields of an integral number of octets as shown. The nomenclature and purpose of the fields is as follows:

- **Preamble 14**: A unique pattern used to synchronize the reception of packets
- **Destination Address 15**: A pattern that specifies the address of the station or stations 4 to receive the packet
- **Source Address 16**: A unique pattern specifying the address of the station 4 originating the transmission
- **Data 17**: The data to be transferred from the source station 4 to the destination station 4
- **FCS 18**: A check sequence over packet (excluding the preamble field) that is used by the destination stations to assess the validity of the received packet

Figure 3 illustrates the formation of the destination address 15 referred to also as DA. For purposes of illustration, two forms of DA may be used. One is the non-broadcast form 19 and the other is the broadcast form 20. A DA 15 consists of 6 octets, or 48 bits, and one of these bits, the Broadcast/Multicast flag 21, is used to differentiate between the two DA forms. When the Broadcast/Multicast flag is zero, the destination address consists of two components: a vendor code 22 and a device code 23. These codes are assigned by a central authority so that each station has a unique station address. The station address is physically associated with the station and is used to identify it, no matter where it may be located on a network, either on a single-segment network or in a larger network composed of multiple segments.

In the case where the Broadcast/Multicast flag 21 is set to one, the DA field 15 is interpreted differently. If the remaining bits of the DA (the Broadcast/Multicast address 24) are all ones, then the destination address is considered to designate all stations in the network, including stations on other segments connected to the bridge 1. In the case where the Broadcast/Multicast flag 21 is one, but the remaining bits of the DA 15 are not all ones, a multicast packet is indicated. The remaining bits then signify a subset of stations in the network that are destinations. Such stations may be attached to any one or different segments. The identification protocol is application dependent and will not be further specified here.

The source address field 16, also referred to as SA, identifies the source station using an addressing scheme as discussed for the DA 15. The SA field does not make use of the Broadcast/Multicast flag, and so the contents of the source address field always consist of the vendor code 22 and device number 23 only and thus uniquely identify the station originating the packet.

Within a single physical network segment 2, such as that composed of stations A0, B0, and C0 of Figure 1, the operation of the packet protocol is straightforward. Stations transmit packets 13 in which the SA 16 contains their unique station address and in which the DA 15 contains the address of the station they wish to communicate with. Alternately, they can form a DA 15 so that it has broadcast address format 20 and the packet 13 will be received by all stations attached to the segment.

Each station attached to the segment listens to all transmissions on that segment and checks the DA of each packet. A packet is intended for a station's address if a non-broadcast DA matches its station address exactly or a Broadcast/Multicast DA is received. In the case of a Broadcast/Multicast DA 20, the station will receive the packet if the Broadcast/Multicast address 24 matches according to application-specific rules.
The purpose of the bridge is to allow stations on different attached network segments to communicate with each other. There are several advantages to using a bridge rather than simply forming one large common network electronically. By use of a bridge, network segments can be smaller physically (i.e., each segment can contain fewer station) and, therefore, each segment's electrical limits can be more easily met. From a performance standpoint, the transmission capacity of a segment is limited, and therefore the rate at which messages can be transferred between stations on a segment is limited. By subdividing a large segment into a collection of smaller segments connected by a bridge, the overall usage of a connected segment will be reduced on average. In the illustrated example (Figure 1), for instance, stations on Port 2, such as A2 and C2, may communicate at full segment speed simultaneously while stations on another port, say Port 3, also use the full capacity of their attached segment.

The bridge comes into play when a station on one segment, such as A0, must communicate with a station (or stations) on another segment, say C3. In this case, the bridge must pass packets for the two communicating stations between appropriate ports, in this case between Port 0 and Port 3. Because a station might be portable and thus might move from one segment to another, it is necessary for the bridge to implement an adaptive algorithm. One such prior-art algorithm is described in U.S. Patent 4,597,078, entitled "Bridge Circuit for Interconnecting Networks." Bridges constructed according to this algorithm are referred to as "learning bridges." The following brief discussion of learning bridge operation is given here, since this is the preferred mode of bridge operation to which the invention applies.

The key to learning bridge operation is that each station always contains the unique address of the originating station in the SA field. In operation, the bridge examines and evaluates all packet transmissions on its attached ports. Using information derived from this process, the bridge builds a Bridging Table, as illustrated in Figure 4. Each Bridging Table Entry consists of a Station Address field and a Port Number field. There is one Bridging Table Entry for each station currently known to the bridge. In the Bridging Table Entry, the Port Number indicates the port to which the corresponding station is attached. Figure 4 illustrates a Bridging Table corresponding to the example bridge and network configuration shown in Figure 1. In the illustrated case, all bridge-attached station addresses are present in the Bridging Table. Because networks have a dynamic character, it is not necessarily the case that all station address/port number pairs will be in the Bridging Table at all times.

In a learning bridge, the Bridging Table is built dynamically by the bridge, as discussed later. Ignoring for now the port monitor feature, the Bridging Table is used to forward received packets to their destinations as follows:

1. If the destination address field of a received packet has the Broadcast/Multicast flag set to one, then the packet is forwarded to all attached ports, except the port on which it was received.
2. If the destination address field of a received packet has the Broadcast/Multicast flag set to zero, then the DA field contains a unique station address. The Bridging Table is accessed using the DA field of the received packet. If the Bridging Table contains an entry with a Station Address field matching the DA field of the received packet, then the corresponding Port Number field is retrieved. There are two cases to consider. If the retrieved port number is identical to the port number on which the packet was received, then the packet is destined for the same network segment as the sending station. In this case, no action is taken as the transmission has already occurred on the proper segment. The alternative case is where the retrieved port number does not match the port number on which the packet was received. In this case, the packet is forwarded to the port number indicated by the retrieved Bridging Table Entry.
3. If during the process outlined in 2 directly above, the destination address field of the received packet does not match the Station Address field of any Bridging Table Entry, then the packet is forwarded to all attached ports, except for the port on which it was received. This ensures that the destination station, if present on any bridge-attached segment, will receive the packet.

In a learning bridge, the Bridging Table is built dynamically, as packets are received. The bridge examines the source address field of each packet received on each port. If the station address in the source address field matches the Station Address field of an entry in the Bridging Table, and the port number on which the packet was received matches the port number field of that entry, then the Bridging Table is not modified. However, if the SA of a received packet matches a Station Address field of a Bridging Table Entry, but the port number on which the packet was received is not equal to the corresponding Port Number field for that entry, then the Port Number field is written with the port number on which the packet was received. Other actions, such as flushing the Bridging Cache may also be required. However, if the source address of the received packet does not match the Station Address field of any Bridging Table entry, then a new entry is added to the Bridging Table. This entry consists of a Station Address field containing the SA of the received packet and a corresponding Port Number field containing the port number of the port on which the packet was received.
[0028] When the bridge is initialized, the Bridging Table 25 is empty. As packets on the attached network segments are examined, Bridging Table Entries 26 are formed and added to the Bridging Table 25. By this process, the bridge "learns" the correspondence between the attached stations and the port to which they are attached. To accommodate for the fact that networks change and stations may be added, removed or moved from one segment to another, the learning bridge incorporates an aging algorithm to periodically remove Bridging Table Entries 26 that have not been used for a period of time.

[0029] It is also possible for a network administrator to configure "permanent entries" in the Bridging Table. This avoids the need for the bridge to learn such entries, and can also be used to enhance network security. For example, the bridge could be configured not to forward packets to any DA on a particular port unless the Bridging Table contains a permanent entry for that DA, matching that port.

[0030] A further complication of bridge operation is that bridge 1 is typically part of a large network consisting of many bridges and their attached segments. The topology of the network might include loops in which there is more than one network path between two bridges. This may be unintentional or intentional, for example where redundancy is required in the network. In the case of broadcast packets or when a received packet has a DA 15 field for which no matching Bridging Table Entry 26 exists, the packet is forwarded to all ports. If network loops are present, this forwarding activity can lead to infinite duplication and propagation of a packet. To prevent this, the learning bridge implements an algorithm, referred to as a "spanning-tree algorithm", that limits the ports to which packets of the type discussed above can be forwarded. This algorithm is well defined by IEEE Standard 802.1d. Operation of the spanning-tree algorithm requires that the bridge 1 form an internal map of the network to which it is attached. This is done by communicating periodically with other bridges attached to the segments that are attached to the bridge 1. Thus, there are instances in which the bridge itself may generate packets for transmission even though it has not received any specific packets instructing it to do so.

CUSTOM FILTERING OF PACKETS

[0031] In the forwarding operation discussed above, the bridge makes forwarding decisions based only on the DA field 15 of a packet 13. However, more useful bridge operation can be had by further qualifying the forwarding decision based on specific contents of each packet. Under this additional regime, the forwarding of certain packets may be suppressed (that is, they are filtered out) if conditions based on packet contents are met. These conditions are referred to here as custom filtering rules (CFRs) and are implemented through the use of templates 29 as shown in Figure 5.

[0032] A template 29 consists of three components, an offset 30, a 32-bit mask 31, and a 32-bit comparator 32. The template defines a test to be applied to a packet according to the following algorithm. First, the offset 30 is used to identify the start of a four-octet field. W. 33 of the packet. Offset 30 is expressed in octets from the start of the destination field 15. The identified field, W. 30 is then logically ANDed bit for bit with the 32-bit mask 31. The 32-bit result 34 is then compared at 35.1 logically (bit for bit) with the comparator 32 of the template yielding a logical result 35 which is true or false. If the result 35 of template evaluation is true (i.e., the result 34 equals the comparator 32), then the packet is not forwarded (i.e., it is filtered). In the preferred embodiment, the filtering algorithm is implemented with software; however, a strictly hardware implementation or a mixed hardware/software implementation is also possible.

[0033] It is intended that the bridge 1 should provide for a plurality of templates and that facilities be provided to allow for multiple templates to be evaluated against a given packet and for the results of such evaluation 35 to be combined according to the well-known rules of Boolean logic. Thus, filtering of a packet can be based on quite complex conditions. These complex conditions are referred to here as "custom filtering rules," or "CFRs." Through the proper construction of templates and CFRs, it is possible to filter out quite specific types of packets. For example all AppleTalk packets with an Appletalk source address of 15 (Hex) could be filtered by setting an offset of 16 (decimal), a mask of FF000000 (Hex), and a comparator of 15000000 (Hex). This might be used to prevent a particular station from communicating via AppleTalk protocol with selected other stations.

[0034] To further enhance the usefulness of CFRs, it is intended that the bridge allow for the association of CFRs with the port on which the packet is received, the SA 16 of the received packet, the DA 15 of the received packet, and the destination port (or ports) to which the packet is forwarded. Various combinations of such associations are also possible.

[0035] In the example bridge implementation, templates 29 and rules are defined through the use of the supervisory access terminal 12.

SUMMARY OF BRIDGE OPERATIONS

[0036] From the discussion above, it will be seen that the bridge is able to handle several situations reflecting the various packet generating and forwarding situations. In summary, these include:

EP 0 710 415 B1
EP 0 710 415 B1

1. Forwarding of a single packet from one port to another.
2. Forwarding of multicast and broadcast packets to more than one port and possibly all ports.
3. Forwarding of management packets generated from within the bridge.
4. Suppression of packet forwarding to particular ports due, for example, to operation of the spanning-tree algorithm or for security purposes.
5. Filtering (suppression) of packet forwarding due, for example, to the evaluation of custom filtering rules (CFRs).

ROUTER OPERATION

[0037] The discussion above has been related explicitly to bridges. However, the invention to be discussed also applies to routing. Packet routing involves receiving a packet on a port (i.e., from an attached network) and retransmitting it to another port based on information contained in the Data field 17. The DA of a packet to be routed will be either the station address of the router or a broadcast/multicast address. The SA 16 is the station address of the station or router originating the packet. The router may be physically and/or logically incorporated in a bridge. (Devices which combined router and bridge functionality are known as "brouters").

[0038] When a packet arrives at a router, the Data field 17 is parsed and examined. Specific protocols are defined for each type of packet to be routed and are indicated by sub-fields in the packet Data field 17. One of the sub-fields may be a network address which is a logical, rather than a physical, address indicating the ultimate destination of the packet. To route the packet, the router modifies the DA 15 to point to the next link or hop in the route and substitutes its own address for SA 16. Sub-fields of the Data field 17 may also be modified. In particular, there is usually a “hop count” indicating the maximum number of hops a packet may traverse before it is considered invalid or mis-routed. Other sub-fields of Data 17 may include control options, length, type, serial number, priority and so forth. These sub-fields are used to further specify the route.

[0039] CFRRs may be applied to routed packets just as they are to bridged packets. It is also the case that some routed packets are consumed by the router or possibly generated internally for transmission to other routers. Thus it will be seen that routed packets can generate packet forwarding situations similar to those arising for bridges, as discussed above under “Summary of Bridge Operations.”

BRIDGE HARDWARE IMPLEMENTATION

[0040] Figure 6 illustrates the hardware of the example bridge 1 in block diagram form. In keeping with the example bridge discussed above, only 6 port controllers 37.0 through 37.5 are illustrated, although those skilled in the art of hardware system design will see that the design may be readily extended to additional ports 3. Each port is based on the ILACC 32-bit Ethernet Controller, available from Advanced Micro Devices (AMD) of Sunnyvale, California. These controllers have the capability of sending and receiving packets directly to and from the shared memory 39 via the shared memory interface 38 without direct intervention of the bridge Main CPU 42 or I/O CPU 43. This process will be discussed further below.

[0041] The bridge contains two processors whose primary function is to examine packets stored in the shared memory and make the appropriate changes to shared memory tables and data structures to allow forwarding to take place. The main CPU 42 is based on the MIPS R3001 25 MHz processor from Integrated Device Technology (IDT) of Santa Clara, California. Associated with the chip is a 256K Byte cache memory in which frequently referenced portions of the real-time packet forwarding code and control data are held. An attached Program Memory 41 contains up to 8 MBytes of additional storage for less time critical software and data, such as that related to the supervisory access function. A serial interface 45 is connected to the Main CPU to provide the Supervisory Access Port 11. Also connected to the Main CPU 42 is a Floppy Disk 44 that provides a convenient means of updating the system software and saving configuration information, such as permanent Bridging Table entries and CFRRs, to be read at system startup.

[0042] A second processor, the I/O CPU 43, is based on a MIPS R3051 33 MHz processor also available from IDT. The primary purpose of this processor is to supervise the sending and receiving of packets 13, manage Packet Buffers in shared memory 39, handle packet reception errors and similar activities. This processor supports an onboard cache 46, which holds all of the I/O CPU code, thus providing very high performance.

[0043] Packets received from the ports and packets generated within the system to support management functions are stored in the shared memory 39, which is based on a 1.5 Mbyte array of SRAMs. The structure of typical shared memory 39 is described in patent application "Methods and Apparatus for Data Transfer Between Source and Destination Modules," Serial Number 07/304,053, now U.S. patent No. 5,237,670. The configured array has an aggregate bandwidth of 400 Mbytes/second. Shared memory is made available to the port controllers 37, the Main CPU 42, and the I/O CPU 43 via the shared memory interface 38. Each Port controller 37 is allocated 32 Kbytes of shared memory for received packets and 64 Kbytes of shared memory for transmitted packets.
PACKET DESCRIPTOR FORMAT

[0044] Packet forwarding is the process by which a received packet (or possibly one generated internally) is transmitted on one or more ports. While the forwarding decisions are made primarily by the Main CPU, the port controllers and the I/O CPU also participate in the mechanics of forwarding.

[0045] Figure 7 shows shared memory data structures involved in the reception, forwarding, and transmission of packets. Portions of these data structures can be manipulated by the port controllers, Main CPU and I/O CPU. Packets being processed are stored in Packet Buffers maintained in the Packet Buffer Pool. Each Packet Buffer is a contiguous shared memory area sufficient to hold an average sized Ethernet packet (of up to 256 octets). When longer packets must be handled, several Packet Buffers are used.

[0046] Because even a minimum-size packet contains a considerable number of bytes (64), it is desirable to handle packets indirectly. This is done by means of a Packet Descriptor, as shown in Figures 7 and 8. A Packet Descriptor is a shared-memory data structure and has five components. The packet pointer points to the actual packet data held in a Packet Buffer in the Packet Buffer Pool. As packets are processed, the Packet Descriptor may be copied or moved. (“Move” means to copy and delete the original.) However, the packet itself is not moved or copied, it is only referred to via the packet pointer. This indirect approach saves considerable shared-memory space and access bandwidth.

[0047] Flags within the Packet Descriptor indicate various conditions related to packet status, such as the presence of errors and their causes. Packet processing is directed by the State field of the Packet Descriptor. Details of packet processing and State field manipulation will be discussed below. The Length field indicates the length of the packet within the Packet Buffer.

[0048] Within the Packet Descriptor is the XMASK-Pointer that points to an XMASK indicating the destination port or ports (if any) to which the packet is to be transmitted. During forwarding, the Main CPU fills in the XMASK-Pointer field based on the forwarding algorithm and conditions in effect at the time a packet is processed. Execution of the forwarding algorithm produces a data quantity referred to as an XMASK and illustrated in Figure 9.

[0049] XMASK is simply a bit vector in which each bit indicates a port to which the packet is to be dispatched. Bit value “0” means, “Do not dispatch to the respective port”, and bit value “1” means, “Dispatch to the respective port”. If multiple bits are set, then the packet will be dispatched to each port indicated. If no bits are set, then the packet will not be dispatched (forwarded) to any port. For purposes of discussion and illustration, XMASK will be represented in binary with the rightmost bit being the least significant bit and designating port 0. Table I shows some examples of XMASK values for the 6 port example system.

<table>
<thead>
<tr>
<th>XMASK</th>
<th>ACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000</td>
<td>Do Not Dispatch</td>
</tr>
<tr>
<td>000001</td>
<td>Dispatch to Port 0 only</td>
</tr>
<tr>
<td>010011</td>
<td>Dispatch to Ports 0, 1 and 4</td>
</tr>
<tr>
<td>111111</td>
<td>Dispatch to All Ports</td>
</tr>
</tbody>
</table>

[0050] A computed XMASK value related to a packet is held in the XMASK Pool, a data structure in shared memory. Within the Packet Descriptor, the XMASK-Pointer field will point to the computed XMASK in the XMASK Pool. This allows multiple Packet Descriptors to point to the same XMASK value and facilitates dispatching the same packet to several ports, as would be required in a Broadcast/Multicast situation or when port monitoring is enabled.

[0051] For purposes of explaining the invention, some simplifications to the form of XMASK have been made and certain optimizations will be evident to those skilled in the art. For example, when XMASK designates only one destination port, the port number itself may be held directly in the XMASK-Pointer if a flag designating the alternate format is provided. This may be more efficient on some hardware systems.

PROCESSING OF PACKETS

[0052] Packet processing will be explained by way of example using Figures 10A and 10B, which illustrate changes to the shared memory data structure as packet processing progresses. Use will also be made of Figure 11 showing the sequence of processing steps. During packet processing, the actual Packet Buffer is not moved or copied in
shared memory 39. Instead, the Packet Descriptor 49 associated with that packet buffer is moved from one shared memory data structure to the next and possibly copied and/or modified. In particular, the State field 52 of the Packet Descriptor 49 is modified according to the sequence outlined in Figure 11 where enclosed text, such as 64, 65, 66, 67, 68, 69 and 70 represent states. What is shown in Figure 11 is the normal sequence of state processing where no errors have occurred.

[0053] In the example provided here, it is assumed that a packet will be received on Port 0 and sent to Ports 2 and 4. Initially, the configuration of memory will be as shown in Figure 10A. Associated with each port controller is a Receive Descriptor Ring (RDR) 72 and Transmit Descriptor Ring (TDR) 71 realized in shared memory 39. Figures 10A and 10B only illustrate the RDR for Port 0 and the TDR for Ports 2 and 4. Receive and Transmit Descriptor Rings (72 and 71) are circular data structures of well known design and are of a fixed size designed to hold an integral number of Packet Descriptors 49. Descriptor ring size is a design choice based on various system parameters of the particular implementation.

[0054] Initially, the RDR 72 will contain one or more Packet Descriptors 49, each with a State field 52 marked "Available for Reception" indicating that the associated Packet Buffers are available for the port controller 37 to fill with received packets. One Packet Descriptor in the RDR will be designated as to the next to be filled packet. Each "Available for Reception" Packet Descriptor 49 in the RDR will point to an empty Packet Buffer 47 in the Packet Buffer Pool 48 which is a data structure held in shared memory. With respect to the state diagram in Figure 11, the Packet Descriptor 49 is in the "Available for Reception" state 64. When a packet arrives at Port 0, the Port 0 controller 37.0 will transfer the data received to the Packet Buffer 47, as directed by the Packet Pointer field 50 of the Packet Descriptor 49. In the preferred implementation, this process is under the control of the Port Controller 37 and occurs independently of other processes on other port controllers and processes on the Main CPU 42 and I/O CPU 43. It will be recognized, however, that other approaches to providing independent processes are possible.

[0055] Once the Port 0 Controller has placed the received packet in the Packet Buffer 47, it will update the Packet Descriptor 49 by supplying the proper Length field 53, setting Flags 51 as required, and changing the State to "Received" 65 as shown in Figure 11. At this point, Port Controller 0 will access the next "Available for Reception" Packet Descriptor 49 in preparation for receiving a new packet.

[0056] Independently of the Port controller operation, the I/O CPU 43 supports a process that polls each Port RDR 72 and inspects the Packet Descriptors 49. When a Packet Descriptor 49 is found to be in the "Received" state 65, the I/O CPU 43 will process the packet checking for packet errors and updating receive statistics (such as number of packets received on this port). Upon completion of this process, the State field 52 of the Packet Descriptor 49 is marked as "Available for Forwarding" 66.

[0057] The Main CPU 42, working independently of the Port Controller 37 and the I/O CPU 43, periodically polls all RDRs 72 to determine if any queued packets are to be forwarded. Based on the SA 16 and DA 15 fields of the Packet 13 and upon the port number of the RDR 72, on which the packet is queued (RPORT), the Main CPU will carry out the Forwarding Algorithm as in Figure 16. The result of this process will be an XMASK value 55 designating the port or ports (possibly none) to which the packet 13 is to be forwarded. This XMASK 55 value will be placed in an available entry in the XMASK Pool 57 and the appropriate pointer to the entry will be entered into the XMASK-Pointer field 54 of the Packet Descriptor 49. The State field 52 of the Packet Descriptor is then changed to "Forwarded" 67.

[0058] Periodically, the I/O CPU 43 will scan the RDRs 72 to determine if any Packet Descriptors 49 are in a "Forwarded" state 67. When such a Packet Descriptor 49 is found, it will be copied to each TDR 71 (if any) as indicated by the set bits in the associated XMASK value 55. The State field 52 of each Packet Descriptor 49 copied to the TDR 71 is changed to "Available for Transmission" 68. Each Packet Descriptor 49 copied to a TDR 71 will contain a Packet Pointer 50 pointing to the packet in the Packet Buffer Pool 48 and an XMASK-Pointer 54 pointing to the XMASK value 55 in the XMASK pool 57. Once the I/O CPU 43 has copied a Packet Descriptor 49 to the appropriate TDRs 71, the Packet Descriptor in RDR 72 is marked "Available for Reception" 64 and linked to an empty Packet Buffer 47 from the Packet Buffer Pool 48. Figure 10B illustrates the situation after the example packet has been forwarded to the TDRs for ports 2 and 4.

[0059] Transmission of packets is carried out independently by the Port Controllers 37. Each Port Controller 37 scans its associated TDR 71 and on encountering a Packet Descriptor 49 with a State field 52 marked "Available for Transmission" 68 will begin transmitting the Packet 13 from Packet Buffer 47 to its associated port. Upon completion of the transmission, the State field 52 is marked "Transmitted" 69. When a packet is sent to two or more ports, it may be transmitted at different times, since packet transmission on a particular port depends on the state of the TDR associated with that port and the traffic queued on that port.

[0060] Clean up of the TDR 71 is carried out by the I/O CPU 43, which periodically scans all TDRs 71. When a Packet Descriptor 49 with a State field 52 marked "Transmitted" 69 is found, the bit in the XMASK 55 designated by the XMASK Pointer 54 corresponding to the port under examination is reset. If the XMASK 55 is now cleared, there are no more outstanding Packet Descriptors 49 associated with the packet 13; therefore, the Packet Buffer 47 is free and may be linked to a position on the Packet Buffer Free List 56 for later reuse. The Packet Descriptor 49 on the TDR 71 is marked
as "Free" making it available for reuse. Similarly, the XMASK is made available for reuse in the XMASK Pool.

Other issues related to packet processing, such as error handling and statistics gathering, have not been detailed here. The appropriate method of handling such issues is dependent upon the particular implementation and will be clear to those skilled in the art.

FORWARDING DATA STRUCTURES

Primary responsibility for forwarding packets resides in the Main CPU program. For purposes of illustrating the present invention, the data structures and operation of the forwarding algorithm will be discussed below. Only those features directly related to the port monitor feature will be explained. Initially, the discussion will be restricted to normal forwarding (i.e. when port monitoring is disabled). It will be seen that the proposed data structures support rapid computation of the XMASK value, which is used to direct packet forwarding. Once the normal case has been presented, the adjustments to the data structures necessary to provide port monitoring will be explained. These adjustments will also be seen to be particularly efficient in terms of implementation and execution.

Forwarding of a packet is based on several inputs and produces, as an output, an XMASK value. Required algorithmic inputs are DA - the destination address of a received packet, RPORT - the port number on which the packet was received, SA - the source address of the received packet, RSTATE - the state of the receiving port (RPORT), NG - the network groups, and the current CFRs in effect.

RSTATE reflects the state of the receiving port. This is a port specific indicator (one per port) and indicates whether packets arriving at a port from its attached segment should be forwarded and whether packets from other ports or the packets generated within the bridge itself (management packets) may be forwarded to the port. RSTATE for a port varies slowly relative to the reception of packets and usually remains static for a long period of time. For example, RSTATE changes during the execution of the spanning-tree algorithm as ports are enabled and disabled to prevent logical loops.

NG, Network Groups, define which bridge-connected ports are allowed to communicate. NG values are defined by a network administrator using the supervisory terminal or its network-based equivalent connection. Like RSTATE, NG values remain static for long periods of time relative to the transmission time of packets.

Because CFRs (Custom Filtering Rules) control packet transmission based on packet contents, possibly Data Field contents, CFRs when specified will have a dynamic effect on packet forwarding. That is, each packet arriving at a port (RPORT) with the same SA and DA may be forwarded differently. Thus CFRs must be evaluated for every packet forwarded between ports and specific addresses (SA and DA) for which CFRs are currently defined. The discussion below initially assumes that CFRs are not in effect.

In operation, forwarding will depend on several data structures, the Bridging Table (Figure 4), the Forwarding Table (Figure 12), the Broadcast/Multicast Table (Figure 13), Management Table (Figure 14), and the Bridging Cache (Figure 15). The structure of the Bridging Table has been discussed above.

Figure 12 illustrates the Forwarding Table. This data structure is a two-dimensional array. One index of the array is RPORT, the number of the port on which the packet to be forwarded was received. The other index is XPORT, the number of the port on which the packet is to be sent based on the DA field. XPORT is determined by accessing the Bridging Table with DA and retrieving the corresponding Port Number field. Entries in the Forwarding Table are XMASK values and reflect the current port-to-port connectivity of the bridge based on NG and RSTATE. For normal forwarding (port monitoring not in effect) XMASK will either be null (all zeroes) or will indicate a single port. Figure 12 illustrates an example Forwarding Table for a typical situation where all ports may communicate with each other. Null (all zeroes) XMASK values along the diagonal of the Forwarding Table indicate that if RPORT is equal to XPORT the packet should not be forwarded since the destination station is on the same port as the source station.

In the Forwarding Table example of Figure 12, it is also assumed that Port 4 is isolated logically from all other ports. In the monitoring examples that follow, the monitoring port will be Port 4. In some embodiments, the Monitoring Port is isolated logically so that only specifically identified monitored packets appear on the attached network segment. As a result "row 4" (i.e., RPORT=4) and "column 4" (i.e., XPORT=4) will contain null XMASK values.

The Broadcast/Multicast Table is illustrated in Figure 13. When a received packet indicates a Broadcast or Multicast address (that is when the Broadcast/Multicast flag is set), the Broadcast/Multicast Table is used in place of the Forwarding Table to develop XMASK. The Broadcast/Multicast Table is a one dimensional array indexed by RPORT. Each array entry is an XMASK value. Figure 13 illustrates a Broadcast/Multicast Table in which all ports are allowed to communicate with one another, except for Port 4, the monitoring port of the example. Therefore, each entry will have a 1 in each XMASK bit position except for bit 4 (the monitoring port) and the bit corresponding to RPORT (thereby preventing broadcast to the source port).

Network Groups (NG) affect the contents of the Forwarding Table and the Broadcast/Multicast Table.
The examples in Figures 12 and 13 assume that all ports are allowed to communicate. If a network administrator has restricted communication by defining network groups, then some of the "1" entries in Figures 12 and 13 will be set to 0. For example, if ports 0 and 1 were defined as belonging to one network group, and ports 2, 3, 4, 5 were defined as belonging to another, then all of the Forwarding Table entries in the outlined regions 90, 92 in Figure 12 would be 000000. Similarly, the Broadcast/Multicast Table bits in the outlined regions 94, 96 in Figure 13 would also be zeros. Subsequent examples do not show Network Groups, but the port-monitoring operations described later take into account the possibility that Network Groups have been defined.

[0072] For packets generated within the bridge or router related to network management and router operation, Management Table 82 (Figure 14) is used. This table is a one dimensional array indexed by MPORT 78, the port number on which the management related packet is to be sent. Figure 14 illustrates an example Management Table 82 in which each port is available for participation in management functions except for Port 4, the monitoring port 10.

[0073] Although the Bridging Table 25 and the Forwarding Table 80 are sufficient for the XMASK 55 calculation, performance of the forwarding process can be improved significantly by the introduction of an additional data structure designated as the Bridging Cache 83 and shown for the preferred embodiment in Figure 15. Conceptually, the Bridging Cache 83 contains multiple logical entries in which specific RPORT 85, SA 16 and DA 15 values are associated with an XMASK 55. Since this association changes slowly, it is usually possible to bypass the normal forwarding calculation and retrieve the XMASK 55 value directly from the Bridging Cache 83. Other factors, such as NG and RSTATE also change slowly and thus do not degrade the operation of the Bridging Cache 83 unduly.

[0074] When an XMASK value is calculated, the RPORT 85, SA 16 and DA 15 value used in the calculation are combined into an entry and placed in the Bridging Cache 83. When a new packet arrives for forwarding, the Bridging Cache 83 is accessed to determine if the RPORT 85, SA 16 and DA 15 associated with the packet match the RPORT 85, SA 16, and DA 15 of a Bridging Cache entry. If a match is found, then the XMASK value 55 from the Bridging Cache 83 can be used. Otherwise, the full forwarding algorithm must be carried out.

[0075] In the preferred embodiment, the Bridging Cache is partitioned into separate sub-caches - one associated with each RPORT 85. Since the maximum number of the receiving ports is relatively small, this is a very efficient method of handling part of the cache look-up. The Bridging Cache is accessed with the 3-tuple <RPORT,SA,DA>. Based on RPORT 85, the appropriate sub-cache associated 77 with the receive port is selected. Next the 96-bit value consisting of SA 16 concatenated with DA 15 is hashed using well-known techniques to produce a pointer to a Bridging Cache entry 79 in the selected sub-cache 77. A comparison is then made between the input SA, DA values and the SA, DA values in the selected Bridging Cache entry 79. If a match is obtained, the XMASK value 55 for that entry is retrieved. If no match occurs, the next Bridging Cache entry 79 is examined in like manner. This process continues until a match is found or a maximum number of attempts is made. Other approaches to accessing of the Bridging Cache 83 achieving the same result will be evident to those skilled in the art.

[0076] Use of the Bridging Cache 83 replaces the need to validate the received SA 16 in the Bridging Table 25, to look-up XPORT 86 in the Bridging Table 25 and to use the Forwarding Table 80 to retrieve XMASK 55. RPORT 85 and SA 16 are both used in the cache access so that changes to the port association of SA can be detected and accommodated as described next.

[0077] Bridging Cache entries 79 must be invalidated or flushed if they no longer reflect the outcome of the Bridging Algorithm. If, for example, the correspondence between SA 16 and RPORT 85 is found to be invalid, all Bridging Cache entries 79 with the corresponding SA 16 value in the RPORT sub-cache 77 must be cleared (the "flush" step in Figure 16). System level events may also cause a cache flush. For example, any change to the CFRs, the network groups NG, or the spanning-tree state may result in Bridging Cache entries 79 becoming invalid. In these cases, the offending Bridging Cache entries 79 must be removed or, if it is more efficient, all cache entries may be invalidated. Commands issued from the Supervisory Access Terminal 12 (or its network equivalent) may also cause a cache flush.

[0078] In some embodiments, any port or address to which a CFR is applied is excluded from the Bridging Cache 83. In other embodiments, Bridging Cache entries 79 include additional fields indicating the presence of a CFR and its point of application (DA, SA, RPORT). In some implementations this may allow CFR-related information to be accessed more quickly, depending on how data structures selected are realized.

[0079] It will be further recognized by those skilled in the art that alternatives to the Bridging Cache 83 data structures are possible while still preserving its performance-enhancing properties. Furthermore, it is also possible to associate the data structures discussed above, such as the Bridging Table 25; the Bridging Cache 83, with separate CPUs and memories, even though in the preferred embodiment they are implemented by code and data in the Main CPU 42 and Program Memory 41.

55 FORWARDING ALGORITHM

[0080] Packets that require forwarding may be incoming packets arriving at the bridge from its ports 3 or internally generated management packets. The forwarding algorithm discussed below operates in both cases and is also inde-
port is then able to
forward or placed on a network where DA is located. Using RPORT values to access the Bridging Table 83 will not contain a valid, static XMASK value in the Bridging Cache.

[0083] Full packet processing (i.e. when no matching cache entry is found), first involves a test of the DA 15 at 160.5 to determine if the Broadcast/Multicast flag 21 is set. If it is set then the XMASK value 55 is retrieved directly at 160.6 from the Broadcast/Multicast Table 81 using RPORT 85.

[0084] If the Broadcast/Multicast bit is not set, then the next step, at 160.7, is to access the Bridging Table 25 using SA 16 to determine if the source address and its associated RP value 85 are currently present and correct. If it is determined that the SA 16, RP 85 relationship has changed or SA 16 is not present, then the Bridging Table 25 must be updated at 160.8 to reflect the new relationship. When this occurs, it is also necessary to search the Bridging Cache 83 and invalidate any entries with a Source Address field 16 equal to the SA 16 from the received packet 13 (step 160.9).

[0085] If the Bridging Table 25 access shows that SA 16 is present and has the correct RP value, then the Bridging Table 25 is re-accessed at 160.10 using DA 15 in an attempt to retrieve the XPORT value 15 corresponding to DA. In the event that an entry corresponding to DA 15 is not found, the RP value will be used to access the Broadcast/Multicast Table 81 to retrieve an XMASK 55. This XMASK will indicate ports to which the packet will be directed in an attempt to place it on a network where DA is located.

[0086] When DA 15 is present in the Bridging Table 25, the XPORT value 86 will be retrieved indicating the port where DA 15 is located. Using RP 85 and XPORT 86, the Forwarding Table 80 is accessed at 160.11 and an XMASK 55 is retrieved.

[0087] After completion of the processing detailed here, an XMASK value 55 is available for use in dispatching. In cases where XMASK 55 is obtained from the Bridging Cache 83, dispatching may be done directly. In all other cases, it is necessary to check for the presence of custom filtering rules (steps 160.12, 160.13). Flags indicating the presence of custom filtering rules are maintained in the Bridge Table 25 for SA 16 and DA 15 and in separate rule tables associated with each port. When indicated, the appropriate CFRs are evaluated and the XMASK 55 is modified as required to produce a final value (step 160.14). This process can be quite involved and may affect performance significantly. When the processed packet is incoming (not generated) with a single station DA (not Broadcast or Multicast) and no CFRs are to be applied, the Bridging Cache 83 is updated at 160.15 from <RP,SA,DA> to reflect the new XMASK value 55.

[0088] In the preferred embodiment, packets with Multicast/Broadcast addresses are not placed in the Bridging Cache 83. There is nothing to prevent this from being done; however such packets are a relatively small part of the total packet traffic. Therefore, the Bridging Cache 83 is better used if entries are devoted exclusively to single address DAs 15. In situations with different traffic profiles from the preferred embodiment, it may be desirable to include multicast and broadcast addresses in the Bridging Cache 83.

DESCRIPTION OF PORT MONITORING FEATURE

[0089] Port monitoring is a process by which packets arriving at the bridge or generated internally may be copied to one or more monitoring ports 10 (Figure 1). A monitoring device 9 attached to the monitoring port 10 is then able to provide analysis of the monitored packets. In the preferred embodiment, the monitoring device 9 would be, for example, a Sniffer™ from Network General or a LANalyzer™ from Novell. These devices analyze packet traffic on a network and provide various diagnostic information enabling the network manager to locate problems, evaluate performance, and determine appropriate adjustments to network parameters.

[0090] Port monitoring is controlled from the supervisory access terminal 12. Using this terminal, the network manager may identify monitored ports 3 and monitoring ports 10. When port monitoring is enabled, packets associated with the monitored ports 3 will be forwarded to monitoring ports 10. In the preferred implementation, these packets are not actually copied, but the packet processing protocol described above is used in which only the Packet Descriptors...
are copied.

Port monitoring is controlled by the Supervisory Access Terminal using a simple command-line language. Table II illustrates the syntax of the commands. For each command, the prefix "pm" indicates that this is a port-monitoring command. There are three basic commands: "view", "viewpair" and "close". The first three commands shown in Table II are of the "view" type as identified by the command word "view". These commands designate a <monitored-port-number> and <monitoring-port-number>. There is also a field to designate the type of monitoring desired, either "incoming", "forwarded" or "generated". Incoming packets are those arriving at a designated monitored port. Forwarded packets are all packets forwarded to the designated monitored port from any other port. Generated packets are those generated internally by the bridge and forwarded to the monitored port. When the view command is given, all packets of the designated type will be "copied" from the port designated by <monitored-port-number> to the port designated by <monitoring-port-number>, in addition to their normal dispatching.

A "viewpair" command specifies a pair of ports and a monitoring port. Packets received on the port designated by the <source-monitored-port-number> and forwarded to the port designated by <destination-monitored-port-number> will be "copied" to the port designated by <monitoring-port-number>.

To terminate port monitoring, the "close" command is issued.

It is intended that the effect of individual commands be cumulative, that is, each processed command (except "close") will enable additional port monitoring. The effects of any commands issued since the previous "close" command will continue unchanged. Thus, through repeated application of the commands above, several ports may be designated as monitored ports, several ports may be designated as monitoring ports or various combinations thereof.

For illustrative purposes, a simple command language has been specified. It will be recognized that the command syntax outlined above could be enhanced using well-known techniques to provide a compound type of command allowing several monitored ports, monitoring ports, and packet forwarding situations to be specified in one command line or through other types of user interfaces.

### IMPLEMENTATION OF PORT-MONITORING COMMANDS

Up to this point, the bridge, its implementation and operation, has been illustrated only for normal operation, where port monitoring is disabled. Based on commands issued from the supervisory access terminal, numerous aspects of port monitoring may be enabled. For the preferred implementation, port monitoring involves modifying the data structures discussed previously to indicate the monitored and monitoring ports.

Modification will be illustrated for each of the monitoring situations: forwarded, incoming, generated, and port pairs.

To illustrate the effects of various port monitoring commands on the forwarding data structures, examples will be provided based on the use of Port as the designated monitoring port. For single-port monitoring, port will be used, and for port-pair monitoring, port will be the source-monitored-port and port will be the destination monitored port.

For all the examples, the assumption has been made that the monitoring port, Port, is used only for monitoring. Therefore, packets will only be forwarded to Port due to the enabling of the port-monitor function. This is the preferred mode of operation for the bridge when port monitoring is enabled, since other stations on the monitoring port may be unable to properly interpret packet traffic arising from the port-monitoring function.
Monitoring of Incoming Packets

[0099] If incoming packets on a port are to be monitored, then all packets received at the designated monitored port must be copied to the monitoring port. Packets are copied to the monitoring port even if they are not to be sent to any other port (i.e. they are consumed by the bridge). When monitoring of incoming packets is required, the Forwarding Table 80 and the Broadcast/Multicast Table 81 are modified. The Management Table 82 is not modified, since it affects only generated packets.

[0100] To enable monitoring of incoming packets on <monitored-port-number>, each entry in the Forwarding Table 80 where RPORT 85 is equal to <monitored-port-number> is modified. For each such entry, the XMASK bit corresponding to <monitoring-port-number> is set. Figure 17A shows the results of executing the command "pm view 2 on 4" on the example Forwarding Table of Figure 12. Since port 2 is to be monitored on Port 4, each XMASK entry 55 in "row 2" 60 will have bit 4 set.

[0101] A similar modification must be made to the Broadcast/Multicast Table 81. For the XMASK entry 55 where RPORT 85 is equal to <monitored-port-number> the XMASK bit corresponding to <monitoring-port-number> is set. Figure 17B illustrates the results of executing "pm view 2 on 4" on the example Broadcast/Multitask table 81 of Figure 13. Due to command execution, the entry 61 for RPORT = 2 has bit 4 corresponding to the monitoring port set. For the other entries, bit 4 is unchanged and remains cleared since Port 4 is isolated to support port monitoring in the preferred manner.

[0102] No modifications are made to the Management Table 82 to support the monitoring of incoming packets since XMASK values 55 in the table apply only to packets generated by the bridge.

Monitoring of Forwarded Packets

[0103] In the case where forwarded packets are to be monitored, it is necessary to modify XMASK entries 55 in the Forwarding Table 80 and Broadcast/Multicast Table 81 so that each packet forwarded to a designated <monitored-port-number> is also copied to the <monitoring-port-number>. No changes are made to the Management Table 82.

[0104] To accommodate monitoring of packets forwarded to <monitored-port-number>, the bit corresponding to <monitoring-port-number> must be set in the XMASK of each entry in the Forwarding Table 80 where XPORT is equal to <monitored-port-number> except for the entry where RPORT is equal to <monitored-port-number>. Figure 18A shows the result of executing the command "pm view forwarded 2 on 4" on the example Forwarding Table 80 of Figure 12. The modified entries are the "column 2" 73 and indicate that packets forwarded to port 2 should also be forwarded to Port 4, the monitoring port. The entry where RPORT=XPORT=2 has a null XMASK (000000) since packets received on port 2 should not be forwarded to that port.

[0105] Broadcast/Multicast packets can also be forwarded to the monitored port 3, thus it is necessary to modify the Broadcast/Multicast Table 81. Each XMASK entry in the Broadcast/Multicast Table 81 is modified by ORing the bit corresponding to <monitored-port-number> with the bit corresponding to <monitoring-port-number> and placing the result in the bit corresponding to <monitoring-port-number>. Figure 18B shows the results of modifying the Broadcast/Multicast Table of Figure 13 according to the command above. The result is that "bit column" 4 63, and the result is returned to bit column 4 63 indicating that each Broadcast/Multicast packet from an RPORT that is forwarded to port 2 should also be forwarded to Port 4.

Monitoring of Generated Packets

[0106] Monitoring of generated packets involves modifying only the Management Table 82. The Forwarding Table 80 and the Broadcast/Multicast Table 81 remain unchanged since they have no effect on the forwarding of generated packets originating within the bridge itself.

[0107] To enable monitoring of generated packets, each XMASK entry 55 in the Management Table 82 is modified so that the bit corresponding to <monitored-port-number> is ORed with the bit corresponding to <monitoring-port-number>, and the result is placed in the bit corresponding to <monitoring-port-number>. Figure 19 illustrates the result of the command "pm view generated 2 on 4" as applied to the example Management Table of Figure 14. "Bit column" 2 75 corresponding to the monitored port 2 has been ORed with "bit column" 4 76 representing the monitoring Port 4, and the result is returned to bit column 4 76.

Monitoring of Port Pairs

[0108] When port-pair monitoring is enabled, packets originating on a source monitored port 3 and forwarded to a destination monitored port 3 will be copied also to the monitoring port 10. To support this option, the Forwarding Table 80 and Broadcast/Multicast Table 81 must be modified but the Management Table 82 is unchanged.
The XMASK entry 55 in the Forwarding Table 80 designated by RPORT= source-monitored-port-number> and XPORT=<destination-monitored-port-number> is modified by setting the XMASK bit corresponding to <monitoring-port-number>. Figure 20A shows the results of applying the command "pm view pair 2 3 on 4" to the example Forwarding Table 80 of Figure 12. The modified entry 84 is highlighted.

In the Broadcast/Multicast Table 81, the entry corresponding to RPORT= source-monitored-port-number> is modified by ORing the XMASK bit corresponding to <destination-monitored-port-number> with the bit corresponding to <monitoring-port-number> and placing the result in the bit corresponding to <monitoring-port-number>. Figure 20B shows the result of applying the above command to the example Broadcast/Multicast Table of Figure 13. Only the RPORT=2 entry 61 corresponding to the source monitored port is modified by ORing XMASK bit 3 (corresponding to <destination-monitored-port-number>) with bit 4 (corresponding to <monitoring-port-number>) and placing the result in bit 4. No change to the Management Table 82 is required to enable monitoring of a port pair.

Close Command

The effects of port monitoring are cumulative. When a "pm close" command occurs, the Forwarding Table 80, Broadcast/Multicast Table 83 and Management Table 82 are simply restored to their original state before the application of any "pm view" or "pm viewpair" command.

Other Issues Related to Port Monitoring

Port Monitoring works naturally with the Bridging Cache 83. XMASK values obtained from the Forwarding Table 80 are placed in the Bridging Cache 83 provided no CFRs are in effect as would be the case in normal processing. Operation of the Bridging Cache 83 is unaffected by port monitoring.

CFRs may be applied to the monitoring port 10. However, in the preferred embodiment this was not allowed so as to improve efficiency.

Because the application of monitoring commands may change XMASK values 55, it is important to flush the Bridging Cache 83 whenever a monitoring command is given.

In some embodiments, packets with errors and those that are too large or too small are not "copied" to monitoring port 10. This could be done if it was desirable in a particular implementation.

The uncertainty as to the monitored packet's originating network segment is reduced. Indeed, the bridge knows precisely on which port each incoming packet was received, even if the packet's SA is wrong due to malfunction or sabotage. Thus the packets received on precisely selected port or ports can be isolated and forwarded to the network monitor even if the packets have wrong source addresses. The bridge debugging is therefore facilitated. In particular, security problems become easier to solve. The uncertainty is reduced also in the router embodiments because the router also determines the packet's receiving port independently of the packet's SA.

In some embodiments, different network segments connected to the bridge use different protocols. The bridge translates packets from one protocol format to another as needed. In particular, each packet transmitted to the monitoring port 10 is translated, if needed, to the format of the segment attached to the monitoring port. The bridge's ability to translate packets allows freedom in selecting the network segment attached to the monitoring port. For example, in some embodiments, some non-monitoring port segments are FDDI segments, and the segment attached to the monitoring port is an Ethernet segment. Using the Ethernet segment allows reducing the network cost because Ethernet network monitors are typically less expensive than FDDI network monitors.

PORT MONITORING IN ROUTERS

In router implementations, many of the basic issues related to port monitoring are also present. Packets are routed based on the content of the Data Field 17. Routing depends on data structures similar to those used in bridging. For instance, there may be a Routing Table for translating network addresses into ports and network destinations. There may also be Address Resolution Tables to translate router and host targets into actual Ethernet addresses, which are in turn used to update the DA 15 of the packet 13 to direct it to the next link or to the final destination. As in bridging, performance can be improved by caching recently calculated results. For example, the Network address, Ethernet address and port number may be cached together with an XMASK 55 value. Because the forwarding decision depends on many factors, such as router state, state of the next hop, and the state of the route it is not possible to compute the XMASK 55 in a static, direct manner as can be done for bridging. When monitoring is enabled, the XMASK 55 derived from the Routing Table and Address Resolution Table is modified algorithmically according to the monitoring currently enabled. This XMASK 55 is then cached for later reference in the Routing Cache.

When forwarding an incoming packet, a router normally modifies a portion of the packet header. For example, it replaces the SA and DA of the received packet with its own SA and the DA of the next hop, and it may update a hop
count. When port monitoring is in effect, the packet forwarded on the monitoring port is the modified packet, not exactly the received packet.

[0121] In some embodiments, in order to forward exactly the received packet to the monitoring port, the router makes a copy of the received packet before modifying it. It will be apparent to those skilled in the art that it may not be necessary to copy the entire packet, only the modified part, if the Port controllers can "gather" multiple buffers into a single packet for transmission. In this case, an extra buffer can be allocated for the copied and modified part of the packet, while the original buffer can be used to forward the packet to the monitoring port (or vice versa).

[0122] While the invention has been described in terms of a preferred implementation based on a specific bridge and network example, those skilled in the art will recognize that the invention can be practiced with modification and extension within the scope of the appended claims.

Claims

1. An apparatus for allowing communication between units connected to different network segments, the apparatus comprising:

   a plurality of ports (3,10) for connection to the networks segments and to one or more monitoring systems; and
   first means (11,12) for transmitting packets of information to one or more of the ports (3,10), wherein each packet of information comprises forwarding information to be used in determining the packet destination;

   wherein the first means (11,12) comprises means for transmitting each of one or more packets: (a) to one or more ports (3) determined from the packet destination if the destination includes a unit other than the apparatus; and, in addition, (b) to one or more monitoring ports (10).

2. The apparatus of Claim 1 wherein each monitoring port is adapted to allow connection to a network segment accessible to a network monitor (9).

3. The apparatus of any one of the preceding claims wherein the first means is adapted to allow, in response to a command, transmission of a packet: (a) to one or more ports determined based on the packet destination and, in addition, (b) to one or more monitoring ports.

4. The apparatus of any one of the preceding claims further comprising second means for specifying to the first means which packets are to be transmitted to one or more monitoring ports, wherein the second means is adapted to allow specifying such packets at any time during the operation of the first means.

5. The apparatus of any one of the preceding claims wherein each of the ports is adapted to allow connection to a network segment.

6. The apparatus of any one of the preceding claims wherein the plurality of ports comprises:

   one or more ports for connection to one or more network segments employing a first protocol format; and
   one or more ports for connection to one or more network segments employing a second protocol format different from the first protocol format;

   wherein the one or more monitoring ports include a first monitoring port for connection to a network segment employing the first protocol format; and
   wherein the first means comprises means to translate packets from the second protocol format to the first protocol format to allow packets received from a network segment employing the second protocol format to be transmitted to the first monitoring port.

7. The apparatus of any one of the preceding claims wherein the first means comprises a store for storing one or more data structures that allow the first means to determine, using a packet's forwarding information, all the ports, if any, to which the packet is to be transmitted; and

   wherein the apparatus further comprises means to modify the data structures in response to commands so as to define which packets would be transmitted to which monitoring ports.

8. The apparatus of Claim 7 wherein the commands include a command to transmit to a monitoring port packets
incoming on a selected port.

9. The apparatus of Claim 7 or 8 wherein the commands include a command to transmit to a monitoring port packets forwarded to a selected port for transmission.

10. The apparatus of Claim 7, 8 or 9 further comprising means for generating packets, wherein the commands include a command to transmit to a monitoring port packets generated by the generating means.

11. The apparatus of any one of Claims 7-10 wherein the commands include a command to transmit to a monitoring port packets incoming on a first selected port and forwarded to a second selected port for transmission.

12. The apparatus of any one of the preceding claims wherein the first means comprises means for applying one or more filtering rules based on the specific contacts of each packet to determine which packets are to be transmitted to a monitoring port.

13. The apparatus of any one of the preceding claims wherein the packets have a variable number of data units.

14. The apparatus of any one of the preceding claims wherein:

   each packet's forwarding information comprises a source address and a destination address;
   the first means comprises a store S1 for storing one or more entries of the form (RPORT, SA, DA, XP) wherein:

   RPORT identifies one or the ports;
   SA is a source address;
   DA is a destination address; and
   XP identifies zero, one, or more than one ports to which a packet which is received on port RPORT and which has a source address SA and a destination address DA is to be transmitted; and
   the apparatus further comprises means M for searching the store S1, when a packet is received, for an entry whose RPORT identifies the port on which the packet is received and whose SA and DA correspond, respectively, to a source address and a destination address of the received packet,

   wherein, if such an entry is found, the means M is adapted to determine from XP of the found entry those ports to which the received packet is to be transmitted.

15. The apparatus of any one of Claims 1-13 wherein:

   each packet's forwarding information comprises a source address and a destination address;
   the first means comprises a store S2 for storing one or more entries each of which comprises a triple (SA, DA, XP)

   wherein:
   
   SA is a source address:
   
   DA is a destination address; and
   
   XP identifies zero, one, or more than one ports to which a packet having a source address SA and a destination address DA is to be transmitted; and
   
   the apparatus further comprises means M for searching the store S2, when a packet is received, for an entry whose SA and DA correspond, respectively, to a source address and a destination address of the received packet, wherein, if such an entry is found, the means M is adapted to determine from XP of the found entry those ports to which the received packet is to be transmitted.

16. The apparatus of Claim 14 or 15 wherein, for each entry, XP is a map in which, for a port, one or more bits specify whether information is to be transmitted to said port.

17. The apparatus of any one of the preceding claims wherein the first means comprises:
a store for storing (a) packets of information in buffers, (b) for each port P1 of one or more of the ports, a first
data structure for containing one or more pointers to one or more buffers that store packets received on the
port P1, and (c) for each port P2 of one or more of the ports, a second data structure for containing one or
more pointers to one or more buffers that store packets to be transmitted on the port P2;
means M2 for transmitting, on each port having a second data structure, packets stored in buffers pointed to
by pointers contained in the second data structure of the port; and
means for copying at least one pointer from the first data structure of a first port to the second data structures
of one or more second ports to forward a packet received on the first port to the one or more second ports.

18. A method for monitoring a network comprising an apparatus interconnecting a plurality of network segments at
least one of which comprises a network monitor (9), the method comprising:

(a) obtaining, from each packet received or generated by the apparatus, forwarding information to be used in
determining the packet destination;
(b) if a packet destination includes a station other than the apparatus, then transmitting the packet, by the
apparatus, to one or more of the network segments in order to deliver the packet to the packet destination;
and
characterised in that
(c) if a packet is to be delivered to a network monitor (9), then transmitting the packet, by the apparatus, to a
network segment comprising the network monitor (9).

19. The method of Claim 18 wherein for at least one packet whose destination includes a station other than the ap-
paratus and which is to be delivered to a network monitor, the steps(b) and (c) are performed in response to a
command.

20. The method of Claim 18 or 19 wherein:
one or more of the network segments employ a first protocol format;
one or more of the network segments employ a second protocol format different from the first protocol format;
at least one network segment comprising a network monitor employs the first protocol format; and
the method further comprises translating one or more packets received on one or more network segments
employing the second protocol format from the second protocol format to the first protocol format and trans-
mitting such packets to a network segment which comprises a network monitor and employs the first protocol
format.

21. The method of any one of Claims 18-20 further comprising:

storing in a store one or more data structures for determining, using a packet's forwarding information, all the
network segments, if any, to which the packet is to be transmitted; and
modifying the data structures in response to a command so as to define which packets should be transmitted
to which network segments comprising network monitors.

22. The method of Claim 21 wherein the command is a command to transmit to a network segment comprising a
network monitor packets incoming from a selected network segment.

23. The method of Claim 21 wherein the command is a command to transmit to a network segment comprising a
network monitor packets transmitted in step (b) to a selected network segment.

24. The method of Claim 21 further comprising generating packets by the apparatus,
wherein the command is a command to transmit to a network segment comprising a network monitor packets
generated by the generating step.

25. The method of Claim 21 wherein the command is a command to transmit to a network segment comprising a
network monitor packets received from a first selected network segment and transmitted in step (b) to a second
selected network segment.

26. The method of any one of Claims 18-25 wherein the step (c) comprises applying one or more filtering rules based
on the specific contents of each packet to determine whether a packet is to be delivered to a network monitor.
27. The method of any one of Claims 18-26 wherein the packets have a variable number of data units.

28. The method of any one of the Claims 18-27 further comprising the steps of:

storing packets of information in buffers; and

providing for each port P2 of said ports a data structure for containing one or more pointers to one or more buffers that store packets to be transmitted on the port P2,

wherein at least one of steps (b) and (c) comprises transmitting on each port P2 packets stored in buffers pointed to by pointers in the data structures of the port P2.

29. The method of Claim 28 wherein each of steps (b) and (c) comprises transmitting on each port P2 packets stored in buffers pointed to by pointers in the data structures of the port P2.

30. The method of Claim 28 or 29 further comprising:

receiving a packet PC1 on one of the ports;

determining one or more ports to which the packet PC1 is to be transmitted; and

inserting a pointer to a buffer storing at least a portion of the packet PC1 into the data structure of each port on which the packet PC1 is to be transmitted.

Patentansprüche

1. Vorrichtung zum Ermöglichen von Übertragung zwischen Einheiten, die mit verschiedenen Netzwerksegmenten verbunden sind, wobei die Vorrichtung folgendes aufweist:

   eine Anzahl an Anschlüssen (3, 10) zur Verbindung mit den Netzwerksegmenten und mit einem oder mehreren Überwachungssystemen; und

   erste Mittel (11, 12) zum Übertragen von Informationspaketen an einen oder mehrere der Anschlüsse (3, 10), wobei jedes Informationspaket Übermittlungsinformation aufweist, die beim Bestimmen des Bestimmungsortes des Pakets zu verwenden ist;

   wobei die ersten Mittel (11, 12) Mittel zum Übertragen jedes der einen oder mehreren Pakete (a) zu einem oder mehreren Anschlüssen (3), die aus dem Paketbestimmungsort bestimmt werden, wenn der Bestimmungsort eine von der Vorrichtung verschiedene Einheit umfasst; und, zusätzlich, (b) zu einem oder mehreren Überwachungsanschlüssen (10) umfassen.

2. Vorrichtung nach Anspruch 1, wobei jeder Überwachungsanschluss (10) dafür ausgelegt ist, eine Verbindung zu einem Netzwerksegment zu ermöglichen, welches für einen Netzwerkmonitor (9) zugänglich ist.

3. Vorrichtung nach einem der vorstehenden Ansprüche, wobei das erste Mittel dafür ausgelegt ist, als Antwort auf eine Anweisung die Übertragung eines Pakets (a) zu einem oder mehreren Anschlüssen, welche basierend auf dem Paketbestimmungsort bestimmt werden, und, zusätzlich, (b) zu einem oder mehreren Überwachungsanschlüssen zu ermöglichen.

4. Vorrichtung nach einem der vorstehenden Ansprüche, weiter aufweisend zweite Mittel, um den ersten Mitteln anzugeben, welche Pakete zu einem oder mehreren Überwachungsanschlüssen zu übertragen sind, wobei das zweite Mittel dafür ausgelegt ist, solche Pakete zu einem beliebigen Zeitpunkt während des Betriebs der ersten Mittel zu bezeichnen.

5. Vorrichtung nach einem der vorstehenden Ansprüche, wobei jeder der Anschlüsse dafür ausgelegt ist, die Verbindung zu einem Netzwerksegment zu ermöglichen.

6. Vorrichtung nach einem der vorstehenden Ansprüche, wobei die Anzahl an Anschlüssen folgendes aufweist:

   einen Anschluss oder mehrere Anschlüsse zur Verbindung mit einem oder mehreren Netzwerksegmenten unter Einsatz eines ersten Protokollformats; und
einen Anschluss oder mehrere Anschlüsse zur Verbindung mit einem oder mehreren Netzwerksegmenten unter Einsatz eines zweiten Protokollformats, das vom ersten Protokollformat verschieden ist;

wobei der Überwachungsanschluss oder die Überwachungsanschlüsse einen ersten Überwachungsanschluss zur Verbindung mit einem Netzwerksegment unter Einsatz des ersten Protokollformats aufweisen; und

wobei das erste Mittel Mittel zum Übertragen der Pakete aus dem zweiten Protokollformat in das erste Protokollformat aufweist, um es zu ermöglichen, dass Pakete, die von einem Netzwerksegment empfangen werden, welches das zweite Protokollformat verwendet, an den ersten Überwachungsanschluss geschickt werden können.

7. Vorrichtung nach einem der vorstehenden Ansprüche, wobei das erste Mittel einen Speicher zum Speichern einer oder mehrerer Datenstrukturen aufweist, welche es ermöglichen, dass das erste Mittel unter Einsatz einer Versendeinformation eines Pakets alle Anschlüsse, an die das Paket zu übertragen ist, wenn solche vorhanden sind, bestimmt; und

wobei die Vorrichtung darüber hinaus Mittel zum Modifizieren der Datenstruktur als Antwort auf Anweisungen umfasst, so dass definiert werden kann, welche Pakete zu welchen Überwachungsanschlüssen zu übertragen sind.

8. Vorrichtung nach Anspruch 7, wobei die Anweisungen eine Anweisung zum Übertragen von Paketen, die auf einem ausgewählten Anschluss ankommen, zu einem Überwachungsanschluss umfassen.

9. Vorrichtung nach Anspruch 7 oder 8, wobei die Anweisungen eine Anweisung zum Übertragen von Paketen, welche zu einem ausgewählten Anschluss für die Übertragung weitergeleitet werden, an einen Überwachungsanschluss aufweisen.

10. Vorrichtung nach Anspruch 7, 8 oder 9, weiter umfassend Mittel zum Erzeugen von Paketen, wobei die Anweisungen eine Anweisung zum Übertragen von Paketen, die vom Erzeugungsmittel erzeugt wurden, an einen Überwachungsanschluss umfassen.

11. Vorrichtung nach einem der Ansprüche 7 bis 10, wobei die Anweisungen eine Anweisung zum Übertragen von Paketen, welche auf einem ersten ausgewählten Anschluss ankommen und für die Übertragung zu einem zweiten ausgewählten Anschluss weitergeleitet werden, an einen Überwachungsanschluss aufweisen.

12. Vorrichtung nach einem der vorstehenden Ansprüche, wobei die ersten Mittel Mittel zum Anlegen einer oder mehrerer Filterregeln basierend auf den spezifischen Kontakten jedes Pakets aufweist, um zu bestimmen, welche Pakete zu einem Überwachungsanschluss zu übertragen sind.

13. Vorrichtung nach einem der vorstehenden Ansprüche, wobei die Pakete eine variable Anzahl an Dateneinheiten umfassen.

14. Vorrichtung nach einem der vorstehenden Ansprüche, wobei:

die Übertragungsinformation jedes Pakets eine Quellenadresse und eine Bestimmungsadresse aufweist; das erste Mittel einen Speicher S1 zum Speichern einer oder mehrerer Einträge der Form (RPORT, SA, DA, XP) aufweist, wobei:

RPORT einen Anschluss oder die Anschlüsse bezeichnet;
SA eine Quellenadresse ist;
DA eine Bestimmungsadresse ist; und
XP null Anschlüsse, einen oder mehr als einen Anschluss bezeichnet, an den ein Paket, welches auf dem Anschluss RPORT empfangen wird und eine Quellenadresse SA und eine Bestimmungsadresse DA aufweist, zu übertragen ist; und

die Vorrichtung darüber hinaus Mittel M aufweist, um den Speicher beim Empfang eines Pakets nach einem Eingang abzusuchen, dessen RPORT den Anschluss bezeichnet, an dem das Paket empfangen wird, und dessen SA und DA der Quellenadresse bzw. der Bestimmungsadresse des empfangenen Pakets entsprechen, wobei das Mittel M dafür ausgelegt ist, aus XP des aufgefundenen Eingangs jene Anschlüsse herauszufinden, an die das empfangene Paket zu übertragen ist, wenn ein solcher Eingang gefunden wurde.
15. Vorrichtung nach einem der Ansprüche 1 bis 13, wobei:

die Übertragungsinformation jedes Pakets eine Quellenadresse und eine Bestimmungsadresse aufweist;
das erste Mittel einen Speicher S2 zum Speichern eines oder mehrerer Eingänge aufweist, von denen jeder
ein Triple (SA, DA, XP) aufweist, wobei:

SA eine Quellenadresse ist;
DA eine Bestimmungsadresse ist; und
XP null Anschlüsse, eins oder mehr als einen Anschluss, an den ein Paket, welches eine Quellenadresse
SA und eine Bestimmungsadresse DA aufweist, zu übertragen ist, bezeichnet; und
die Vorrichtung darüber hinaus Mittel M aufweist, um den Speicher S2 beim Empfang eines Pakets nach einem
Eingang abzusuchen, dessen SA und DA einer Quellenadresse bzw. einer Bestimmungsadresse des emp-
fangenen Pakets entsprechen, wobei das Mittel M dafür ausgelegt ist, aus XP des aufgefundenen Eingangs
jene Anschlüsse zu bestimmen, an die das empfangene Paket zu übertragen ist, wenn ein solcher Eingang
aufgefunden wurde.

16. Vorrichtung nach Anspruch 14 oder 15, wobei XP für jeden Eingang ein Abbild ist, auf dem ein oder mehrere Bits
für einen Anschluss angeben, ob Information an jenen Anschluss zu übertragen ist.

17. Vorrichtung nach einem der vorstehenden Ansprüche, wobei die ersten Mittel folgendes aufweisen:

 einen Speicher zum Speichern (a) von Informationspaketen in Puffern, (b) einer ersten Datenstruktur für jeden
Anschluss P1 der einen oder mehrerer Anschlüsse, um einen oder mehrere Zeiger auf eine oder mehrere
Pufferspeicher zu enthalten, welche an dem Anschluss P1 empfangene Pakete speichern, und (c) einer zwei-
ten Datenstruktur für jeden Anschluss P2 der einen oder mehreren Anschlüsse, um einen oder mehrere Zeiger
auf einen oder mehrerer Pufferspeicher zu enthalten, welche Pakete speichern, die auf den Anschluss P2 zu
übertragen sind;
Mittel M2 zum Übertragen von Paketen, welche in Pufferspeichern gespeichert sind, auf welche die Zeiger
gerichtet sind, die in der zweiten Datenstruktur des Anschlusses enthalten sind, auf jedem Anschluss, der
eine zweite Datenstruktur aufweist; und
Mittel zum Kopieren mindestens eines Zeigers aus der ersten Datenstruktur eines ersten Anschlusses auf die
zweiten Datenstrukturen eines oder mehrerer zweiter Anschlüsse, um ein Paket, das auf dem ersten Anschluss
empfangen wurde, zu dem zweiten Anschluss oder den zweiten Anschlüssen zu übertragen.

18. Verfahren zum Überwachen eines Netzwerks, aufweisend eine Vorrichtung, welche eine Anzahl an Netzwerkseg-
menten miteinander verbindet, von denen zumindest eines einen Netzwerkmonitor (9) umfasst, wobei das Ver-
fahren die folgenden Schritte aufweist:

(a) Gewinnen von Übertragungsinformation aus jedem Paket, das von der Vorrichtung empfangen oder er-
zeugt wurde, welche Übertragungsinformation beim Bestimmen des Paketbestimmungsortes zu verwenden
ist;
(b) wenn ein Paketbestimmungsort eine Station umfasst, die von der Vorrichtung verschieden ist, Übertragen

des Pakets durch die Vorrichtung zu einem oder mehreren der Netzwerksegmente, um das Paket an den
Paketbestimmungsort zu übermitteln; **dadurch gekennzeichnet, dass**
(c) wenn ein Paket an einen Netzwerkmonitor (9) zu übermitteln ist, das Paket durch die Vorrichtung auf ein
Netzwerksegment übertragen wird, das den Netzwerkmonitor (9) enthält.

19. Verfahren nach Anspruch 18, wobei für mindestens ein Paket, dessen Bestimmungsort eine Station enthält, die
vom der Vorrichtung verschieden ist, und das an einen Netzwerkmonitor zu übermitteln ist, die Schritte (b) und (c)
as Antwort auf eine Anweisung ausgeführt werden.

20. Verfahren nach Anspruch 18 oder 19, wobei:

(eines oder mehrere der Netzwerksegmente ein erstes Protokollformat verwenden;
eines oder mehrere der Netzwerksegmente ein zweites Protokollformat verwenden, das vom ersten Proto-
kollformat verschieden ist;
mindestens ein Netzwerksegment, das einen Netzwerkmonitor umfasst, das erste Protokollformat verwendet;
und das Verfahren darüber hinaus das Übertragen eines oder mehrerer Pakete, die auf einem oder mehreren Netzwerksegmenten empfangen wurden, welche das zweite Protokollformat verwenden, aus dem zweiten Protokollformat in das erste Protokollformat und das Übermitteln solcher Pakete an ein Netzwerksegment, welches einen Netzwerkmonitor umfasst und das erste Protokollformat verwendet, aufweist.

21. Verfahren nach einem der Ansprüche 18 bis 20, weiter aufweisend:

Speichern einer Datenstruktur oder mehrerer Datenstrukturen zum Bestimmen, unter Verwendung der Übertragungsinformation eines Pakets, aller Netzwerksegmente, an die das Paket zu übertragen ist, falls es solche gibt, in einem Speicher; und

Modifizieren der Datenstrukturen als Antwort auf eine Anweisung, um zu definieren, welche Pakete zu welchen Netzwerksegmenten, die Netzwerkmonitor umfassen, zu übertragen sind.

22. Verfahren nach Anspruch 21, wobei die Anweisung eine Anweisung ist, aus einem ausgewählten Netzwerksegment hereinkommende Pakete an ein Netzwerksegment zu übertragen, welches einen Netzwerkmonitor umfasst.

23. Verfahren nach Anspruch 21, wobei die Anweisung eine Anweisung ist, Pakete, die in Schritt (b) an ein ausgewähltes Netzwerksegment übertragen wurden, an ein Netzwerksegment zu übertragen, welches einen Netzwerkmonitor umfasst.

24. Verfahren nach Anspruch 21, weiter umfassend das Erzeugen von Paketen durch die Vorrichtung, wobei die Anweisung eine Anweisung ist, Pakete, die im Erzeugungsschritt erzeugt wurden, an ein Netzwerksegment zu übertragen, welches einen Netzwerkmonitor aufweist.

25. Verfahren nach Anspruch 21, wobei die Anweisung eine Anweisung ist, Pakete, die von einem ausgewählten ersten Netzwerksegment empfangen wurden und in Schritt (b) auf ein zweites ausgewähltes Netzwerksegment übertragen wurden, zu einem Netzwerksegment zu übertragen, welches einen Netzwerkmonitor aufweist.

26. Verfahren nach einem der Ansprüche 18 bis 25, wobei der Schritt (c) das Anwenden einer oder mehrerer Filterregeln basierend auf den spezifischen Inhalten jedes Pakets aufweist, um zu bestimmen, ob ein Paket an einen Netzwerkmonitor zu übertragen ist.

27. Verfahren nach einem der Ansprüche 18 bis 26, wobei die Pakete eine variable Anzahl an Dateneinheiten aufweisen.

28. Verfahren nach einem der Ansprüche 18 bis 27, weiter aufweisend die folgenden Schritte:

Speichern von Paketen mit Informationen in Pufferspeichern; und

Bereitstellen einer Datenstruktur für jeden Anschluss P2 der Anschlüsse, um einen oder mehrere Zeiger auf einen oder mehrere Pufferspeicher zu enthalten, welche Pakete speichern, die auf dem Anschluss P2 zu übertragen sind,

wobei zumindest einer der Schritte (b) und (c) das Übertragen von Paketen, welche in Pufferspeichern gespeichert sind, auf welche die Zeiger in den Datenstrukturen des Anschlusses P2 gerichtet sind, auf jedem Anschluss P2 umfasst.

29. Verfahren nach Anspruch 28, wobei jeder der Schritte (b) und (c) das Übertragen von Paketen, die in Pufferspeichern gespeichert sind, auf welche Zeiger in den Datenstrukturen des Anschlusses P2 gerichtet sind, auf jedem Anschluss P2 umfasst.

30. Verfahren nach Anspruch 28 oder 29, weiter aufweisend:

Empfangen eines Pakets PC1 auf einem der Anschlüsse;
Bestimmen eines Anschlusses oder mehrerer Anschlüsse, an die das Paket PC1 zu übertragen ist; und
Einschieben eines Zeigers in einen Pufferspeicher, der zumindest einen Teil des Pakets PC1 speichert, in die Datenstruktur jedes Anschlusses, auf dem das Paket PC1 zu übertragen ist.
Revendications

1. Appareil pour permettre des communications entre des unités connectées à différents segments de réseaux, l’appareil comprenant :

- une pluralité d’accès (3, 10) permettant une connexion aux segments de réseaux et à un ou plusieurs systèmes de contrôle ; et
- un premier moyen (11, 12) pour transmettre des paquets d’informations à un ou plusieurs des accès (3, 10), dans lequel chaque paquet d’informations comprend des informations de renvoi devant être utilisées dans la détermination de la destination du paquet ;

- dans lequel le premier moyen (11, 12) comprend un moyen pour transmettre chacun des un ou plusieurs paquets :
  (a) vers un ou plusieurs accès (3) déterminé(s) à partir de la destination du paquet si la destination comporte une unité autre que l’appareil ; et, en outre, (b) vers un ou plusieurs accès de contrôle (10).

2. Appareil selon la revendication 1, dans lequel chaque accès de contrôle (10) est rendu apte à permettre une connexion à un segment de réseau accessible à un contrôleur de réseau (9).

3. Appareil selon l’une quelconque des revendications précédentes, dans lequel le premier moyen est rendu apte à permettre, en réponse à une commande, la transmission d’un paquet : (a) vers un ou plusieurs accès déterminés sur la base de la destination du paquet et, en outre, (b) vers un ou plusieurs accès de contrôle.

4. Appareil selon l’une quelconque des revendications précédentes, comprenant en outre un second moyen pour indiquer au premier moyen quels paquets doivent être transmis vers un ou plusieurs accès de contrôle, dans lequel le second moyen est rendu apte à permettre la spécification de ces paquets à un instant quelconque pendant le fonctionnement du premier moyen.

5. Appareil selon l’une quelconque des revendications précédentes, dans lequel chacun des accès est apte à permettre une connexion à un segment de réseau.

6. Appareil selon l’une quelconque des revendications précédentes, dans lequel la pluralité d’accès comprend :

- un ou plusieurs accès pour la connexion à un ou plusieurs segments de réseaux en utilisant un premier format de protocole ; et
- un ou plusieurs accès pour la connexion à un ou plusieurs segments de réseaux en utilisant un second format de protocole différent du premier format de protocole ;

- dans lequel les un ou plusieurs accès de contrôle comportent un premier accès de contrôle pour la connexion à un segment de réseau utilisant le premier format de protocole ; et
- dans lequel le premier moyen comprend un moyen pour traduire des paquets du second format de protocole en le premier format de protocole pour permettre la transmission de paquets reçus d’un segment de réseau utilisant le second format de protocole vers le premier accès de contrôle.

7. Appareil selon l’une quelconque des revendications précédentes, dans lequel le premier moyen comprend une mémoire pour mémoriser une ou plusieurs structures de données qui permettent au premier moyen de déterminer, en utilisant des informations de renvoi d’un paquet, tous les accès, s’il en existe, vers lesquels le paquet doit être transmis ; et

- dans lequel l’appareil comprend en outre un moyen pour modifier les structures de données en réponse à des commandes afin de définir quels paquets doivent être transmis vers quels accès de contrôle.

8. Appareil selon la revendication 7, dans lequel les commandes comportent une commande de transmission vers un accès de contrôle de paquets entrants sur un accès sélectionné.

9. Appareil selon la revendication 7 ou 8, dans lequel les commandes comportent une commande de transmission vers un accès de contrôle de paquets renvoyés vers un accès sélectionné pour la transmission.
10. Appareil selon la revendication 7, 8 ou 9, comprenant en outre un moyen pour générer des paquets, dans lequel les commandes comportent une commande de transmission vers un accès de contrôle de paquets générés par le moyen générateur.

11. Appareil selon l'une quelconque des revendications 7 à 10, dans lequel les commandes comportent une commande de transmission vers un accès de contrôle de paquets entrants sur un premier accès sélectionné et renvoyés vers un second accès sélectionné pour la transmission.

12. Appareil selon l'une quelconque des revendications précédentes, dans lequel le premier moyen comprend un moyen d'application d'une ou plusieurs règles de filtrage ayant pour base les contacts spécifiques de chaque paquet pour déterminer quels paquets doivent être transmis vers un accès de contrôle.

13. Appareil selon l'une quelconque des revendications précédentes, dans lequel les paquets ont un nombre variable d'unités de données.

14. Appareil selon l'une quelconque des revendications précédentes, dans lequel :

   les informations de renvoi de chaque paquet comprennent une adresse de source et une adresse de destination ;

   le premier moyen comprend une mémoire S1 pour mémoriser une ou plusieurs entrées de la forme (RPORT, SA, DA, XP), où :

   RPORT identifie un ou les accès ;
   SA est une adresse de source ;
   DA est une adresse de destination ; et
   XP identifie zéro, un ou plusieurs accès auxquels doit être transmis un paquet qui est reçu sur l'accès RPORT et qui a une adresse de source SA et une adresse de destination DA ; et

   l'appareil comprend en outre un moyen M pour rechercher dans la mémoire S1, lorsqu'un paquet est reçu, une entrée dont le paramètre RPORT identifie l'accès sur lequel le paquet est reçu et dont les paramètres SA et DA correspondent, respectivement, à une adresse de source et à une adresse de destination du paquet reçu, dans lequel, si cette entrée est trouvée, le moyen M est rendu apte à déterminer à partir du paramètre XP de l'entrée trouvée les accès auxquels le paquet reçu doit être transmis.

15. Appareil selon l'une quelconque des revendications 1 à 13, dans lequel :

   les informations de renvoi de chaque paquet comprennent une adresse de source et une adresse de destination ;

   le premier moyen comprend une mémoire S2 pour mémoriser une ou plusieurs entrées dont chacune comprend un triplet (SA, DA, XP) dans lequel :

   SA est une adresse de source ;
   DA est une adresse de destination ; et
   XP identifie zéro, un ou plusieurs accès auxquels doit être transmis un paquet ayant une adresse de source SA et une adresse de destination DA ; et

   l'appareil comprend en outre un moyen M pour rechercher dans la mémoire S2, lorsqu'un paquet est reçu, une entrée dont les paramètres SA et DA correspondent respectivement à une adresse de source et à une adresse de destination du paquet reçu, dans lequel, si cette entrée est trouvée, le moyen M est rendu apte à déterminer à partir du paramètre XP de l'entrée trouvée les accès auxquels le paquet reçu doit être transmis.

16. Appareil selon la revendication 14 ou 15, dans lequel, pour chaque entrée, XP est une carte dans laquelle, pour un accès, un ou plusieurs bits indiquent si des informations doivent être transmises vers ledit accès.

17. Appareil selon l'une quelconque des revendications précédentes, dans lequel le premier moyen comprend :

   une mémoire pour mémoriser (a) des paquets d'informations dans des tampons, (b) pour chaque accès P1 d'un ou plusieurs accès, une première structure de données destinée à contenir un ou plusieurs pointeurs sur un ou plusieurs tampons qui mémorisent des paquets reçus sur l'accès P1, et (c) pour chaque accès P2
d'un ou plusieurs des accès, une seconde structure de données pour contenir un ou plusieurs pointeurs sur un ou plusieurs tampons qui mémorisent des paquets devant être transmis sur l'accès P2 ;
un moyen M2 pour transmettre, sur chaque accès ayant une seconde structure de données, des paquets stockés dans des tampons sur lesquels pointent des pointeurs contenus dans la seconde structure de données de l'accès ; et
un moyen pour copier au moins un pointeur de la première structure de données d'un premier accès vers la seconde structure de données d'un ou plusieurs seconds accès pour renvoyer un paquet reçu sur le premier accès vers les un ou plusieurs seconds accès.

18. Procédé de surveillance d'un réseau comprenant un appareil interconnectant une pluralité de segments de réseaux dont au moins l'un comprend un contrôleur de réseau (9), le procédé comprenant les étapes qui consistent à :

(a) obtenir de chaque paquet reçu ou généré par l'appareil des informations de renvoi devant être utilisées dans la détermination de la destination du paquet ;
(b) si une destination d'un paquet contient une station autre que l'appareil, faire alors en sorte que l'appareil transmette le paquet vers un ou plusieurs des segments de réseaux afin de délivrer le paquet à la destination du paquet ; et caractérisé en ce que :
(c) si un paquet doit être délivré à un contrôleur de réseau (9), l'appareil transmet alors le paquet à un segment de réseau comprenant le contrôleur de réseau (9).

19. Procédé selon la revendication 18, dans lequel, pour au moins un paquet dont la destination comporte une station autre que l'appareil et qui doit être délivré à un contrôleur de réseau, les étapes (b) et (c) sont effectuées en réponse à une commande.

20. Procédé selon la revendication 18 ou 19, dans lequel :

un ou plusieurs des segments de réseaux utilisent un premier format de protocole ;
un ou plusieurs des segments de réseaux utilisent un second format de protocole différent du premier format de protocole ;
au moins un segment de réseau comprenant un contrôleur de réseau utilise le premier format de protocole ; et
le procédé consiste en outre à traduire un ou plusieurs paquets reçus sur un ou plusieurs segments de réseaux utilisant le second format de protocole du second format de protocole en le premier format de protocole et à transmettre ces paquets à un segment de réseau qui comprend un contrôleur de réseau et qui utilise le premier format de protocole.

21. Procédé selon l'une quelconque des revendications 18 à 20, comprenant en outre les étapes qui consistent à :

mémoriser dans une mémoire une ou plusieurs structures de données pour déterminer, en utilisant les informations de renvoi d'un paquet, tous les segments de réseaux, s'il en existe, auxquels le paquet doit être transmis ; et
modifier les structures de données en réponse à une commande afin de définir quels paquets doivent être transmis à quels segments de réseaux comprenant des contrôleurs de réseau.

22. Procédé selon la revendication 21, dans lequel la commande est une commande de transmission, à un segment de réseau comprenant un contrôleur de réseau, de paquets entrants provenant d'un segment de réseau sélectionné.

23. Procédé selon la revendication 21, dans lequel la commande est une commande de transmission, à un segment de réseau comprenant un contrôleur de réseau, de paquets transmis lors de l'étape (b) à un segment de réseau sélectionné.

24. Procédé selon la revendication 21, consistant en outre à faire en sorte que l'appareil génère des paquets, dans lequel la commande est une commande de transmission à un segment de réseau comprenant un contrôleur de réseau, de paquets générés par l'étape de génération.

25. Procédé selon la revendication 21, dans lequel la commande est une commande de transmission, à un segment de réseau comprenant un contrôleur de réseau, de paquets reçus d'un premier segment de réseau sélectionné et transmis lors de l'étape (b) à un second segment de réseau sélectionné.
26. Procédé selon l'une quelconque des revendications 18 à 25, dans lequel l'étape (c) consiste à appliquer une ou plusieurs règles de filtrage ayant pour base le contenu spécifique de chaque paquet pour déterminer si un paquet doit être délivré à un contrôleur de réseau.

27. Procédé selon l'une quelconque des revendications 18 à 26, dans lequel les paquets ont un nombre variable d'unités de données.

28. Procédé selon l'une quelconque des revendications 18 à 27, comprenant en outre les étapes consistant à :

mémoriser des paquets d'informations dans des tampons ; et
fournir, pour chaque accès P2 desdits accès, une structure de données destinée à contenir un ou plusieurs pointeurs sur un ou plusieurs tampons qui mémorisent des paquets devant être transmis sur l'accès P2,

dans lequel au moins l'une des étapes (b) et (c) consiste à transmettre sur chaque accès P2 des paquets mémorisés dans des tampons sur lesquels pointent des pointeurs dans les structures de données de l'accès P2.

29. Procédé selon la revendication 28, dans lequel chacune des étapes (b) et (c) consiste à transmettre sur chaque accès P2 des paquets stockés dans des tampons sur lesquels pointent des pointeurs dans les structures de données de l'accès P2.

30. Procédé selon la revendication 28 ou 29, comprenant en outre les étapes qui consistent à :

recevoir un paquet PC1 sur l'un des accès ;
déterminer un ou plusieurs accès auxquels le paquet PC1 doit être transmis ; et
insérer un pointeur sur un tampon mémorisant au moins une partie du paquet PC1 dans la structure de données de chaque accès sur lequel le paquet PC1 doit être transmis.
Figure 4
Figure 10A

Port 4 Controller

71.4 Port Transmit Descriptor Ring

Port 2 Controller

71.2 Port 2 Transmit Descriptor Ring

Packet Buffer Pool

Packet Buffer Free List

Packet Buffer
<table>
<thead>
<tr>
<th>REPORT</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>101110</td>
<td>101101</td>
<td>10101</td>
<td>100111</td>
<td>000000</td>
<td>001111</td>
</tr>
</tbody>
</table>

Figure 13
<table>
<thead>
<tr>
<th>M</th>
<th>P</th>
<th>O</th>
<th>R</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>0</td>
<td>0</td>
<td>00001</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>0</td>
<td>0</td>
<td>00010</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0</td>
<td>0</td>
<td>00100</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0</td>
<td>0</td>
<td>01000</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0</td>
<td>0</td>
<td>00000</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>1</td>
<td>0</td>
<td>10000</td>
</tr>
</tbody>
</table>

Figure 14
### Figure 17A

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>000000</td>
<td>000010</td>
<td>001000</td>
<td>001000</td>
<td>000000</td>
<td>100000</td>
</tr>
<tr>
<td>1</td>
<td>000001</td>
<td>000000</td>
<td>000100</td>
<td>001000</td>
<td>000000</td>
<td>100000</td>
</tr>
<tr>
<td>2</td>
<td>010001</td>
<td>010010</td>
<td>010000</td>
<td>011000</td>
<td>010000</td>
<td>110000</td>
</tr>
<tr>
<td>3</td>
<td>000001</td>
<td>000010</td>
<td>000100</td>
<td>000000</td>
<td>000000</td>
<td>100000</td>
</tr>
<tr>
<td>4</td>
<td>000000</td>
<td>000000</td>
<td>000000</td>
<td>000000</td>
<td>000000</td>
<td>000000</td>
</tr>
<tr>
<td>5</td>
<td>000001</td>
<td>000010</td>
<td>000100</td>
<td>001000</td>
<td>000000</td>
<td>000000</td>
</tr>
</tbody>
</table>

### Figure 17B

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>101110</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>101101</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>111011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>100111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>001111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Figure 18A

<table>
<thead>
<tr>
<th>REPORT</th>
<th>XPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>000000</td>
</tr>
<tr>
<td>1</td>
<td>000001</td>
</tr>
<tr>
<td>2</td>
<td>000001</td>
</tr>
<tr>
<td>3</td>
<td>000001</td>
</tr>
<tr>
<td>4</td>
<td>000000</td>
</tr>
<tr>
<td>5</td>
<td>000001</td>
</tr>
</tbody>
</table>

### Figure 18B

<table>
<thead>
<tr>
<th>REPORT</th>
<th>REPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>111110</td>
</tr>
<tr>
<td>1</td>
<td>111101</td>
</tr>
<tr>
<td>2</td>
<td>101011</td>
</tr>
<tr>
<td>3</td>
<td>110111</td>
</tr>
<tr>
<td>4</td>
<td>000000</td>
</tr>
<tr>
<td>5</td>
<td>011111</td>
</tr>
</tbody>
</table>
**Figure 20A**

<table>
<thead>
<tr>
<th>XPORT</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>000000</td>
<td>000010</td>
<td>000100</td>
<td>001000</td>
<td>000000</td>
<td>100000</td>
</tr>
<tr>
<td>1</td>
<td>000001</td>
<td>000000</td>
<td>000100</td>
<td>001000</td>
<td>000000</td>
<td>100000</td>
</tr>
<tr>
<td>2</td>
<td>000001</td>
<td>000010</td>
<td>000000</td>
<td>011000</td>
<td>000000</td>
<td>100000</td>
</tr>
<tr>
<td>3</td>
<td>000001</td>
<td>000010</td>
<td>000100</td>
<td>000000</td>
<td>000000</td>
<td>100000</td>
</tr>
<tr>
<td>4</td>
<td>000000</td>
<td>000000</td>
<td>000000</td>
<td>000000</td>
<td>000000</td>
<td>000000</td>
</tr>
<tr>
<td>5</td>
<td>000001</td>
<td>000010</td>
<td>000100</td>
<td>001000</td>
<td>000000</td>
<td>000000</td>
</tr>
</tbody>
</table>

**Figure 20B**

<table>
<thead>
<tr>
<th>REPORT</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>11110</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>101101</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>111011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>100111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>011111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>