PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 12/46

A2

(11) International Publication Number: **WO 97/24841**

(43) International Publication Date: 10 July 1997 (10.07.97)

(21) International Application Number: PCT/US96/20205

(22) International Filing Date: 18 December 1996 (18.12.96)

(30) Priority Data:
08/581,134 29 December 1995 (29.12.95) US

(71) Applicant: CISCO SYSTEMS, INC. [US/US]; 170 W. Tasman Avenue, San Jose, CA 95134 (US).

(72) Inventors: CHERITON, David, R.; 131 Cowper Street, Palo Alto, CA 94301 (US). BECHTOLSHEIM, Andreas, V.; 1140 Hamilton Avenue, Palo Alto, CA 94301 (US).

(74) Agents: D'ALESSANDRO, Kenneth et al.; D'Alessandro & Ritchie, P.O. Box 640640, San Jose, CA 95164-0640 (US).

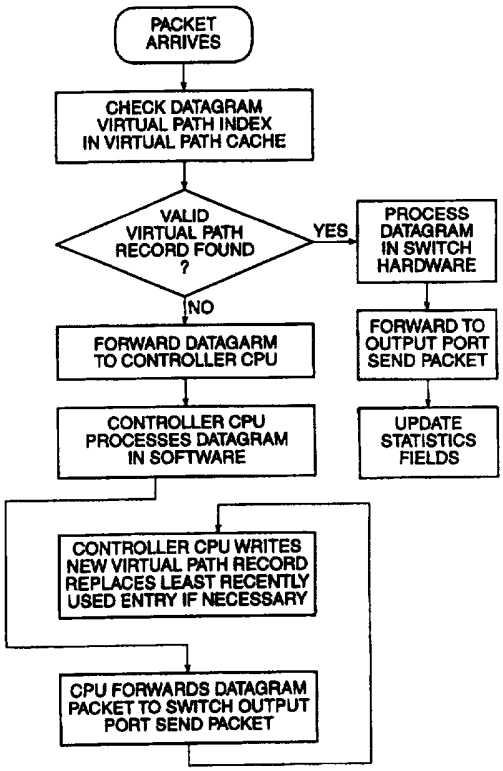
(81) Designated States: AU, CA, JP, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: DATAGRAM TRANSMISSION OVER VIRTUAL CIRCUITS

(57) Abstract

A method and apparatus for an enhanced datagram packet switched computer network is disclosed. The invention processes network datagram packets in network devices as separate flows, based on the source-destination address pair contained in the datagram packet itself. As a result, the network can control and manage each flow of datagrams in a segregated fashion. The processing steps that can be specified for each flow include traffic management, flow control, packet forwarding, access control and other network management functions. The ability to control network traffic on a per flow basis allows for the efficient handling of a wide range and a large variety of network traffic, as is typical in large-scale computer networks, including video and multimedia type traffic. The amount of buffer resources and bandwidth resources assigned to each flow can be individually controlled by network management. In the dynamic operation of the network, these resources can be varied based on actual network traffic loading and congestion encountered. The invention also teaches an enhanced network access control method which can selectively control flows of datagram packets entering the network and traveling between network nodes. This new network access control method also interoperates with existing media access control protocols, such as used in the Ethernet or 802.3 local area network. An important aspect of the invention is that it does not require any changes to existing network protocols or network applications. This is accomplished by specifying the flow control, traffic management and network control functions via network management. Applications or network protocols that require the network to provide a certain level of bandwidth or performance can request such services via the network management function.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AM	Armenia	**GB**	United Kingdom	**MW**	Malawi
AT	Austria	**GE**	Georgia	**MX**	Mexico
AU	Australia	**GN**	Guinea	**NE**	Niger
BB	Barbados	**GR**	Greece	**NL**	Netherlands
BE	Belgium	**HU**	Hungary	**NO**	Norway
BF	Burkina Faso	**IE**	Ireland	**NZ**	New Zealand
BG	Bulgaria	**IT**	Italy	**PL**	Poland
BJ	Benin	**JP**	Japan	**PT**	Portugal
BR	Brazil	**KE**	Kenya	**RO**	Romania
BY	Belarus	**KG**	Kyrgystan	**RU**	Russian Federation
CA	Canada	**KP**	Democratic People's Republic of Korea	**SD**	Sudan
CF	Central African Republic			**SE**	Sweden
CG	Congo	**KR**	Republic of Korea	**SG**	Singapore
CH	Switzerland	**KZ**	Kazakhstan	**SI**	Slovenia
CI	Côte d'Ivoire	**LI**	Liechtenstein	**SK**	Slovakia
CM	Cameroon	**LK**	Sri Lanka	**SN**	Senegal
CN	China	**LR**	Liberia	**SZ**	Swaziland
CS	Czechoslovakia	**LT**	Lithuania	**TD**	Chad
CZ	Czech Republic	**LU**	Luxembourg	**TG**	Togo
DE	Germany	**LV**	Latvia	**TJ**	Tajikistan
DK	Denmark	**MC**	Monaco	**TT**	Trinidad and Tobago
EE	Estonia	**MD**	Republic of Moldova	**UA**	Ukraine
ES	Spain	**MG**	Madagascar	**UG**	Uganda
FI	Finland	**ML**	Mali	**US**	United States of America
FR	France	**MN**	Mongolia	**UZ**	Uzbekistan
GA	Gabon	**MR**	Mauritania	**VN**	Viet Nam

DATAGRAM TRANSMISSION OVER VIRTUAL CIRCUITS

Field of the Invention

The present invention relates to the field of computer networks. More particularly, the present invention relates to the field of computer networks which are based on datagram packet switching.

Background of the Invention

Computer Networks are used to interconnect computers and peripherals to allow them to exchange and share data such as files, electronic mail, databases, multimedia/video, and other data.

Packet Switching

Nearly all computer networks use packet switching, which divides longer units of data transfer into smaller packets which are sent separately over the network. This allows each packet to be processed independently from another packet without having to wait for the entire data transfer to be completed. It also enables communications between a plurality of computer systems to be intermixed on one network. Host interfaces connect the computer systems to a network allowing each computer system to act as the source and destination of packets on the network.

A first key issue in packet switched networks is addressing. The addressing in packet switched networks is conventionally performed by one of two approaches, known as virtual circuit packet switching or datagram packet switching.

Virtual Circuit Packet Switching

In the virtual circuit approach, before any data can be transmitted, a virtual circuit must be first established along the path from the source to the destination in advance

of any communication. After the virtual circuit is setup, the source can then send packets to the destination. Each packet in the virtual circuit approach has a virtual circuit identifier, which is used to switch the packet along the path from source to the destination.

The virtual circuit approach reduces the size of the identification required in each packet header. It also allows additional information about the packet handling to be established as part of the virtual circuit setup operation. Another claimed benefit- is that forwarding and switching of virtual circuit packets can be made more efficient because of the virtual setup process. However, the virtual circuit approach incurs the cost of delay to setup the virtual circuit before sending any data, and it incurs the cost of maintaining the virtual circuit state in each network device along the virtual circuit path, even if a virtual circuit is idle. Also, in practice the memory space for virtual circuit state in network devices has limited the number of circuits that are available, which complicates the behavior of network nodes that need to create virtual circuits to communicate.

Datagram Packet Switching

In the datagram approach, each datagram packet is a self-contained unit.of data delivery. A typical datagram packet includes a globally unique source address, a globally unique destination address, a protocol type field, a data field, and a cyclic redundancy checksum ("CRC") to insure data integrity.

Datagrams can be sent without prior arrangement with the network, i.e. without setting up a virtual circuit or connection. Each network device receiving a datagram packet examines the destination address included in the datagram packet and makes a local decision whether to accept, ignore, or forward this packet.

Various conventional network devices learn information from observing datagram packet traffic in data networks. For example, a conventional network switch device that interconnects multiple network segments can "learn" the location of network stations connected to its ports by monitoring the source address of packets received on its ports. After it has associated a station address with a certain port, the network switch can then forward datagram packets addressed to that station to that port. In this type of device, the datagram source address is used

to learn the location of a station on the network, whereas the forwarding decision is made on basis of the datagram destination address alone.

Datagram packet switching has the advantage that it avoids the overhead and cost of setting up virtual circuit connection in network devices. However, it incurs the expense of transmitting a larger packet header than required for virtual circuit switching, and it incurs the cost for processing this larger packet header in every network device to which it is delivered. Also, there is no virtual circuit setup process to establish additional information for datagram packet processing. Another disadvantage of datagram packet switching is that it is difficult to control packet flow to the same degree as with virtual circuits because there is, in the conventional case, no state in the network devices associated with the traffic flow.

The datagram packet switching approach has been extensively used in shared media local area networks. Shared media networks provide for a multiplicity of stations directly connected to the network, with the shared media providing direct access from any transmitter to any receiver. Since the receivers need to be able to distinguish packets addressed specifically to them, each receiver needs to have a unique address. In addition, since the unit of access to the shared medium is one packet, each packet needs to contain the unique address capable to identify the receiver. As a result, all commonly used local area networks are based on datagram packet switching and have no provisions for virtual circuit setup.

Media Access Control Protocol

The network access mechanism in shared media local area network will now be further described. This function, commonly known as the media access control or MAC protocol, defines how to arbitrate access among multiple stations that desire to use the network. Individual stations connected to the network have to adhere to the MAC protocol in order to allow proper network operation.

A number of different media access control protocols exist. The MAC protocol, in conjunction with the exact packet format, is the essence of what defines a local area network standard. The following is a brief overview of local area network standards that are in wide use today.

The most widely used local area network is commonly known as Ethernet and employs an access protocol referred to as Carrier Sense Multiple Access with Collision Detection (CSMA/CD). [see U.S. Patent 4,063,220, issued Dec 13, 1977, for a Multipoint Data Communication System with Collision Detection, Inventors Metcalfe, Boggs, Thacker, and Lampson]. The current definition of the Ethernet CSMA-/CD protocol is defined in IEEE Standard 802.3, published by the Institute of Electrical and Electronics Engineers, 345 East 45th Street, New York, NY 10017. The Ethernet standard specifies a data transmission rate of 10 or 100 Megabits/second.

Another widely used local area network standard is Tokenring, also known as IEEE Std. 802.5, transmitting at a speed of 4 or 16 Mbits/sec and FDDI, or Fiber-Distributed-Data-Interface, which sends data at a speed of 100 Mbits/sec. Both Tokenring and FDDI are based on a circulating token granting access to the network, although their respective datagram packet formats and other operating aspects are unique to each standard.

What is common to all these media access control mechanisms is that they do not include provisions for virtual circuit setup and have no provisions to specify attributes that relate to virtual circuits, such as traffic management or flow control for specific connections. This limits the ability of conventional local area networks to accommodate higher level network functions or to support virtual connection oriented traffic mechanisms.

Devices for Interconnecting Local Area Networks Another key issue with datagram packet switched networks is how to interconnect individual network segments into larger networks. The size and usage of datagram packet switched networks has grown much beyond what was envisioned when these networks were designed. Devices such as bridges, switches, and routers, have been used to interconnect individual LAN segments into larger networks, but each have their own set of problems in scaling to higher performance.

Bridges forward datagram packets between network segments by learning the location of the devices on the network by observing the source address contained in datagram packets passing by. once the bridge has learned which network device ia located on which network segment, it can then forward datagram

packets addressed to that network device to the appropriate network segment. one of the limitations of bridges is that they do not filter traffic beyond the data link level.

Switches are basically multi-port network bridges that can forward datagram packets among a multiplicity of network ports. Frequently, switches provide additional capabilities for assisting with network management, including traffic filtering and segmenting networks into virtual LANS. As in the case of bridges, switches have to forward broadcasts to all ports configured into one virtual LAN. In addition, conventional switches cannot provide fair service or priority service to individual traffic flows, and they require significant amount of memory to avoid dropping packets in the case of network congestion.

Routers also interconnect several network segments, but they operate primarily at the network protocol layer, rather than at the datagram packet layer. Routers participate in all network protocol functions, which traditionally requires general purpose processing. As a result, traditional routers are more expensive and have less throughput than switches. -In addition they are more difficult to administer.

Finally, virtual circuit packet switched networks, in particular ATM, have been proposed to interconnect local area network segments. However, it has turned out to be very difficult to map existing network protocols that are based on datagram packets to the ATM network architecture.

In summary, bridges and switches transparently extend the domain of networks, and allow for cost-effective and high-performance implementations. However, they cannot segment a network effectively in terms of traffic management and broadcast traffic. Routers, on the other hand, can segment networks very effectively, but are much more expensive and are performance bottleneck in high-speed networks. ATM has been very difficult to map to current network protocols.

The ideal network device for interconnecting network segments would have the high-speed and cost-effectiveness of a switch, with the ability of segment and manage network traffic similar to a router.

Traffic Management

Another key issue in packet switched networks is traffic control or traffic management.

In a packet switched network, each link at every switching node in the network represents a queue. As the traffic arrival rate at a link approaches its transmission rate, the queue length grows dramatically and the data in the queue needs to be stored in the attached network nodes. Eventually, a network node will run out of packet buffer capacity which will cause further packets arriving to be discarded or dropped. Dropped packets are eventually retransmitted by the source, causing the traffic load to increase further. Eventually, the network can reach a state where most of the packets in the network are retransmissions.

Conventionally, two types of traffic control mechanism are used in packet switched networks: flow control and congestion control. Flow control is concerned with matching the transmission rate of a source station to the reception rate of a destination station. A typical networks flow control mechanism uses a window techniques to limit the number of packets a source can transmit which are not yet confirmed as having been received by the destination. Conventional flow control is an end-to-end mechanism that exists in certain network protocols, in particular connection oriented network protocols such as TCP/IP. However, conventional flow control between source and destination does not solve the network congestion problem, since it does not take the utilization of buffer resources within the network into account. In addition, non-connection oriented network protocols do not use window based flow control. Also, continuous rate traffic sources such as real-time video don't match the nature of destination controlled behavior since the transmission rate is determined by the source.

Problem Statement

What is needed is an improved method and apparatus for high-speed datagram packet switched networks that can support a large number of network stations, a wide range of network transmission speeds, a wide variety of source traffic behavior including video and multimedia, while maintaining compatibility with existing network protocols and applications.

SUMMARY OF THE INVENTION

Methods and apparatus for an enhanced datagram packet switched computer network are disclosed.

The invention processes network datagram packets in network devices as.separate flows, based on the source-destination address pair contained in the datagram packet itself. As a result, the network can control and manage each flow of datagrams in a segregated fashion. The processing steps that can be specified for each flow include traffic management, flow control, packet forwarding, access control, and other network management functions.

The ability to control network traffic on a per flow basis allows for the efficient handling of a wide range and a large variety of network traffic, as is typical in large-scale computer networks, including video and multimedia type traffic.

The amount of buffer resources and bandwidth resources assigned to each flow can be individually controlled by network management. In the dynamic operation of the network, these resources can be varied based on actual network traffic loading and congestion encountered.

The invention also includes an enhanced network access control method which can selectively control flows of datagram packets entering the network and traveling between network nodes. This new network access control method interoperates with existing media access control protocols, such as used in the Ethernet or 802.3 local area network.

An important aspect of the invention is that it can be implemented in network switching devices at very high performance and at low cost. High performance is required to match the transmission speed of datagram packets on the network. Low cost is essential such that it is economical to use the invention widely.

In the preferred implementation, both high-performance and low cost is achieved by partitioning the task of datagram flow processing between dedicated network switch hardware-and dedicated network switch software that executes on a high-speed controller CPU.

The network switch hardware provides a multiplicity of network ports, a shared memory buffer for storing datagram packets, a virtual path cache that stores the state and processing instructions specific to the active datagram packet flows.

Datagram packets received on an input port are buffered in the shared memory buffer. The source-destination address pair in the datagram packet header is used to index the virtual path cache to find a matching entry. If a matching entry is found in the virtual path cache, then the switch hardware performs all the packet processing steps indicated in the virtual path record, including traffic management and packet routing.

If no matching entry is found in the virtual path cache, then the datagram packet is forwarded to the controller CPU for general purpose processing. The controller CPU determines, through network management data structures and software, how to process further datagram packets with this source-destination address in the switch hardware. The controller CPU then loads an appropriate entry into the virtual path cache. If all entries in the virtual path cache are in use, then the CPU removes the least recently used entry before loading the new entry.

Brief Description of the Drawings

Other features and advantages of the present invention will become more apparent to those skilled in the art from the following detailed description in conjunction with the appended drawings in which:

Figure 1 is a block diagram of a computer communication system;

Figure 2 is a block diagram of the Ethernet datagram packet format;

Figure 3 illustrates the Virtual Path Record data structure;

Figure 4 is a block diagram of the network switching device;

Figure 5 is a flow diagram of the network switching device operation;

Figure 6 is a block diagram of the virtual path cache;

Figure 7 is a block diagram of the virtual path hash function; and

Figure 8 is a block diagram of the transmit data structure.

Detailed Description of the Invention

An enhanced computer network communication system is disclosed.

To help understand the invention, the following definitions are used:

A "datagram packet" is a self-contained unit of packet delivery on a packet switched network, which includes a destination address field, a source address field, an optional type field, and a data field.

The "destination address" and the "source address" refer to the physical network device addresses contained in a datagram packet, both of which are unique within a network.

A "flow" is a plurality of datagram packets each packet containing an identical source-destination address pair.

A "virtual path" is the communication path from a source to a destination in a datagram packet switched network.

In the following description, for purposes of explanation, specific numbers, times, signals, and other parameters are set forth in order to provide a thorough understanding of the present invention. However, it will be apparent to anyone skilled in the art that the present invention may be practiced without these specific details. In other instances, well known circuits and devices are shown in block diagram in order not to obscure the present invention unnecessarily.

The datagram packet switched communication system is illustrated in Figure 1 by way of four network switching devices 101, 102, 103, and 104 interconnected with each other via backbone links 105, 106, 107, and 108. In addition, switch 103 is

interconnected via links 110 through 112 to client computers 120 through 122, switch 104 is interconnected via links 113 through 115 to client computers 123 to 125 and switch 102 is interconnected via links 116 and 117 to server computers 126 and 127.

The network shown in Figure 1 is small for purposes of illustration. In practice most networks would include a much larger number of host computers and network switching devices.

The basic unit of transmission and processing in this network is a datagram. For purposes of Illustration, we will be using the Ethernet datagram packet format in this description. it will be apparent to anyone skilled in the art that other datagram packet formats can also be used to practice this invention, including the different datagrams described in the IEEE 802 family of network standards.

As illustrated in Figure 2, the Ethernet datagram 200 contains a 48-bit destination address field 201, a 48-bit source address field 202, a 16-bit type field 203, a variable size data field 204 ranging from 46 bytes to 1500 bytes, and a 32-bit CRC field 205.

A key aspect of the present invention is the virtual path method. A virtual path is specified by the source-destination address pair contained in a datagram packet. In the Ethernet packet datagram, the source-destination address pair can be thought of as a 96-bit circuit identifier, which specifies a unidirectional circuit from the source to the destination. This 96-bit circuit identifier will subsequently be referred to as a virtual path in order to avoid confusion with virtual circuit networks. While this 96-bit virtual path identifier may appear large as compared to the much smaller circuit identifiers in virtual circuit networks, it has the significant advantage that it is globally unique and thus does not require to be mapped to different identifiers as a datagram packet travels along the path from source to destination.

Virtual Path Record

Each datagram packet arriving at a network switching device is recognized as traveling on a particular virtual path by the source-destination address pair contained in this header. The network switch maps this source-destination pair to a

virtual path record in the switch. The virtual path record specifies how the datagram packet is to be processed, including its routing, priority, scheduling and buffer resource allocation.

Figure 3 illustrates the data structure of the virtual path record.

It will be apparent to those skilled in the art that other data structures from the one shown can be successfully used, including but not limited to fields of different size, arranging the fields in different order, and additional fields not present in Figure 3.

Turning now to the specific virtual path record illustrated in Figure 3, there are four groups of fields: the tag field 310, the forwarding field 320, the state field 330, and the statistics field 340. The function and meaning of these fields will now be described.

Tag Field

The purpose of the tag field is to match an incoming datagram packet against a virtual path record. The tag field 310 has four subfields: the destination address field 311, the source address field 312, the optional type field 313, and the input port field 314.

Since the virtual path index is quite large, 96 bits in the case of Ethernet datagrams, it is not practical to provide a full array indexed by the virtual path number. Instead, each virtual path record is keyed, with the virtual path number to allow lookup by search or partial index. One method for organizing the virtual path records and looking them up will be further described below.

For the lookup method to locate the correct virtual path record, the destination address field 311 and the source address field 312 in the virtual path record must match the destination address field 201 and the source address field 202 in the datagram header.

Type field 313 allows for optional type filtering. If the type field is set to 0, any type field 203 in the datagram header will match this virtual path record.

However, if the type field is not 0, then the type field has to match the type field 203 in the datagram header exactly for the match to be successful.

The input port field 314 allows input port filtering. The input port field has to match the actual input port number at which the datagram packet has been received.

Forwarding Field

The forwarding field 320 determines how the datagram packet should be forwarded. Output port field 321 specifies the output port on which this datagram will be transmitted. Priority field 322 specifies the traffic management priority of this virtual path compared to other virtual paths. Real time field 323 forces the switch to process this packet in real time mode, which includes the act of dropping the packet if it cannot be sent within a certain time.

Store Forward field 324 selects the store-forward mode of operation. Normally the switch operates in cut-through mode where an incoming datagram is sent on to the output port as soon as feasible, even before it is completely received. In store-forward mode, an incoming packet must be completely received before it is sent. This is a requirement in case of speed conversion from a slower input port to a faster output port. Store-forward mode is also used to insure the correctness of the complete packet received before sending it on.

Multicast field 325 selects the multicast mode of operation. Multicast mode involves scheduling of a single datagram packet on multiple outputs, which are determined by a bit vector in the output port field, with "1" bits indicating output ports to which the Multicast should be sent.

Field 326 is the Snoop Mode. Snoop mode when selected sends a copy of the datagram packet to the CPU for general purpose processing.

Field 327 is the Buffer Size field, which specifies the maximum number of packet buffers allocated to this path for buffering purposes.

The state field 330, includes the following fields; Head Pointer Field 331, which points to the beginning of the buffer area associated with this virtual path;

Tail Pointer 332 points to the end of the buffer area of this virtual path; Uplink Pointer 333 points to the next virtual path record to transmit to the source, downlink pointer 334 points to.the next virtual path record to transmit to the destination.

The statistics field 340 maintains traffic statistics regarding the traffic received on this virtual path. Field 341 counts the number of packets received and field 342 counts the number of bytes received on this virtual path.

Figure 4 Preferred Implementation

Referring to Figure 4, a virtual path network switching device 400 is illustrated with Network Input Ports 401 through 404, network output ports 405 through 408, switch hardware 409, where shared buffer memory 410, controller CPU 411, CPU interface 412, CPU read-and-write memory 413, Flash PROM 414, and virtual path cache 415. Using Figure 4 and the flow chart in Figure 5, a cycle of operation will be described.

Datagram packets arriving through network ports 401 through 404 are temporarily stored in shared buffer memory 410. As soon as the datagram packet header has arrived, which in the case of Ethernet datagrams is after the first 14 bytes of the datagram packet, the virtual path cache is looked up to check whether a virtual path cache entry exists for this path. If a matching entry is found in the virtual path cache 415, then switch hardware 409 starts processing the datagram packet as specified in the virtual path cache entry which in the typical case will forward the datagram packet on one of the output ports 405 through 408.

If no entry matching the datagram was found in the virtual path cache 415, then the datagram packet is forwarded to controller CPU 411 via CPU interface 412 for general purpose processing. Controller CPU 411 processes datagram packet according to instructions and data stored in main memory 413 and optionally in Flash PROM 414. Said instructions and data structures used for datagram processing have been created previously by network management, network configuration, network statistics, and network behavior.

The result of the datagram general purpose processing is that the CPU determines how future datagram packets on this virtual path should be processed by

the switch hardware 409 and loads an appropriate entry into the virtual path cache 415. If all entries in the virtual path cache 415 are in use, then controller CPU 409 removes the least recently used entry in virtual path cache 415 before loading the new entry. CPU 411 then forwards the datagram packet to the switch hardware 409 via CPU interface 412 for transmission.

When the controller CPU loads a new virtual path cache entry, it sets the tag field 310 to the desired virtual path index, the forwarding field 320 to the desired forwarding function, and it initializes the state field 330 and the statistics field 340. The switch hardware will then automatically update the state and the statistics fields as the path is used. The switch hardware does not modify the information in the tag field and the forwarding field.

Figure 5 Flow Diagram

A method in accordance with this invention is shown in the flow diagram of Figure 5.

1. Packet Arrives at the invented system.
2. Check Datagram Virtual Path Index against Virtual Path Cache.
3. If valid virtual path entry is found, process packet in Switch Hardware which includes the steps of forwarding the datagram packet to output port and updating the statistics field 340.
4. If No Entry is found, forward packet to Controller CPU to process datagram packet in software. Controller CPU sends datagram packet back to switch hardware for transmission on the appropriate output port.
5. If Controller CPU determines that future datagram packets of this virtual path should be processed by switch hardware, Controller CPU then creates new virtual path record and writes it into virtual path cache, replacing the least recently used entry if necessary.

Virtual Path Cache organization

Figure 6 illustrates an example of virtual path cache organization.

It will be apparent to anyone skilled in the art that other cache sizes and organizations from the one shown can be successfully used, including but not limited to caches of different size, associativity, alternative hash-function, and content-addressability.

The virtual path cache illustrated in Figure 6 is organized as a 4-set associative cache built with four banks of high-speed static memory 601 through 604, each equipped with a set of comparators 611..614 that control a set of tri-state buffers 621 through 624.

The virtual path index 630, which is the source-destination address pair of the incoming datagram, enters hash function 631 which in turn produces a virtual patch cache index 632 which in turn looks up the four parallel sets of the virtual path cache SRAms 601 through 604. The tag field 310 from each set of SRAMs will be compared against the virtual path index 630 and only that virtual path record that matches will be output on the virtual path record databus 633 via tri-state drivers 621 through 624. Combinatorial logic 634 will generate a high signal 635 to indicate a hit.

If no tag field matches, then combinatorial logic 634 will generate a low signal 635 to indicate a miss; i.e. that no valid virtual path record was found in the virtual path cache.

Hash Function

Figure 7 illustrates a specific hash function embodied in hash function logic 631. Again, this hash function is used for illustration only. It will be apparent to anyone skilled in the art that other hash functions from the one shown can be used.

Referring now to Figure 7, the specific hash function logic is the bitwise Exclusive-OR 703 between the low-order 15 bits of the destination address 701 and the source address 702 of the Virtual Path Index 630, producing the 15 bit virtual path cache index 632. A bit-wise Exclusive-OR function is used because it is simple and fast to implement. The low-order address bits are used from both source and destination address since they change with the highest frequency.

Packet Transmission

Figure 8 shows a data structure that illustrates how an output port transmits datagram packets buffered in the switch shared memory. This particular structure design is for purposes of illustration only; it will be apparent to anyone skilled in the art that other data structures from the one shown can be used successfully.

The data structure in Figure 8 is for one output port only. Referring briefly to Figure 4, each output port 405 through 408 in the switch hardware 409 has a similar data structure to that shown in Figure 8.

Output port 801 is to transmit datagrams buffered and waiting for transmission on virtual paths 810-1, 810-2 through 810-*n*, where *n* is a selected integer.

The output port 801 has a head pointer 802 and current pointer 803. Head pointer 802 points to the first entry 810-1 in the transmit list 804, which links to the next entry 810-2. Current pointer 803 points to the entry from which datagrams are to be transmitted next, which is virtual path 810-2 in this example. The transmit list 804 is formed by the link fields 811-1 through 811-n. Each link field points to the next path in the transmit list. The last link entry 811-n in the transmit list 804 has a link field value of 0. The actual length of the transmit list 804 will vary as a function of the number of paths that have datagrams pending for transmission on output port 801. If no path is waiting to transmit on output port 801 then the value of both head pointer 802 and current pointer 803 is 0.

The next datagram,to be sent on output port 801 is determined by current pointer 803 which points to the next entry in the transmit list 804 of linked virtual path entries. This method of organizing the output list 804 as a chain of all virtual paths waiting to transmit on output port 801 has the effect of giving round robin priority to datagram packets waiting to be transmitted from different virtual paths.

A mechanism is also provided to send datagram packets from selected virtual paths at a higher priority than other virtual paths. If the priority field 322 in the virtual path record 300 is set, then the value in the priority field 322 indicates the

number of packets to be transmitted from a virtual path before transmitting a packet from the next path in the transmit list 804.

Overall Switch and Network Operation

Referring to Figures 1 through 8, the overall operation of the system shall now be described.

For purposes of illustration, assume that client station 120 (Figure 1) wants to send a datagram packet 200 (Figure 2) to server station 126. Datagram 200 will be received on switch input port 401 (Figure 4). Switch hardware generates virtual path index 630 from the datagram destination and source address fields and sends virtual path index signals on bus 421 to virtual path cache 415 for lookup.

The virtual path index in cache 415 will be converted by hash logic 631 to a virtual path cache index 632 (Figures 6 and 7) that indexes the four set associative virtual path cache 601 through 604 (Figure 6). The virtual path index is further compared in parallel against the outputs from the four set associate cache 601 through 604 via the four comparators 611 through 614. Assuming a valid virtual path tag was found in SRAM cache 601 then comparator 611 will indicate a "hit" signal on hit/miss wire 635 and enable tri-state buffer 621 to output the virtual path record stored in SRAM cache 601 on virtual path record bus 633.

Switch hardware 409 then forwards and processes the datagram packet according to the fields of the virtual path record on bus 633 (Figures 6 and 4).

If the virtual path cache had not contained a valid virtual path cache entry then it would indicate miss on the hit/miss wire 635 (Figures 6 and 4). This causes the switch hardware to forward the packet to the CPU 411 via CPU interface 412 for processing. The controller CPU 411 then processes the packet in software and sends it back to the switch hardware 409 to output on the appropriate output port. if controller CPU 411 determines that future datagram packets of this virtual path should be processed by switch hardware, controller CPU 411 then creates a new virtual path record for said virtual path and writes it into virtual path cache 415 replacing the least recently used entry if necessary.

Several advantages flow from this invention. For example, the invented method and structure:

1. supports reliable and efficient data communication in datagram networks without dropping datagram packets due to lack of network resources;

2. allows the specification of the attributes of datagram packet flows similar to the capabilities available in a virtual circuit packet switched network, but without the disadvantages of having to set up, maintain, and tear down virtual circuits; and

3. is compatible with existing network protocols and applications, and interoperates with the installed base of datagram network interfaces.

The other embodiments of this invention may be obvious to those skilled in the art. The above description is illustrative only and not limiting.

CLAIMS

We claim:

1. In a network device, a method for processing datagram packets where the source destination address pair contained in the datagram packet determines the processing action to be taken.

2. The method of Claim 1 including the steps of comparing said source destination address pair contained in the datagram packet to a listing of source destination pairs, each source destination pair in the listing being associated with a virtual path; and

should a match be found between the source destination address pair contained in the datagram packet and a source destination address pair contained in the listing, selecting the virtual path associated with the source destination address pair for transmitting the datagram packet.

3. The method of Claim 2 wherein the virtual path for transmission of the datagram packet is determined based upon the actual network traffic loading and the frequency of transmission of datagram packets utilizing the same source-destination address pair.

4. In a network device, a method for processing datagram packets where the source destination address pair contained in the datagram packet determines the processing actions to be taken based on specifications and state stored in the network device.

5. In a network device that processes datagram packets,'a method for selecting the processing steps to be applied to a datagram packet, said method comprising the steps of:

determining from the source-destination address pair contained in the datagram packet a virtual path record comprising specifications for processing the datagram packet; and

processing the datagram packet according to the processing steps associated with said virtual path record.

6. In a network device, a method for processing datagram packets having a source and a destination address comprising the steps of:

receiving a datagram packet;

determining from the source-destination address pair contained in.the datagram packet a virtual path record comprising specifications for processing the datagram packet; and

processing the datagram packet according to the processing steps associated with said virtual path record.

7. The method of processing datagram packets using a network device to receive and transmit said datagram packets, which comprises:

organizing the network device such it can process datagram packets as flows;

initializing the network device state such that the network device is ready to actually process datagram packets as flows; and

processing a specific datagram packet as an element of a flow based on the source-destination address pair contained in the datagram packet.

8. The method of Claim 7 including:

modifying the network device state based on the datagram packets received and other processing actions chosen.

9. The method of Claim 1 wherein the determination of the processing action to be taken is determined by the source-destination address pair and the type field contained in the datagram packet.

10. The method of Claim 1 wherein the determination of the processing action to be taken is determined by the source-destination address pair and contents of the data field contained in the datagram packet.

11. The method of Claim 1 wherein the datagram packet is one of the datagrams defined in the IEEE 802 family of network standards.

12. The method of Claim 1 wherein the datagram packet is of the type commonly known as an Ethernet datagram packet and wherein the source address is the Ethernet packet source address field and the destination address is the Ethernet packet destination address field.

13. The method of Claim 1 wherein the step of processing the datagram packets comprises an action chosen from the group consisting of: dropping the packet buffering the packet, forwarding the packet to an output port, forwarding the packet to a multiplicity of output ports, forwarding a copy of the packet to another network device, and generating a response back to the sender of the packet.

14. The method of Claim 1 wherein processing includes the step of forwarding the datagram packet to an output port.

15. The method of Claim 1 wherein processing includes the step of forwarding the datagram packet to a multiplicity of output ports.

16. The method of Claim 1 wherein processing includes the step of sending the datagram packet to a controller CPU for further processing.

17. The method of Claim 1 wherein processing includes the step of forwarding a copy of the datagram packet to another network device for processing.

18. The method of,Claim 1 wherein processing includes the step of discarding the datagram packet.

19. The method of Claim 1 wherein processing includes the step of notifying the sender of the datagram packet that the datagram packet that was sent has not been accepted.

20. The method of Claim 1 wherein processing includes the step of notifying the sender of the datagram packet that the datagram packet that was sent has not been accepted, wherein said datagram is an Ethernet datagram, and wherein the response back to the source comprises an Ethernet collision signal.

21. The method of Claim 1 wherein the step of forwarding a datagram packet includes determining whether the datagram is sent from an authorized source to an authorized destination.

22. The method of Claim 1 wherein processing includes the step of storing the datagram packet in the packet buffer memory means, with separate buffer resources provided for each separate virtual path.

23. The method of Claim 1 wherein processing involves determining the availability of buffer space on a particular virtual path and not accepting additional datagram packets that would exceed the amount of buffer space allowed for this virtual path.

24. The method of Claim 1 wherein processing includes communicating the availability of packet buffer resources available to the upstream sender of datagram packets for this virtual path.

25. The method of Claim 1 further comprising allowing the network device to specify the maximum rate of transmission of a flow of datagrams.

26. The method of Claim 1 further comprising a method providing for specifying the rate of transmission of one flow of datagrams relative to the rate of transmission of other flows of datagrams.

27. A network device, comprising:
at least one port for receiving and transmitting datagram packets;
memory means for buffering said datagram packets; and
processing means, said device being capable of processing each of said datagram packets according to instructions stored in the network device that are specific to the source-destination address pair of each datagram packet.

28. A network device as in Claim 27 wherein said at least one port comprises a plurality of ports.

29. A network device as in Claim 27 wherein said processing defines an action to be taken with respect to each datagram packet, said action being chosen from the group consisting of: dropping the packet, buffering the packet, forwarding the packet to an output port, forwarding the packet to a multiplicity of output ports, forwarding a copy of the packet to another network device, and generating a response back to the sender of the packet.

30. The network device of Claim 27 wherein each datagram packet is one of the datagrams defined in the IEEE 802 family of network standards.

31. The network device of Claim 27 wherein each datagram packet is of the type commonly known as an Ethernet datagram packet and wherein the source address is the Ethernet packet source address field and the destination address is the Ethernet packet destination address field.

32. The network device of Claim 27 wherein the step of processing each datagram packet comprises a routing action chosen from the group consisting of: dropping the packet, buffering the packet, forwarding the packet to an output port, forwarding the packet to a multiplicity of output ports, forwarding a copy of the packet to another network device, and generating a response back to the sender of the packet.

33. The network device of Claim 27, further comprising means for storing information in the virtual path record, said information specifying the amount of buffer resources provided for each separate flow of datagram packets.

34. The network device of Claim 27, further comprising flow control means, said flow control means providing the capability to communicate to the upstream sender of datagram packets the availability of packet buffer resources available.

35. The network device of Claim 27, further comprising means for allowing,the network system to control the relative rate of transmissions among multiple flows of datagrams.

36. The network device of Claim 27, further comprising:

a virtual path cache memory for providing storage for the most recently used virtual path records;

virtual.path cache lookup means that allows switch hardware to determine if a certain virtual path entry is located in the virtual path cache;

means for performing processing actions specified in the virtual path record if a valid virtual path record is found in the virtual path cache memory; and

means for processing datagram packets for which no valid entry is found in the virtual path cache memory.

37. The network device of Claim 27, further comprising a controller CPU and controller memory for processing datagram packets that have no valid virtual path record in the virtual path cache, said controller CPU also being able to read and write virtual path cache entries.

38. In a computer network system comprised of a multiplicity of network devices, a method for controlling network operation that specifies for each network device the processing of datagram flows, where each datagram flow is identified by a unique source-address destination pair which is contained in the datagram packets themselves.

39. The method of Claim 38 wherein specifying the processing includes specifying which flows of datagrams receive access to the network.

40. The method of Claim 38 wherein specifying the processing includes the step of specifying how to forward flows of datagrams packets.

41. The method of Claim 38 wherein specifying the processing includes the step of specifying how to buffer flows of datagram packets and the buffer resources to be used.

42. The method of Claim 38 wherein specifying the processing includes the step of specifying how to schedule the rate of transmission for an individual flow of datagram packets.

43. The method of Claim 38 wherein specifying the processing includes the step of specifying how to schedule the rate of transmission among a multiplicity of flows of datagram packets.

44. The method of Claim 38 wherein specifying the processing includes monitoring the rate of flows of datagram packets and taking one of several actions as a function of the flow rate.

45. In a datagram packet switched computer network, comprised of a plurality of connected network devices and host computers, a method for improved datagram packet switching, comprising the steps of:

processing datagram packets as separate flows, each flow being identified by a unique source-destination address pair contained in the datagram itself; and

maintaining in the network devices information that specifies the processing of each flow of datagram packets, said information being initially specified during network configuration.

46. The method of Claim 45 wherein processing includes granting access to a flow of datagram packets based on availability of buffer resources for such a flow.

47. The method of Claim 45 wherein processing includes specifying the forwarding function of the flow of datagram packets at each network device, which is based on the connectivity of the overall network.

48. The method of Claim 45 wherein processing includes specifying the scheduling of each flow of datagram packets at each network device, relative to resource allocation and traffic congestion at each network node.

49. In a network device that processes datagram packets, a method for forwarding datagram packets, said method comprising the steps of:

determining from the source-destination address pair contained in each datagram packet a virtual path record comprising specifications for forwarding the datagram packet; and

forwarding each datagram packet according to the specifications in said virtual path record.

50. In a network device that processes datagram packets, a method for buffering datagram packets, said method comprising the steps of:

determining from the source-destination address pair contained in each datagram packet a virtual path record comprising specifications for buffering the datagram packet; and

buffering the datagram packet according to the specification in said virtual path record.

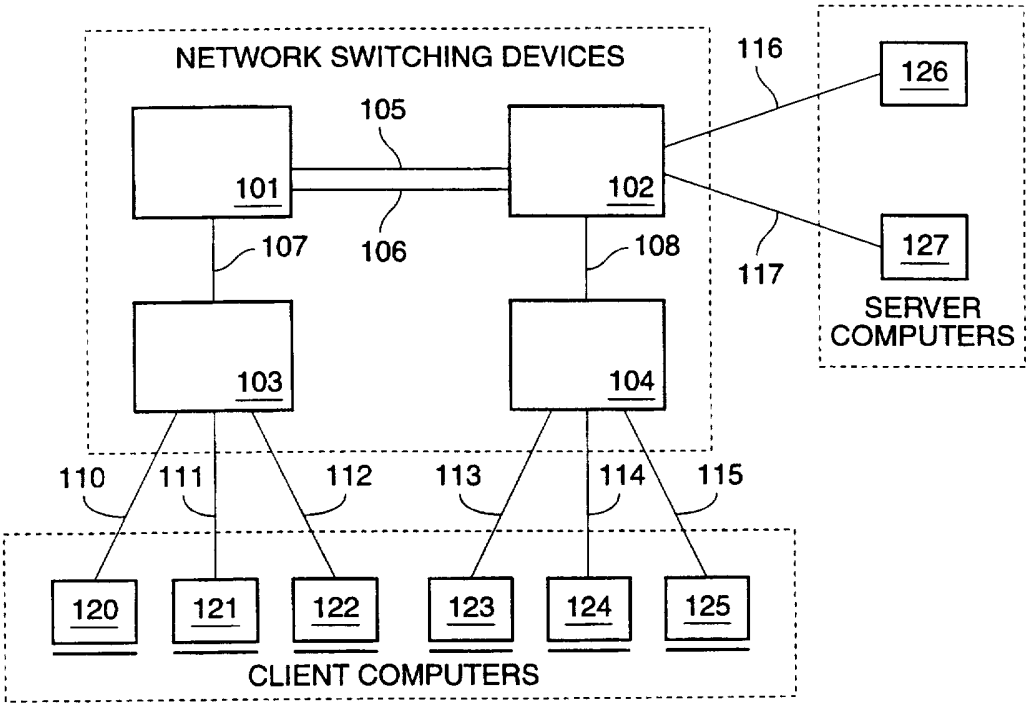

NETWORK SWITCHING DEVICES
101
102
103
104
105
106
107
108
116
117
126
127
SERVER COMPUTERS
110
111
112
113
114
115
120
121
122
123
124
125
CLIENT COMPUTERS

FIG. 1

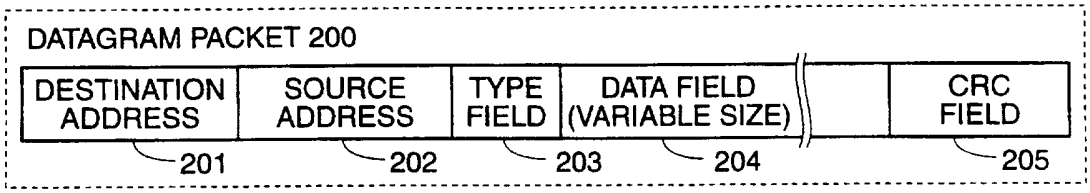

DATAGRAM PACKET 200
DESTINATION ADDRESS
SOURCE ADDRESS
TYPE FIELD
DATA FIELD (VARIABLE SIZE)
CRC FIELD
201
202
203
204
205

FIG. 2

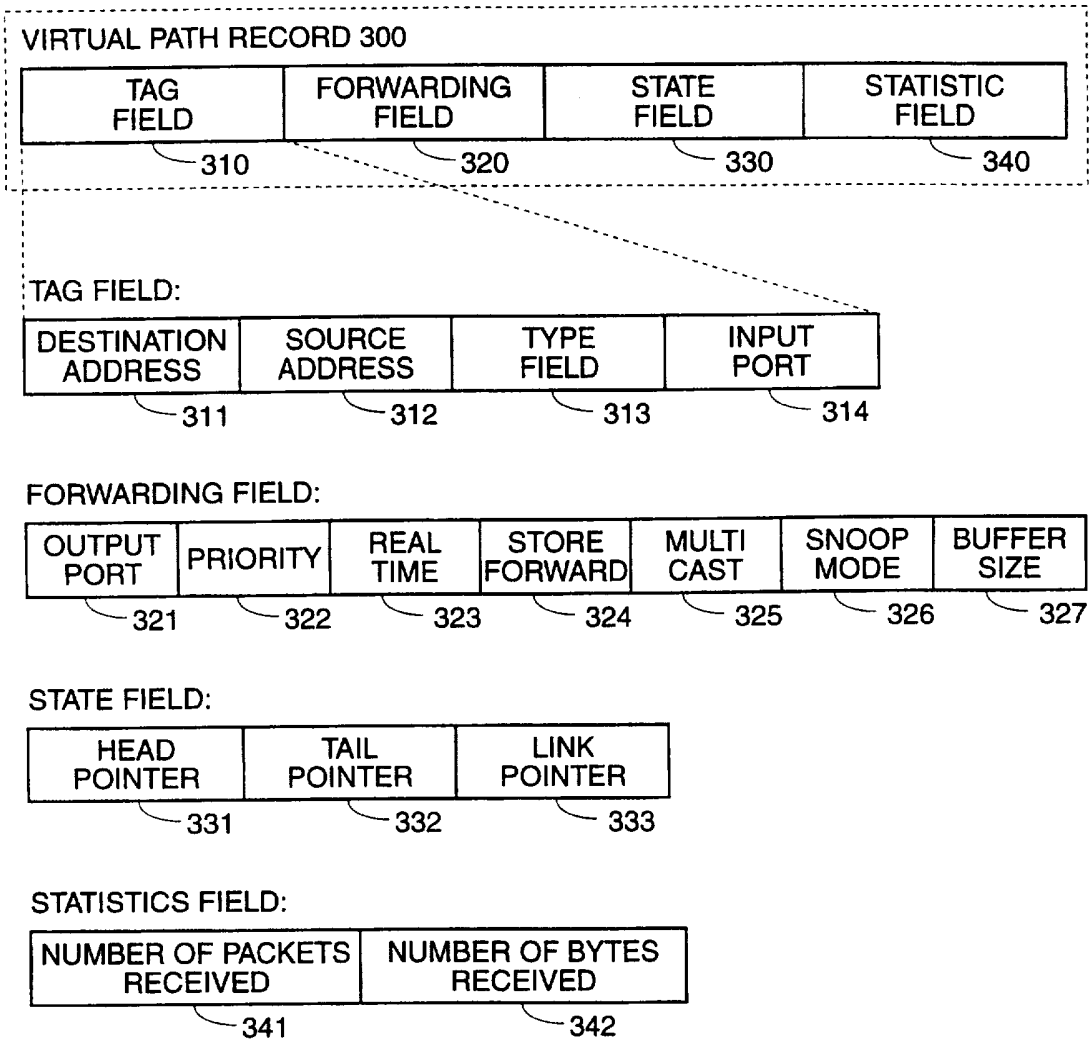

VIRTUAL PATH RECORD 300
TAG FIELD
FORWARDING FIELD
STATE FIELD
STATISTIC FIELD
310
320
330
340
TAG FIELD:
DESTINATION ADDRESS
SOURCE ADDRESS
TYPE FIELD
INPUT PORT
311
312
313
314
FORWARDING FIELD:
OUTPUT PORT
PRIORITY
REAL TIME
STORE FORWARD
MULTI CAST
SNOOP MODE
BUFFER SIZE
321
322
323
324
325
326
327
STATE FIELD:
HEAD POINTER
TAIL POINTER
LINK POINTER
331
332
333
STATISTICS FIELD:
NUMBER OF PACKETS RECEIVED
NUMBER OF BYTES RECEIVED
341
342

FIG. 3

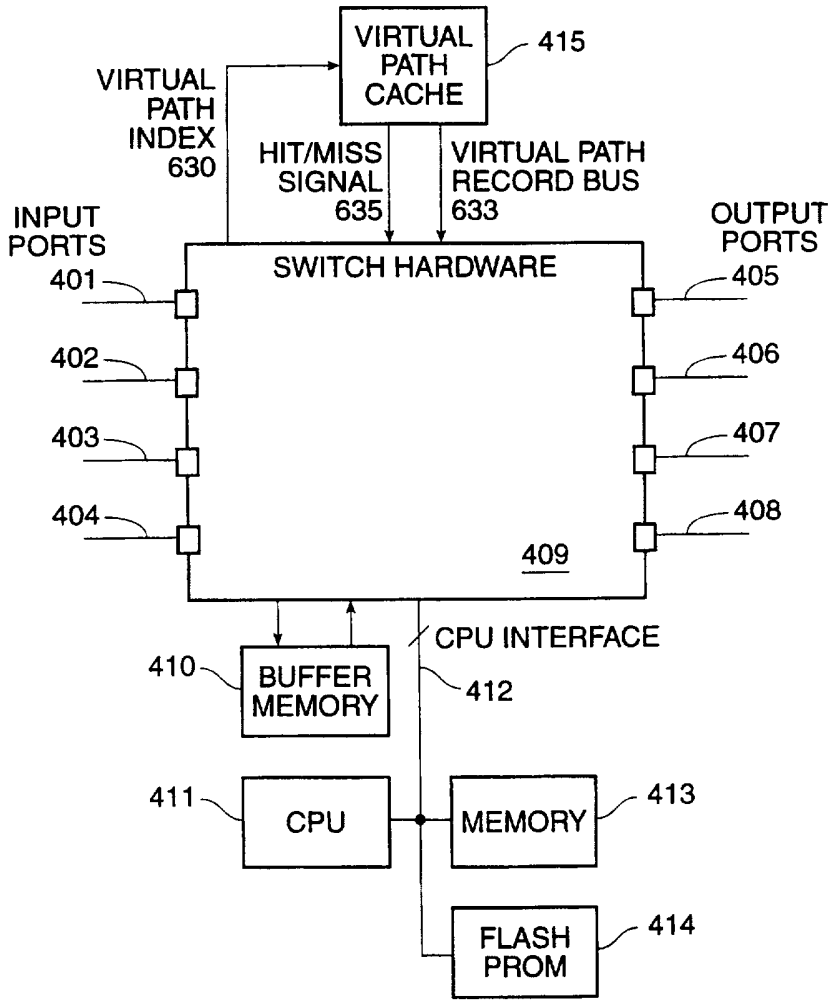

VIRTUAL PATH CACHE
415
VIRTUAL PATH INDEX 630
HIT/MISS SIGNAL 635
VIRTUAL PATH RECORD BUS 633
INPUT PORTS
OUTPUT PORTS
SWITCH HARDWARE
401
402
403
404
405
406
407
408
409
CPU INTERFACE
410
BUFFER MEMORY
412
411
CPU
MEMORY
413
FLASH PROM
414

FIG. 4

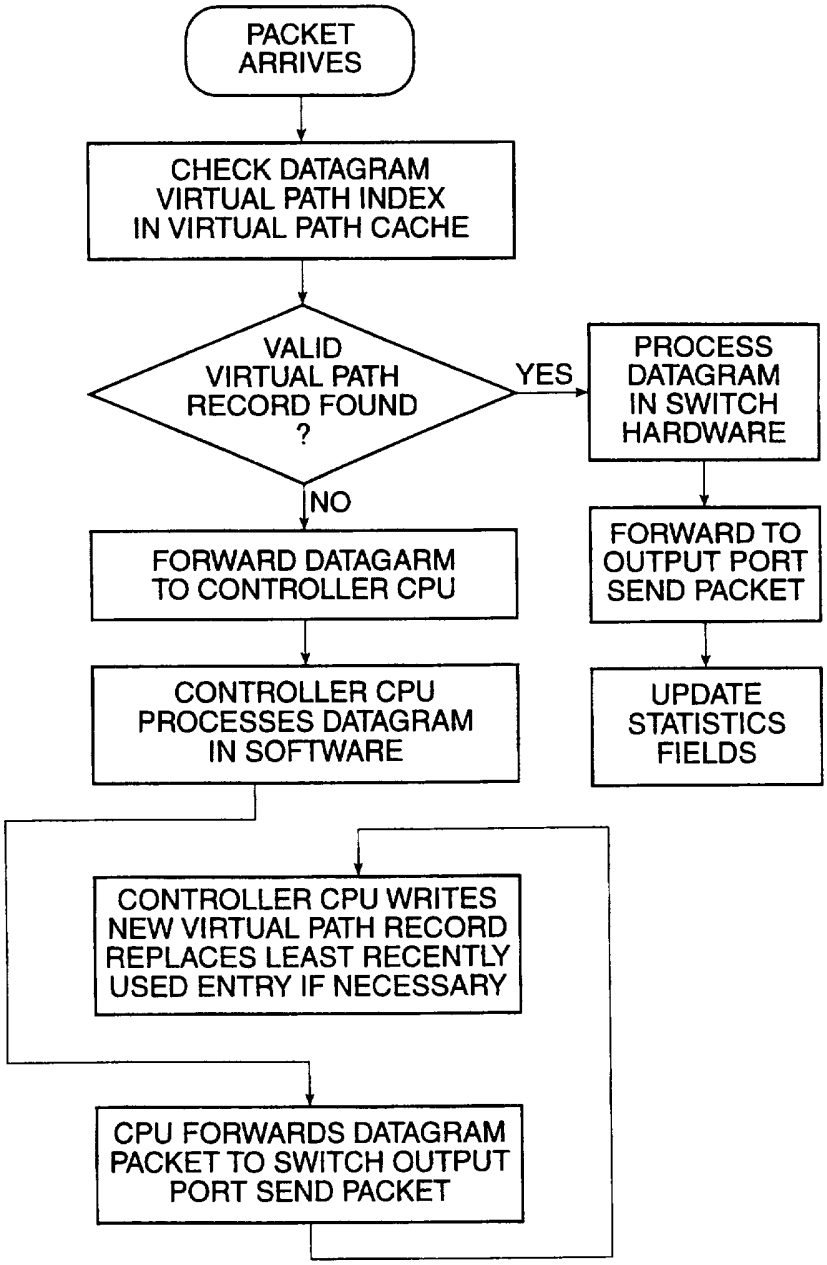

PACKET ARRIVES
CHECK DATAGRAM VIRTUAL PATH INDEX IN VIRTUAL PATH CACHE
VALID VIRTUAL PATH RECORD FOUND ?
YES
NO
PROCESS DATAGRAM IN SWITCH HARDWARE
FORWARD TO OUTPUT PORT SEND PACKET
UPDATE STATISTICS FIELDS
FORWARD DATAGARM TO CONTROLLER CPU
CONTROLLER CPU PROCESSES DATAGRAM IN SOFTWARE
CONTROLLER CPU WRITES NEW VIRTUAL PATH RECORD REPLACES LEAST RECENTLY USED ENTRY IF NECESSARY
CPU FORWARDS DATAGRAM PACKET TO SWITCH OUTPUT PORT SEND PACKET

FIG. 5

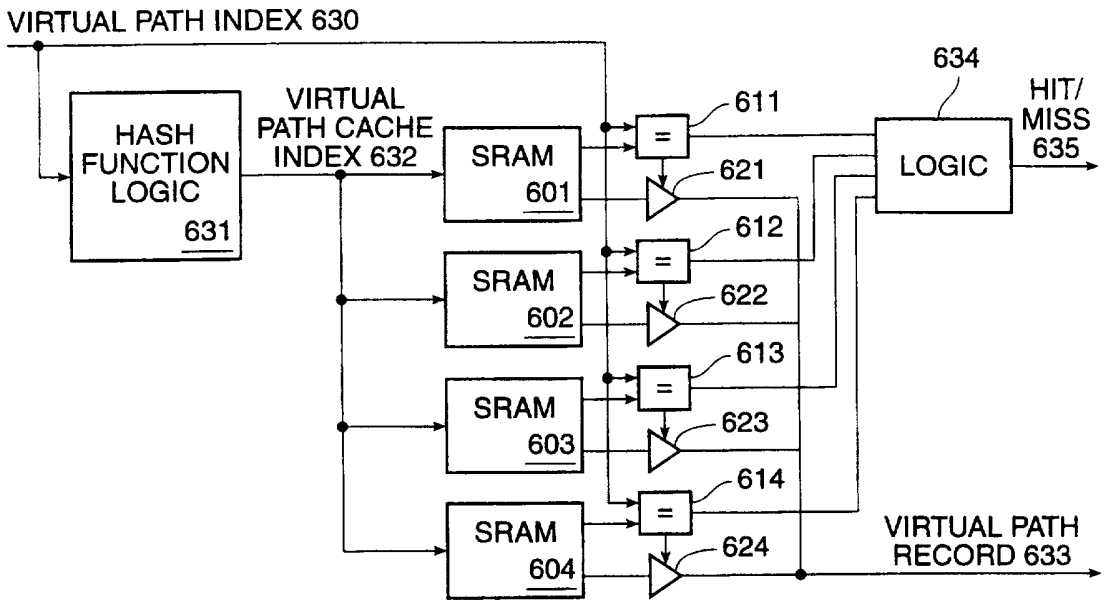


FIG. 6

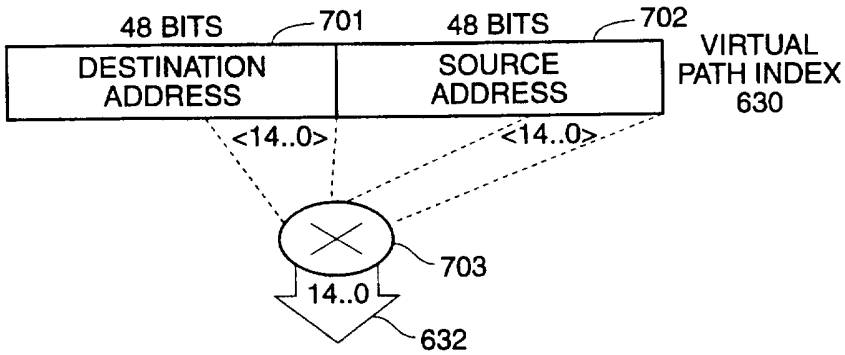


FIG. 7

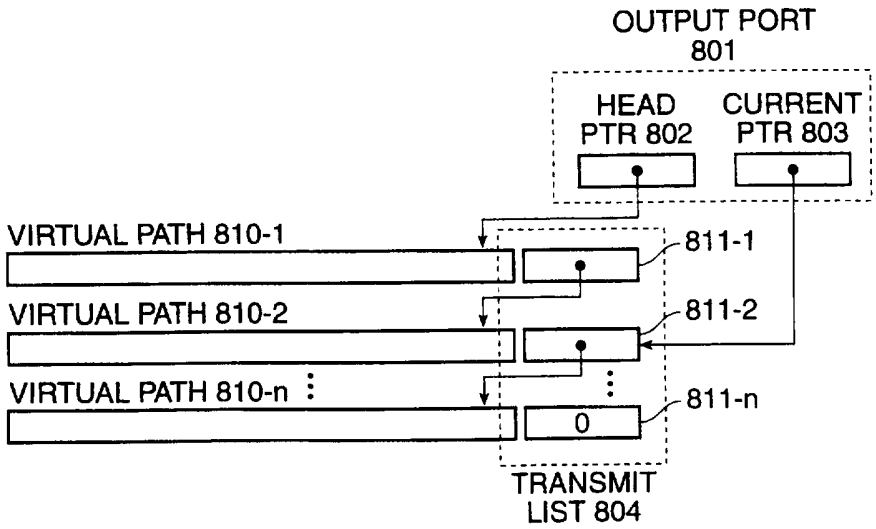


FIG. 8