

(12) PATENT
(19) AUSTRALIAN PATENT OFFICE

(11) Application No. AU 199923174 B2
(10) Patent No. 736780

(54) Title
Method for providing delays independent of switch size in a crossbar switch with speedup

(51)⁶ International Patent Classification(s)
H04L 012/56 H04Q 011/04

(21) Application No: 199923174 (22) Application Date: 1999.01.12

(87) WIPO No: WO99/35792

(30) Priority Data


(31) Number	(32) Date	(33) Country
09/005740	1998.01.12	US

(43) Publication Date : 1999.07.26
(43) Publication Journal Date : 1999.09.30
(44) Accepted Journal Date : 2001.08.02

(71) Applicant(s)
cabletron Systems, Inc.

(72) Inventor(s)
Anna Charny; Pattabhiraman Krishna; Naimish Patel; Robert J. Simcoe

(74) Agent/Attorney
PHILLIPS ORMONDE and FITZPATRICK, 367 Collins Street, MELBOURNE VIC 3000

23174/99 

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

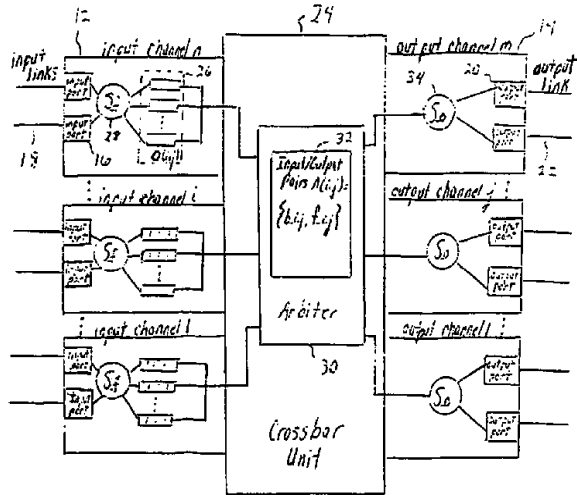
(51) International Patent Classification ⁶ : H04L 12/56, H04Q 11/04		A1	(11) International Publication Number: WO 99/35792
			(43) International Publication Date: 15 July 1999 (15.07.99)
(21) International Application Number: PCT/US99/00684		(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).	
(22) International Filing Date: 12 January 1999 (12.01.99)		<p>Published With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</p>	
(30) Priority Data: 09/005,740 12 January 1998 (12.01.98) US			
(71) Applicant: CABLETRON SYSTEMS, INC. [US/US]; 35 Industrial Way, Rochester, NH 03867 (US).			
(72) Inventors: CHARNY, Anna; 408 Dutton Road, Sudbury, MA 01776 (US). KRISHNA, Pattabhiraman; 21 Royal Crest Drive #9, Marlboro, MA 01752 (US). PATEL, Naimish; 32 Monteiro Way, N. Andover, MA 01845 (US). SIMCOE, Robert, J.; 11 Brookway Road, Westboro, MA 01581 (US).			
(74) Agent: SORKIN, Paul, D.; Wolf, Greenfield & Sacks, P.C., 600 Atlantic Avenue, Boston, MA 02210 (US).			

IP AUSTRALIA
27 JUL 1999
RECEIVED

(54) Title: METHOD FOR PROVIDING DELAYS INDEPENDENT OF SWITCH SIZE IN A CROSSBAR SWITCH WITH SPEEDUP

(57) Abstract

A scheduling and arbitration scheme in an input-buffered switch with speedup deterministic bandwidth and delay performance independent of switch size is presented. Within the framework of a crossbar architecture having a plurality of input channels and output channels, the scheduling and arbitration scheme determines the sequence of fixed-size packet or cell transmissions between the input channels and output channels satisfying the constraint that only one cell can leave an input channel and enter an output channel per phase in such a way that the arbitration delay is bounded for each cell awaiting transmission at the input channel. If the fixed-sized packets result from fragmentation of variable size packets, the scheduling and arbitration scheme implies that if delay guarantees are provided at the cell level, they are also provided to the initial variable size packets, re-assembled at the output channel, as well.



**METHOD FOR PROVIDING DELAYS INDEPENDENT OF SWITCH SIZE IN A
CROSSBAR SWITCH WITH SPEEDUP**

5 **FIELD OF THE INVENTION**

The present invention relates generally to variable and fixed size packet switches, and more particularly, to an apparatus and method for scheduling packet cell inputs through such packet switches.

10 **BACKGROUND OF THE INVENTION**

15 In the field of Integrated Services Networks, the importance of maintaining Quality of Service (QoS) for individual traffic streams (or flows) is generally recognized. Thus, such capability continues to be the subject of much research and development. Of particular interest is the delay experienced by an individual packet or cell. Good delay performance must be provided to all flows abiding to their service contract negotiated at connection setup, even in the presence of other potentially misbehaved flows. Many different methods have
20 been developed to provide such performance in non-blocking switch architectures such as output buffered or shared memory switches. Several algorithms providing a wide range of delay guarantees for non-blocking architectures have been disclosed in the literature.

25 Typically, output-buffered or shared memory non-blocking architectures require the existence of high-speed memory. For example, an output-buffered switch requires that the speed of memory at each output must be equal to the total speed of all inputs. Unfortunately, the rate of the increase in memory speed available with current technology has not kept pace with the rapid growth in demand for providing large-scale integrated services networks. Because
30 there is a growing demand for large switches with total input capacity on the order of tens and hundreds of Gb/s, building an output buffered switch at this speed has become a daunting task given the present state of technology. Similar issues arise with shared memory switches as well.



Thus, there exists a present need in the art to provide adequate delay performance to guaranteed flows while utilizing the scalability of a crossbar architecture with speedup independent of switch size.

5

SUMMARY OF THE INVENTION

According to one aspect of the present invention there is provided a method of providing delay performance independent of switch size in an input-buffered switch with a speedup S of greater than two having input channels and
10 output channels for transferring cells therebetween, including the steps of:

providing to each of the input channels per-output-channel queues for buffering cells awaiting transfer to the output channels, each per-output-channel queue being associated with a respective input channel and output channel, and having an assigned rate and an ideal service associated therewith;

15 providing an arbiter for controlling the transmission of buffered cells from input channels to output channel, the arbiter having a rate controller for scheduling at a given cell slot the queues in the input channels, the rate controller being capable of guaranteeing to each queue an amount of actual service that is within fixed bounds from the ideal service of the queue, the fixed
20 bounds each being equal to one cell;

for each per-output-channel queue, maintaining a pair of state variables including a first and a second the state variable, the first state variable corresponding to an ideal beginning time of the next cell of the per-output queue and the second state variable corresponding to an ideal finishing time of
25 transmission of the next cell of the per-output queue;

initializing the first and second state variables, the first state variable being equal to one and the second state variable being equal to one divided by the assigned rate;

initializing an arbiter clock counter for counting switch phases to zero;

30 providing a set-match set and a set_queues set;

initializing the set-match set to include an empty set and the set-queues set to include all said pairs of state variables;

running the rate controller to select from the set_queues set one of the pairs having the smallest eligible finish time first and, for the selected pair,



updating the first state variable with the ideal finish time and second state variable with the ideal beginning time plus one divided by the assigned rate;
adding the selected pair to the set-match set;
removing from the set_queues set those pairs corresponding to the same
5 input channel and output channel as the selected pair;
determining whether or not the set_queues set is empty;
if the set_queues set is determined to be empty, then notifying the input channels of the per-output-channel queues corresponding to those pairs added to the set-match set, incrementing the counter by one and returning to the step
10 of initializing the set-match and set_queues sets; and
if the set_queues is determined to be not empty, then returning to the step of running the rate controller.

According to a further aspect of the present invention there is provided a switching method for transferring data between input ports and output ports,
15 including:

providing input channels including the input ports for receiving data into the apparatus and output channels including the output ports for transmitting data from the apparatus;

20 providing, for each said input channel, per-output-channel queues for buffering data units awaiting transfer to said output channels, each said per-output-channel queue being associated with at least one said output channel and having an assigned rate and an ideal service associated therewith;

providing an arbiter for controlling transfer of said buffered data units between said input channels and said output channels, said arbiter including a
25 rate controller for scheduling the transfer and guaranteeing to each said per-output-channel queue an amount of actual service within fixed bounds from the ideal service;

maintaining, for each said per-output-channel queue, a first state variable and a second the state variable, said first state variable corresponding to an
30 ideal begin time of a next data unit of said per-output queue and said second state variable corresponding to an ideal finish time of transmission of the next data unit of said per-output queue; and

5
10
15
20
25
30



selecting for transfer by said rate controller, based on said state variables, said data unit buffered in one of said per-output queues having the smallest eligible finish time.

5 According to a still further aspect of the present invention there is provided a data switching apparatus for transferring data between input ports and output ports, including:

input channels including the input ports for receiving data into the apparatus and output channels including the output ports for transmitting data from the apparatus;

10 each said input channel including per-output-channel queues for buffering data units awaiting transfer to said output channels, each said per-output-channel queue being associated with at least one said output channel and having an assigned rate and an ideal service associated therewith;

15 an arbiter for controlling transfer of buffered data units between said input channels and said output channels, said arbiter including a rate controller constructed and arranged to schedule the transfer and guarantee to each said per-output-channel queue an amount of actual service within fixed bounds from the ideal service;

20 said rate controller arranged to maintain, for each said per-output-channel queue, a first state variable and a second the state variable, said first state variable corresponding to an ideal begin time of a next data unit of said per-output queue and said second state variable corresponding to an ideal finish time of transmission of the next data unit of said per-output queue; and

25 said rate controller arranged to select for transfer, based on said state variables, said data unit buffered in one of said per-output queues having the smallest eligible finish time.

30 The present invention may provide per cell/packet delay independent of the switch size comparable to delay guarantees associated with non-blocking output-buffered architectures, while utilizing the scalability of a crossbar. It may allow arbitrary assignment of rates (as long as the rates are feasible in the sense that the sum of all rates does not exceed the total available bandwidth at any input or any output). Additionally, it may allow the flexibility to quickly admit new flows and change the rate assignment of existing flows. Moreover, it ensures protection of well-behaved flows against misbehaved flows.



More specifically, simulations indicate that such a system may be capable of providing delays comparable to those of an output buffered switch, with any speedup of greater than or equal to two, and the delays observed are independent of switch size.

5 While the invention is primarily related to providing per-packet/cell delays to guaranteed flows, it can be used in conjunction with best-effort traffic as well. If best effort traffic is present, it is assumed that the invention as described herein is run at an absolute priority over any scheduling algorithm for best effort traffic.

10

BRIEF DESCRIPTION OF THE DRAWINGS

Preferred embodiment of the present invention will now be described with reference to the accompanying drawings, wherein:

15

FIG. 1
FIG. 2
FIG. 3
FIG. 4
FIG. 5



FIG. 1 is block diagram depicting an input-buffered crossbar switch capable of utilizing per-output-channel queue scheduling and arbitration schemes in accordance with the present invention; and

FIG. 2 is a flow diagram illustrating a queue scheduling and arbitration scheme for providing delays independent of the switch size in accordance with the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring to FIG. 1, with like reference numerals identifying like elements, there is shown an input-buffered crossbar switch 10 implementing a crossbar arbitration scheme in accordance with the present invention. As illustrated in FIG. 1, the underlying architecture of the input-buffered crossbar switch 10 is represented as an $n \times m$ crossbar. Here, " n " is the number of input channels i ($1 \leq i \leq n$) 12 and " m " is the number of output channels j ($1 \leq j \leq m$) 14. Each input channel has one or more input ports 16, each of which corresponds to a physical input link 18. Similarly, the output channels each have one or more output ports 20, each corresponding to a physical output link 22. The input channels 12 are connected to the output channels 14 by way of a crossbar unit 24. It will be understood by those skilled in the art that the crossbar unit as depicted in FIG. 1 includes a crossbar switch fabric of known construction, the details of which have been omitted for purposes of simplification. It is the crossbar switch fabric that is responsible for transferring cells between input and output channels.

In the embodiment shown, the total capacity of all input channels and all output channels is assumed to be the same, although the capacity of individual links may be different. Hereinafter, the capacity of a single channel is denoted by r_c . The speed of the switch fabric, denoted by r_{sw} , is assumed to be S times faster than the speed of any channel. In general, the switch and the channel clocks are not assumed to be synchronized. The speedup values may be arbitrary (and not necessarily integer) values in the range of $1 \leq S \leq n$. It is further assumed that the switch operates in phases of duration T_{sw} defined as the time needed to transmit a unit of data at speed r_{sw} . Such phases are referred to as matching phases. In this disclosure, a unit of data shall be referred to as a *cell*. Accordingly, a switch can move at most one cell from each input channel and at most one cell to each output channel at each matching phase. Therefore, on average, a switch with speedup S can move S cells from each input channel and S cells to each output channel. At $S=n$, the switch is equivalent to the output buffered switch.

Although not shown in FIG. 1, packets received on a given input link 18 are typically buffered at the input ports. Also, each flow to which the received packets correspond may be allocated a separate buffer or queue at the input channel. These "per-flow" queues may be located in an area of central memory within the input channel. Alternatively, flow queues may be located in a memory in the input ports associated with the input channel. When the packets received from the input links are of variable length, they are fragmented into fixed-size cells. If the packets arriving at the switch all have a fixed length, e.g., a cell in ATM networks, no fragmentation is required. In packet switching networks, where arriving packets are of different sizes, the implementation is free to choose the size of the cell as is convenient. The tradeoff in the choice of this size is that the smaller the cell, the better delays can be provided, but the faster the arbitration must be (and therefore the more expensive the switch). In addition, small cell size causes larger fragmentation overhead. Upon arrival and after possible fragmentation, cells are mapped to a corresponding flow (based on various classifiers: source address, destination address, protocol type, etc.). Once mapped, the cells are placed in the appropriate "per-flow" queue.

Associated with each guaranteed flow is some rate r_f , which is typically established at connection setup time (e.g., via RSVP). Rates assigned to guaranteed flows can also be changed during a renegotiation of service parameters as allowed by the current RSVP specification. It is assumed that the rate assignment is feasible, i.e., the sum of the rates of all flows at each input port channel does not exceed the capacity of this input port channel, and the sum of rates of all flows across all input ports destined to a particular output port does not exceed the capacity of that output port. If the sum of port capacities equals the channel capacity as assumed here, the feasibility of rates across all input and output ports implies the feasibility of rates across all input and output channels. Included in the rate r_f assigned to the flow is any overhead associated with packet fragmentation and re-assembly. The actual data rate negotiated at connection setup may therefore be lower. For networks with fixed packet size, such as ATM, however, no segmentation and re-assembly is required. Thus, no overhead is present.

As shown in FIG. 1, each input channel i 12 has m virtual output queues (VOQs) or per-output-channel queues 26 (also referred to as per-output or virtual output queues), denoted by $Q(i,j)$, $1 \leq j \leq m$, one for each output channel j 14. In the embodiment shown in FIG. 1, the input channel maintains a single flow-level scheduler $S_{f(i)}$ 28, which needs to schedule only a single flow per cell time. Once scheduler $S_{f(i)}$ schedules some flow f , it adds the index of this

flow f (or, alternatively, the head of the line (HOL) cell of flow f) to the tail of queue $Q(i,j)$. Thus, depending on the implementation, $Q(i,j)$ may contain either cells or pointers to cells of individual flows. Any known QoS-capable scheduler, such as those described above, can be used for the S_f scheduler.

5 In another variation, each input channel could maintain one flow-level scheduler $S_f(i,j)$ for each output. When the input channel i needs to transmit a cell to a given output j , it invokes scheduler $S_f(i,j)$ to determine which flow destined to j should be chosen. Unlike the option described above, in which scheduler $S_f(i)$ can run at link speed, the flow-level schedulers $S_f(i,j)$ must be capable of choosing up to S cells per cell time as it is possible that this input may need
10 to send a cell to the same output in all S matching phases of the current cell slot. In yet another approach, the input can run m parallel S_f schedulers, one per output. Each of these schedulers may schedule $1 \leq k \leq S$ cells per cell time. When a flow is scheduled by S_f , an index to this flow is added to $Q(i,j)$.

Also included in the input-buffered crossbar switch 10 is an arbiter 30 as shown in FIG.

15 1. It is the arbiter's responsibility to determine which of the input channels should be able to transmit a cell to particular output channels, i.e., cells from which per-output-channel queues should be transmitted. It is assumed that arbiter 24 operates in matching phases. The duration of each phase is equal to the duration of the channel cell slot divided by the speedup S . The goal of the arbiter is to compute a maximal (conflict-free) match between the input and output
20 channels so that at most one cell leaves any input channel and at most one cell enters any output channel during a single matching phase. Although the term "maximal match" (or, alternatively, "maximal matching") is well understood by those skilled in the art, a definition may be had with reference to papers by N. McKeown et al. and Stiliadis et al., cited above, as well as U.S. Patent No. 5,517,495 to Lund et al.

25 As explained above, during each of its matching phases, the arbiter decides which input can send a cell to which output by computing a maximal matching between all inputs and all outputs. The algorithm used to compute the maximal match is described in detail in paragraphs to follow. Once the matching is completed, the arbiter notifies each input of the output to which it can send a cell by sending to the input channel the index of the per-output queue from which
30 the cell is to be transmitted. The input channel then picks a cell to send to that output channel and the cell is transmitted to the output channel. As shown in FIG. 1, the arbiter 30 maintains

for each input/output pair i,j , a pair of variables $(b_{i,j}, f_{i,j})$ denoted as $A(i,j)$ 32. How the arbiter utilizes these input/output pairs will be described in detail later with reference to FIG. 2.

When an input channel 12 receives from the arbiter 30 the index of the $Q(i,j)$ corresponding to the output channel 14 for the current matching phase, it forwards the HOL cell of $Q(i,j)$ (or, alternatively, the cell pointed to by the HOL pointer in $Q(i,j)$) to the output channel 5 j . If $Q(i,j)$ is empty that is, there is no cell of a guaranteed flow in the queue, then a cell of a lower-priority service destined to the same output is sent instead. If there is no best effort traffic at this input matching phase, then no cell is sent.

Although not shown in FIG. 1, a cell forwarded by an input channel i to an output channel 10 j is added to a queue maintained by the output channel. A variety of queuing disciplines can be used, such as FIFO, per-input-port, or per flow. If the queue is not a simple FIFO, each output has an additional scheduler, shown in FIG. 1 as output scheduler S_o 34. This output scheduler determines the order in which cells are transmitted onto the output link from the output channel. It is assumed that any required reassembly occurs before S_o is used, so that S_o schedules 15 packets rather than cells.

Any known QoS-capable scheduler such as those mentioned above can be used for the schedule S_o .

Since each scheduler S_f, S_o operates independently of the other, the delay of an individual cell in the switch is the sum of the delay of this cell under its input and output 20 schedulers S_f and S_o , plus the delay due to the potential arbitration conflicts. The delay of a packet segmented in cells is comprised of the delay experienced by its last cell plus the segmentation and re-assembly delays.

Still referring to FIG. 1, it can now be appreciated that, with respect to each input channel, each of the queues $Q(i,j)$ contains cells (or pointers to cells) which have already been scheduled 25 by S_f but which have not yet been transmitted to their destination output channel with which the VOQ is associated due to arbitration conflicts. The present invention undertakes the task of determining the sequence of transmissions between input channels and output channels satisfying the crossbar constraint that only one cell can leave an input channel and enter an output channel per phase in such a way that the arbitration delay is bounded for each cell awaiting its 30 transmission at the input channel.

Now referring to FIG. 2, there is illustrated the actions of the arbiter with respect to scheduling the per-output-channel queues 40 in accordance with the present invention. As



previously indicated, the arbiter maintains a pair of variables $(b_{i,j}, f_{i,j})$ or $A(i,j)$ for each input/output pair i,j . These variables $b_{i,j}$ and $f_{i,j}$ will be referred to as starting time and finish time, respectively. The starting time is the ideal beginning time of transmission of the next cell of the queue with which the input/output pair is associated. The finish time is the ideal finishing time of transmission of the next cell of the queue with which the input/output pair is associated. At initial step 42, the arbiter obtains for each input/output pair i,j the rate $r_{i,j}$, which is the sum of the assigned rates of all flows going from input i to output j . Also, in the same step, variables $b_{i,j}$ and $f_{i,j}$ are initialized (to zero and $1/r_{i,j}$, respectively) and a count value *time* is set to zero.

As further illustrated in FIG. 2, at each matching phase the arbiter computes the maximal match as follows. In step 44, the arbiter initializes a *Set_Match* set to an empty set and a *Set_Queues* set to all $A(i,j)$. Now referring to step 46 in FIG. 2, the arbiter selects the pair $A(i,j)$ having the smallest finish time $f_{i,j}$ among all eligible pairs, where eligible pairs are defined as those whose starting time $b_{i,j}$ is at or before the current time. In step 48, the arbiter adds the pair selected in step 46 to *Set_Match*, updates the variables such that $b_{i,j}=f_{i,j}$ and $f_{i,j}=f_{i,j}+1/r_{i,j}$ as indicated in step 50 and, in step 52, removes from set *Set_Queues* all pairs corresponding to the input and/or output of the $A(i,j)$ selected in step 46. If there are any pairs remaining in *Set_Queues* (step 54), the arbiter returns to step 46 and performs the next iteration of the matching process. Otherwise, the matching is complete. In step 56, for each $A(i,j)$ in the match, the arbiter informs the input i to send to all output j . As can be seen, the $A(i,j)$ in the match correspond to the per-output-channel queues $Q(i,j)$ from which a cell should be transmitted in the current matching phase. The arbiter then proceeds to the next matching phase, incrementing count *time* by one (step 58) and updating the rates $r_{i,j}$ as necessary (step 60) before returning to step 44.

In an alternative input-buffered switch algorithm described in a co-pending application which runs a separate version of the rate controller per input and performs arbitration using the scheduling times of the rate controllers, the delay bound is a function of the size of the switch (i.e., the number of input/channels). In contrast, the arbiter of the present invention runs a single rate controller across all queues regardless of the input or output channels to which they correspond and uses finish times (rather than scheduled times) as described above. Also, in the above-referenced co-pending application, the input rate controllers which schedule per-output queues at each input are oblivious to potential arbitration conflicts. The arbitration conflicts are resolved at the arbiter using timestamps of the scheduling times of the input rate controllers.



Here, in the present invention, the rate controller which is run in the arbiter uses ideal start and finish times of all input/output pairs directly and explicitly resolves arbitration conflicts as part of the operation of the rate controller. Hence, the advantage of the present invention is that the observed delays are independent of the size of the switch and depends only on the rate of the flow. However, in the present invention the rate controller must operate at the faster speed of the switch fabric whereas the input channel rate-controllers in the co-pending application need to operate at a slower channel speed. Likewise, the size of the input to the rate-controller in the present invention is $n \times m$, whereas in the co-pending invention the input to each of the rate-controllers is only m . As a result, the implementation of the co-pending invention may be less expensive, especially at high speeds.

While the disclosed input-buffered switch and scheduling method has been particularly shown and described with reference to the preferred embodiments, it will be understood by those skilled in the art that various modifications in form and detail may be made therein without departing from the scope and spirit of the invention as set forth by the claims. Accordingly, modifications such as those suggested above, but not limited thereto, are to be considered within the scope of the claims.

THE CLAIMS DEFINING THE INVENTION ARE AS FOLLOWS:

1. A method of providing delay performance independent of switch size in an input-buffered switch with a speedup S of greater than two having input channels and output channels for transferring cells therebetween, including the steps of:

5 providing to each of the input channels per-output-channel queues for buffering cells awaiting transfer to the output channels, each per-output-channel queue being associated with a respective input channel and output channel, and having an assigned rate and an ideal service associated therewith;

10 providing an arbiter for controlling the transmission of buffered cells from input channels to output channel, the arbiter having a rate controller for scheduling at a given cell slot the queues in the input channels, the rate controller being capable of guaranteeing to each queue an amount of actual service that is within fixed bounds from the ideal service of the queue, the fixed bounds each being equal to one cell;

15 for each per-output-channel queue, maintaining a pair of state variables including a first and a second the state variable, the first state variable corresponding to an ideal beginning time of the next cell of the per-output queue and the second state variable corresponding to an ideal finishing time of transmission of the next cell of the per-output queue;

20 initializing the first and second state variables, the first state variable being equal to one and the second state variable being equal to one divided by the assigned rate;

25 initializing an arbiter clock counter for counting switch phases to zero;

providing a set-match set and a set_queues set;

initializing the set-match set to include an empty set and the set-queues set to include all said pairs of state variables;

30 running the rate controller to select from the set_queues set one of the pairs having the smallest eligible finish time first and, for the selected pair, updating the first state variable with the ideal finish time and second state variable with the ideal beginning time plus one divided by the assigned rate;

adding the selected pair to the set-match set;



removing from the set_queues set those pairs corresponding to the same input channel and output channel as the selected pair;
determining whether or not the set_queues set is empty;
if the set_queues set is determined to be empty, then notifying the input
5 channels of the per-output-channel queues corresponding to those pairs added to the set-match set, incrementing the counter by one and returning to the step of initializing the set-match and set_queues sets; and
if the set_queues is determined to be not empty, then returning to the step of running the rate controller.

10

2. The method of claim 1 further including providing to each of the output channels a queue arranged to receive cells.

3. The method of claim 1 or 2 wherein said cells are ATM cells.

15

4. A switching method for transferring data between input ports and output ports, including:

providing input channels including the input ports for receiving data into the apparatus and output channels including the output ports for transmitting data from the apparatus;

20

providing, for each said input channel, per-output-channel queues for buffering data units awaiting transfer to said output channels, each said per-output-channel queue being associated with at least one said output channel and having an assigned rate and an ideal service associated therewith;

25

providing an arbiter for controlling transfer of said buffered data units between said input channels and said output channels, said arbiter including a rate controller for scheduling the transfer and guaranteeing to each said per-output-channel queue an amount of actual service within fixed bounds from the ideal service;

30

maintaining, for each said per-output-channel queue, a first state variable and a second state variable, said first state variable corresponding to an ideal begin time of a next data unit of said per-output queue and said second state variable corresponding to an ideal finish time of transmission of the next data unit of said per-output queue; and



selecting for transfer by said rate controller, based on said state variables, said data unit buffered in one of said per-output queues having the smallest eligible finish time.

5 5. The switching method of claim 4, wherein said selecting for transfer includes:

initializing the first and second state variables forming a pair of state variables, the first state variable being equal to one and the second state variable being equal to one divided by the assigned rate;

10 initializing an arbiter clock counter for counting switch phases to zero;

providing a set-match set and a set_queues set;

initializing the set-match set to include an empty set and the set_queues set to include all said pairs of state variables;

15 running the rate controller to select from the set_queues set one of the pairs having the smallest eligible finish time first and, for the selected pair, updating the first state variable with the ideal finish time and second state variable with the ideal begin time plus one divided by the assigned rate;

adding the selected pair to the set-match set;

20 removing from the set_queues set those pairs corresponding to the same input channel and output channel as the selected pair;

determining whether or not the set_queues set is empty;

25 if the set_queues set is determined to be empty, then notifying the input channels of the per-output-channel queues corresponding to those pairs added to the set-match set, incrementing the counter by one and returning to the step of initializing the set-match and set_queues sets; and

if the set_queues is determined to be not empty, then returning to the step of running the rate controller.

30 6. The switching method of claim 4 or 5 wherein said data units buffered in said per-output-channel queues are pointers to data stored in a shared memory.

7. The switching method of claim 4, 5 or 6 wherein said data units are fixed length cells.



8. The switching method of claim 7 wherein said cells are ATM cells.

9. The data switching apparatus of any one of claims 4 to 8 wherein said data transferred between input ports and output ports are a variable length packets and said method including fragmenting, in said input channel, said variable length packets into said data units to be buffered in said per-output-channel queues, and assembling, in said output channel, said fragmented data units to form said variable length packets.

10. A data switching apparatus for transferring data between input ports and output ports, including:

input channels including the input ports for receiving data into the apparatus and output channels including the output ports for transmitting data from the apparatus;

each said input channel including per-output-channel queues for buffering data units awaiting transfer to said output channels, each said per-output-channel queue being associated with at least one said output channel and having an assigned rate and an ideal service associated therewith;

an arbiter for controlling transfer of buffered data units between said input channels and said output channels, said arbiter including a rate controller constructed and arranged to schedule the transfer and guarantee to each said per-output-channel queue an amount of actual service within fixed bounds from the ideal service;

said rate controller arranged to maintain, for each said per-output-channel queue, a first state variable and a second the state variable, said first state variable corresponding to an ideal begin time of a next data unit of said per-output queue and said second state variable corresponding to an ideal finish time of transmission of the next data unit of said per-output queue; and

said rate controller arranged to select for transfer, based on said state variables, said data unit buffered in one of said per-output queues having the smallest eligible finish time.

11. The data switching apparatus of claim 10 wherein said rate controller is further arranged to use a set-match set and a set_queues set, said set-match



- set being first initialized to include an empty set and said set-queues set to include all pairs of said state variables; said rate controller being further arranged to select from said set_queues set one of the pairs having the smallest eligible finish time first and, for the selected pair, update said first state
- 5 variable with the ideal finish time and second state variable with the ideal begin time plus one divided by the assigned rate, add the selected pair to said set-match set, and remove from said set queues set those pairs corresponding to a same one of said input and output channels as the selected pair.
- 10 12. The data switching apparatus of claim 10 wherein said input channel further includes one flow-level scheduler arranged to schedule said data units for storing in said per-output queues.
- 15 13. The data switching apparatus of claim 12 wherein said flow-level scheduler is a QoS capable scheduler.
14. The data switching apparatus of any one of claims 10 to 13 wherein said input channel further includes several flow-level schedulers.
- 20 15. The data switching apparatus of claim 12, 13 or 14 wherein said output channel further includes one output scheduler arranged to schedule said data units.
- 25 16. The data switching apparatus of claim 15 wherein said output scheduler is a QoS capable scheduler.
17. The data switching apparatus of claim 15 wherein said output channel further includes output buffers.
- 30 18. The data switching apparatus of any one of claims 10 to 17 wherein said data units stored in said queues are pointers to data stored in a shared memory.



19. The data switching apparatus of any one of claims 10 to 18 wherein said data units are fixed length cells.
20. The data switching apparatus of claim 19 wherein said cells are ATM cells.
21. The data switching apparatus of any one of claims 10 to 20 wherein said data transferred between input ports and output ports are a variable length packets.
22. The data switching apparatus of claim 21 wherein said input channel further includes a fragmentation circuitry, and said output channel further includes an assembly circuitry.
23. A method of providing delay performance independent of switch size in an input buffered switch substantially as herein described with reference to the accompanying drawings.
24. A switching method for transferring data between input ports and output ports substantially as herein described with reference to the accompanying drawings.
25. A data switching apparatus for transferring data between input ports and output ports substantially as herein described with reference to the accompanying drawings.

DATED: 31 August, 2000

PHILLIPS ORMONDE & FITZPATRICK
Attorneys for:
CABLETRON SYSTEMS, INC.



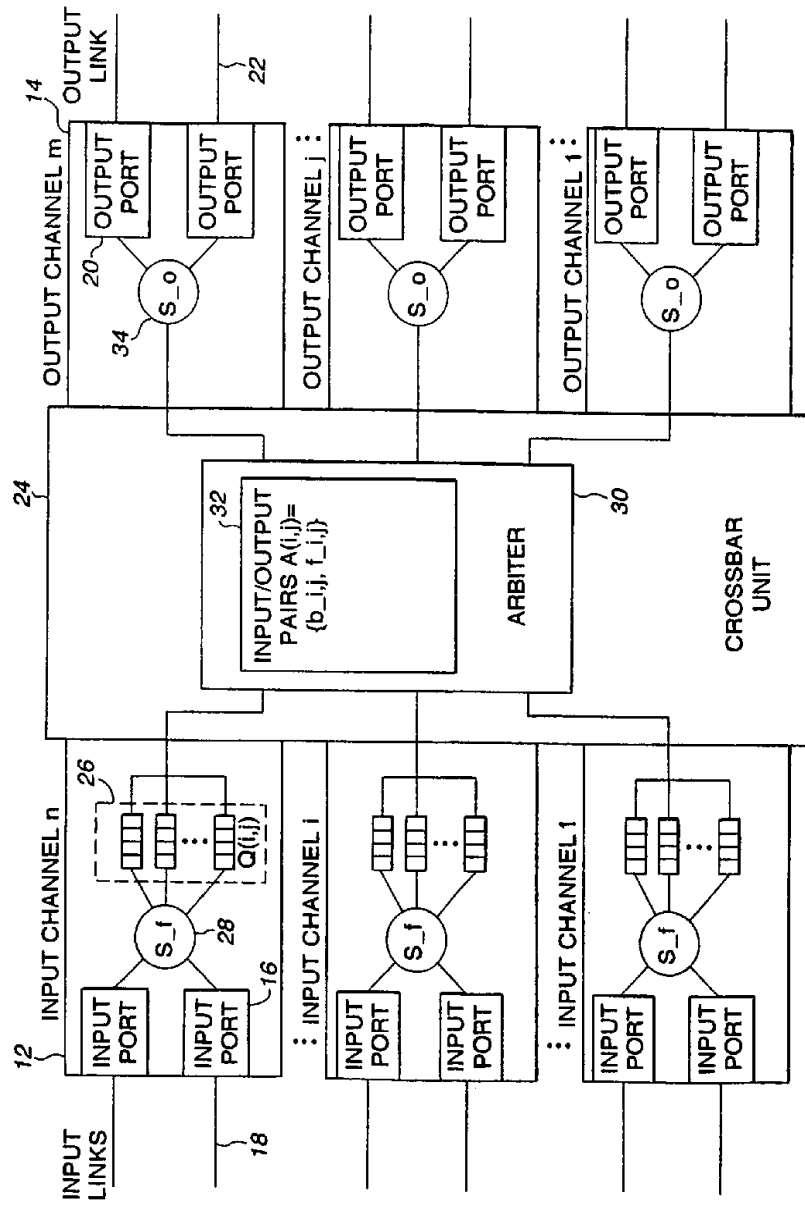


FIG. 1

10

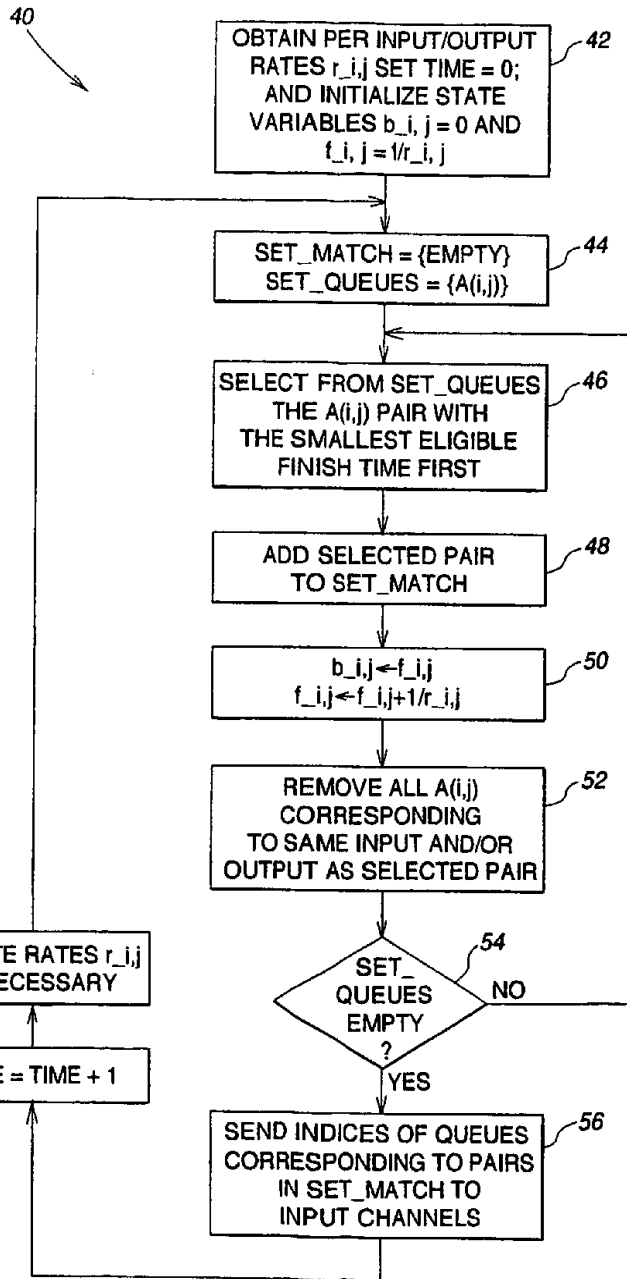


FIG. 2
SUBSTITUTE SHEET (RULE 26)