96/0332-0
IPOS

Cloonan-Richards 20-32

TERABIT PER SECOND ATM PACKET SWITCH HAVING OUT-OF-BAND CONTROL WITH MULTICASTING

Technical Field

The invention relates to large telecommunication switches that use data packets in order to communicate at aggregate throughputs at the one terabit per second level, and more particularly to such a method and apparatus for using a large telecommunication switch to provide multicasting.

Description of the Prior Art

Telecommunications have long used digital switching to encode, multiplex, transmit and decode audio frequencies in order to carry the millions of telephone voice calls of the world. Telecommunication switches for voice calls have grown to very large sizes to keep pace with the demand. Most of the switching systems that route and control voice call traffic are called circuit switches, which means that for each call a type of bi-directional audio circuit is set up between the calling party and the called party. The circuit that is set up has the bandwidth and transport timing necessary to simulate a face-to-face conversation without objectionable distortion or time delays.

An alternative to circuit switching is called packet switching. For packet switching, the calling party is responsible for converting the information into one or more packets. This information could be encoded voice, it could be encoded computer data, or it could be encoded video. The number of the called party is typically included in a packet header to guide the packet to its destination. The packet switching network has the task of routing each packet to its respective destination without undue delay. The called party usually has the equipment to receive the packets and decode the information back into an appropriate form.

The extremely rapid growth of packet switching traffic carrying voice, computer (LAN/WAN) , facsimile, image and video data to an ever widening variety of locations, along with the proposals for a National Information Infrastructure, has challenged both the packet switch protocols and system architectures.

Many vendors and service providers have joined forces to define a global standard that would permit packet switching services to be provided in a ubiquitous

This Sheet Has Been Replaced

REPLACED SHEETS SG
Description : paged 1 to paged 33
Claim(s) : paged 34 to paged 40
Drawing(s) : paged ..... to paged .....
Abstract : paged 41 to paged 1
DATE OF RECEIPT .....

fashion. The result of this coordinated effort has been the rapid development and deployment of an Asynchronous Transfer Mode (ATM) as a means of efficiently routing and transporting data packets that have stochastically-distributed arrival rates according to the recent ATM standard. ATM is thus a packet-oriented standard, but unlike most of its data packet predecessors (X.25, frame relay, etc.), ATM uses short, fixed-length, 53-byte packets that are called cells. ATM also uses a very streamlined form of error recovery and flow control relative to its predecessors. In fact, the ATM standard essentially eliminates most error protection and flow control at the link level, leaving these functions to higher level protocols at the edges of the network. This approach permits rapid routing of the short cells with minimal network delay and jitter, making ATM compatible with voice, data and video services. ATM has been embraced by the computer, LAN, and WAN industries, so a seam-less packet communication from the source computer through LANs, WANs, and the public-switched network is a reality.

If this level of connectivity becomes available to the average consumer and if advanced broad band services that combine voice, broad band data and video are similarly available at a reasonable price, then the volume of ATM traffic that may be generated in the future is virtually limitless. As a result, the number and size of the switches and cross-connects required to route this ATM packet traffic may also grow by phenomenal rates within the next decade. ATM switches and cross-connects for toll and gateway applications may require aggregate bandwidths ranging from 155 gigabits per second (1000 inputs at SONET OC-3 155 megabits per second rates) to 2.4 terabits per second (1000 inputs at SONET OC-48 2.4 gigabits per second rates). Additionally, if demand for broad band services to the home and/or LAN/WAN connectivity through the public-switched network grows as some experts believe, then local telephone exchange carriers may require ATM switches and cross-connects for metropolitan area network (MAN) applications having aggregate bandwidths ranging from 100 gigabits per second (50,000 inputs at Ethernet 10 Megabits per second rates and 20 percent occupancy) to 775 gigabits per second (50,000 inputs at SONET OC-3 155 Megabits per second rates and 10 percent occupancy).

This Sheet Has Been Replaced

By necessity, most of the current architectural research and hardware/software development for ATM switches has concentrated on switches with much smaller aggregate bandwidths that will meet the more near-term needs of the marketplace. For example, most of the proposals within the LAN/WAN community have supported aggregate bandwidths ranging from 150 Megabits per second to 12 Gigabits per second, and most of the published proposals within the telecommunications community have supported aggregate bandwidths ranging from 20 Gigabits per second to 160 Gigabits per second. Extensions of most of these architectures to larger sizes usually produce systems that are cost prohibitive, size prohibitive, and/or physically unrealizable because of limits of the underlying to technology.

For example, very common designs for large, high-throughput switches use a multi-stage interconnection network containing multiple stages of switching nodes (node-stages) interconnected by stages of links (link-stages) to provide multiple paths between input ports and output ports. Clos, Banyan and Benes networks are examples of such networks. A multiple stage network design can yield networks with very high levels of performance (low blocking probabilities, low delay, high degrees of fault tolerance, etc.), and may result in low system-level costs, because network resources (nodes and links) are time-shared by the many different paths that can be set up within the network. Physically realizing a multistage network for ATM because of the very short packet duration time is, however, a problem.

The design of any large, high-throughput ATM switching architecture must address two fundamental issues that profoundly effect the overall performance of the resulting ATM switch. The first of these issues is cell loss due to blocking within the internal links of the distribution network (also known as the switching fabric), and the second is cell loss due to contention for output ports by two or more ATM cells that pass through the switch at the same moment in time. The first issue can usually be solved by designing a network with sufficient switching fabric (nodes and links) so that multiple paths exist between input ports and output ports. As a result, if two or more ATM cells attempt to use the same shared resource (nodes or links) within the switching fabric, the cells can usually find two disjoint paths that eliminate the

This Sheet Has Been Replaced

internal network blocking problem. The second issue requires the switch designer to identify a technique for handling simultaneous cells.

A general design technique for a switch to handle cells destined for the same output port is analyzed in an article, A Growable Packet Switch Architecture, IEEE Transactions on Communications, February, 1992, by Eng et al. and in another article The Knockout Switch, ISS AT&T Technical Papers, 1987, by Yeh et al. This general design technique segments the switch into two distinct parts, as shown in Fig. 1. A Nx(FN) distribution network (which provides for N input ports) and a bank of K mxn output packet modules (which provide for a total of M=Kn output ports). Given that each of the links emanating from the distribution network is required to be terminated at one of the inputs to an output packet module, it can be seen that the equation FN=Km must be satisfied. The switching fabric is a memory-less Nx(FN) fanout switch whose function is to route an arriving ATM cell to any of the m inputs on the output packet module connected to the cell's desired output port. The output packet module is a mxn switch with buffers that are available for storing cells that must be delayed when two or more cells contend for a particular output port. If the arriving traffic is uniformly distributed across all output ports and if the buffers within the output packet modules are sufficiently large, then the ratio m:n can always be chosen large enough to force the cell loss probability within the network to be below any desired cell loss probability level. In fact, if the network size (N) is large and if R represents the switch loading, then the cell loss probability of a network with mxn output packet modules as shown by Eng et al is given by:

$$P(\text{cell loss}) = [1-m/(nR)][1-\sum_{k=0}^{m}\{(nR)^k e^{(-nR)}\}/k!]+(nR)^m e^{(-nR)}/m!$$

Present packet switches have acceptable cell loss probabilities of approximately 10^{-12}, so any loss probability smaller than present units are considered acceptable.

Besides the ATM cell losses because of internal contentions, in an ATM Packet Switch where all of the N cells arrive simultaneously at the inputs of the distribution network, the cells must be processed by each stage of the path hunt processing pipeline before the next group of N cells arrives at the network input ports. For example, if the incoming transmission lines support SONET OC-48 2.5 gigabits

This Sheet Has Been Replaced

per second bit-rates, then the group of N ATM cells that arrive together must be processed and sent on to the next stage of the pipeline every 176 nsec (the duration of an ATM cell on a 2.5 gigabits per second link). For large values of N, a substantial amount of processing power will therefore be required to complete the path hunt operations for all N cells. (Note: If N=256, then 1.45×10^9 path hunts must be completed every second, which corresponds to an average processing rate of one path hunt every 684 psec). Present commercial microprocessors can process approximately 100 million instructions per second. If each path hunt took only one instruction, at least 15 of microprocessors would spend 100% of their processing time performing these path hunts. Thus, a controller based on something other than a single microprocessor will be necessary for a large ATM packet switch.

Two approaches to solving the path hunt problem can be envisioned. One approach uses in-band, i.e., self-routing, control techniques to perform the path hunts. The Clos, Banyan and Benes networks multistage networks mentioned above typically use in-band controllers and control techniques. For in-band control techniques, the connection requests are pre-pended to the ATM cells and routed through the switch along the same paths used by the following ATM payload. The in-band control approach typically requires parallel processing elements to be distributed throughout all of the nodes in the network, resulting in relatively complicated hardware to be duplicated in every node of the network in order to perform localized path hunting operations (on only the cells that pass through that node) when determining how to route the arriving connection requests and ATM cells. The second approach uses an out-of-band controller and out-of-band control techniques whereby the controller and switch fabric are logically separated, so connection requests must be routed to the controller, which performs the path hunting, before the resulting control signals are inputted to the switch fabric to set the paths. This second approach requires that the out-of-band controller have tremendous processing power, (as mentioned above), because of the many path hunting operations that must be performed in a very short period of time.

Since the path hunting operations in switches using in-band control techniques are only based on localized traffic information and not on global information

This Sheet Has Been Replaced

regarding all of the switch traffic, the routing that in-band controllers perform may not always optimal with regard to internal blocking. As a result, networks based on in-band control techniques often require more switch fabric (stages and nodes) in order to provide the same operating characteristics as a less expensive switch based on out-of-band control techniques. Thus, there is a need in the art for a packet switch controller that provides a global view of all paths to provide optimal path hunting for each packet.

For individual users, the path hunted and used is from one input to one desired output. For these one-to-one users, it is important to find non-blocking paths for their packets quickly and efficiently. Beyond the one-to-one uses, there are a number of uses for one input to multiple outputs capabilities, such operation is known as multicasting.

Multicasting is the ability of a switch to route a single input ATM cell to multiple output ports. Multicasting is an important feature for applications that require a single source to transmit to many destinations. Common applications of this feature can be found in the distribution of video traffic from a video server in a CATV head-end office to all of the homes that are tuned to a particular television show. The multicasting feature may also be useful for the distribution of an e-mail message from a user on a single computer to users on many spatially distributed computers in a telecommuting environment. Although the protocols and signaling formats required for multicast routing are not clearly defined by an ATM standard, one very likely approach is for the protocol to require the user to set up a connection to all of the output ports using a special VPI/VCI address that points to a linked list containing all of the desired output ports. In many ATM switches based on self-routing control techniques, the switch design of choice is for the input line cards to make multiple copies of the arriving ATM cell to be multicast before injecting the multiple copies of the cell into the network. This known design requires relatively complicated additions to the hardware in the input line cards, because a single input ATM cell might request transmission to all of the N output ports, so each input line card must be capable of creating and injecting N copies of the ATM cell into the network. This puts a tremendous design burden one the line cards. Thus, it is desirable to have an ATM

This Sheet Has Been Replaced

switch that provides multicasting without burdening the design of the line cards. It is also desirable to have a distributed, out-of-band controller that controls an ATM packet switch to provide multicasting of a single data cell to multiple outputs.

Summary of the Invention

Briefly stated, in accordance with one aspect of the invention, the foregoing problems are overcome by providing a distributed, out-of-band controller for an ATM packet switch having line interfaces with one ATM cell period storage, a single stage, reduced complexity switch fabric, an out-of-band controller that provides for transmission to multiple output packet modules to which destination output lines are attached, and output packet modules which provides a copy of the desired ATM cell to each destination output line attached thereto.

In accordance with another aspect of the invention, the foregoing problems are overcome by providing a controller for a packet switch for locating multiple paths for each multicast telecommunication packet, this controller includes a memory for storing a busy-or-idle status of each connection of the packet switch between its switch fabric and its output modules. This memory is divided into a number of busy-or-idle status tables, and these busy-or-idle tables are accessed concurrently for rapidly locating each required path for the multicast packet.

In accordance with another aspect of the invention, the foregoing problems are overcome by providing a controller for a packet switch which has a switch fabric, which in turn has a number of pipes with each pipe having a number of crossbar switches. This controller is for locating a plurality of paths for a multicast telecommunication packet received on one of a plurality of input lines of the packet switch. The controller includes a number of pipe hunting controllers, with each of these pipe hunting controllers controlling a respective pipe out of the number of pipes. Each of these pipe hunting controllers has a plurality of crossbar switch controllers, with each crossbar switch of a pipe having a respective crossbar switch controller associated with it. The controller also includes a store for a storing busy-or-idle status table. Each bit of the busy-or-idle status table in the store is associated with a respective connection of a respective crossbar switch and stores a respective busy-or-

This Sheet Has Been Replaced

idle bit to control the operation of that connection. Furthermore, the controller can access each bit of the busy-or-idle status table to request at least one path to at least one output packet module for one or more output lines attached thereto for a multicast packet concurrently. Thus, allowing very rapid and very efficient multicasting packet switching operations.

Brief Description of the Drawing

FIG. 1 is a block diagram of a generalized growable packet switch;

FIG. 2 is a slightly re-drawn Fig. 1.

FIG. 3 is a block diagram of a growable packet switch in which the switch fabric is partitioned into L multiple pipes according to the present invention.

FIG. 4 is a block diagram, similar to Fig. 3, of a specific embodiment of the present invention having four pipes (L=4) and showing a configuration for the pipes.

FIG. 5 is a simplified block diagram of the embodiment shown in FIG. 4 which shows greater details of the controller.

FIG 6. illustrates the timing sequences of requests to the controller shown in FIG. 5.

FIG. 7 is a simplified block diagram of an embodiment of an output module.

FIG. 8 is an illustrative example of rolling and its operation in a plan view of an amusement park and its satellite parking lots.

FIG. 9 shows plots of calculated values of various ATM cell loss probabilities both with and without the assignment of preferences.

FIG. 10 is a simplified block diagram of a representative switch controller and its link controllers.

FIG. 11 is a detailed logic diagram of a link controller.

FIG. 12 is a state table for the link controller shown in FIG. 11.

FIGs. 13A-13D when joined together show the operation of a switch controller in response to a sequence of requests.

FIG. 14 illustrates the rolling of path hunting requests through a switch having four pipe controllers according to the present invention.

FIG. 15 illustrates the multiple paths taken to the destination output lines for a multicast cell.

This Sheet Has Been Replaced

Detailed Description

Referring now to FIG. 2, a large, generalized switch 10 for ATM communications, is shown in block diagram form. ATM switch 10 has a number of input interfaces 12_0 - 12_{N-1}, a switch fabric 14, and buffered output modules 16_0 - 16_{V-1}. For ATM operation, input interfaces 12_0 - 12_{N-1} , are high speed digital amplifiers that serve as a matching networks and power amplifiers for fanning out information received on their inputs to multiple input ports of the switch fabric 14. Each of the input interfaces 12_0 - 12_{N-1} also needs a capability to store one ATM cell, as will be explained below. Similarly for ATM operation, buffered output modules 16_0-16_{V-1} are concentrators that are buffered to reduce packet loss when two or more packets are directed to and contend for the same output of outputs Out_0-Out_{N-1}.

Switch fabric 14 includes a fanout F where each of the outputs from the input interfaces. 12_0-12_{N-1} is fanned out to F inputs within switch fabric 14, such that if ATM switch 10 is an NxN switch then switch fabric 14 will have FN internal inputs and FN outputs to output modules 16_0-16_{V-1}. Output Modules 16_0-16_{V-1} have a fanin or concentration factor of F in order to convert the FN outputs of the switch fabric 14 to N output module outputs Out_0-Out_{N-1}. Each output module 16_0-16_{V-1} stores arriving ATM packets in FIFO queues, and then routes the ATM packets at the front of each of these FIFO queues to their desired outputs Out_0-Out_{N-1} when the output ports are available.

Switch fabric 14 is a general distribution network which may be a network of switches, specifically crossbar switches, to provide multiple paths from each of its input ports 17_0-17_{N-1} to each of its output ports 19_0-19_{FN-1}. However, it becomes highly impractical to make an NxN switch out of a single crossbar to operate as the switching component of switch fabric 14 when the size of N exceeds 32. Thus, some other way is needed to realize the general architecture shown in FIG. 2.

Referring now to FIG. 3, an ATM switch 10A that is both practical and possible for N inputs where the size of N is at least 256, is shown. Multiple paths from each input 17_0-17_{N-1} through the switch fabric 14A are provided to prevent blocking. These multiple paths are partitioned into groups called pipes with each pipe providing exactly one path between each input port 17_0-17_{N-1} and each output port

IThis Sheet Has Been Replaced

19_0-19_{FN-1} within the network. Thus, switch fabric 14A is made up of multiple pipes 18_0-18_{L-1}. The output modules 16_0-16_{V-1} are essentially the same as the output modules shown in FIG. 2.

Switch fabric as seen in copending and commonly assigned application entitled **TERABIT PER SECOND PACKET SWITCH,** EPC No. 95309013.1, filed December 12, 1995, which is hereby incorporated by reference, is a single stage, memoryless, and non-self routing network. Since the switch fabric 14A is not unconditionally non-blocking as a full NxN crossbar switch would be, a controller 20 is included to hunt for a path through the four pipes for each ATM cell. Since each of the pipes 18_0-18_3 contains a path that could transport the ATM cell, the real purpose of the controller 20 is to find a path that is not blocked.

For ATM switch 10A, if the number of input lines, N is equal to 256 and if each input line is operated at a standard 2.5 Gigabits per second data rate, its aggregate throughput will be 0.640 Terabits per second. Scaling or growing such an ATM switch by a factor of two to 512 input lines and output lines should be straightforward and result in aggregate throughputs of greater than 1 Terabits per second. Scaling to an ATM switch size of 1024x1024 is considered within the present technology, and the architecture of the present invention is believed to be extensible even further as the speed of commercially available components increases and as new, faster technologies are developed.

Referring now to FIG. 4, a specific embodiment of an ATM switch 10A is shown. In this specific embodiment ATM switch 10A has two hundred fifty six input interfaces 12_0-12_{255} which are connected to two hundred fifty-six ATM input lines In_0-In_{255}. The outputs of the input interfaces are connected to the input ports 17_0-17_{N-1} of the switch fabric 14A. The switch fabric 14A contains a total of sixty-four 16x16 crossbar switches 15_0-15_{63} which are partitioned into four pipes 18_0-18_3. The fanout F is equal to four which if the number of output ports = FN results in 1024 output ports 19_0-19_{1023}. The output ports 19_0-19_{1023} are respectively connected to the inputs of sixteen 64x16 output packet modules 16_0-16_{15}. The sixteen 64x16 output packet modules are connected to two hundred fifty six outputs Out_0-Out_{255}. Those skilled in the art will recognize that other combinations of components could

This Sheet Has Been Replacec

have been used, for example thirty two 32x8 output modules could have been used instead of the 64x16 output modules shown in FIG. 4.

ATM switch 10A also has a controller 20 which has the tasks of hunting and finding an available pipe through the switch fabric 14A for each ATM packet. The controller 20 uses the fact that the switch fabric 14A is partitioned into four pipes to break the pipe hunting tasks into four parallel pipe hunting tasks that are each temporally shifted by an acceptable amount. Details of one embodiment of such a controller 20 are shown in FIG. 5.

For the 0.640 Terabits per second, N=256 embodiment mentioned previously and shown in FIGs. 4 and 5, the controller 20 may be contained on approximately eight printed circuit boards. Controller 20 would accept up to 256 sixteen-bit request vectors from up to 256 line input interfaces 12_0-12_{255} and perform path hunts on each of these request vectors within each 176 nanosecond. ATM cell interval to create the 1024 sixteen-bit connect vectors used to established connections within the switch fabric 14A. This requires that controller 20 operate with a processor clock rate of at least 46 Megabits per second. This moderate clock rate permits the logic within the controller 20 to be implemented with off-the-shelf CMOS EPLD's or similar devices, thus making the cost of the controller 20 (in large quantities) very reasonable.

The movement of request vectors from the input interfaces 12_0-12_{255} to the controller 20 and the movement of connect vectors from the controller 20 to the crossbar switches 15_0-15_{63} of the switch fabric 14A is a challenging task, because large amounts of control information must be transported every 176 nsec ATM cell interval. For example, in an ATM switch containing 256 input interfaces, 256 16-bit request vectors must be transported to the controller 20 every 176 nsec, leading to an aggregate bandwidth of 23 Gigabits per second between the input interfaces sub-system and the controller 20 sub-system. In addition, 1024 16-bit connect vectors must be transported to the switch fabric 14A every 176 nsec to control the crossbars switches 15_0-15_{63}. This requires an aggregate bandwidth of 93 Gigabits per second between the controller 20 sub-system and the switch fabric 14A sub-system. This 93 Gigabits per second connect vector information can be compressed into 29 Gigabits per second (given that only one input can be routed to an output during each ATM cell

This Sheet Has Been Replaced

interval) by standard compression techniques. However, since this control information should be delivered with high reliability, all of the control connections or control links between these sub-systems should be dually redundant (not shown in FIG 4), so there is actually 46 Gigabits per second of data moving between the input interfaces cards and the controller 20 and 58 Gigabits per second of data moving between the controller 20 and the switch fabric 14A. Preferably, high-speed serial links 22 will be used to transmit this control information. For such a case, input interfaces 12_0-12_{255} would be grouped by fours such that only sixty-four serial links would be required to move request vectors from the input interfaces 12_0-12_{255} to the controller 20, and 128 serial links would be required to move the resulting connect vectors from the controller 20 to the pipes 18_0-18_3 (assuming the aforementioned data compression techniques are applied to the connect vectors).

While the use of out-of-band control techniques does require the additional hardware cost of these high-speed serial control links 22, these links 22 cause very little increase the overall system hardware cost. Considering that the 256-input ATM switch 10A of FIGs. 4 and 5 already has 1024 high-speed serial links required to route ATM cells between the input interfaces 12_0-12_{255}, and the switch fabric 14A (when the fanout of four is included) and 1024 more high-speed serial links are used to route ATM cells from the switch fabric outputs 19_0-19_{1023} to the output packet modules 16_0-16_{15}. Thus, the addition of the 192 serial links 22 for routing of the control information increases the total number of high-speed serial links within the system by merely nine percent.

Applying the calculations of Yeh et al. from the article "The Knockout Switch" the ATM cell loss probability of the ATM switch 10A shown in FIGs. 4 and 5 is 4.34×10^{-3}, assuming that the connections of the inputs is symmetrical and not independent as set forth in our copending application entitled "TERABIT PER SECOND PACKET SWITCH," EPC No. 95309013.1, filed December 12, 1995. This cell loss probability falls short of the acceptable ATM cell loss probability of less than 1×10^{-12} mentioned previously.

To reduce the ATM cell loss probabilities, controller 20 applies a temporal spreading technique known as rolling, which provides many statistical advantages.

This Sheet Has Been Replaced

Rolling involves and fulfills three fundamental goals that are aimed at providing more evenly distributed traffic loads. These goals are: (1) spatially distribute the traffic evenly across all pipes 18_0-18_3 so that one pipe will only carry its proportional fraction of the traffic load, (2) spatially distribute the traffic evenly across all of the 16x16 crossbar switches 15_0-15_{63} within each pipe 18_0-18_3 so that each of the crossbar switches is equally loaded, and (3) temporally distribute the traffic that arrives in a given ATM cell period across two ATM cell periods so that the traffic load can be effectively decreased in an occasional ATM cell period when an unusually high volume of traffic exists and is destined for a particular output packet module. This effective lowering of the traffic load is accomplished by delaying some of the ATM cells arriving during a congested ATM cell interval. The cells are delayed until the next consecutive ATM cell interval when the traffic load competing for the popular resources, i.e. connections to popular output packet modules, will most likely be lower, so the delayed cells should have a higher probability of being routed in the next ATM cell interval. Since the switch fabric 14A is memoryless, the ATM cells that must wait for the next ATM cell interval are stored in their respective input interfaces 12_0-12_{255}.

In addition to satisfying these three fundamental goals of packet traffic control to distribute the load, rolling also satisfies two further very important ATM system goals. First, goal (4) is that the ATM switch 10A must guarantee that ATM cell ordering can be simply maintained when an ATM stream is re-constructed at an output packet module 16_0-16_{15} even if rolling causes some of the ATM cells within the stream to be delayed differently than others. Secondly, goal (5) is that rolling must also guarantee that the controller 20 will attempt to route every ATM cell through each of the four paths to its desired output packet module, but each of the successive path hunt attempts must occur in a more lightly-loaded 16x16 crossbar switch so that the first attempt occurs in a 16x16 crossbar switch with many previously-routed ATM cells (and very few available paths to output packet modules) while the fourth and final path hunt attempt occurs in an 16x16 crossbar switch that is virtually empty (thereby providing many available paths to output packet modules). The rolling technique is similar to spatial path hunt techniques that pack as many calls as possible

IThis Sheet Has Been Replacec

in one portion of a spatial network, which by forcing near 100% occupancy in parts of a system results in the remainder of the calls having a very high probability of being successfully routed through the remainder of the system if usage is below 100%. Thus, rolling in its fourth and final path hunt attempt provides a very high probability of an ATM cell successfully being routed. Goal (5), by packing many ATM cells in one portion of the network, superficially seems to conflict with goal (1) that requires the traffic be spatially distributed across the network. However, as will be explained below, temporal spreading provided by the rolling technique permits the network to simultaneously satisfy both goals (1) and (5).

Assuming that each of the 256 input ports 17_0-17_{N-1} of FIG. 4 has an ATM cell that needs to be routed through the distribution network, and assuming that the switch fabric 14A is composed of four pipes 18_0-18_3, then the out-of-band controller 20 may be required to perform 256x4=1024 unique path hunts for the ATM cells before the cells can be routed. To distribute the ATM cells evenly across all four pipes, the 256 ATM cells requesting connections, the rolling technique divides the requests into four groups of equal size. The first group will have path hunts performed for its ATM cells in pipe 18_0 first, then in pipe 18_1, then in pipe 18_2, and finally in pipe 18_3. The second group will have path hunts performed for its ATM cells in pipe 18_1 first, then in pipe 18_2, then in pipe 18_3, and finally in pipe 18_0. The third group will have path hunts performed for its ATM cells in pipe 18_2 first, then in pipe 18_3, then in pipe 18_0, and finally in pipe 18_1. The fourth group will have path hunts performed for its ATM cells in pipe 18_3 first, then in pipe 18_0, then in pipe 18_1, and finally in pipe 18_2. This ring-like ordering of the path hunts guarantees that the routed ATM cells are distributed evenly across all four pipes. In addition, if the ATM cells within each of the four equally sized groups are selected such that the ATM cells within a single group can be routed into exactly four of the 16 inputs on any 16x16 crossbar switch, then the routed ATM cells will also be evenly distributed across all of the 16x16 crossbar switches.

Referring now to FIGs. 5 and 6, a timing diagram for a rolling technique according to the present invention is described. To satisfy goals (1), (2), and (5) simultaneously, the out-of-band controller 20 uses the time delay/time distribution

This Sheet Has Been Replacec

described in goal (3), and these ATM cell delays required by goal (3) must be provided during each ATM cell interval. In all cases, when a group of ATM cells is passed around the ring-like structure of controller 20 from pipe 18_3 to pipe 18_0, the controller 20 re-assigns the cells to the next ATM cell interval (period) which requires that the ATM cells be delayed by one cell period. Because of this re-assignment and delay, each cell group encounters a very lightly-loaded set of 16x16 crossbar switches for its fourth and final path hunt. An additional advantage of this rolling technique using re-assignment and delay of ATM cell intervals is that it also allows more than 64 simultaneously arriving ATM cells to be routed through the switch fabric 14A to any single output packet module 16_0-16_{15} (even though there are only 64 connections or links from the switch fabric 14A to each output packet module 16_0-16_{15}). This is occurs with the rolling technique because all of the ATM cells do not need to be routed during the same ATM cell interval. Thus, the rolling technique when used in the out-of-band controller 20 results in extremely low cell loss probabilities both within the switch fabric 14A and the output modules 16_0-16_{15}, even during a transient cell interval that has an extraordinarily high traffic load.

The one ATM cell period delays incurred by some of the ATM cells as they are routed through the switch fabric 14A would normally lead to the conclusion that there would be difficulties in satisfying goal(4) of maintaining proper cell ordering. However, the ring-like ordering of the path hunts within the out-of-band controller 20 guarantees that delayed cells in a stream of ATM cells will always be routed through lower-numbered pipes than non-delayed cells (where pipe 18_0 is the lowest-numbered pipe and pipe 18_3 is the highest-numbered pipe). This information, coupled with the fact that ATM cells are delayed by at most one cell period, ensures that proper cell ordering will be maintained if the cells are extracted from the switch fabric 14A and loaded into first-in-first-out queues 174_0-174_{63} (shown in FIG. 7) of each output module of the output modules 16_0-16_{15} in the order of the lowest numbered pipe to the highest numbered pipe: pipe 18_0, pipe 18_1, pipe 18_2, and pipe 18_3.

Referring now to FIG. 7, the output module 16_0 (and the fifteen other output modules 16_1-16_{15}) may be a 64x16 embodiment of the concentrator described in U.S. Patent No. 5,412,646, entitled "ASYNCHRONOUS TRANSFER MODE SWITCH

This Sheet Has Been Replacec

ARCHITECTURE", issued May 2, 1995, by Cyr et al. and commonly assigned to the assignee of the present invention, which application is hereby incorporated by reference. The output module 16_0 in FIG. 7 is a specific case of the generalized concentrator shown in FIG. 4 of the above-referenced patent application of Cyr et al. Since the output modules 16_0-16_{15} are well described in the above referenced application, in the interest of brevity they will not be further described here.

To provide a better understanding the equation of the rolling technique, a real-life analogy will be described with respect to FIG. 8, which is a plan view of an amusement park system 500. Consider the problem of transporting a large number of people from amusement park parking lots 511, 512, 513, or 514 to the amusement park 520 using trams to shuttle the people between the two points. Tram system 530 is composed of four tram shuttle trains each with a predetermined route, which is analogous to the four pipes of switch fabric 14A. Each tram shuttle train contains sixteen cars (representing the 16x16 crossbar switches within a particular pipe), and each shuttle car is equipped with sixteen seats (representing the output links emanating from a single 16x16 crossbar switch). In this analogy, each customer (representing an ATM cell) arrives in one of four parking lots 511, 512, 513, or 514 surrounding the amusement park 520. As a result, each customer is instantly placed in one of four groups, and since the parking lots 511-514 are the same size, each group contains an equal number of customers on the average. The customers in any single parking lot 511, 512, 513, or 514 must then divide up and stand in one of sixteen lines, where each line is associated with a respective car of the tram shuttle train. The amusement park 520 is sub-divided into sixteen different theme areas (The Past Land, The Future Land, etc.), and each of the sixteen seats of a particular tram car is labeled with the theme area to which that seat's occupant will be given admission. Before arriving in the parking lot, each customer must randomly chose one of the sixteen theme areas (representing the sixteen output packet modules 16_0-16_{15}) where he or she wishes to spend the day. Customers must then find an available seat associated with their desired theme area on one of the four trams that passes by the loading area 531, 532, 533, or 534 of their parking lot. If a customer has not found an available seat after four trams have passed by, then he or she is not permitted to enter the amusement park

This Sheet Has Been Replaced

during that day (This harsh condition represents the loss of an ATM cell due to blocking in all four pipes of the distribution network, a small but finite possibility).

The first tram that stops at the loading area that the customer can try has already visited three other parking lot loading areas, so the customer's pre-specified seat may be full. However, if the customer does find his or her seat to be vacant on that tram, then the tram will deliver him or her straight to the amusement park 520. If the customer fails to get on the first tram, he or she must wait and try the second tram which has already visited two other parking lot loading areas. If the customer is successful at finding his or her pre-specified seat on the second tram, that tram will deliver the customer to the amusement park 520 after one more parking lot stop. If the customer fails to get on the first tram and the second tram, then he or she must wait and try the third tram which has only visited one other parking lot loading area. If the customer is successful at finding his or her seat on the third tram, that tram will deliver him or her to the amusement park 520 after two additional parking lot stops. If the customer fails to get on any of the first three trams, then the customer must wait and try the fourth and final tram. Fortunately, this tram has not visited any parking lots yet, so the arriving tram is empty, and the customer 's seat will be taken only if another customer in his/her parking lot line is also trying for the same seat. The system 530 satisfies goal (5), because each of the successively arriving trams is more lightly-loaded than the previous one. Thus, a controller 20 rolling ATM cells indeed can fulfill goals (1), (2), and (5).

The rolling technique if used by itself improves the ATM cell loss probability of ATM switch 10A from 4.34×10^{-3} to approximately 10^{-11}. Using the analysis techniques of the article "A Growable Packet Switch Architecture" the cell loss probabilities for an ATM switch 10A that has independent connections to the inputs of the switch fabric 14 according to Galois field theory and also has an out-of-band controller 20 that incorporates rolling techniques can be analytically modeled and calculated. Each of the 16x16 crossbar switches in pipe 18_0 receives an offered traffic load equal to $Ra=R_L/4+Rres$, where Rres is defined to be the fraction of the 16 inputs to a 16x16 crossbar switch that are blocked in pipe 18_3 and routed to pipe 18_0 for a re-attempt. For a first attempt at solving for the cell loss probability, let us assume that

This Sheet Has Been Replaced

Rres=R_L/16. Thus, the cell loss probability of a single 16x16 crossbar switch in pipe 18_0 can be determined using the equation of Eng et al.

$$P(\text{cell loss})=[1-m/(nR_L)]\,[1-\sum_{k=0}^{m}\{(nR_L)^k e^{(-nR_L)}\}/k!]+(nR_L)^m e^{(-nR_L)}/m!;$$

where m=1, n=1, and the switch loading is given by Ra=R_L+/4+R_L+/16. Using these assignments, the resulting cell loss probability for a fully-loaded (R_L+=1.0) pipe 18_0 16x16 crossbar switch can be calculated to be:

$$P(\text{cell loss in pipe } 18_0)=1.3\text{x}10^{-1}.$$

Thus, the fraction of the 16 inputs to a 16x16 crossbar that are passed to the second pipe after the first attempt is given by:

$$f1\text{-}2=Ra \times P(\text{cell loss in pipe } 18_0) = (3.13\text{x}10^{-1})\,(1.3\text{x}10^{-1}) = 4.06\text{x}10^{-2}.$$

By symmetry, this should have also been the same as the fraction of inputs that are passed from pipe 18_3 to pipe 18_0, so the residue assumption of R_L+/16=0.062 above was incorrect. By refining this assumption and performing a second attempt, and now assuming that Rres=R_L+/32. Thus, the cell loss probability of a single 16x16 crossbar switch in pipe 18_0 can be determined again using the equation of Eng et al., where m=1, n=1, and the switch loading is given by Ra=R_L/4+R_L/32. Using these assignments, the resulting cell loss probability for a fully-loaded (R_L=1.0) pipe 18_0 16x16 crossbar switch is calculated to be:

$$P(\text{cell loss in pipe } 18_0)=1.2\text{x}10^{-1}.$$

Thus, the fraction of the 16 inputs to a 16x16 crossbar that are passed to the second pipe after the first attempt is given by:

$$f1\text{-}2 =Ra \times P(\text{cell loss in pipe } 18_0) = (2.81\text{x}10^{-1})\,(1.2\text{x}10^{-1}) = 3.37\text{x}10^{-2}.$$

This calculation result is very close to the assumed value of Rres=R_L/32=3.13x10^{-2}, so the assumption is considered to be satisfactory. The blocked cells are sent to pipe 18_1 for subsequent path hunting, and they encounter a negligible number of ATM cells from previous attempts. Thus, the 16x16 crossbar switch in pipe 18_1 can be modeled for analysis as a growable packet switch, with m=1, n=1, and Ra = f1-2, and the resulting cell loss probability of this model is 1.4x10^{-21}. The fraction of the 16 inputs to the 16x16 crossbar in pipe 18_1 that are passed to the pipe 18_2 is 4.2x10^{-4}. Similar arguments can be used to show that the

This Sheet Has Been Replaced

resulting cell loss probability for cells entering pipe 18_2 is $1.9x10^{-4}$, and the resulting fraction of the 16 inputs to a 16x16 crossbar passed to pipe 18_3 is $7.9x10^{-8}$. The resulting ATM cell loss probability in pipe 18_3 is $3.7x10^{-8}$, and the fraction of the 16 inputs to a 16x16 crossbar not routed in pipe 18_3 (and therefore not routed in all four pipe attempts) is $2.9x10^{-15}$. Thus, through the use of the rolling techniques within the out-of-band controller 20, the ATM cell loss probability of an ATM switch 10A with independent connections at the inputs of its switch fabric 14A can be decreased from an unacceptable value of $1.47x10^{-6}$ to an acceptable value of $2.9x10^{-15}$.

A preference technique may be used in conjunction with the rolling technique described above to decrease the cell loss probability of an ATM switch 10A even further. Referring back to FIG. 8 and the amusement park analogy, some form of arbitration was required at the tram loading areas to determine which of the customers in the line will be given a particular seat on the tram when more than one customer is requesting the same seat. Similarly, the out-of-band controller 20 must provide an arbitration scheme for selecting which of the arriving ATM cells will be assigned a particular link whenever two or more cells request access to the same link. The arbitration scheme used can have an advantageous effect on the ATM cell loss probabilities.

One possible arbitration scheme is a random scheme to determine which of the ATM cells is assigned the link. The random selection scheme is the scheme assumed for the analysis of the rolling technique presented above. However, other arbitration schemes are possible, and one particular arbitration scheme that has advantageous results is called the preference scheme. The preference arbitration scheme assigns a preference weight to each of the ATM cells in a particular grouping. ATM cells with higher preference weights are given precedence over ATM cells with lower preference weights whenever two or more cells request access to the same link. As a result, an effective hierarchy is created within the groupings of ATM cells.

The creation of a hierarchy may superficially seem to produce undesirable characteristics within the switch fabric 14A; because customers with high preference weights will be offered better service than customers with low preference weights. In fact, the one customer with the highest preference weight within each group can never

This Sheet Has Been Replaced

have his or her ATM cell blocked by another customer's ATM cell. Although this may seem unfair, a detailed analysis of the effects of imposing this hierarchy indicates that it actually leads to improved performance, i.e. lower cell loss probabilities, for all customers- even for the customer at the bottom of the hierarchy with the very lowest preference weight.

The results of this analysis are summarized in FIG. 9, where the probability of loss of an ATM cell ; i.e., the probability of a cell not being assigned to an available path, is shown as a function of the number of path hunts that were attempted in different pipes by the out-of-band controller 20. In this analysis, it was assumed that the group sizes were four- i.e., up to four ATM cells could simultaneously compete for access to the same link. As a result, four different preference weights were assigned to create a hierarchy for the four input ports associated with each group. The preference weight associated with a particular input port is assumed to be a fixed constant that does not vary with time. The resulting plots 901, 902, 903 and 904 in FIG. 9 indicate that the cell loss probability decreases as more path hunts in more pipes are performed, but it also shows that the inputs with the lower preference weights 903, 904 have higher cell loss probabilities than the inputs with higher preference weights 901, 902, as might be expected. Super-imposed on these plots is a similar plot 910 which indicates the probability of not being served when a random selection arbitration scheme is used instead of the hierarchy arbitration scheme. The surprising and unexpected results are that after path hunt attempts in four different pipes, the random selection arbitration scheme produces cell loss probabilities which are higher than the average of the cell loss probabilities for the hierarchy arbitration scheme. In fact, the plot 910 of the random selection arbitration scheme shows an average cell loss probabilities for all of the input ports which are notably higher than the plots 903 and 904 which are the average cell loss probabilities for even the input ports with the lowest preference weights within the hierarchy arbitration scheme. This phenomenon can be explained by the fact that after three sets of path hunts in three different pipes, the distribution of ATM cell requests entering the fourth pipe is very different depending on whether the random or preferences arbitration scheme is used. In the random selection arbitration scheme, there is a small but equal probability that all of

This Sheet Has Been Replaced

the ATM cells are requesting a path. However, in the hierarchy arbitration scheme, most of the ATM cells with higher preference weights will be requesting a path with a probability of practically zero, while the ATM cell with the lowest preference weight will be requesting a path with a sizable probability, because that particular ATM cell may have been denied access to links in all three of its previous path hunt attempts. However, a single request arriving with a high probability at the fourth and last path hunter in the controller will lead to more routed ATM cells than many requests arriving with low probability, because the single request can always be satisfied since contention for an output link will never occur.

As a result, it seems apparent from the plots in FIG. 9 that by assigning preference weights to the input ports and by using a hierarchy arbitration method to resolve link contention and route paths in the out-of-band controller, the worst-case cell loss probability of the switch fabric 14A can be decreased from 2.9×10^{-15} that was achieved by the introduction of the rolling technique to an even lower value of 2.4×10^{-16}. It is worth noting that input ports that are assigned higher preference weights will encounter even lower cell loss probabilities as indicated in FIG. 9.

Referring back to FIG. 5, in order to provide a physical embodiment of the rolling and preference methods, the ATM switch 10A is segmented in to four basic sub-systems. These four sub-groups consist of the input interfaces 12_0-12_{255}, the output modules 16_0-16_{15}, the switch fabric 14A, and the out-of-band controller 20.

The input interfaces 12_0-12_{255} within the network provide the necessary interfaces between the incoming transmission links and the links connected to the switch fabric 14A and the out-of-band controller 20. As a result, the input interfaces 12_0-12_{255} must provide a termination for the input transmission line. For example, if the input transmission line is a SONET link, then the input interface must provide for clock recovery, link error detection, SONET pointer processing and frame delineation, ATM cell extraction, and an elastic storage function to synchronize the arriving ATM cells to the system clock within the distribution network. The extracted ATM cells are then loaded into a FIFO buffer of the input interface. The input interface must also read ATM cells from the FIFO buffer and extract the ATM header from the cell. The VPI/VCI field of each ATM header is then used as an address into a translation table

This Sheet Has Been Replaced

located on the input interface. The output of the translation table provides a new VPI/VCI field and the address of the output packet module to which the ATM cell is to be routed. The new VPI/VCI field is written into the ATM cell as a replacement for the old VPI/VCI field, while the output module address is routed as a request vector to the out-of-band controller 20 for the controller fabric 14A. Since the amount of processing time required by the out-of-band controller 20 is a fixed value, the input interface simply holds the ATM cell in a buffer until the out-of-band controller 20 has completed its path hunt and has relayed the results into the switch fabric 14A. Once the switch fabric 14A is loaded with the new switch settings to appropriately route the ATM cell, the input interface can inject the ATM cell into the switch fabric 14A and it will be automatically routed through the switch fabric 14A to its desired output module 16_0-16_{15}. It should be noted that each input interface 12_0-12_{255} actually is provided with one link to each of the four pipes 18_0-18_3 of the switch fabric 14A. In addition, the use of rolling (i.e. temporal spreading) within the switch fabric 14A may require a copy of the ATM cell to be injected into each of the four links during any one of two consecutive ATM cell intervals. As a result, the timing within the input interfaces 12_0-12_{255} must be tightly coupled and synchronized to the timing of the rest of the sub-systems within the ATM switch 10A.

Each of the two hundred fifty six input interfaces 12_0-12_{255} in FIG. 5 are numbered with an address ranging from 0 to 255, but each input interface is also assigned an alias address given by a letter between A and P. These alias addresses are used to identify which input port the input interfaces will connect to within the switch fabric 14A. The actual set of four crossbar switches to which a particular input interface is connected is determined by the Galois field techniques that were described previously. These techniques guarantee independence between all of the inputs on any 16x16 crossbar switch of any pipe.

Each of the sixteen output modules 16_0-16_{15} in FIG. 5 is labeled with addresses ranging from AA to PP, and each output module performs an important function within the ATM switch 10A. Each of the output modules 16_0-16_{15} within FIG. 5 provides terminations for a respective set of sixty-four links emanating from the switch fabric 14A. Each output module 16_0-16_{15} also provides two basic

This Sheet Has Been Replaced

functions: it provides a small degree of space switching to route each ATM cell arriving on one of the sixty-four inputs to the desired one of the sixteen output ports, and it provides buffering of ATM cells to handle the problems associated with multiple packets that are simultaneously destined for the same output Out_0-Out_{255}.

There are many ways for these two functions to be implemented. The most straight-forward approach would probably construct a shared memory switch that could perform sixty-four memory writes and sixteen memory reads within an ATM cell interval (176 nsec). The memory could then be treated as sixteen disjoint linked lists (one for each output Out_0-Out_{255}) along with a seventeenth linked list containing idle memory locations. Although simple, this approach requires eighty memory accesses every 176 nsec, so it would demand memories with 2.2 nsec access times. An alternate approach would split each 64x16 output module 16_0-16_{15} into a 64x16 concentrator and a 16x16 shared memory switch. The concentrator would be a memory system that provides for sixty-four writes and sixteen reads every ATM cell interval, but the memory size could be small (and memory speeds could be fast) since the buffering required for output contention problems is not provided in this memory. In addition, the 64x16 concentrator could be implemented as a single linked list spread out across sixty-four distinct memory chips. As a result, each memory chip would require only one write and up to sixteen reads for every ATM cell interval. The 16x16 shared memory switch only performs thirty-two memory accesses every ATM cell interval, so slower (and larger) memories could be used, and the buffering for output contention problems could be provided in this shared memory portion of the output module. Thus, this latter arrangement is the more practical alternative for an output module.

The switch fabric 14A is essentially a group of small circuit switches that provide the required connectivity between the input interfaces and the output modules in response to the control signals generated by the out-of-band controller 20. In the embodiment of the ATM switch 10A shown in FIG. 5, the switch fabric 14A is composed of sixty-four 16x16 crossbar switches, where disjoint groups of sixteen switches comprise a pipe. The four pipes are labeled pipe 18_0, pipe 18_1, pipe 18_2, and pipe 18_3, and the sixteen 16x16 crossbar switches within a given pipe are labeled

This Sheet Has Been Replaced

switch 0-15. The crossbar switches must be capable of receiving the control signals generated by the out-of-band controller 20 and must reconfigure all of the switch settings during a guard-band interval between consecutive ATM cells. Each 16x16 crossbar switch supports sixteen inputs labeled input A through input P, and each 16x16 crossbar switch also supports sixteen outputs labeled output AA to output PP. It was noted above that each input interface connects to a different 16x16 crossbar in each of the four pipes 18_0-18_3, but it should now be noted that an input interface that connects to input X in pipe 18_0 is required to be connected to input X in the other three pipes 18_1-18_3 as well, where X is an element of the set {A,B,...,P}. The actual connections between the input interfaces 12_0-12_{255} and the crossbar switches within the switch fabric 14A are determined using Galois field theory techniques that were referenced above. These techniques guarantee independence between input ports for routing within switches in each pipe of the switch fabric 14A. FIG. 5 also illustrates that output YY from each of the sixty-four crossbar switches is routed to one of the sixty-four inputs on the 64x16 output module labeled YY, where YY is an element of the set {AA,BB,..,PP}.

The basic function of the out-of-band controller 20 for the switch fabric 14A is to determine through which of the four pipes 18_0-18_3 a particular ATM cell may be routed. Once the out-of-band controller 20 has successfully determined a pipe through which the ATM cell is to be routed without being blocked, the task of setting up the path through the pipe is simple, because by the definition of a pipe, there will exist only one path within the pipe between the input port of the arriving ATM cell and the desired output module. As a result, the fundamental path hunting task of a switching network is essentially reduced to the simpler task of pipe hunting in the ATM switch 10A.

The out-of-band controller 20 still requires a large busy-idle table to identify the status of each of the intermediate (FN) links between the 16x16 crossbar switches of the switch fabric 14A and the output modules 16_0-16_{15} as busy and unavailable or idle and available. However, this large busy-idle table may be sub-divided into many small busy-idle tables that the controller 20 can access in parallel, and thereby perform many pipe hunting operations in parallel. There are many ways to implement the

This Sheet Has Been Replaced

controller 20 for a large switch having the general growable packet switch architecture. In the extreme case, four levels of parallelism may be applied to the architecture of the controller 20 to perform pipe hunting. One embodiment that uses three levels of parallelism will be described in detail, first and then a fourth level of parallelism for the controller 20 will be discussed.

The first level of parallelism is obtained by providing each of the four pipes 18_0-18_3 with a respective pipe hunt controller 24_0-24_3. This level of parallelism allows pipe hunting to be carried out in all four pipe hunt controllers 24_0-24_3 simultaneously. The second level of parallelism is obtained by providing switch controllers 26_0-26_{63}, with sixteen switch controllers within each pipe hunt controller 24_0-24_3. A unique switch controller 26_0-26_{63} is respectively associated with each of the 16x16 switches within each pipe of the switch fabric 14A. As a result, pipe hunting operations can be carried out in parallel within all sixteen of the switch controllers of each pipe hunt controller 24_0-24_3. The third level of parallelism is obtained by permitting each of the switch controllers 26_0-26_{63} to perform parallel processing over all sixteen of the output links attached to its respective 16x16 crossbar switch. Effectively, each of the switch controllers 26_0-26_{63} reads sixteen busy-idle bits from its busy-idle memory in parallel, performs parallel pipe hunting operations based on those sixteen bits, and then writes the sixteen resulting busy-idle bits into its respective busy-idle memory in parallel with the other busy-idle memories. A representative switch controller 26_0 of the sixty four switch controllers 26_0-26_{63} is shown in FIG. 10. The concurrent processing of sixteen busy-idle bits is accomplished by providing switch controller 26_0 sixteen unique link controllers AA-PP, each of the link controllers AA-PP is assigned the task of processing busy-idle bits for one intermediate link between its portion of the switch fabric 14A and its respective output modules. In the embodiment shown in FIG. 10, the large busy-idle memory required to control switch 10A has been divided into many single bit memories, busy-idle flip-flops, with each single bit, busy-idle memory being logically and physically associated with its respective link controller AA-PP.

The general data flow for request vectors generated by the input interfaces 12_0-12_{255} is shown in Fig. 5. For example, input interface 12_0 in Fig. 5 routes its request

This Sheet Has Been Replaced

vector via connection 21_0 to pipe hunting controller 24_0 where it is poked into the pipe hunting ring (i.e. controller 20), and the rolling scheme requires the request vector to be looped through pipe hunt controller 24_1, pipe hunt controller 24_2, and pipe hunt controller 24_3 as it circulates around the ring. In general, each of the input interfaces 12_0-12_{255} produces one request vector, and each request vector will contain a number of bits equal to the number of output modules within the system. The request vector from a single input interface in FIG. 5 is thus a sixteen-bit data word, where each bit of the request vector points to one of the sixteen output modules. If an ATM cell within a input interface is requesting a connection to an output port on the i-th output module, then bit i within the request vector will be set to a logic "1" and all other bits within the request vector will be set to a logic "0". When the controller 20 receives this particular request vector from the input interface, it can then identify that a path is required between the source input interface and the i-th output module.

The entire sixteen-bit request vector from a input interface is routed via a respective control connection 21_0-21_{255} to one of the four pipe hunt controllers 24_0-24_3, and the controller 20 pokes the vector into one of the sixteen switch controllers associated with that particular pipe hunt controller. As shown in FIG. 10, the sixteen bits of the request vector are injected into a switch controller and are distributed across all sixteen of the link controllers within that particular switch controller. Each link controller is associated with a single link between the crossbar switches and the output modules, and it essentially processes one bit of the sixteen-bit request vector. This finite state machine circuitry that is associated with a single link controller consists of one flip-flop (the single-bit memory required to store the busy-idle bit associated with this link controller's link) and four logic gates. A state table description of the link controller operation is given in FIG 12, where the state variable is defined by the busy-idle bit. The link controller hardware provides for one request vector input bit, designated request-in; one request vector output bit, designated request-out; and one connection vector output bit, designated connect. The request vector input bit is a logic "1" if the input desires a connection through the link associated with this link controller- otherwise, it is a logic "0". The request vector output bit is a logic "1" if the logic "1" input request vector bit was not satisfied by this particular link controller-

This Sheet Has Been Replaced

otherwise, it is a logic "0". The connect vector output bit is a logic "1" if the logic "1" input request vector bit was satisfied by this particular link controller indicating the ATM cell will be routed to its desired output module through the link associated with this link controller- otherwise, it is a logic "0". The busy-idle flip-flop in Fig. 10 is reset to the logic "0" (idle) state at the beginning of each ATM cell slot, so the first request vector bit that enters the link controller with a logic "1" request is assigned the link (creating a logic "1" connect vector bit and a logic "0" output request vector bit) and sets the busy-idle flip-flop to the logic "1" (busy) state. Any subsequent request vector bits that enter the link controller during this particular ATM cell slot will be denied a connection through this link (forcing a logic "0" output on the connect vector bit and creating an output request vector bit that is identical to the input request vector bit). A time-lapsed view of several consecutive sixteen-bit 'monocast' request vectors passing through a single switch controller is shown in Fig. 12, along with the resulting states of the busy-idle bits stored within the switch controller. The resulting output request vectors and output connect vectors illustrate the general operation of each of the pipe hunt controllers 24_0-24_3.

The use of rolling within the controller 20 requires a very precise temporal ordering of two fundamental events: poking and busy-idle flip-flop clearing. The timing diagram of FIG. 13 illustrates the synchronization and data flow that might be used for the logic within the controller 20. As indicated by the timing diagram, the flow of data around the ring of controller 20 is from pipe controller 24_0 to pipe controller 24_1 to pipe controller 24_2 to pipe controller 24_3 and back to pipe controller 24_0. Request vectors generated by input interfaces with alias addresses A, B, C, and D are poked into pipe controller 24_0. Request vectors generated by input interfaces with alias addresses E, F, G, and H are poked into pipe controller 24_1. Request vectors generated by input interfaces with alias addresses I, J, K, and L are poked into pipe controller 24_2. Request vectors generated by input interfaces with alias addresses M, N, O, and P are poked into pipe controller 24_3. The poking times and busy-idle bit clearing times take place at different moments within each of the pipe hunt controllers 24_0-24_3. From the point of view of any pipe controller, the request vector bits flow through the pipe controller in alphabetical order (A to P) if one ignores the busy-idle

This Sheet Has Been Replaced

bit clearing times. This ordering guarantees that the aforementioned advantages of preferences will be realized within the controller 20, because the request vector generated from a input interface with alias address A will always be given precedence over the request vectors generated from input interfaces with alias addresses B, C, and D, etc.

The benefits derived from forced independence between the inputs on a particular 16x16 crossbar switch produce a slight increase in the complexity of the pipe hunter circuitry. Because of the independent connections between the input interfaces and the switch fabric 14A, which independence is assured by the use of Galois field theory, a request vector from a single input interface must be appropriately routed to several different switch controllers in each of the stages in the pipe hunting ring. The mixing nature of the Galois field theory generated connections requires each input interface 12_0-12_{255} to be connected to a different set of 16x16 crossbar switches within the switch fabric 14A, and as a consequence, it also requires request vectors generated on different input interfaces to be routed through entirely different sets of switch controllers within the controller 20. Since request vectors are time-multiplexed on links within the controller 20, all of the request vectors (within a particular ATM cell slot) that are expelled from a particular switch controller in one pipe hunter stage must (by definition) be routed to different switch controllers in the next pipe hunter stage. To provide this dynamic routing of the request vectors, each pipe hunt controller 24_0, 24_1,24_2 and 24_3 is connected to a respective small switching network 30_0, 30_1, 30_2 and 30_3, shown in FIG. 5. Alternatively, simple multiplexers may be used instead of switching networks 30_0, 30_1, 30_2 and 30_3, thereby greatly decreasing costs for the controller 20. Fortunately, the required configurations of these small switching networks 30_0, 30_1, 30_2 and 30_3,(or multiplexers) are cyclic with a period equal to the ATM cell period, and the required configurations can be determined *a priori* and can therefore be "hard-coded" into the small switching networks (multiplexers) during the design of the circuitry of the controller 20.

As mentioned previously, ATM switch 10A shown in FIG. 5 might be scaled such that the number of input lines were 512, 1024 or even higher. For those size switches, assuming that the input lines are carrying 2.5 Gigabits per second data rates,

This Sheet Has Been Replaced

the aggregate throughput would be over 1.0 Terabits per second. For switches of that size, a fourth level of parallelism may be needed to provide sufficient processing power for the controller 20 to hunt for all those paths through all those pipes. For ATM switches with 512 and 1024 input lines, the data rates on wires within their respective controllers are 204 Megabits per second and 386 Megabits per second, which is considerably higher that the 113 Megabits per second rate of the 256 input line version of ATM switch 10A.

The basic idea behind the fourth level of parallelism is a modification of the previously described controller 20 design which requires that request vectors be routed through the pipe hunter stages in parallel. In particular, all of the request vectors that are poked into a particular pipe are routed through the pipe hunter stages together, and these request vectors are said to comprise a poke group. In the embodiment shown in FIG. 5, this approach to the design of controller 20 creates four poke groups of sixteen-bit request vectors, so each poke group contains sixty-four bits. The four poke groups can be labeled with a concatenation of the four alias labels on the request vectors. As a result, the four poke groups for the re-designed pipe hunter of FIG. 5 are called ABCD, EFGH, IJKL, and MNOP. It is important to note that whenever a single sixty-four bit ABCD poke group is being routed through one of the switch controllers in pipe controller a of FIG. 5, there is also a sixty-four bit ABCD poke group being routed through each of the other fifteen switch controllers in pipe controller 24_0. As a result, there are a total of 1024 request vector bits associated with sixteen ABCD poke groups that are being routed through pipe 18_0 at a single instant of time. The modified controller 20 processes the request vectors for all N input ports (by passing them through all four pipe hunt controllers 24_0-24_3) every eight clock cycles, and since this task must be completed within a single 176 nsec ATM cell interval, the required clock rate within the controller 20 is 46 Megabits per second regardless of the size (aggregate throughput) of the NxN ATM switch. As a result, since the controller 20 must perform eight processing steps (regardless of the network size), the process is said to be an O(1) path hunt algorithm. During the execution of this O(1) path hunt algorithm for the N=256 input ATM switch 10A of FIG. 5, the equivalent of 16,384 link controller path hunts and 16,384 link controller path hunt

This Sheet Has Been Replaced

checks are performed every 176 nsec, so if each path hunt is considered to be an instruction execution and each path hunt check is considered to be an instruction execution, then the controller 20 can be viewed as a parallel processor capable of sustaining a 186 giga instructions per second processing rate. The trade-off for maintaining a reasonable data rate in the controller 20 (regardless of size) is an increase in link controller logic complexity and an increase in signal connections passing between successive stages of the controller as the size is increases. ATM switch designs with aggregate throughputs in excess of 1 Terabits per second will require between 4096 and 32,768 signals (at 46 Megabits per second) to be routed between successive pipe controller stages.

In addition to increasing the number of signals between pipe controller stages, the use of parallelism within the controller 20 also requires a slight increase in the hardware requirements for each link controller, because each link controller must now support a parallel path hunt on four bits within the poke group. The extra hardware added to the controller 20 by the fourth level of parallelism should be offset by the resulting lower processing rate.

Multicasting Operation

The output packet modules, because the functionality within them, have a complexity that is approximately equivalent to the distribution network and the distributed controller. For example, each output packet module must provide a concentrator function that requires many FIFO-like memories, and the necessary access speeds of these memories is directly proportional to the number of inputs and outputs of the concentrator. Additionally, each output packet module must also provide a large shared buffer memory in order to accommodate packets that are temporarily blocked from their desired output port. The necessary access speeds of this buffer memory is proportional to the number of outputs from the output packet module. The output packet module must also provide a means of guiding the packets it receives to their desired output ports in the correct time sequence. Further, each output packet module must provide for multicasting of cells between its inputs and outputs by: making multiple copies of the multicast cells, injecting new VPI/VCI fields in each of the multicast cells, and routing them appropriately. From this set of

This Sheet Has Been Replaced

functions provided, it should be evident that larger output packet modules will make the task of providing all of these functions difficult and expensive. Thus, tradeoff techniques that help decrease the required size of the output packet module can be very beneficial.

In the system shown in FIG. 5, each of the output packet modules is designed to support sixty-four input links and sixteen output links. This size is directly related to the number of crossbar switches used in the distribution network 14A, because each output packet module must have exactly one link routed to it from each of the crossbar switches in the distribution network 14A. The number of outputs emanating from the output packet module is always 1/F times the number of inputs to the output packet module, where F is the fanout provided by the distribution network 14A. The fanout F is four in the distribution network shown in FIG. 5.

The ATM switch architecture of the present invention which is based on out-of-band control techniques permits the use of global information with regard to the status of the switching fabric 14A in many different ways. Likewise, the manner in which paths for arriving ATM cells are rapidly routed within the sections of the pipe hunter also provide opportunities for several features that would probably be very difficult in an architecture based on in-band/self-routing control techniques. Multicasting is one of these features..

Multicasting is defined for ATM as the ability of an ATM switch to route a single input ATM cell to multiple output ports. Multicasting is an important feature for applications that require a single source to transmit to many destinations. On common application of this feature would in the distribution of video traffic from a video server in a CATV head-end office to all of the homes that are tuned to a particular television show. The multicasting feature may also be useful for the distribution of an e-mail message from a user on a single computer to users on many spatially distributed computers in a telecommuting environment. Although the protocols and signaling formats required for multicast routing have not yet been clearly defined by the ATM standards committees, it is likely that the ATM protocol in the future will require the user to set up a connection to all of the output ports using

This Sheet Has Been Replaced

a special VPI/VCI address that points to a linked list containing all of the desired output ports.

Referring now to FIG. 15, multicasting according to the present invention is illustrated. For The task of multicasting is divided into two parts: the distribution network 14A provides multicasting among the output packet modules 16_0-16_{15}, and each of the output packet modules 16_0-16_{15} provides multicasting between its input and output ports. As a result, the output packet modules 16_0-16_{15} (which may implement multicasting using a cell copying technique) only need to make a few copies of the ATM cell arriving at one of its inputs. The distribution network 14A can easily provide the multicast function between any of the input line cards 17_0-17_{255} and each output packet module involved in the multicast. FIG. 15 shows an ATM cell being multicast from line card 17_0 to output packet modules 16_0, 16_7 and 16_{15} to output lines 0, 1, 120 and 240. If the request vector generated by the line card has more than one bit set to a logic "1", then the pipe hunter 20 will automatically attempt to route the ATM cell to multiple output packet modules (without requiring any changes to the previously described hardware). The ATM cell will simply flow from line card 17_0 through more than one path through the distribution network 14A, so no copying of the cell by the line card(s) is required. As for the output packet modules 16_0-16_{15}, if multicasting within an output packet module is required, as it is in output packet module 16_0 in FIG. 15, then the output packet module must either copy the packet into multiple places in a receiving buffer for delivery to the output lines, or the method of operating the output packet modules during multicasting must access the stored cell multiple times. Thus, with almost no extra hardware, the ATM switch 10A can multicast ATM cells.

Multicasting that causes one packet from one input line or a group of packets from a group of input lines to be transferred to all or nearly all of the output lines of the output packet modules 16_0-16_{15} could have a very significant effect on cell loss probabilities if the total data bandwidth through the switch system 10A is not controlled by input limitations. Such input limitations are understood in the art but not shown in the figures as they are beyond the scope of this invention. It can be appreciated that if one input cell is allowed to use all output lines during one cell

This Sheet Has Been Replaced

period, any traffic on the other input lines for that cell period and the next cell period will have to contend for what is left. Simulations have shown the switch 10A to have very low cell loss probabilities for non-multicast ATM cell traffic. For a fully loaded switch as shown in FIG. 5, the average cell loss probability is much better than $1x10^{-12}$, and this cell loss probability decreases rapidly as the input line occupancy is decreased from 100 per cent. During multicasting, with the input packets limited to not exceed the output bandwidth of switch system 10A, the switch system 10A should provide a cell blocking probability that is as good or better than the blocking probability it provides during non-multicasting use. The reason for this is that during multicasting there are fewer paths required through the distribution network 14A since copying is provided as necessary at the output packet modules 16_0-16_{15}.

While the invention has been particularly illustrated and described with reference to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form, details, and applications may be made therein. For example, more than four pipes may be used in the switch fabric requiring modifications to the controller.

This Sheet Has Been Replaced

This Sheet Has Been Replaced

Claims:

1. A packet switch for switching packets from at least one of a plurality of input lines to a plurality of output lines, comprising:

a plurality of input interfaces, each having an input port connected to a respective input line of said plurality of input lines, and each of said input interfaces having an output port;

a network for switching a plurality of I input ports to a plurality of P output ports;

each of said plurality of input interface output ports is fanned out to a respective group of F of said I input ports of said network

said network having a plurality of C pipes, where C is an integer of a value equal to P/I;

a plurality of output modules, said output modules together having a plurality of inputs, each of said output module inputs connected to respective output port of said plurality of P output ports, and together having a plurality of outputs, each of said output module outputs connected to a respective output line of said plurality of output lines;

each pipe of said C pipes having a path from each of the plurality of inputs lines that is connectable to a respective output line of the plurality of output lines; and

an out of band controller multicasting at least one packet from said plurality of input lines to multiple output lines.

2. The packet switch as set forth in claim 1, wherein:

* C L M S *

This Sheet Has Been Replaced

each of said input interfaces has a store for storing a packet; and

said out-of-band controller rolls a request for a path for the packet which was unable to find an unblocked path through a first pipe to an input of a pipe controller of a second pipe and the packet is stored in said input interface while the controller is hunting an unblocked path to a respective output line for each multicast packet.

3. The packet switch as set forth in claim 1, wherein:

each of said input interfaces has a store for storing a packet; and

said out-of-band controller rolls a request for a path for the packet which was unable to find an unblocked path through both a first pipe and a second pipe to an input of a pipe controller of a third pipe and the packet is stored in said input interface while the controller is hunting an unblocked path to a respective output line for each multicast packet.

4. The packet switch as set forth in claim 1, wherein:

each of said input interfaces has a store for storing a packet; and

said out-of-band controller rolls a request for a path for the packet which was unable to find an unblocked path through a first pipe, a second pipe and a third pipe to an input of a pipe controller of a fourth pipe and the packet is stored in said input interface while the controller is hunting an unblocked path to a respective output line for each multicast packet.

5. A controller for a packet switch for locating a plurality of paths for a multicast data packet, comprising:

This Sheet Has Been Replaced

memory means for storing a busy-or-idle status for each connection between the switch fabric and the output modules;

said memory means is divided into a plurality of busy-or-idle status tables; and

means for accessing each of said plurality of busy-or-idle status tables concurrently for locating a path to each output module to which a destination output line is connected for each multicast data packet.

6. A controller for a packet switch having a switch fabric having a plurality of pipes with each pipe having a plurality of crossbar switches for locating a plurality of paths for a multicast data packet, comprising:

a plurality of pipe hunting controllers, with each of said pipe hunting controllers controlling a respective pipe of said plurality of pipes;

each of said pipe hunting controllers having a plurality of crossbar switch controllers, with each crossbar switch of a pipe having a respective crossbar switch controller;

a plurality of storage means for storing busy-or-idle tables;

each of said plurality of storage means is associated with a respective crossbar switch controller of said plurality of crossbar switch controllers and stores a respective busy-or-idle table therefor; and

means for accessing each of said busy-or-idle status tables concurrently with a multicast request vector.

7. The controller for a packet switch as set forth in claim 6, wherein each of said busy-or-idle tables has a plurality of busy-or-idle status bits with each busy-or-

This Sheet Has Been Replaced

idle status bit corresponding to a busy-or-idle status of a respective output of its respective crossbar switch.

8. The controller for a packet switch as set forth in claim 7, wherein each of said crossbar switch controllers processes its plurality of busy-or-idle status bits concurrently.

9. The controller for a packet switch as set forth in claim 7, wherein each busy-or-idle status bit is stored in a respective output controller corresponding to the respective output of its respective crossbar switch.

10. The controller for a packet switch as set forth in claim 9, wherein each output controller processes one busy-or-idle bit for controlling its respective output for each packet processing time cycle.

11. A controller for a packet switch having a switch fabric having plurality of pipes with each pipe having a plurality of crossbar switches for locating multiple paths for each multicasting telecommunication packet, comprising:

a plurality of pipe hunting controllers, with each of said pipe hunting controllers controlling a respective pipe of said plurality of pipes;

each of said pipe hunting controllers having a plurality of crossbar switch controllers, with each crossbar switch of a pipe being controlled by its respective crossbar switch controller;

a plurality of storage means for storing busy-or-idle tables;

each of said plurality of storage means is associated with a respective crossbar switch controller of said plurality of crossbar switch controllers and stores a respective busy-or-idle table therefor; and

This Sheet Has Been Replaced

means for accessing each of said busy-or-idle status tables concurrently;

wherein each of said storage means comprises:

means for storing a busy-or-idle bit;

means for reading said busy-or-idle bit from said storage means;

means for logically operating on said busy-or-idle bit with a request bit to produce a resulting busy-or-idle bit that is stored in said storage means, a request-out bit and a connect bit; and

means responsive to said connect bit of said logically operating means for connecting a path through its respective crossbar switch if said connect bit is set.

12. The controller for a packet switch as set forth in claim 11, further comprising:

means for forwarding said request-out bit to another of said plurality of storage means in another pipe where said request-out bit is inputted to the respective logical operating means of said other storage means and logically operated on with another busy-or-idle bit to produce a resulting busy-or-idle bit, another request-out bit and another connect bit.

13. The controller for a packet switch as set forth in claim 11, wherein said request bits are transmitted as a request vector in parallel.

* C L M S *

This Sheet Has Been Replaced

14. The controller for a packet switch as set forth in claim 11, wherein sixteen of said request bits are transmitted as a request vector to sixteen storage means in parallel.

15. A method for controlling a packet switch having a switch fabric having a plurality of pipes with each pipe having a plurality of crossbar switches, and a plurality of output modules during each packet cycle period, comprising the steps of:

setting all busy-or-idle bits to idle at a beginning of each packet cycle period;

reading a plurality of busy-or-idle bits equal in number to a number of output ports of each crossbar switch from a busy-or-idle memory of a crossbar switch controller in parallel;

performing parallel pipe hunting operations based on said plurality of busy-or-idle bits and an input multicast request vector producing an updated busy-or-idle vector, a connect vector and an output multicast request vector;

storing said updated busy-or-idle vector into the busy-or-idle memory in parallel; and

forwarding said connect vector to a crossbar switch of said plurality of crossbar switches to set a path from an input of said switch fabric to an output of said switch fabric for a packet.

16. The method of claim 15, further comprising the step of:

forwarding said output multicast request vector which contains any unfulfilled requests from said input request vector to a second crossbar switch controller in another pipe;

This Sheet Has Been Replaced

using said output multicast request vector as an input multicast request vector for said second crossbar switch;,

reading a plurality of busy-or-idle bits equal in number to a number of output ports of each crossbar switch from a busy-or-idle memory of said second crossbar switch controller in parallel;

performing parallel pipe hunting operations based on those sixteen busy-or-idle bits and said input multicast request vector producing a second updated busy-or-idle vector, a second connect vector and a second output multicast request vector;

storing said second updated busy-or-idle vector into the busy-or-idle memory of said second crossbar switch in parallel; and

forwarding said second connect vector to a crossbar switch of said plurality of crossbar switches to set a path from an input of said switch fabric to an output of said switch fabric for a packet.

17. The method of claim 15, wherein said performing parallel pipe hunting operations step includes the step of concurrently processing a plurality of busy-or-idle bits by providing a plurality of link controllers within each switch controller with each link controller processing busy-or-idle bits for a respective intermediate link between the switch fabric and an output module of said plurality of output modules.

18. The method of claim 15, wherein said performing parallel pipe hunting operations step includes the step of concurrently processing sixteen busy-or-idle bits by providing sixteen link controllers within each switch controller with each link controller processing busy-or-idle bits for a respective intermediate link between the switch fabric and an output module of said plurality of output modules.

* C L M S *