OPIC
Office de la propriété intellectuelle du Canada

CIPO
Canadian Intellectual Property Office

(12) (19) (CA) **Demande-Application**

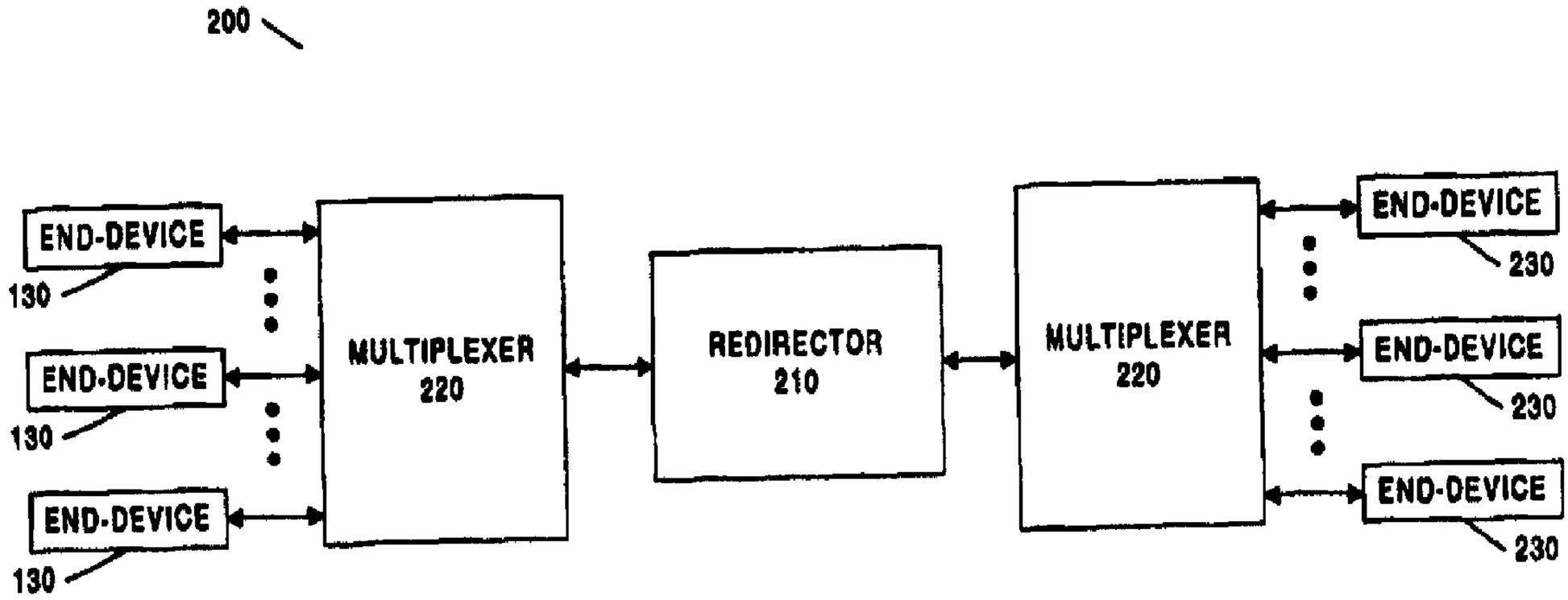
(21) (A1) **2,308,219**
(86) 1998/10/27
(87) 1999/05/06

(72) MCKEOWN, NICHOLAS W., US
(71) ABRIZIO, INC., US

(51) Int.Cl.[6] H04L 12/44
(30) 1997/10/28 (60/063,562) US

(54) **RESEAU DE DONNEES RATIONALISE**

(54) **STREAM-LINE DATA NETWORK**



(57) Selon un mode de réalisation préféré de cette invention, le système de réseau se présente comme un système en temps réel à largeur de bande élevée qui permet à plusieurs dispositifs terminaux de communiquer entre eux de manière simple et efficace en termes de coûts. En particulier, un module redirecteur fonctionne comme un point centralisé du réseau qui achemine les paquets de données, reçus d'un signal rationalisé en amont qui est associé aux dispositifs terminaux sources, vers les signaux rationalisés en aval qui sont associés aux dispositifs terminaux sources qui sont les destinataires désirés des paquets de données. A chaque branche du réseau, on utilise un module de multiplexage pour multiplexer les signaux de voies en des signaux rationalisés en amont et pour démultiplexer les signaux rationalisés en aval en des signaux de voie.

(57) The network system of a preferred embodiment of the present invention achieves a real-time, high bandwidth system that enables a plurality of end-devices to communicate with one another in a simple cost-effective manner. In particular, a redirector module operates as a centralized point in the network, which routes data packets received from an upstream stream-line signal associated with source end-devices to downstream stream-line signals associated with end-devices that are the intended recipients of the data packets. A multiplexer module is utilized in each branch of the network to multiplex channel signals into upstream stream-line signals and demultiplex downstream stream-line signals into channel signals.

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
H04L 12/44

A1

(11) International Publication Number: WO 99/22496

(43) International Publication Date: 6 May 1999 (06.05.99)

(21) International Application Number: PCT/US98/22744

(22) International Filing Date: 27 October 1998 (27.10.98)

(30) Priority Data:
60/063,562 28 October 1997 (28.10.97) US

(71) Applicant: ABRIZIO, INC. [US/US]; 501 B Ellis Street, Mountain View, CA 94043 (US).

(72) Inventor: MCKEOWN, Nicholas, W.; 864 Cedro Way, Stanford, CA 94305 (US).

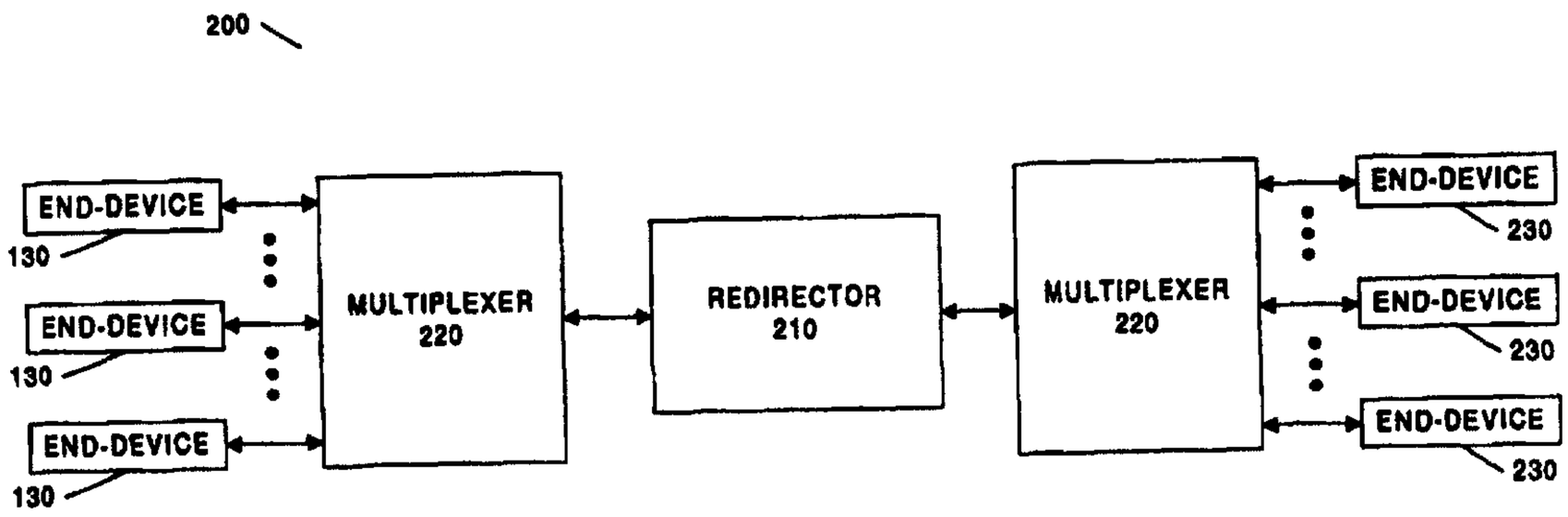
(74) Agents: OKAMOTO, James, K. et al.; Fenwick & West LLP, Two Palo Alto Square, Palo Alto, CA 94306 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published

With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: STREAM–LINE DATA NETWORK



(57) Abstract

The network system of a preferred embodiment of the present invention achieves a real–time, high bandwidth system that enables a plurality of end–devices to communicate with one another in a simple cost–effective manner. In particular, a redirector module operates as a centralized point in the network, which routes data packets received from an upstream stream–line signal associated with source end–devices to downstream stream–line signals associated with end–devices that are the intended recipients of the data packets. A multiplexer module is utilized in each branch of the network to multiplex channel signals into upstream stream–line signals and demultiplex downstream stream–line signals into channel signals.

STREAM-LINE DATA NETWORK

Inventor

Nicholas W. McKeown

RELATED APPLICATIONS

This application claims the benefit under 35 U.S.C. § 119(e) of U.S. Provisional Application Serial No. 60/063,562, inventor Nick McKeown, filed on October 28, 1997, entitled "Stream-Line LANs".

BACKGROUND OF INVENTION

Field of Technology

The present invention generally relates to the field of communication systems and more particularly to the field of computer data network systems.

The evolution of computer data network systems has been very rapid and dynamic. In particular, much progress has occurred in the field of data networks such as local area networks (LANs). LANs are systems, which interconnect a plurality of end-devices, such as computers, through usually one of a variety of different networking protocols. One of the more popular networking protocols is CSMA/CD (IEEE 802.3), which is more popularly known as Ethernet.

An Ethernet-based LAN usually exhibits a data transfer rate between 10Mb/s and 1Gb/s. In particular, in a hub-based Ethernet LAN each end-device within the LAN is connected to a repeater (hub), which receives a data packet having a destination address header and broadcasts the data packet to the remaining end-devices connected to the LAN. Only the end-device having the same address header will process the transmitted data packet and accordingly respond. If two end-devices transmit data packets at the same time to the repeater, a collision will occur, which results in the corruption of the transmitted data packets and the need for both end-devices to retransmit their data packets at a later time. To attempt to avoid further collisions of these same data packets by the same two end-devices, each end-device pauses for a randomly generated amount of time before retransmitting its specific data packet to the same repeater.

Although Ethernet is very simple to implement, this networking scheme suffers from an inherent problem of exhibiting non-deterministic characteristics. In particular, Ethernet does not provide a predictable means for determining how long it will take for an end-device to transmit a data packet through the network to another end-device. Instead, the data transfer rate for the

data packets directly depends upon the number of other active end-devices, which are attached to the network, and the particular arrival time of the data packets at the repeater. The more end-devices, which are transmitting data packets, the more likely that there will be a collision between at least two data packets, thereby requiring each end-device to retransmit its data packets at a later time.

An additional problem with this type of network system is that in recent years some of the new software applications, such as digital video and audio in intranet applications, are requiring even larger amounts of data to be transferred across these same network systems within very time-sensitive bandwidth performance constraints. In addition, data traffic patterns have shifted away from local workgroup traffic toward more backbone-centric server-based data traffic. Such a shift has caused a greater bandwidth burden on the backbone of the LAN, which must be managed efficiently by the network system.

In an attempt to reduce this congestion of LANs, repeaters often are replaced by Ethernet switches that more closely manage the bandwidth of the system. Figure 1 illustrates such a conventional Ethernet switch-based system 100. In particular, a conventional router 110 is used to link each Ethernet switch 120 with the other Ethernet switches 120 in the network 100. In particular, each Ethernet switch 120 represents a central management scheme for a specific branch of the network system 100. Each of these branches of the network system includes a plurality of end-devices 130, which are coupled to a specific Ethernet switch 120. If the end-device 130 must communicate with an end-device 130 on another branch of the network, the data packets must be forwarded via the router 110 to the Ethernet switch 120 that manages the destination end-device 130.

The Ethernet switch 120 has two primary advantages over other conventional network systems. First, by being able to switch amongst dedicated, private Ethernet connections for each of the end-devices 130, which are connected to the Ethernet switch 120, only data packets destined for a specific end-device 130 are specifically forwarded to that end-device 130. Such a scheme avoids the unnecessary additional bandwidth burdens associated with requiring a repeater to broadcast each data packet to every end-device 130 on the network system 100. Second, by utilizing the Ethernet switching architecture, if two or more data packets arrive at the same Ethernet switch 120 at the same time for different end-devices, all of the data packets can be simultaneously forwarded without the risk of a collision and the need to retransmit these data packets. Such a reduction in collisions results in the network system becoming less non-deterministic and more appealing for time-sensitive software applications.

This Ethernet switch architecture, however, has two inherent problems. First, if two or more data packets simultaneously arrive at the same Ethernet switch 120 for the same destination end-device 130, only one data packet can be forwarded at any one time to the same destination end-device 130. To avoid this unwanted generation of a collision signal, which contributes to non-deterministic delays in the system, the Ethernet switch 120 as well as the router 110 must buffer the data packets in one of a plurality of data packet buffers. Each of these data packets then is sequentially retransmitted to the same destination end-device 130, thereby ensuring that the originating end-device 130 does not have to retransmit the same data packet. Since this problem repeatedly occurs within any conventional Ethernet switch-based network system, Ethernet switches 120 and the router 110 typically must contain a large and expensive pool of such data packet buffers to ensure that collisions are minimized within the system 100.

The second problem with the conventional Ethernet switch network system is that to maintain a certain quality of service (QoS) throughout the system 100 a significant amount of management of the system must occur in each Ethernet switch 120 as well as the router 110. In particular, to efficiently manage each branch of the network, each Ethernet switch 120 and the router 110 must rely upon a technique such as packet scheduling, resource reserved protocol (RSVP), weighted fair queuing (WFQ), the assignment of virtual LANs (VLANs), the configuration of ports, the allocation of buffers, the configuration of a spanning tree protocol, IP switching, multiprotocol over ATM (MPOA), tag switching or a combination thereof. Such management of the network system 100 results in a decentralized architecture where the router 110 and each Ethernet switch 120 must perform the same buffering and management functions, thereby complicating the deployment and management and also increasing the overall cost of such a conventional large, high bandwidth network.

What is needed is a network system, which centralizes the management of the network, while still maintaining a predictable high performance bandwidth for servicing high bandwidth and real-time based applications.

SUMMARY OF INVENTION

The network system of a preferred embodiment of the present invention includes a redirector module and a plurality of multiplexer modules, which achieves a real-time, high bandwidth system that enables a plurality of end-devices to communicate with one another in a simple cost-effective manner. The multiplexer modules, which are individually disposed between the redirector module and a plurality of end-devices, multiplexes channel signals,

which are received from source end-devices, and transmits upstream stream-line signals to the redirector module. In addition, the multiplexer modules demultiplex downstream stream-line signals, which are received from the redirector module, into channel signals that are transmitted to the corresponding destination end-devices. By utilizing multiplexer modules to generate stream-line signals, the redirector module becomes the central location for routing and buffering of these stream-line signals. In particular, the redirector module receives from each multiplexer module upstream stream-line signals that contain data packets from multiple source end-devices that are coupled to each multiplexer module. The redirector module routes each of these data packets to a downstream stream-line signal, which is associated with the specific branch of the network where the end-device which is to receive the data pocket, is located.

By relying upon the redirector module to primarily manage the routing of the data packets between the different end-devices, the current network system avoids the problems associated with the conventional decentralized network system schemes.

DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates a conventional Ethernet switch network system.

Figure 2 illustrates an overview of a network system of a preferred embodiment of the present invention.

Figure 3 illustrates a high level block diagram of a redirector module of a preferred embodiment.

Figure 4 illustrates a block diagram of a multiplexer module of a preferred embodiment.

Figure 5 illustrates a statistical multiplexing scheme for a multiplexer module of an alternative embodiment.

Figure 6 illustrates a time division multiplexing scheme for a multiplexer module of a preferred embodiment.

Figure 7 illustrates a timing scheme for a multiplexer module of a preferred embodiment.

Figure 8 illustrates a flow control methodology of a preferred embodiment.

Detailed Description of Preferred Embodiments of the Present Invention

Preferred embodiments of the present invention are now described with reference to figures where like reference numbers indicate identical or functionally similar elements and the left most digit of each reference number corresponds to the figure in which the reference number is first used.

A network system of a preferred embodiment of the present invention contains the management simplicity and costs similar to a repeater and the performance equal to or

exceeding that of an Ethernet switch. In particular, by applying a high performance (e.g., an approximate aggregate bandwidth of 0.6 Terabit/Second and an approximate 1microsecond latency) Internet protocol (IP) router to a preferred embodiment of the present invention, a very simple, powerful and cost-effective network system can be achieved, which can be integrated into different network topologies. For example, a preferred embodiment of the present invention relies upon a simple and inexpensive multiplexed configuration to create a scaleable super router (SSR) network system that contains a large number of Ethernet ports, which operate at, for example, either 10Mbps, 100Mbps or 1Gbps. The benefits of such a network design is that a high level of performance can be guaranteed while still maintaining a simple centralized network administration.

Figure 2 illustrates an overview of a preferred embodiment of a network system 200 of a preferred embodiment of the present invention. In particular, this network system 200 includes a redirector module 210 and a plurality of multiplexer modules 220. The redirector module 210 serves as a centralized module for managing the routing of multiplexed data packets (stream-line signals) between different branches of the network system 200. The multiplexer modules 220, which are each associated with a specific branch of the network system 200, are disposed between a plurality of end-devices 130 and the redirector module 210. When each channel signal, which is comprised of a plurality of data packets that are transmitted from a specific end-device 130, is received by the multiplexer module 220, the multiplexer module 220 mechanically multiplexes each data packet within the channel signal into an "upstream" stream-line signal. This upstream stream-line signal then is transmitted to the redirector module 210 for data packet routing. When a "downstream" stream-line signal is received from the redirector module 210, the multiplexer module 220 mechanically demultiplexes the downstream stream-line signals into channel signals that are transmitted to each corresponding end-device 130. In a preferred embodiment, the multiplexer module 220 utilizes time division multiplexing (TDM), which will be discussed in more detail with regard to Figure 6, to multiplex and demultiplex the data packets.

Since each multiplexer module 220 does not actively manage its corresponding branch of the network 200 like an Ethernet switch, the redirector module 210 becomes the primary component that manages the performance of the network system 200. Such a simple configuration ensures that there is no need for complicated distributed management functions, such as VLANs, or for distributed buffering schemes as is necessary in conventional network

systems. The redirector module 210 functions as a centralized hub where all primary network management and buffering is performed.

Figure 3 illustrates in more detail the redirector module 210, which includes a packet switch module 310 and a plurality of port modules 320. The packet switch module 310 also includes a router module 315 and a routing table 317. Each port module 320 also includes a management unit 360, a buffer module 355 and a scheduler unit 365. In a preferred embodiment, the packet switch module 310 is a LAN switch, such as a Catalyst series LAN switch from Cisco Systems of San Jose, California, and the routing table 317 is a database stored within a memory module (not illustrated) within the packet switch module 310. In addition, the management unit 360 and the scheduler unit 365 are logic circuitry and the buffer module 355 is random access memory. In an alternative embodiment, the packet switch module is a router, such as a Cisco 7500 or 12000 series router from Cisco Systems of San Jose, California.

Each port module 320, which is coupled to the packet switch module 310, operates as an interface between a specific branch of the network system 200 and the packet switch module 310. In particular, each port module 320 may operate as either a source network node or a destination network node for a specific branch of the network 200.

When operating as a source network node, the management unit 360 of the port module 320 receives the upstream stream-line signal from the multiplexer module 220 that is coupled to the port module 320. The management unit 360 then demultiplexes the upstream stream-line signal into its corresponding data packets and sequentially transmits each of these data packet to the packet switch module 310. In a preferred embodiment, with the port module 320 operating as a source network node, the buffer module 355 and the scheduler module 365 remain primarily idle.

When the packet switch module 310 receives the data packets from the management unit 360, the router module 315 reads the header of each data packet and utilizes the routing table 317, which is coupled to the router module 315, to determine the location of the destination end-device 130. In particular, the routing table 317 maintains a list of end-devices 130 and the corresponding branch of the network 200 where each of these end-device 130 is located. When a specific end-device 130 is identified as the destination for a specific data packet, the router module 315 utilizes the routing table 317 to determine the specific port module 320 (destination network node) that is associated with the branch of the network 200 where the destination end-device 130 resides. Once a specific port module 320 is identified as the destination network

node, the packet switch module 310 routes each data packet corresponding to that branch of the network 200 to the management unit 360 of that port module 320.

In a preferred embodiment, the management unit 360 receives each of the data packets associated with this branch of the network and multiplexes them onto the downstream stream-line signal. When bandwidth problems develop in transmitting these data packets to their corresponding destination end-devices 130, the management unit 360 relies upon the buffer module 355 and the scheduler module 365 to manage the buffering of these delayed data packets for later integration into a subsequent downstream stream-line signal that is to be transmitted to the multiplexer module 220.

There are several reasons why a data packet cannot be included in the current downstream stream-line signal. First, as in the conventional Ethernet switch, when multiple data packets simultaneously arrive at the port module 320 (possibly from different port modules 320, but all destined to the same end-device 130), if the downstream stream-line signal data rate necessary to transmit these data packets exceeds the data rate capacity of the multiplexer module 220 to transmit data packets to a specific destination end-device 130, these data packets must be buffered in the buffer module 355 of the port module 320 until some bandwidth becomes available. Second, if the transmission rate between the multiplexer module 220 and the end-devices 130 is half-duplex, only one channel signal can be transmitted either to or from the end-device 130. If an end-device 130 is currently transmitting a first channel signal to the multiplexer module 220 and the multiplexer module 220 attempts to transmit a second channel signal to the same end-device 130, a collision signal between data packets will occur and the multiplexer module 220 will trigger the buffering of the remaining portions of the data packet in both the port module 320 and the multiplexer module 220. A more detailed discussion concerning the triggering and the buffering of data packets will be discussed with regard to Figure 4 and 7.

Upon storing a portion of a data packet in the buffer module 355 for either of the before-mentioned reasons, the scheduler module 365 utilizes one of several different scheduling algorithms to determine the order in which these buffered data packets will be added to subsequent downstream stream-line signals. In a preferred embodiment, a scheduling algorithm, such as a first come, first served (FCFS) (also known as first in, first out (FIFO)), a weighted fair-queuing, or a virtual clock algorithm, is used to determine the order of integration of the buffered data packets into the next downstream stream-line signals.

Figure 4 illustrates in more detail the multiplexer module 220, which is disposed between the port module 320 of the redirector module 210 and a plurality of end-devices 130. The multiplexer module 220 includes a multiplexer/demultiplexer (mux/demux) unit 410, a control unit 420, a plurality of channel modules 430 and a plurality of buffer modules 440. Unlike Ethernet switches, the multiplexer module 130 contains no direct management or address look-up functionality. In addition, the multiplexer module 130 does not have the same level of buffering as that of Ethernet switches. This simplicity of design ensures that the complexity of managing the network system 200 is shifted from each major component within the network 200 to one single, central location, the redirector module 210.

The mux/demux module 410, which is disposed between the port module 320 and the channel modules 430, multiplexes channel signals, which are received from end-devices 130, into upstream stream-line signal data packets and demultiplexes downstream stream-line signals, which are received from the port module 320, into channel signal data packets. To minimize the need for significant buffering in the multiplexer module 220, the connection between the mux/demux module 410 and the port module 320 is full-duplex and equal in the capacity of all connections to/from the end-devices 130. In a preferred embodiment, the mux/demux module 410 is logic circuitry or a software module executed within a general purpose computer.

Figures 5 and 6 illustrate two alternative embodiments for the mux/demux module 410 to multiplex the channel signals into an upstream stream-line signal. It should be noted, however, that each data packet within this upstream stream-line signal is of a fixed size unit (e.g., one bit at a time, one byte at a time, or any other fixed size unit). In a preferred embodiment, the fixed-size unit is one byte. However, in alternative embodiments, any fixed-size unit may be employed.

In a first embodiment, Figure 5 illustrates the statistical (packet) multiplexing of three channel signals, P_1, P_2 and P_3, which are received from three different end-devices 130, into a single upstream stream-line signal. In particular, each channel signal is serially appended to the end of the preceding channel signal to form the upstream stream-line signal. To ensure that the router module 315 later can properly redirect the data packets that correspond to the channel signals to the appropriate end-device 130 destinations, the multiplexer module 220 transmits a repeating sequence of framing bytes (e.g., one byte for each of the channel modules associated with a source end-device), which indicates which byte in the sequence belongs to which channel module 430. In an upstream stream-line signal, this repeating framing byte corresponds to the

channel module 430 on which the data packet arrived. In a downstream streamline signal, this repeating framing byte corresponds to the channel module 430 to which the data packet is destined.

Figure 6 illustrates a preferred embodiment of the present invention where time division multiplexing (TDM), rather than statistical multiplexing, is used by the mux/demux module 410 to multiplex data packets of the channel signals into stream-line signals. Unlike statistical multiplexing, TDM implicitly encodes the channel module information into the stream-line signal. For example, consider a downstream stream-line that uses TDM with byte-size units and a multiplexer module that has N channel modules. Every Nth byte on the stream-line signal would be destined to channel module P_1, followed by a byte for channel module P_2, etc.

Each of the plurality of channel modules 430 within the multiplexer module 220 represents a dedicated channel to a specific end-device 130. In a preferred embodiment, each end-device 130 may be a computer, multiple electronic devices, a network repeater (hub) or a LAN switch. Each channel module is either logic circuitry or a software module. In an alternative embodiment, for fault tolerance, a multiplexer module 130 also may have additional redundant connections via other channel modules 430 to end-devices 130, other redirector modules 210 or other multiplexer modules 220. Each of the channel modules 430 communicates with a dedicated end-device 130 using a conventional local area network (LAN) protocol, such as 10 Mb/s Ethernet, 100Mb/s (fast) Ethernet, or 1Gb/s Ethernet. Such protocols rely upon either half-duplex (e.g., data packets only flow in one direction at a time) or full duplex (e.g., data packets simultaneously flow in both directions) communication, which can contain an additional flow control protocol. To minimize the possible delays associated with demultiplexing downstream stream-line signals and transmitting the data packets within channel signals from the mux/demux module 410 to the end-devices 130 via the channel modules 430, each channel module 430 of a preferred embodiment of the present invention communicates in full-duplex with the end-devices 130.

In an alternative embodiment, however, the channel module 430 may communicate in half-duplex with the end-devices 130, thereby requiring the system 200 to respond to collisions between channel signals transmitted to and channel signals simultaneously transmitted from the end-devices 130. In such a situation, more significant buffering of data packets in the multiplexer module 220 and the port module 320 will be necessary to minimize the possible delays associated with such a configuration. This degree of buffering, however, is still significantly less than what would be necessary in a conventional decentralized network system.

Figure 7 illustrates in more detail the necessary size of the buffer modules 440 in the multiplexer module 220 and its timing relationship with the port module 320. In particular, in a preferred embodiment, the exact number of bits of storage in the multiplexer module 220 is a design decision based on the data rate of the stream-line signal, the maximum propagation delay along the link between the multiplexer module 220 and the port module 320 (which is dictated by the length of the link and the speed of the propagation along the link), the time for the control unit 420 in the multiplexer module 220 to generate a flow-control message for the management unit 360 of the port module 320, and the time for the management module 360 to process that message. Based upon these factors, an adequate amount of buffering within the multiplexer module 220 can be established to ensure that, when possible, a channel signal will be continuously transmitted to the end-devices 130.

For example, in Figure 7 each buffer $|B_i|$ has a channel module data rate b_i, where

$$b=\Sigma b_i$$

is the total data rate of the stream-line signal ($0 \leq i \leq n$ with n equal to the total number of channel modules 430 in the multiplexer module 220). Assume that the round-trip propagation delay of a bit from the control unit 420 of the multiplexer module 220 to the management unit 360 of the port module 320 and back to the control unit 420 equals T seconds (this time includes the time for the control unit 420 to transmit a flow-control message, and for the management module 360 to process the message). The amount of buffering required for the channel module 430, therefore, is then approximately equal to:

$$|B_i| = b_i \text{ X } T \text{ bits},$$

while the total buffering required by each multiplexer module 220 is:

$$|B| = b \text{ X } T \text{ bits}.$$

In addition, further buffering will be needed to accommodate retransmissions over the multiplexer module 220 to the end-devices 130. In particular, the size of this buffering is calculated in the same manner as above, but with regard to the propagation round trip delay between the multiplexer module 220 and the end-devices 130.

For example, if the stream-line signal operates at 1Gb/s (10^9 bits per second), and a single bit of information travels from the port module 320 to the multiplexer module 220 in 1μs (10^{-6} seconds), then the multiplexer module 220 must have the ability to buffer at least 10^{-6} X $10^9 = 10^3$ bits of the data packets.

Figure 8 illustrates a flow control methodology of a preferred embodiment of the present invention between the port module 320 of the redirector module 210 and the multiplexer module

220. In particular, when data packets have been received by the port module 320 in a preferred embodiment, the port module 320 attempts to continuously transmit 810 these data packets as part of the downstream stream-line signal to the corresponding multiplexer module 220. The multiplexer module 220 demultiplexes this downstream stream-line signal back into the separate data packets and forwards each data packet to its predesignated channel module 430. Each channel module 430 then transmits 820 the received data packet to the end-device 130, which is coupled to the channel module 430. If, after beginning the transmission of the data packet, the channel module 430 receives 830 a collision signal from the end-device 130, the channel module 430 triggers the control unit 420 to transmit 840 a flow control signal STOP message, such as XOFF, to the management unit 360 of the port module 320. This STOP message triggers the management unit 360 of the port module 320 to stop transmitting data packets associated with this specific channel module 430.

To ensure the integrity of each data packet, which has experienced a collision, the buffer module 440 coupled to the destination channel module 430 and the buffer module 355 within the port module 320 jointly store portions of the data packet. In particular, the buffer module 440 stores the portion of the collided data packet that had been in transit between the channel module 430 and the end-device 130. The buffer module 355 stores the portion of the collided data packet that was in transit between the port module 320 and the multiplexer module 220 as well as the remaining portion of the data packet that has yet to be transmitted to the multiplexer module 220.

After a random amount of time has elapsed, the channel module 430 that received the collision signal will begin to retransmit 850 the portion of the data packet that collided. Once the remaining amount of the data packet within the buffer module 440 is equal to the amount of the data packet that will be transmitted to the end-device 130 within a round trip propagation delay, the control unit 420 transmits 860 a flow control START message to the management unit 360 of the port module 320 to reinitiate the transmission of the remaining portion of the data packet to the channel module 430. Such timing will ensure that the end-device 130 coupled to this channel module 430 will receive a continuous stream of data packets without any disruption due to the collision.

There are several ways that the flow-control messages may be sent to the management module 360. For example, in one embodiment, the control unit 420 can send flow-control messages for the channel module 430 by sending explicit control bytes in the corresponding channel of the stream-line signal. The control messages are distinguished from normal data

packet bytes by either (a) sending an additional bit with each byte to indicate whether the byte is a control byte or a data byte, (b) using a line code (e.g. a 4B/5B block code as used in 100 Base TX and FDDI) with certain reserved line codes used to indicate that a byte contains a control message. In an alternative embodiment, a single bit may be added to each data byte from the multiplexer module 220 to the port module 320. The bit always indicates the polarity of the flow control. For example, if the bit is equal to '1', the port module may be prevented from sending data on this channel, whereas if the bit is equal to '0' then the port module 320 may resume transmission on this channel.

To better illustrate the overall flow of the transmission of information from a source end-device 130 to a destination end-device 130 in a preferred embodiment of the present invention, an example of a first end-device A 130, which is coupled to multiplexer R 220, transmitting data to end-device B 130, which is coupled to multiplexer S 220, will now be discussed. In particular, end-device A 130 transmits a channel signal including a source end-device address A and a destination end-device address B. When the packet is received by multiplexer module R 220, the mux/demux module 410 does not examine the address header or content of the data packets of the channel signal, but rather simply TDM multiplexes the data packets onto the upstream stream-line signal in the dedicated channels associated with the source end-device 130. The buffer module 440 of multiplexer module R 220 only buffers the data packets until the next opportunity to transmit the data packets in the appropriate stream-line signal channel. Once the channel signal is TDM multiplexed, the data packets, which are now part of the upstream stream-line signal, are transmitted to the port module 320 associated with this branch of the network. Once received by the port module 320, the packet switch module 310 of the redirector module 210 utilizes the destination address information within each data packet to determine to which destination port module 320 this specific data packet is to be routed. Once the data packet is forwarded to the destination port module 320 coupled to multiplexer module S 220, the data packet is integrated into a downstream stream-line signal and transmitted to multiplexer module S 220, which is disposed between the port module 320 and the end-device B 130. As previously discussed, when the link between the channel port 430 and end-device B 130 is half-duplex and a collision has occurred between two data packets, the data packets that are destined for end-device B 130 must be buffered in the buffer module 355 of port module 320 until the channel module 430 begins to retransmit data packets to end-device B 130. In this instance, the channel module 430 communicates with end-devices 130 in full-duplex, thereby avoiding this need for buffering.

Once the data packet is multiplexed onto the downstream stream-line and transmitted to multiplexer module S, the data packet is TDM demultiplexed by the mux/demux module 410 and transmitted over the channel signal dedicated to the channel module 430, which is connected to end-device B 130. Since the data rate of the channel module 430 is equal to the data rate between the mux/demux module 410 and the channel module 430, the data packets do not need to be buffered.

In an alternative embodiment, however, the mux/demux module 410 may have an aggregate bandwidth that is in excess of the data rate of the channel module 430. Such an over-subscribing of the channel modules 430 may be desirable in cost-sensitive environments to enable the channel modules 430 to be pooled and allocated as needed. The number of channel modules 430 in this alternative embodiment is determined by the acceptable probability of contention for the upstream stream-line signal. In this embodiment, arriving data packets are allocated a free channel module 430, rather than transmitted to a dedicated channel module 430. If all channel modules 430 are occupied, the data packet is dropped, or the end-device 130 is informed. Conventional methods exist to stop the end-device 130 from transmitting, or forcing it to retransmit the data packet. These methods include, but are not limited to: (1) sending an IEEE 802.3 Flow Control message or (2) asserting a collision signal to stop the end-computer from transmitting. When channel modules 430 are allocated on demand, a small header may be added to each data packet to indicate to the port module 320 on which channel port 430 the data packet should arrived. Likewise, when data packets are transmitted by the port module 320, a small header may be used to indicate to which channel module 430 the packet is to be sent.

While the present invention has been particularly shown and described with reference to various preferred embodiments, and several alternate embodiments, it will be understood by persons skilled in the relevant art that various changes in form and details can be made therein without departing from the spirit and scope of the invention.

CLAIMS

What is claimed is:

1. A network system having a plurality of end-devices, the system comprising:

a first multiplexer module, coupled to a first end-device and a second end-device of the plurality of end-devices, for receiving a first data packet signal from the first end-device and a second data packet signal from the second end-device and for transmitting a first stream-line signal with the first data packet signal and the second data packet signal; and

a redirector module, coupled to the first multiplexer module, for receiving the first stream-line signal and for transmitting a second stream-line signal with the second data packet signal and a third stream-line signal with the second data packet signal.

2. The network system of claim 1 wherein the redirector module includes a first port module coupled to the first multiplexer module for receiving the first stream-line signal, retrieving the first data packet signal and the second data packet signal from the first stream-line signal and for transmitting the first data packet signal and second data packet signal.

3. The network system of claim 1 wherein the redirector module includes a first port module for integrating the first data packet signal into the second stream-line signal and for transmitting the second stream-line signal.

4. The network system of claim 3 wherein the first port module includes a management unit for controlling the buffering and integration of the first data packet signal into the second stream-line signal.

5. The network system of claim 3 wherein the first port module includes a buffer module for storing the first data packet signal that cannot be integrated into the second stream-line signal.

6. The network system of claim 3 wherein the first port module includes a scheduler module for scheduling the integration of the first data packet signal into a third stream-line signal.

7. The network system of claim 1 wherein the redirector module includes a packet switch module for routing the first data packet signal and the second data packet signal.

8. The network system of claim 7 wherein the packet switch module includes a router module for routing the first data packet signal and the second data packet signal.

9. The network system of claim 7 wherein the packet switch module includes a routing table for storing the location of each of the plurality of end-devices in the system.

10. The network system of claim 7 wherein the packet switch module includes a local area network switch.

11. The network system of claim 1 further including a second multiplexer module, disposed between a third end-device of the plurality of end-devices and the redirector module, for receiving from the redirector module the second stream-line signal and for transmitting the first data packet signal to the third end-device.

12. The network system of claim 1 wherein the first multiplexer module includes a multiplexer/demultiplexer unit for multiplexing the first data packet signal and the second data packet signal into the first stream-line signal.

13. The network system of claim 1 wherein the first multiplexer module includes a multiplexer/demultiplexer unit for time-division multiplexing the first data packet signal and the second data packet signal into the first stream-line signal.

14. The network system of claim 1 wherein the first multiplexer module includes a multiplexer/demultiplexer unit for statistical multiplexing the first data packet signal and the second data packet signal into the first stream-line signal.

15. The network system of claim 1 wherein the first multiplexer module includes a first channel port coupled to the first end-device for receiving the first data packet signal from the first end-device.

16. The network system of claim 15 wherein the first channel port utilizes an Ethernet protocol to receive the first data packet signal from the first end-device.

17. A network system having a plurality of end-devices, the system comprising:

a first multiplexer module, coupled to a first end-device and a second end-device of the plurality of end-devices, for receiving a first data packet signal from the first end-device and a

second data packet signal from the second end-device and for transmitting a first stream-line signal including the first data packet signal and the second data packet signal;

a second multiplexer module coupled to a third end-device of the plurality of end-devices for receiving a second stream-line signal including the first data packet signal and for transmitting the first data packet signal to the third end-device; and

a redirector module disposed between the first multiplexer module and the second multiplexer module for receiving the first stream-line signal from the first multiplexer module and for transmitting the second stream-line signal to the second multiplexer module.

18. A method for transmitting data from a first end-device to a second end-device, the method comprising the steps of:

multiplexing a first data packet signal from the first end-device into a first stream-line signal comprising a plurality of data packet signals;

transmitting the first stream-line signal to a redirector module;

transmitting a second stream-line signal comprising the first data packet signal from the redirector module to a second multiplexer module, coupled to a second end-device; and

demultiplexing the second stream-line signal into the first data packet for the second end-device.

19. The method of claim 18 further comprising the step of transmitting the first data packet to the second end-device.

20. The method of claim 19 further comprising the step of storing a copy of a first portion of the first data packet signal that is currently being transmitted to the second end-device.

21. The method of claim 18 further comprising the step of scheduling the combining of the first data packet signal into the second stream-line signal.

22. A multiplexer module, disposed between a plurality of end-devices and a redirector module, the multiplexer module comprising:

a multiplexer/demultiplexer module for receiving a first data packet signal and a second data packet signal and for transmitting a first stream-line signal comprising the first data packet signal and the second data packet signal;

a first channel module, disposed between the multiplexer/demultiplexer module and a first end-device of the plurality of end-devices, for receiving the first data packet signal from the first end-device; and

a second channel module, disposed between the multiplexer/demultiplexer module and a second end-device of the plurality of end-devices, for receiving the second data packet signal from the second end-device.

23. The multiplexer module of claim 22 wherein the first channel module communicates by Ethernet protocol with the first end-device to receive the first data packet signal and the second channel module communicates by Ethernet protocol with the second end-device to receive the second data packet signal.

24. The multiplexer module of claim 22 further comprising a control unit, coupled to the multiplexer/demultiplexer module for controlling the transmission of the second stream-line signal from the redirector module.

25. The multiplexer module of claim 22 further comprising a buffer module, coupled to the first channel module, for storing a first portion of a first data packet signal that is unable to be transmitted by the first channel module to the first end-device.

26. The multiplexer module of claim 22 further comprising

a control unit, coupled to the multiplexer/demultiplexer module for controlling the transmission of the second stream-line signal from the redirector module; and

a buffer module, coupled to the first channel module, for storing a first portion of a first data packet signal that is unable to be transmitted by the first channel module to the first end-device.

27. A redirector module disposed between a first multiplexer module and a second multiplexer module, the redirector module comprising:

a first port module coupled to the first multiplexer module for receiving a first stream-line signal, retrieving a first data packet signal and a second data packet signal from the first stream-line signal and for transmitting the first data packet signal and the second data packet signal;

a packet switch module, coupled to the first port module, for routing the first data packet signal and the second data packet signal; and

a second port module, coupled to the packet switch module, for receiving the first data packet signal, integrating the first data packet signal into a second stream-line signal and for transmitting the second stream-line signal comprising the first data packet signal.

28. A port module disposed between a multiplexer module and a packet switch module, the port module comprising:

a management unit for controlling the receiving of a first stream-line signal comprising a first data packet signal and a second data packet signal from the multiplexer module and the transmitting of the first data packet signal and the second data packet signal to the packet switch module, for controlling the receiving of a third data packet signal and a fourth data packet signal from the packet switch module, the buffering of the third data packet signal and fourth data packet signal and the transmitting of a second stream-line signal comprising the third data packet signal and the fourth data packet signal to the multiplexer module; and

a buffer module for storing a first portion of the third data packet signal that is transmitted to the second stream-line signal.

29. The port module of claim 28 further comprising a scheduler module for determining the order and timing of transmitting the third data packet within a third stream-line signal to be transmitted to the multiplexer module.

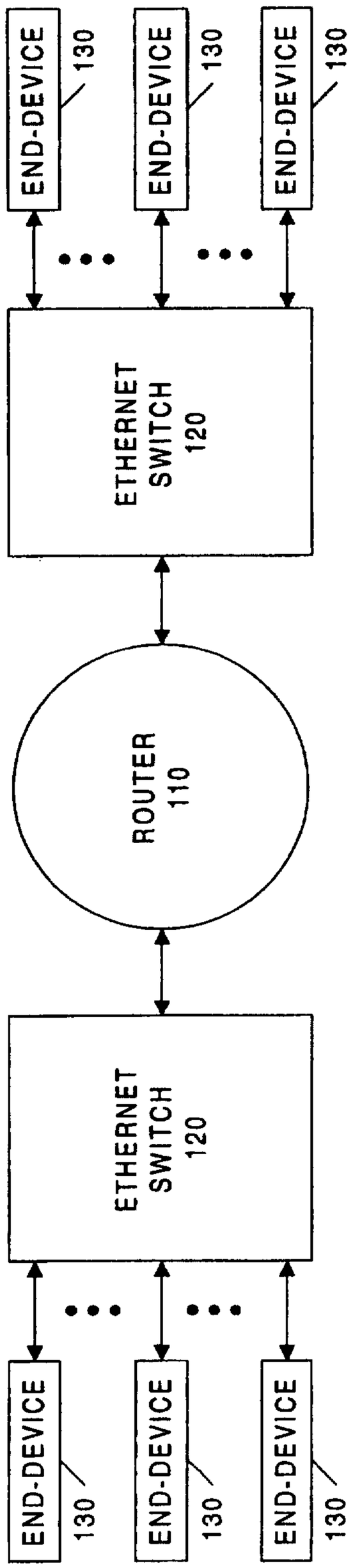


FIGURE 1 (PRIOR ART)

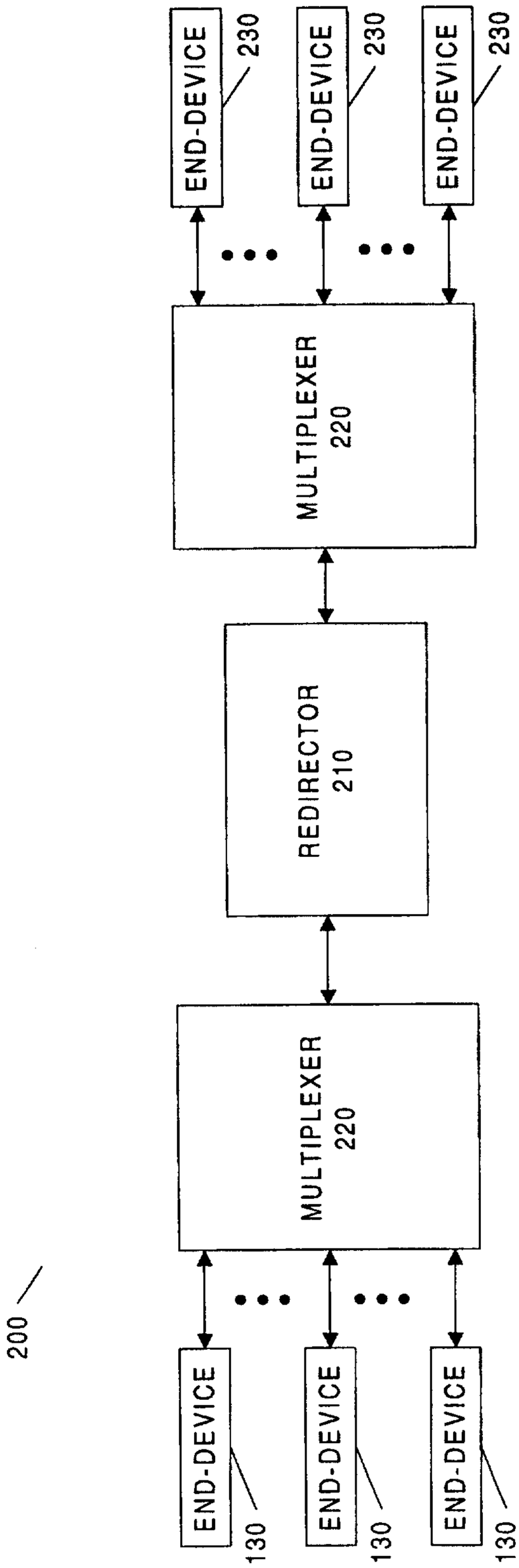

END-DEVICE
230
END-DEVICE
230
END-DEVICE
230
MULTIPLEXER
220
REDIRECTOR
210
MULTIPLEXER
220
200
END-DEVICE
130
END-DEVICE
130
END-DEVICE
130

FIGURE 2

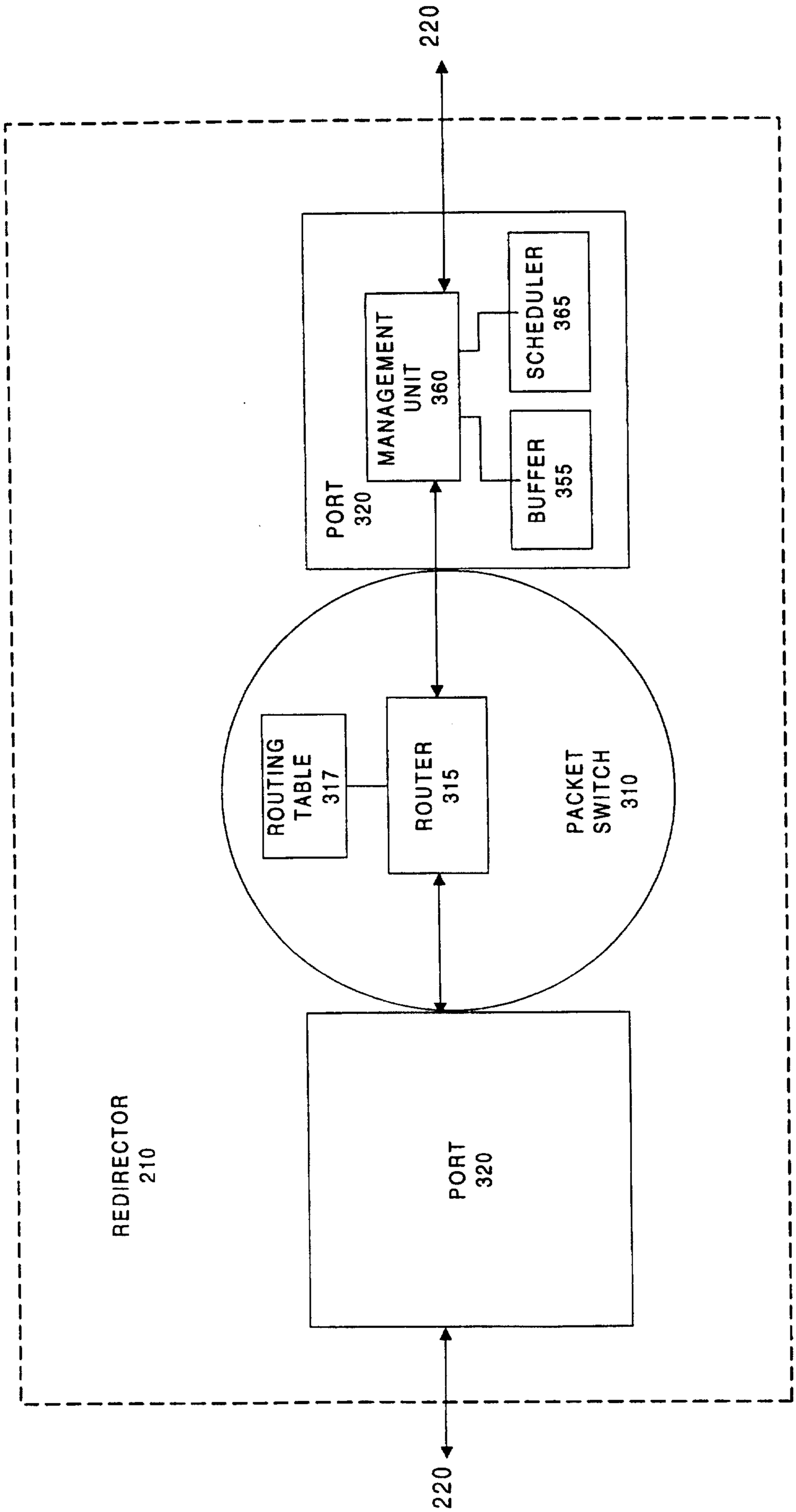

220
REDIRECTOR
210
PORT
320
MANAGEMENT
UNIT
360
SCHEDULER
365
BUFFER
355
ROUTING
TABLE
317
ROUTER
315
PACKET
SWITCH
310
PORT
320
220

FIGURE 3

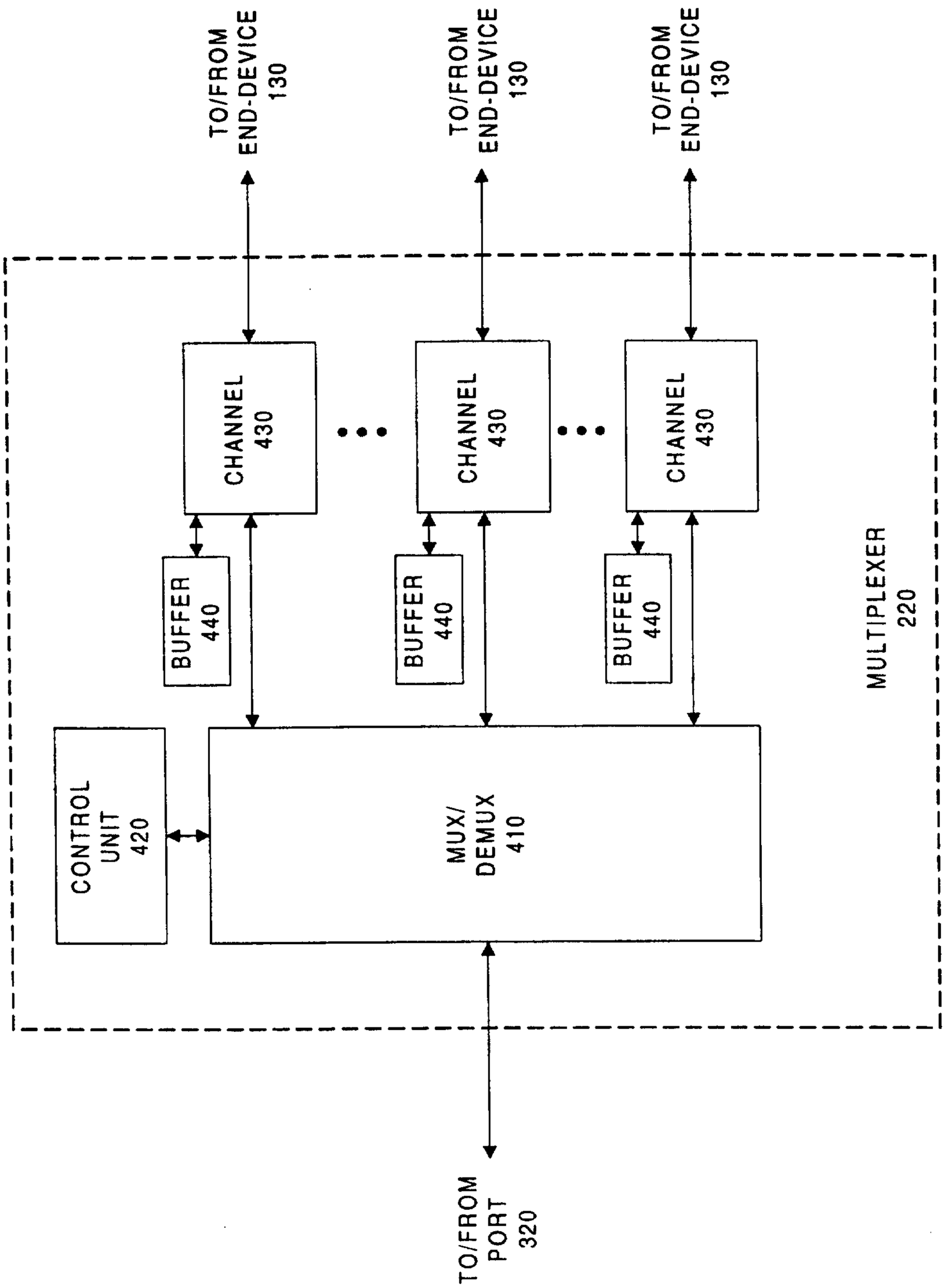

TO/FROM END-DEVICE 130
TO/FROM END-DEVICE 130
TO/FROM END-DEVICE 130
CHANNEL 430
CHANNEL 430
CHANNEL 430
BUFFER 440
BUFFER 440
BUFFER 440
CONTROL UNIT 420
MUX/ DEMUX 410
MULTIPLEXER 220
TO/FROM PORT 320

FIGURE 4

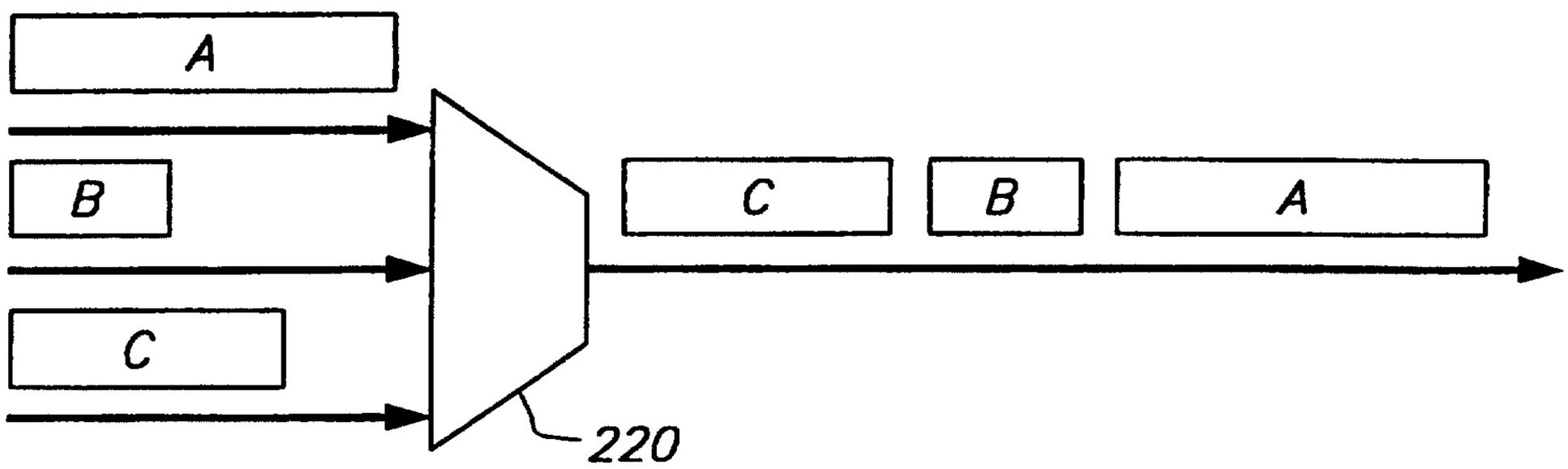


FIG. 5

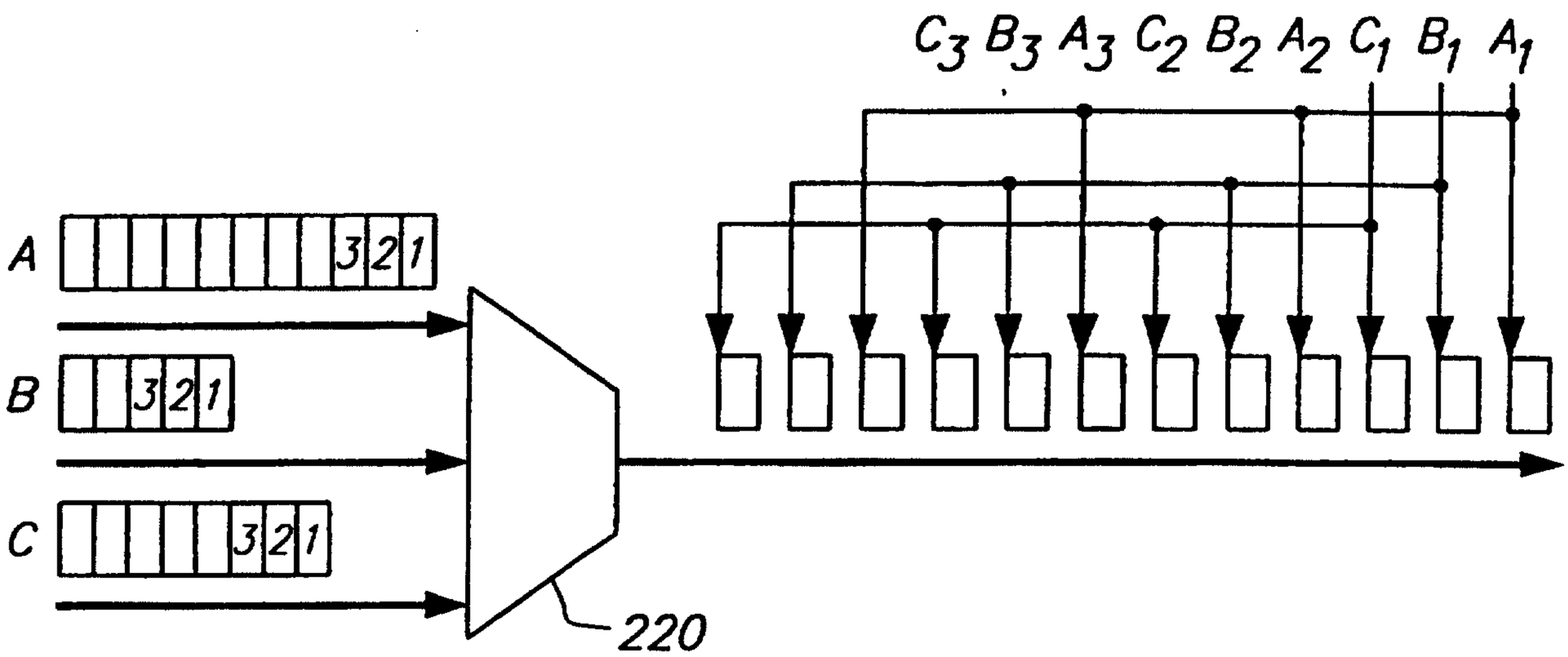


FIG. 6

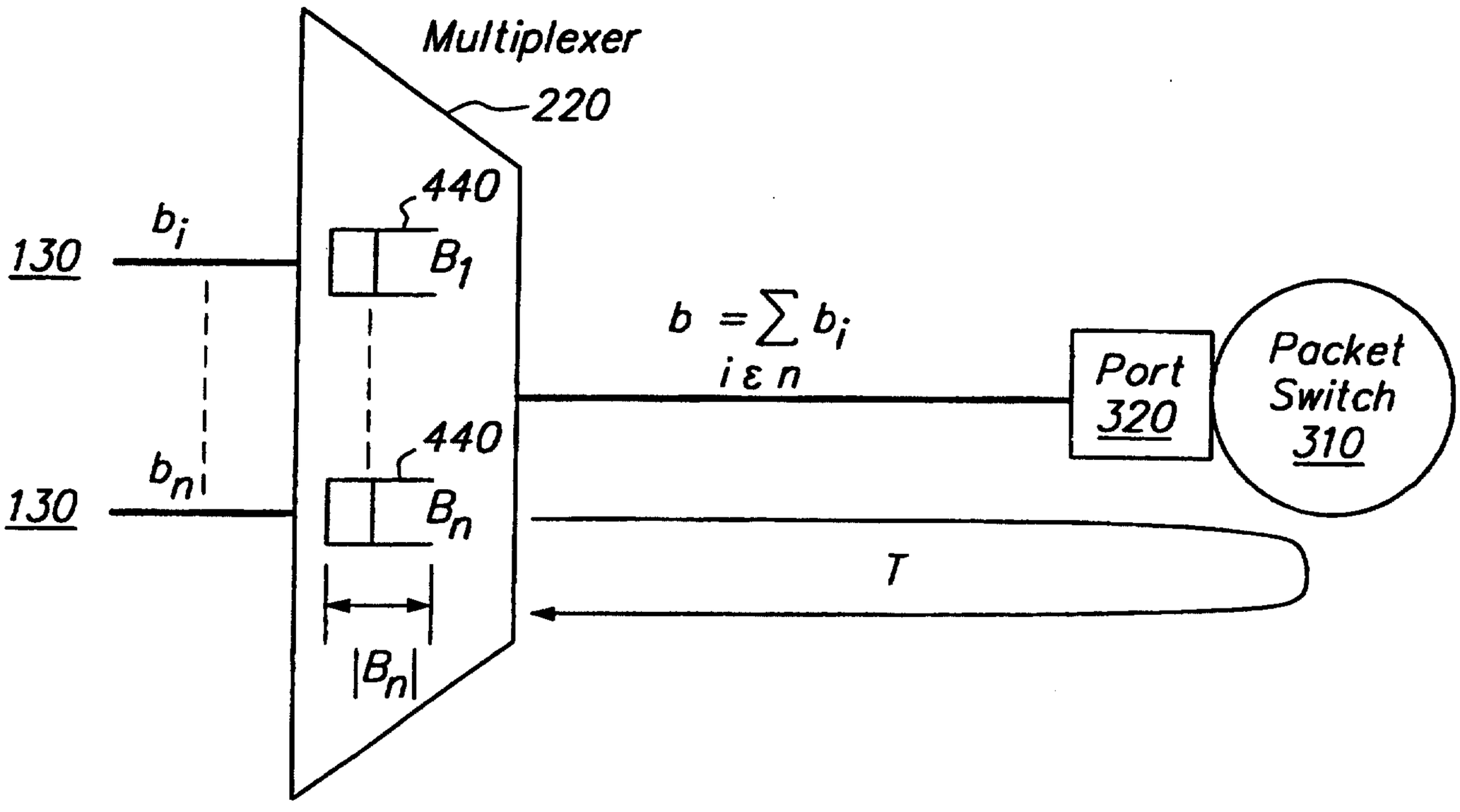

Multiplexer
220
440
b_i
130
B_1
$b = \sum_{i \varepsilon n} b_i$
Port
320
Packet
Switch
310
440
b_n
130
B_n
T
$|B_n|$

FIG. 7

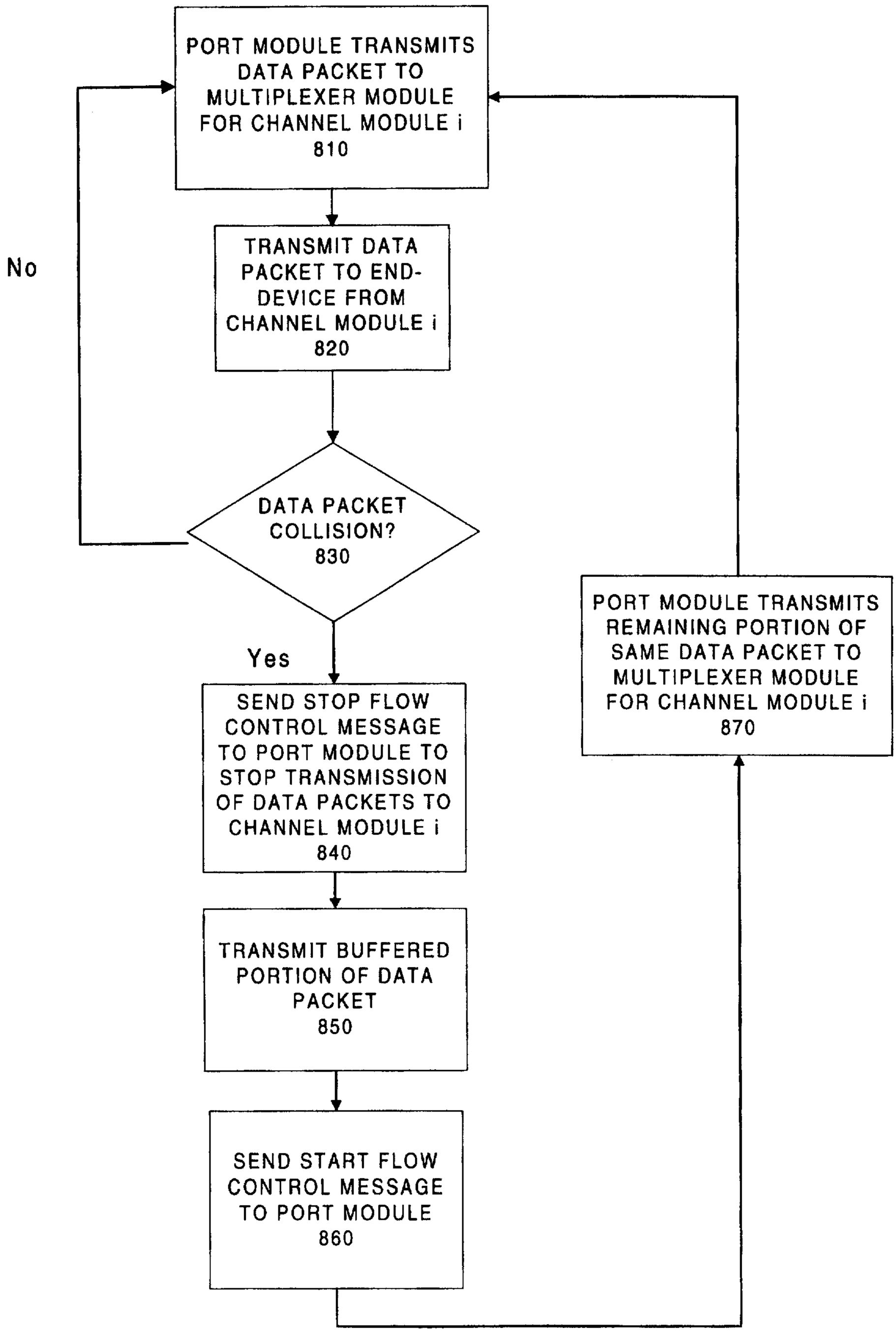

PORT MODULE TRANSMITS DATA PACKET TO MULTIPLEXER MODULE FOR CHANNEL MODULE i 810
TRANSMIT DATA PACKET TO END-DEVICE FROM CHANNEL MODULE i 820
DATA PACKET COLLISION? 830
No
Yes
SEND STOP FLOW CONTROL MESSAGE TO PORT MODULE TO STOP TRANSMISSION OF DATA PACKETS TO CHANNEL MODULE i 840
TRANSMIT BUFFERED PORTION OF DATA PACKET 850
SEND START FLOW CONTROL MESSAGE TO PORT MODULE 860
PORT MODULE TRANSMITS REMAINING PORTION OF SAME DATA PACKET TO MULTIPLEXER MODULE FOR CHANNEL MODULE i 870

FIGURE 8