An Enhanced Timed-Round-Robin Traffic Control Scheme for ATM Networks

Qin Zheng

TR94-01   December 1994

Abstract

ATM (asynchronous transfer mode) aims at providing both guaranteed bandwidth to support real-time communications and dynamic bandwidth sharing to accommodate bursty data traffic. Achievement of this goal is vital to make ATM an enabling technology for the future integrated digital networks. To realize this objective, good algorithms for controlling network traffic are required. This paper proposes a traffic control scheme which uses a timed-round-robin cell transmission scheduling algorithm enhanced with a simple feedback flow control mechanism to realize the promised advantages of ATM networks. Specifically, with the proposed scheme, a network is able to provide each user with (1) a guaranteed bandwidth, (2) immediate access to the full link bandwidth when there is no contention from other users, and (3) free of cell losses. An ATM switch design is also presented which shows the feasibility of implementing the proposed scheme with todayś switch architectures without adding much extra cost.

Mitsubishi Electric Research Laboratories
Cambridge Research Center

Technical Report 94-01 March 25, 1994

AN ENHANCED TIMED-ROUND-ROBIN TRAFFIC CONTROL SCHEME FOR ATM NETWORKS

by

Qin Zheng

Abstract

ATM (asynchronous transfer mode) aims at providing both guaranteed bandwidth to support real-time communications and dynamic bandwidth sharing to accommodate bursty data traffic. Achievement of this goal is vital to make ATM an enabling technology for the future integrated digital networks. To realize this objective, good algorithms for controlling network traffic are required. This paper proposes a traffic control scheme which uses a timed-round-robin cell transmission scheduling algorithm enhanced with a simple feedback flow control mechanism to realize the promised advantages of ATM networks. Specifically, with the proposed scheme, a network is able to provide each user with (1) a guaranteed bandwidth, (2) immediate access to the full link bandwidth when there is no contention from other users, and (3) free of cell losses. An ATM switch design is also presented which shows the feasibility of implementing the proposed scheme with today's switch architectures without adding much extra cost.

Submitted to *ACM Multimedia '94.*

201 Broadway
Cambridge Massachusetts 02139

Publication History:-

1. First printing, TR 94-01, March 25, 1994

1 Introduction

ATM (asynchronous transfer mode) has gained wide acceptance in both industry and academia over the past several years [1]. It is now almost unanimously agreed that ATM will become an enabling technology for the future integrated digital networks and thus nearly all large computer and communication companies have invested in developing ATM products [2, 3, 4, 5]. However, most existing ATM products have been developed without attention to a fundamental problem of the ATM: traffic control to maintain a satisfactory network performance. This has made today's ATM networks incapable of providing quality of service guarantees for real-time applications and dynamic bandwidth sharing for bursty data traffic in an integrated manner [6, 7], making the ATM technology fall far behind with what it promised as a technology to integrate today's telecommunication and computer networks.

There are several reasons for the lack of traffic control functions in today's ATM products. First, traffic control in fast cell-switched networks is a very difficult problem. Although many approaches have been proposed, there is no general agreement as to which one can really solve the problem [8, 9, 10]. As a result of this, no standards have been established for traffic control in ATM networks, which makes it very difficult for vendors to decide on a right traffic control scheme to implement. Secondly, an addition of a traffic control function usually increases the complexity and cost of an ATM product significantly. This is highly undesirable for vendors to make their products competitive in today's very competitive ATM market. Finally, in a networked environment, interoperability is of the upmost importance. Nobody is willing to add functions to make their products incapable of talking with others. These factors have made most vendors simply build "raw" ATM products which do not have any traffic control capacities.

However, this situation will not last long. First, ATM will not survive without showing some convincing advantages over other networking technologies. As today's experimental ATM networks become larger and larger, and high-performance workstations and multimedia applications generate more and more traffic, one can no longer solely rely on high transmission bandwidth to provide a satisfactory network performance. Sooner or later, all ATM networks will have to use some traffic control schemes. Secondly, after demonstration that the current VLSI technology is capable of implementing cell-based fast switching of ATM (which only a few years ago was believed to be not practical), most vendors do have the technology and interest in adding traffic control functions to their products. Finally, at the time when no standard traffic control schemes are available, the ones who make the first tries will very likely make their approaches the *de facto* standard of the industry.

Based on these observations, this paper aims at developing a traffic control scheme which is effective enough to provide a satisfactory network performance, but also simple enough to be implemented in the next generation ATM products.

Traffic control in an ATM network can be described as a problem of controlling the amount of traffic transferred over the network such that a satisfactory quality of service and network utilization can be realized. Allowing too much traffic into a network will cause network congestion which may result in cell losses and reduce quality of service. On the other hand, overly limiting traffic into a network is also not desirable because this may cause a low network utilization.

People have tried to solve the traffic control problem with two approaches: *preventive control* and *reactive control*. Preventive control tries to prevent a network from reaching an unacceptable level of congestion [11, 12]. A common approach is to control traffic flow at entry points of a network with admission control and bandwidth enforcement. Admission control determines whether or not to accept a new connection at the connection setup time.

Bandwidth enforcement monitors each individual connection to ensure that the actual traffic flow conforms to that specified at the time of connection setup. With the use of a proper cell transmission scheduling policy at switches [13, 14, 15, 16, 17, 18, 19], preventive control can often provide users with a certain type of quality of service, ranging from a guaranteed bandwidth to delay bounds. But a major problem with preventive control is that it is difficult to achieve high network utilization for variable bit-rate traffics. A user is usually not allowed to send more messages than specified even when a network load condition allows this to happen. So preventive traffic control is suitable for constant bit-rate transmissions, but may cause unacceptable low efficiency for bursty data traffics. Also, many traffic policing and scheduling schemes turned out to be very costly to implement in practice [20].

Reactive control, on the other hand, admits traffic according to the *current* network load condition with a feedback mechanism [21, 22, 23, 24]. A network can accept traffic with its best effort and there is no pre-set limit on the amount of traffic that each connection can carry. Thus it is possible to achieve a very high degree of dynamic bandwidth sharing and a high network utilization with reactive control, making it suitable for bursty data traffics like those exhibited in computer networks. However, with reactive control, it is difficult to provide quality of service guarantees since the bandwidth that each connection can use depends on the actual network load condition. Lack of quality of service guarantees makes reactive control incapable of supporting real-time applications like interactive audio/video transmissions.

Thus, it is clear that to make ATM a true enabling technology to integrate today's data and telecommunication networks, a traffic control scheme which combines the advantages of both preventive and reactive control must be used. Also, the scheme should be simple enough to be implemented with today's technology without adding too much cost to a final product. Specifically, we want to achieve the following objectives:

1. Guaranteed bandwidth transmission.
   This is to allow users to establish connections with guaranteed bandwidths. In an ATM network, the bandwidth guarantee is provided in a form of (N_{cell}, T), which means that one can send at least N_{cell} of ATM cells over a connection during a time period of length T. Guaranteed bandwidth transmission is a necessity to make an ATM network capable of supporting services provided by today's circuit-switched telecommunication networks.

2. Dynamic bandwidth sharing and lossless transmission.
   This means that a connection can instantaneously exceed its guaranteed bandwidth whenever resources are available. Specifically, any bandwidth not allocated or bandwidth allocated to one connection but not used should be made available to all others, and it should also be guaranteed that no buffer overflows will occur due to this dynamic bandwidth sharing. Achievement of this objective will allow an ATM network to make efficient use of transmission bandwidth which is essential to support variable bit-rate traffics of today's data communication networks.

3. Easy implementation.
   Adding any traffic control functions will inevitably increase the implementation complexity. But we hope to develop a traffic control scheme which does not require some costly circuits like a fast sequencer to sort cells in a particular order as needed by many proposed traffic control schemes [25]. Low cost is important to make the scheme implementable in commercial products (not just laboratory ones).

In addition to making ATM a true enabling technology for the future integrated digital networks, we believe that integration of guaranteed bandwidth transmission and dynamic bandwidth sharing as described above could bring immediate benefits to both network users and network providers if a proper billing scheme is used. One possible scheme is to calculate the cost of a connection partially based on its guaranteed bandwidth and partially based on the actual number of cells transmitted. In this way, a video conferencing user may be able to save money by turning off the video when a meeting gets boring. Since the connection is still there, he/she can monitor the meeting with audio and restart the video transmission immediately when some thing interesting happens. Network providers, on the other hand, would benefit from the increased network utilization and potential "double charging" of the guaranteed bandwidth.

We propose in this paper a traffic control scheme which uses a timed-round-robin cell transmission algorithm enhanced with a simple feedback flow control mechanism to achieve the above objectives. The paper is organized as follows. Section 2 discusses the providing of guaranteed bandwidth transmission with a timed-round-robin cell transmission scheduling algorithm, and Section 3 shows how the enhancement with a simple feedback flow control mechanism can help achieving dynamic bandwidth sharing and lossless transmission. Implementation of the proposed scheme in ATM switches is discussed in Section 4, and the paper concludes with Section 5.

2 Guaranteed bandwidth transmission

Guaranteed bandwidth transmission has long been realized in telecommunication networks with an STM (Synchronous Transfer Mode) technology. STM divides time into fixed length frames each of which is further divided into smaller time slots. Guaranteed bandwidth transmission is realized by allocating certain numbers of time slots in each frame to individual connections. Although STM has been quite successful in telecommunication networks, it is not suitable for the future integrated networks due to its inflexibility in bandwidth allocation and inefficiency in bandwidth utilization. Specifically, with STM, it is difficult to manage time slots to provide connections with different guaranteed bandwidths and allow connections to dynamically share transmission bandwidth.

ATM uses a cell-based transmission approach which does not statically allocate time-slots to connections. Each cell is a self-contained unit which can be transmitted at any time. In this way, users can share transmission bandwidth efficiently and a high network utilization can be achieved. However, a consequence of this dynamic bandwidth sharing is that it becomes difficult to provide bandwidth guarantees to connections.

One approach to solve this problem in ATM networks is to use a Stop-and-Go cell transmission scheme [11]. Similar to STM, Stop-and-Go divides time into frames. But instead of allocating fixed time-slots within each frame to connections, Stop-and-Go uses a stop-queue and a go-queue to limit the number of cells that each connection can transmit during each time frame. In this way, bandwidth can be flexibly managed (i.e., connections with different bandwidths can be easily accommodated). But limiting the number of cells in each frame diminishes the potential for dynamic bandwidth sharing.

We propose to use a Timed-Round-Robin cell transmission scheduling algorithm to provide bandwidth guarantees in ATM networks. Timed-Round-Robin scheduling can realize similar bandwidth allocation ability as Stop-and-Go, but it has the advantage of being easier to incorporate a feedback flow control scheme to achieve dynamic bandwidth sharing and lossless transmission by using a concept of synchronous and asynchronous transmissions similar to that used for the Medium Access Control (MAC) of the FDDI [26, 27]. Specifically, we divide time into fixed-size frames called Round-Robin Period (RRP). In one RRP, each connection is given a certain amount of Guaranteed Transmission Time (GTT) with which cells are transmitted without being controlled by a feedback flow control scheme. This is called synchronous transmission since each connection is periodically granted a certain amount of transmission time. Any remaining time is used to transmit cells asynchronously subject to a feedback flow control scheme. In this way, one can simultaneously achieve guaranteed bandwidth allocation and dynamic bandwidth sharing.

For the convenience of presentation, we assume that time is measured in a unit of the transmission time of one ATM cell and RRP and GTT's are all integers. Also, the term "connections" will be used instead of virtual circuits (VCs) or virtual paths (VPs) as used in ATM standards to indicate that a connection can actually represent one or many VCs/VPs. The proposed timed-round-robin cell transmission algorithm is given below and illustrated in Fig. 1:

Algorithm 2.1 (Timed-round-robin transmission) .

1. *Each link is assigned a round-robin period, RRP, which is the time of completing one round of round-robin transmission, and a round-robin timer, RRT, which controls the time that each connection can use to transmit its cells.*

2. *According to its requested bandwidth, a connection is assigned a guaranteed transmission time, GTT, over each link it passes through.* $GTT/RRP \times$ *link bandwidth is the bandwidth allocated to the connection over a link.*

3. *Suppose there are n connections,* $CN_1, \cdots, CN_n$*, established over a transmission link with guaranteed transmission times* $GTT_1, \cdots, GTT_n$*, respectively. Let* $T_i = GTT_1 + \cdots + GTT_i$ *be the accumulated guaranteed transmission time of the first i connections, and* $\delta = (RRP - T_n)/n$ *be each connection's share of unallocated bandwidth. Then, cell transmissions are scheduled in a timed-round-robin manner as follows:*

   Step 1: *Set* $i := 1$ *and* $RRT := 0$.

   Step 2: *Transmit up to* GTT_i *cells of* CN_i*, or more specifically, transmit cells of* CN_i *until RRT reaches* T_i *or all cells of* CN_i *have been transmitted. Cells transmitted in this way are marked as synchronous cells.*

   Step 3: *Transmit cells with an asynchronous transmission algorithm described in the next Section until RRT reaches* $T_i + \lfloor i\delta \rfloor$*. Cells transmitted in this way are marked as asynchronous cells.*

   Step 4: *If* $i < n$*, set* $i := i + 1$ *and go to Step 2. Otherwise, go to Step 1.*

Fig. 1 shows a queuing structure for round-robin transmission and a pattern of alternations between synchronous and asynchronous transmissions when each connection uses all its guaranteed transmission time transmitting synchronous cells. If a connection does not use all its assigned guaranteed transmission time, the remaining time is used for asynchronous transmission as indicated by the small arrows in the time chart.

Since connection i is guaranteed to have GTT_i units of time to transmit its cells during a time period of RRP, it is guaranteed to be able to use $B_i = (GTT_i/RRP) \times B$ average bandwidth over the link. A further observation is that if no more than GTT_i cells of connection i arrive at a node in any RRP units of time (recall that 1 time unit equals the transmission time of one cell), connection i is always able to transmit all its cells queued at the node each

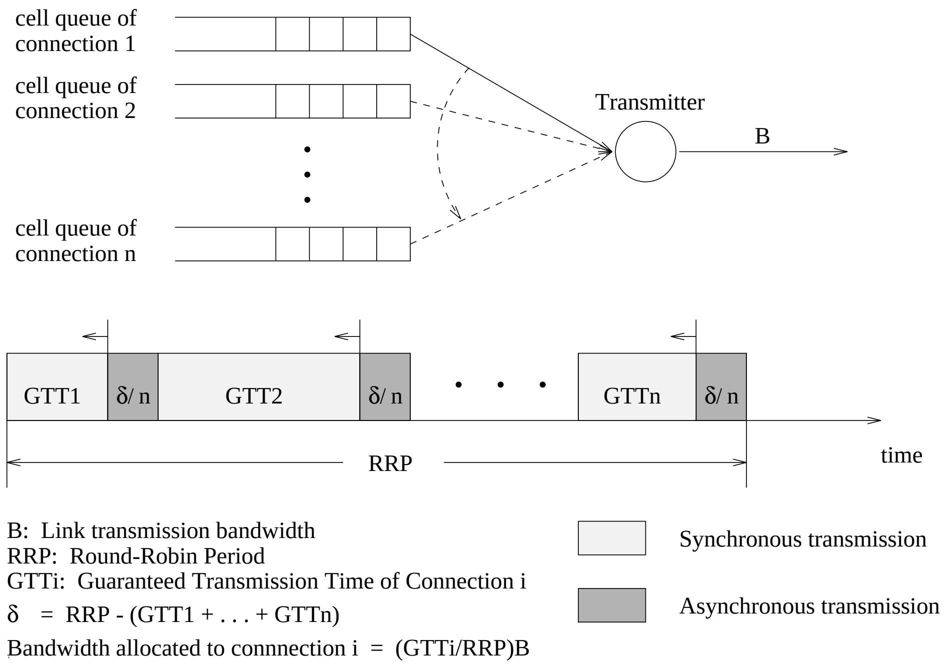
cell queue of connection 1
cell queue of connection 2
cell queue of connection n
Transmitter
B
GTT1
δ/ n
GTT2
δ/ n
GTTn
δ/ n
RRP
time
B: Link transmission bandwidth
RRP: Round-Robin Period
GTTi: Guaranteed Transmission Time of Connection i
δ = RRP - (GTT1 + . . . + GTTn)
Bandwidth allocated to connnection i = (GTTi/RRP)B
Synchronous transmission
Asynchronous transmission

Figure 1: Timed-round-robin cell transmission.

time it gets its turn to transmit. This leads to the following two properties of Algorithm 2.1 under a condition that the average cell arrival rate of a connection over a time period of RRP does not exceed the allocated bandwidth of the connection:

1. The worst-case cell queuing delay at one node is bounded by RRP.
2. The maximum buffer needed for connection i is bounded by the GTT_i.

These two features make ATM networks capable of supporting services supported by today's circuit-switched networks since bounded delay and lossless transmission are guaranteed as long as an application does not exceed its allocated communication bandwidth.

The round-robin period RRP is an important parameter which affects the performance of Algorithm 2.1. On one hand, it is desirable to set a small RRP to achieve small cell delays and low buffer requirement. On the other hand, RRP determines the grain of bandwidth allocation. If the link bandwidth is B, then the minimum bandwidth that can be allocated to a connection is B/RRP. For example, to set up a 64 Kb/s connection over a 150 Mb/s link, RRP should be no smaller than $150,000/64 = 2344$, resulting in a maximum cell delay of about 6.3 ms over the link. Of course, one can easily make some enhancements to the basic timed-round-robin algorithm discussed above. For example, a smaller worst-case delay can be achieved by splitting a high-bandwidth connection into several low-bandwidth connections, and a finer bandwidth allocation grain can be achieved by multiplexing several low-bandwidth connections into a single high-bandwidth connection. Also, there are no particular reasons for interleaving the unallocated transmission time δ with the guaranteed transmission times in the way shown in Fig. 1. One may distribute δ in some other ways for some purpose (e.g., easy implementation or low-latency communication [28]). We are not going to discuss these enhancements further in this paper.

As expained in Algorithm 2.1, $GTT/RRP\times$ link bandwidth is the bandwidth guaranteed to a connection over a link. So to establish a connection of bandwidth B_c over a link of bandwidth

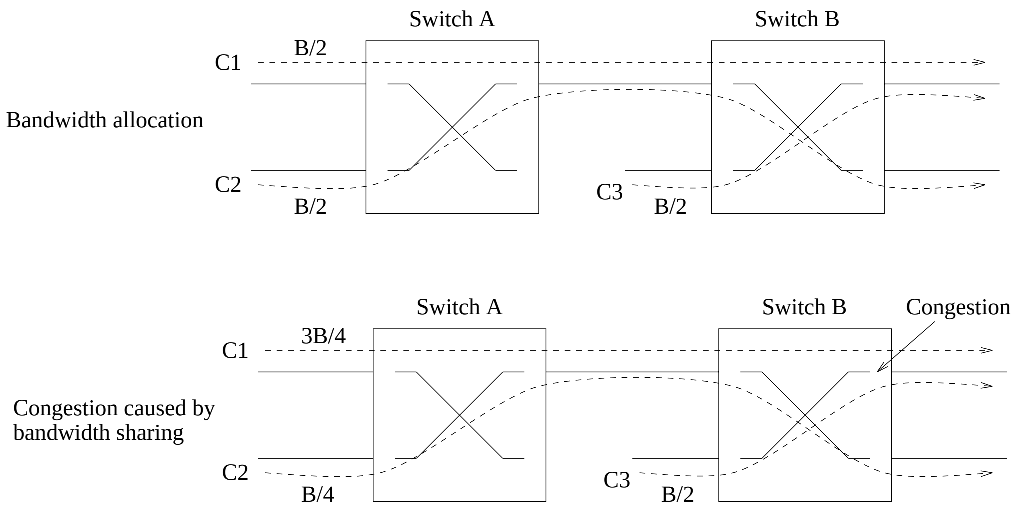

Figure 2: Dynamic bandwidth sharing may cause congestion at a down stream node.

B with a round-robin period RRP, the required guaranteed transmission time for the connection is $GTT = \lceil B_c RRP/B \rceil$. The connection can be accepted over a link if the summation of $GTTs$ over all connections passing through the link does no exceed RRP. An addition or deletion of a connection needs to change the values of T_i's and δ, which can be done at the end of a round-robin transmission period.

One problem with the above connection establishment procedure is that due to the ceiling operation (i.e., rounding up to an integer) in calculating $GTTs$, a connection may be allocated slightly different bandwidths over links with different $RRPs$. If the bandwidth allocated to a connection over an upstream link is larger than that over a downstream link, cell losses might happen [29]. Since a connection is usually established sequentially over links from its source to destination, this problem can be avoided by using the bandwidth actually allocated to a connection at an upstream link to calculate the GTT required at the next downstream link. In this way, lossless synchronous transmission is guaranteed.

3 Dynamic bandwidth sharing and lossless transmission

Algorithm 2.1 would behave very much like STM or Stop-and-Go if no asynchronous transmission were allowed in Step 3. Each connection is allocated a certain amount of bandwidth, but any unused or unallocated bandwidth is wasted. So the objective of this section is to develop an asynchronous transmission algorithm to make efficient and safe use of the remaining bandwidth. By "efficient" we mean that one should be able to transmit as many asynchronous cells as possible, and by "safe" we mean that one should not transmit too many cells as to overflow a downstream node and cause cell losses.

Fig. 2 shows an example that an uncontrolled dynamic bandwidth sharing may cause buffer overflows. Suppose that three connections, C1, C2, and C3, are each allocated one half of the link transmission bandwidth as shown in the top half of Fig. 2. No congestion will occur if each connection uses its own allocated bandwidth only. However, if at a certain time, C2 uses only one half of its allocated bandwidth and the unused half is dynamically shared by C1, then cell losses may happen due to a congestion at a downstream node as shown in the lower half of the figure. So, to completely avoid cell losses, dynamic bandwidth sharing (i.e., asynchronous

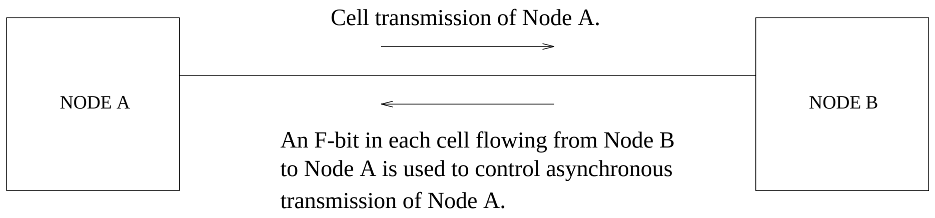
Cell transmission of Node A.
NODE A
NODE B
An F-bit in each cell flowing from Node B to Node A is used to control asynchronous transmission of Node A.

Figure 3: Feedback flow control of asynchronous transmissions.

transmission in Algorithm 2.1) should be permitted only when there is enough buffer space at a downstream node.

We propose to use a simple link-by-link feedback flow control scheme to enable or disable asynchronous transmissions of a node. As shown in Fig. 3, an F-bit in each cell flowing from a downstream node B to a current node A is used to control asynchronous transmissions of Node A (for simplicity, we assume that transmission bandwidths in both directions are the same). The F-bit is marked either as ENABLED or DISABLED. Node A is allowed to transmit asynchronous cells to Node B only when it is receiving F=ENABLED cells from node B. With this feedback flow control scheme, an appropriate F-bit setting algorithm can be used at Node B to ensure

1. no buffer overflow — disable the asynchronous transmission of node A when otherwise it will cause a buffer overflow at node B, and
2. no buffer underflow — enable the asynchronous transmission of node A when it will not cause a buffer overflow at node B.

For simplicity, we assume that a cell inserting scheme is used which transmits idle cells containing the feedback information only over otherwise idle transmission links. Thus Node A continuously receives either F=ENABLED or F=DISABLED cells from Node B. Also suppose that a buffer space of M cells is allocated at node B to store cells from node A as shown in Fig. 4 (dynamic buffer sharing among incoming links is a subject for further study). Since synchronous transmission is not control by a feedback control scheme, according to Section 2, in the worst case a buffer of RRP cells is needed to store synchronous cells, resulting a buffer of $M - RRP$ cells for asynchronous cells. Let R be the round-trip propagation delay from node B to node A. Then after node B starts marking F-bits as DISABLED, there still could be up to R asynchronous cells arriving from node A. Thus a simple F-bit setting algorithm would be to set F=DISABLED when the number of asynchronous cells reaches $M - RRP - R$ and set F=ENABLED when the number goes below $M - RRP - R$ as shown below.

Algorithm 3.1 (F-bit setting algorithm 1) .

1. *A counter C is used to record the number of asynchronous cells arrived from node A which are currently being buffered at node B.*
2. *Set $F := DISABLED$ when $C \geq M - RRP - R$, and $F := ENABLED$ otherwise.*

However, there are two problems with this simple approach,

1. Counting of asynchronous cells.
   In an ATM network, cells belonging to one connection are transmitted in a FIFO (First In

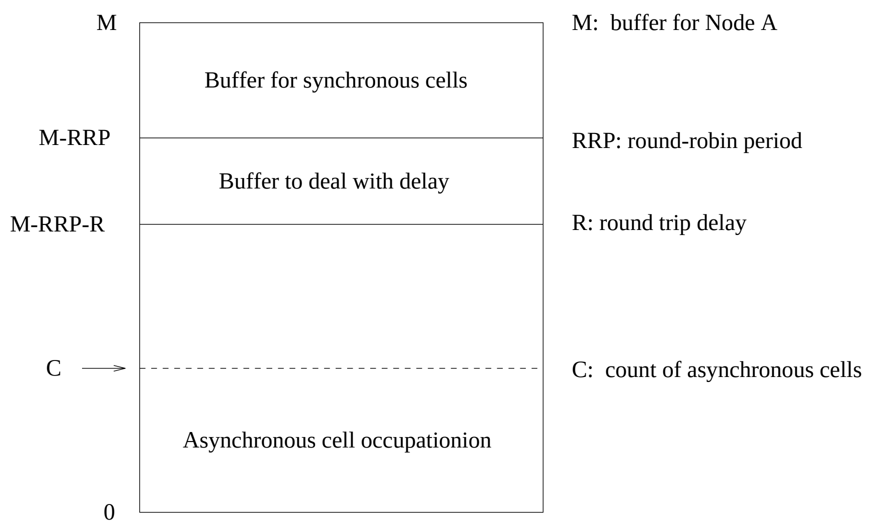
M
M-RRP
M-RRP-R
C
0
Buffer for synchronous cells
Buffer to deal with delay
Asynchronous cell occupationion
M: buffer for Node A
RRP: round-robin period
R: round trip delay
C: count of asynchronous cells

Figure 4: Buffer management at node B.

First Out) order. Thus asynchronous cells arrived from Node A may be transmitted with synchronous transmission times (i.e., GTT_i's) at Node B. This complicates the problem since it reduces the transmission time allocated to synchronous cells and may cause them to stay at Node B for a time period longer than RRP. A consequence of this is that a buffer space of more than RRP cells might be required to store synchronous cells. Thus counting asynchronous cells only can not prevent buffer overflows.

2. Buffer underflow.
   Algorithm 3.1 can not prevent buffer underflows. It is possible that node B has a free buffer of up to R cells, but no asynchronous transmission is allowed from node A. This may cause significant buffer and bandwidth waste for networks with large inter-node propagation delays. In the worst-case where $R > M - RRP$, the asynchronous transmissions will be completely blocked.

To solve the first problem, we observe that instead of doing FIFO transmission for each connection, if Node B uses a scheduling algorithm which alway transmits synchronous cells first with guaranteed transmission times and asynchronous cells first when in an asynchronous transmission mode, then C can be simply calculated as the number of asynchronous cells arrived from Node A which are currently stored in Node B. The reason for this to be true is that with this scheduling policy, no synchronous cells will experience delays longer than RRP, thus a buffer space of RRP cells is enough to accommodate all synchronous cells from Node A. Consequently, one only needs to make sure that the number of asynchronous cells in Node B does not exceed M-RRP-R.

However, the above described scheduling algorithm can not be actually used since ATM requires to maintain a cell transmission order for each connection. In other words, the FIFO scheduling has to be used for cells belonging to the same connection. We solve this problem by "swapping" cell types. Specifically, when Node B transmits an asynchronous cell of connection i with the guaranteed transmission time GTT_i and there is another synchronous cell of connection i in Node B, the types of these two cells are "swapped" by treating this situation as if a synchronous cell were transmitted and the asynchronous cell still remained in the buffer. This idea is detailed in the following algorithm:

Algorithm 3.2 (Management of counter C) .

1. *In addition to using a counter C to record the number of asynchronous cells, a counter S_i is used to record the number of synchronous cells of the connection i, $i = 1, \cdots, n$.*

2. *C is increased by 1 when an asynchronous cell arrives from Node A, and S_i is increased by 1 when a synchronous cell of connection i arrives from Node A.*

3. *When Node B transmits a cell of connection i with the guaranteed transmission time GTT_i, then*

   (a) *if the cell is a synchronous one, decrease S_i by 1,*

   (b) *if the cell is an asynchronous one, decrease S_i by 1 if $S_i > 0$, otherwise, decrease C by 1.*

4. *When Node B transmits a cell of connection i with the asynchronous transmission algorithm (i.e., Step 3 in Algorithm 2.1), then*

   (a) *if the cell is an asynchronous one, decrease C by 1,*

   (b) *if the cell is a synchronous one, decrease C by 1 if $C > 0$, otherwise, decrease S_i by 1.*

To solve the buffer underflow problem, we need an algorithm which does not always set F=DISABLED when $C \geq M - RRP - R$. The reason for setting F=DISABLED when C reaches $M - RRP - R$ in Algorithm 3.1 is that due to a round trip propagation delay R, node B may receive up to R asynchronous cells from node A after it sets F=DISABLED. However, a further observation is that node B will not receive any asynchronous cells from node A after it has set F=DISABLED for R units of time. So if the value of C at this time is not larger than $M - RRP$, node B can safely send out $M - RRP - C$ cells with F-bits set to ENABLED to node A which would allow node A to transmit up to this number of asynchronous cells. This idea can be realized with the following F-bit setting algorithm.

Algorithm 3.3 (F-bit setting algorithm 2) .

1. *A counter C is maintained using Algorithm 3.2.*

2. *Let t be the current time, t_{de} and t_{ed} be the last time that F changes its value from DISABLED to ENABLED and from ENABLED to DISABLED, respectively. Let C_{de} be the value of C at time t_{de}. Then the F-bit is set as follows:*

   Step 1: *If $C < M - RRP - R$, set $F := ENABLED$. Otherwise, go to Step 2.*

   Step 2: *If $C < M - RRP - R$ at time $t - 1$ (i.e., C is increased to $M - RRP - R$ at t), set $t_{ed} := t$ and $F := DISABLED$. Otherwise, go to Step 3.*

   Step 3: *If $t < t_{ed} + R$, set $F := DISABLED$. Otherwise, go to Step 4.*

   Step 4: *If $C = M - RRP$, set $F := DISABLED$. Otherwise, go to Step 5.*

   Step 5: *If $F = DISABLED$ at time $t-1$, set $t_{de} := t$, $F := ENABLED$, and $C_{de} := C$. Otherwise, go to Step 6.*

   Step 6: *If $t < t_{de} + M - RRP - C_{de}$, set $F := ENABLED$. Otherwise, set $F := DISABLED$ and $t_{ed} := t$.*

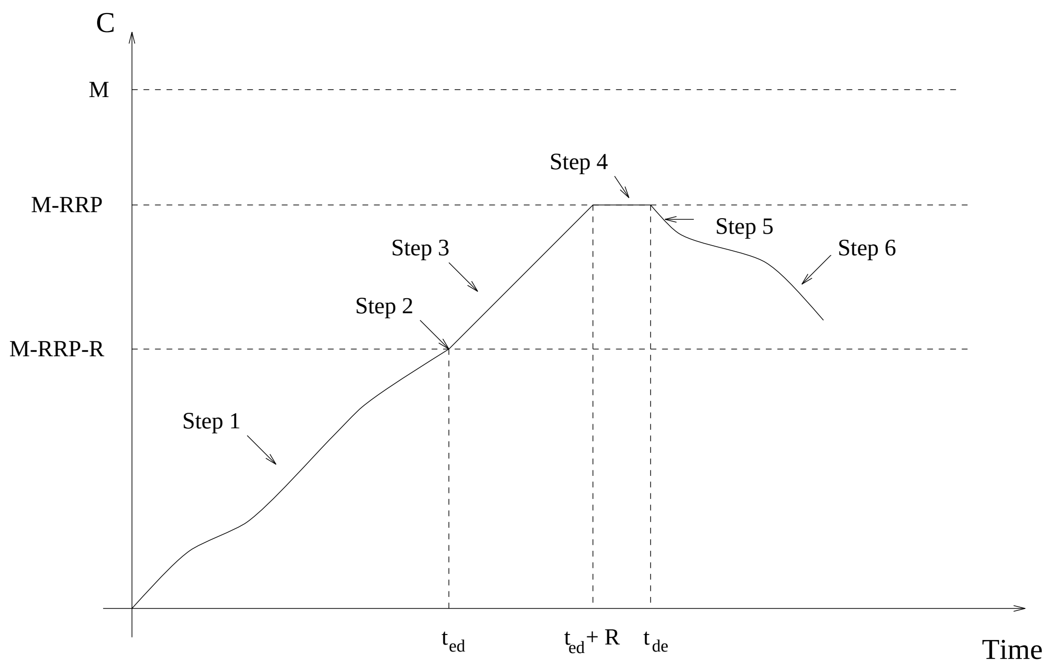

Figure 5: Operation of Algorithm 3.3.

The above algorithm is illustrated in Fig. 5. When C is below $M - RRP - R$, it is always safe to allow node A to send asynchronous cells. So in Step 1, cells sending from node B to node A are marked with F=ENABLED. Step 2 detects the time when C is increased to $M-RRP-R$ and sets t_{ed} and F=DISABLED to stop asynchronous transmission of Node A. However, due to the round trip propagation delay R, node B may still receive asynchronous cells from Node A. So Step 3 continues to set F=DISABLED. In the worst case, up to R extra asynchronous cells arrive at node B which may make C reach $M - RRP$ at time $t_{ed} + R$. Step 4 deals with this situation by continuing setting F=DISABLED. Step 5 detects the time when F can be set to ENABLED and sets t_{de}, C_{de}. Then, up to $M - R - C_{ed}$ cells can be set with F=ENABLED (Step 6). If $C \geq M - RRP - R$ after that, F is set to DISABLED and t_{ed} to the current time, which will make F be set to DISABLED for R units of time in Step 3 if C keeps above the $M - RRP - R$ level.

Algorithm 3.3 is still conservative since it is assumed that up to R extra asynchronous cells may arrive from Node A after t_{ed} (i.e., when F is changed from ENABLED to DISABLED). This is not always true. For example, under certain conditions, only up to $M - RRP - C_{de}$ extra asynchronous cells may arrive after F is set to DISABLED in Step 6. So it is possible to develop a more efficient F-bit setting algorithm than Algorithm 3.3 at an extra implementation cost.

Another issue about the asynchronous transmission algorithm is the transmission scheduling policy to be used while Node A is receiving F=ENABLED cells from node B. Stating in another way, when Node B allows node A to do asynchronous transmission, what cells should Node A pick up to transmit? The answer to this question depends on the desired performance of asynchronous transmission. If the fairness among connections is required, then a pure round-robin scheduling policy can be used. Also, priority or timed-round-robin scheduling can be used to distinguish importance between connections. We are not going to elaborate further on this issue in this paper.

We conclude this section with some comparisons of our feedback flow control scheme with others. Traditionally, telecommunication people had been quite reluctant to accept the concept of feedback flow control since it is in general not needed for constant bit-rate traffics. However, with the introduction of ATM and the requirement for supporting variable bit-rate traffics, more and more people have realized that it would not be possible to achieve a satisfactory network utilization without a feedback flow control mechanism. Many schemes for implementing such a mechanism have been proposed [21, 22, 24, 23], and among them the most similar ones to ours are the link-by-link per-connection credit-based feedback flow control schemes [21, 22]. Comparing with these schemes, the one proposed in this paper has been significantly simplified in the following two ways:

1. Per-link instead of per-connection feedback control.

   [21, 22] used a per-connection feedback control scheme to avoid a problem that one connection may block transmissions of all other connections over a link. Per-connection control means that each connection must maintain its own credit count which is updated using the information transmitted individually from a downstream node. Also, the buffer space is statically allocated to each connection. A consequence of this is the complexity of the credit management mechanism and the requirement for a large buffer to achieve dynamic bandwidth sharing.

   With a timed-round-robin cell transmission scheme, each connection is guaranteed a certain amount of minimum bandwidth and no connections will be blocked completed. Thus we can use a *per-link* feedback control scheme which treats all connections over a link as a single one. This significantly simplifies the feedback mechanism and reduces the buffer requirement.

2. State-based instead of credit-based feedback control.

   Another simplification which has been made is the usage of just 1 bit in a feedback cell to indicate the current buffer occupation state (instead of the credit count value) at a downstream node, which makes it possible to encode the feedback information in regular cells flowing from a downstream node to a current node and relieves a node from the burden of creating special feedback cells. Also, letting regular cells carry feedback information enables us to realize continuous feedback control by using a cell insertion scheme.

The above two simplifications make our scheme easier to implement and requiring smaller buffer space than that of [21, 22] while maintaining a comparable bandwidth sharing capacity.

4 Implementation

In this section, we present an implementation of the proposed traffic control scheme in ATM switches. Our objective is to show that our scheme can be implemented without adding much complexity to the existing switches. For this purpose, only an architectural design is given and many details are left out.

For simplicity, we assume a shared-buffer ATM switch as shown in Fig. 6. Incoming cells are multiplexed and stored in a Shared Buffer Memory (SBM) which are later demultiplexed to their corresponding outgoing links and transmitted to their next nodes. A Control Unit (CNT) provides idle addresses in the SBM to the multiplexor (MUX) to store incoming cells.

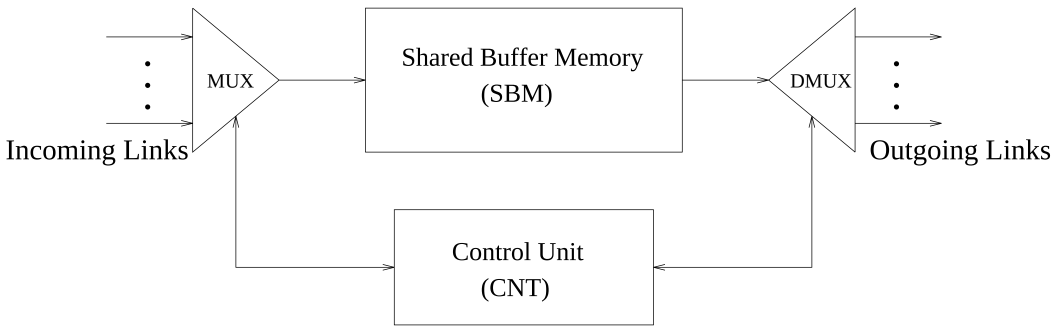

Figure 6: A shared-buffer ATM switch architecture.

The CNT also provides addresses of cells stored in the SBM to the demultiplexor (DMUX) for cell transmissions. This shared-buffer switch architecture has been used in many existing ATM switches.

A key requirement to implement a round-robin cell transmission scheme is to maintain a FIFO cell queue for each connection. One way of doing this is to use a linked-list queueing structure as shown in Fig. 7. A tag is added to each cell stored in the SBM which serves as a pointer to the location of the next cell in a queue. The addresses of the first and last cells of queues are stored in a Connection Lookup Table (CLT) in CNT. The architecture of CNT is shown in Fig. 8 which manages the cell queues and schedules cells to be transmitted in the

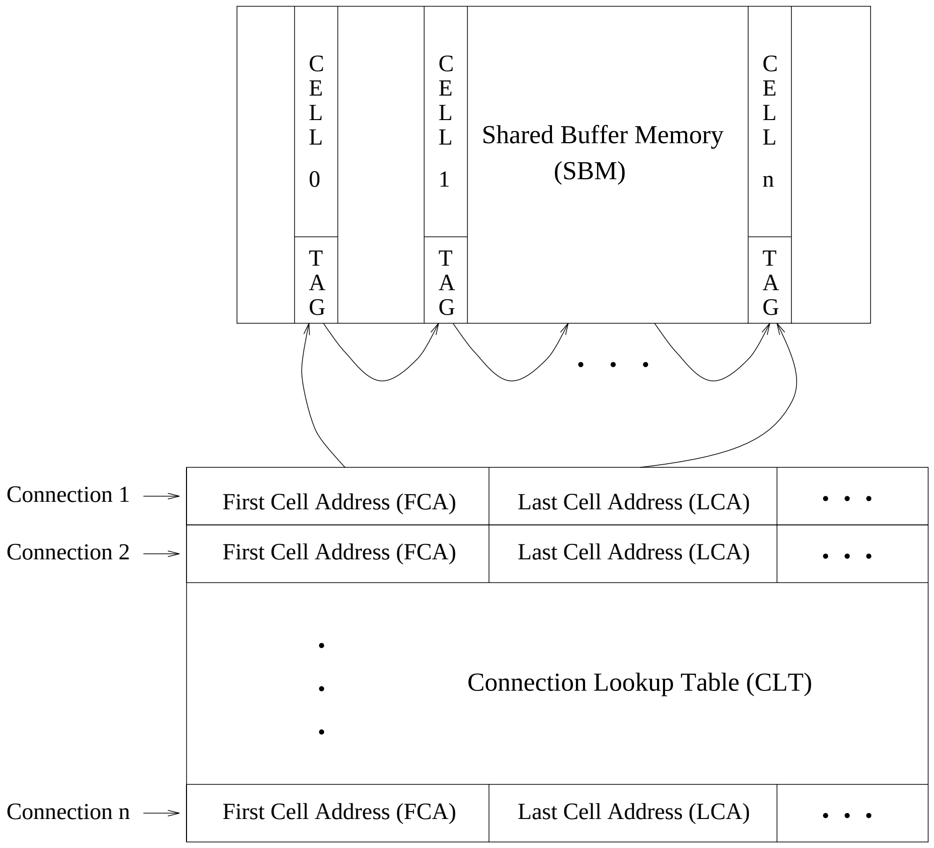

Figure 7: A cell queue realized in a linked list data structure.

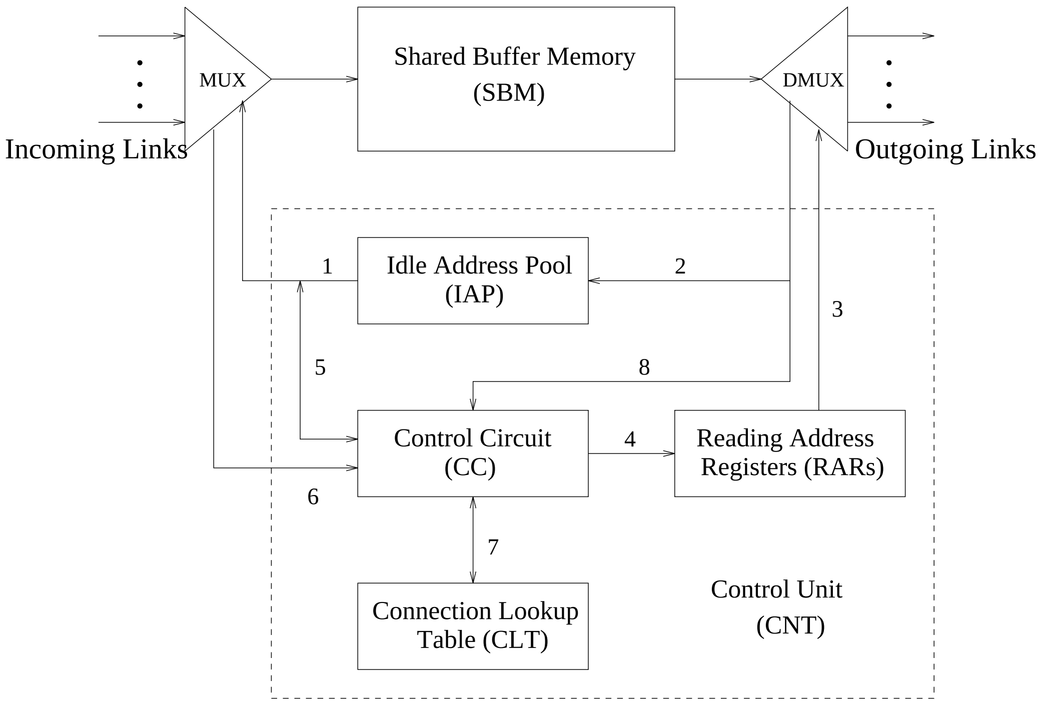

Figure 8: Architecture of CNT.

following way:

1. An Idle Address Pool (IAP) is used to store idle addresses in the SBM where incoming cells can be stored. One address is allocated from IAP to each incoming cell (see path 1 in Fig. 8). When a cell is transmitted, its address in SBM is returned to IAP (path 2).

2. One Reading Address Register (RAR) is used for each outgoing link which provides the address of the cell in SBM which should be transmitted next (path 3). RARs are updated by a Control Circuit (CC) (path 4) according to the cell transmission scheduling algorithm used.

3. A Connection Lookup Table (CLT) is used to store information for each connection. The information useful for implementing the proposed traffic control scheme includes: addresses of the first and last cells of cell queues, allocated Guaranteed Transmission Times GTT_i's, synchronous cell counts S_i's, pointers to the next and last connections to realize a desired round-robin order, etc.. The CLT can be constructed as an individual component, or more conveniently, be combined with the existing virtual-circuit routing tables in ATM switches.

4. Management of cell queues is done as follows:

   - When a cell arrives at an incoming link, an Idle Address (IA) is read from the IAP where the cell is to be stored. This IA and the header of the incoming cell which contains the connection identification number are also forwarded to a Control Circuit (CC) (paths 5 and 6). If the First Cell Address (FCA) of the connection in the CLT is not set (meaning the queue is empty), both FCA and Last Cell Address (LCA) are set to be IA (path 7). Otherwise, only LCA is set to be IA and the original value

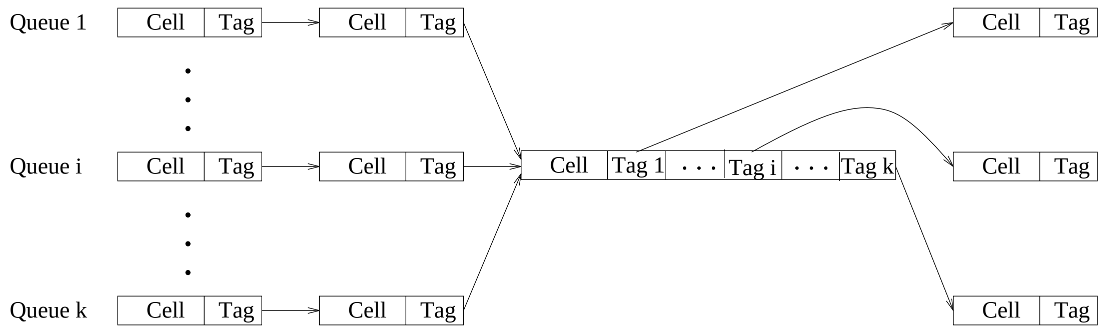

Figure 9: An enhanced linked-list data structure.

of LCA is sent to MUX so that the tag of the cell stored at ALC can be updated to be IA (paths 7 and 5).

- When a cell is transmitted, the FCA field should be updated to the address of the next cell in the queue. This is done by writing the tag of the cell transmitted into the the FCA field (paths 8 and 7). To reduce the frequency that CLT is accessed, this operation only needs to be performed at a connection switching time (i.e., when a transmitter stops transmitting cells of one connection and switches to another connection).

Once a FIFO cell queue is maintained for each connection, it would be straightforward to design a CC to implement the proposed timed-round-robin cell transmission algorithm (Algorithm 2.1). Since time is measured in a unit of one cell transmission time, a counter can be used to act as the round-robin timer RRT for each outgoing link. Let $Cell_0$ be the cell currently being transmitted, and $Cell_1$ the next cell to be transmitted. According to the rules specified in Algorithm 2.1, CC performs one of the following three actions to provide the address of $Cell_1$, denoted by p_1, to RARs:

1. If $Cell_1$ and $Cell_0$ belong to the same connection, set $p_1 :=$ tag of $Cell_0$.
2. If $Cell_1$ belongs to a connection C_i which is different from that of $Cell_0$, set $p_1 := FCA$ of connection i which is stored in the CLT.
3. If no cell is to be transmitted next, set p_1 to be the address of an idle cell stored in SBM which carries feedback information only (i.e., cell insertion).

Implementation of the proposed feedback flow control scheme is also easy. One bit in an ATM cell header is needed to serve as the F-bit and another one to distinguish a synchronous cell from an asynchronous one. Since the usage of three Generic Flow Control (GFC) bits in the ATM cell header has not been defined, we assume that two of them can be used for our purpose. The values of these two bits are determined by the CC according to Algorithm 2.1 and Algorithm 3.3 which are then transferred to the DMUX via paths 4 and 3 in Fig. 8, and inserted into a cell at the time when it is transmitted.

From the above discussion, we see that with the usage of a linked-list queueing structure, the proposed traffic control scheme can be easily implemented in an exiting ATM switch architecture. However, one problem with the conventional linked-list structure is its inability to support multicast transmissions in a shared-buffer switch where one cell may appear in several queues [30]. We propose to solve this problem by using an enhanced linked-list data structure

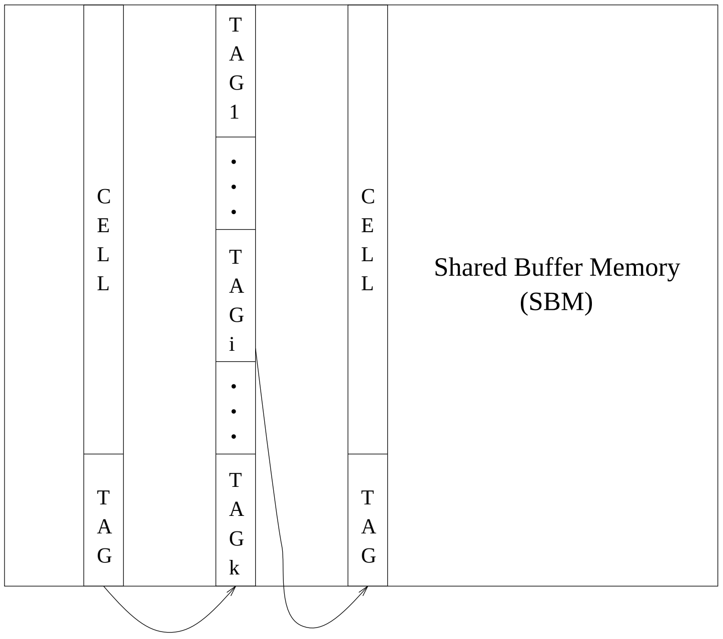

Figure 10: Implementation of the enhanced linked-list data structure in the SBM.

as shown in Fig. 9 where a cell belonging to multiple queues uses multiple tags each pointing to the next cell of a queue. By associating each tag with a queue either explicitly (e.g., putting a queue ID in a tag) or implicitly (e.g., the ith tag is for ith queue), the enhanced linked-list would be able to handle multicast cells. Also, if one clear tag i when the cell has been transmitted in queue i, it would be easy to determine when a multicast cell can be removed from the buffer by checking whether or not all tags have been cleared.

Implementation of the enhance linked-list queueing structure in the SBA can be done in one of the following two ways,

1. Enlarge the tag field.
   One can make the tag field of every cell large enough to hold the maximum number of pointers (i.e., the number of ports of a switch). For example, for a 16×16 switch with an SBM capable of storing 64K cells (i.e., 2 bytes addressing space), the tag field of a cell needs to be 2 bytes $\times$ 16 ports $=$ 32 bytes long which represents a $32/53 = 60\%$ of buffer overheads.

2. Use an indirect addressing scheme.
   Since not every cell is a multicast one, one can reduce the buffer overheads by making the tag field of a cell just large enough to hold one pointer (2 bytes in the above example). For a unicast cell, its tag directly points to the next cell in a queue as shown in Fig. 7. For a multicast cell, this tag is served as an indirect pointer to a place where actual pointers are stored as shown in Fig. 10. This approach can reduce the average buffer overheads, but requires a faster SBM since it needs two buffer accesses to transmit a multicast cell.

Finally, it should be noted that the enhance linked-list structure is also useful for fault-tolerance since multiple tags can be used to serve as redundant pointers to the next cell in a queue.

5 Conclusions

We proposed in this paper a traffic control scheme which can simultaneously realize (1) guaranteed bandwidth transmission, (2) dynamic bandwidth sharing, (3) lossless transmission, and (4) easy implementation, by using an innovative timed-round-robin cell transmission scheduling algorithm enhanced with a simple feedback flow control mechanism. The proposed scheme provides ATM with an ability of supporting real-time traffic and bursty data traffic in an integrated manner which makes it qualify as a true enabling technology for the future integrated digital networks.

Acknowledgement

The author would like to thank Dr. John Howard of Mitsubishi Electric Research Laboratories for his careful review of the paper and many valuable comments and suggestions on possible extensions of the results presented in this paper.

References

[1] M. Prycker, *Asynchronous transfer mode: solution for broadband ISDN*, Ellis Horwood Limited, Chichester, West Sussex, PO191EB, England, 1991.

[2] H. Kondoh, H. Notani, H. Yamanaka, K. Higashitani, H. Saito, I. Hayashi, S. Kohama, Y. Matsuda, K. Oshima, and M. Nakaya, "A 622-Mb/s 8X8 ATM switch chip set with shared multibuffer architecture," *IEEE Journal of Solid-State Circuits*, vol. 28, no. 7, pp. 808–815, July 1993.

[3] T. Kozaki, N. Endo, Y. Sakurai, O. Matsubara, M. Mizukami, and K. Asano, "32 X 32 shared buffer type ATM switch VLSI's for B-ISDN's," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 8, pp. 1239–1254, October 1991.

[4] H. J. Chao, "A recursive modular Terabit/second ATM switch," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 8, pp. 1161–1172, October 1991.

[5] T. R. Banniza, G. J. Eilenberger, B. Pauwels, and Y. Therasse, "Design and technology aspects of VLSI's for ATM switches," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 8, pp. 1255–1264, October 1991.

[6] P. Strauss, "ATM WANs: rocket science or lost in space," *Datamation*, pp. 20–28, January 1994.

[7] C.-T. Lea, "What should be the goal for ATM," *IEEE Network*, pp. 60–66, September 1992.

[8] The ATM Forum, "ATM User-Network Interface Specification,", September 1993. Version 3.0.

[9] M. Decina and V. Trecordi, "Traffic Management and Congestion Control for ATM Networks," *IEEE Network*, Sept. 1992.

[10] J. J. Bae and T. Suda, "Survey of traffic control schemes and protocols in atm networks," *Proceedings of the IEEE*, vol. 79, no. 2, pp. 170–189, February 1991.

[11] S. J. Golestani, "A stop-and-go queueing framework for congestion management," in *Proc. SIGCOMM Symposium*, pp. 8–18. ACM, September 1990.

[12] J. S. Turner, "Manageing bandwidth in ATM networks with bursty traffic," *IEEE Networks*, pp. 50–58, September 1992.

[13] H. Zhang and S. Keshav, "Comparison of Rate-Based Service Discipline," *Proceedings of ACM SIGCOMM, Zurich, Switzerland*, pp. 113–121, Sept. 1991.

[14] D. Ferrari and D. C. Verma, "A scheme for real-time channel establishment in wide-area networks," *IEEE Journal on Selected Areas in Communications*, vol. SAC-8, no. 3, pp. 368–379, April 1990.

[15] Q. Zheng and K. G. Shin, "On the ability of establishing real–time channels in point–to–point packet–switched networks," *IEEE Transactions on Communication* (in press), March 1994.

[16] L. Zhang, "Virtual Clock: A new traffic control algorithm for packet-switched networks," *ACM Trans. Computer Systems*, vol. 9, no. 2, pp. 101–124, May 1991.

[17] A. K. J. Parekh, *A generalized processor sharing approach to flow control in integrated services networks*, PhD thesis, Massachusetts Institute of Technology, February 1992.

[18] C. R. Kalmanek, H. Kanakis, and S. Keshav, "Rate controlled servers for very high-speed networks," in *Proceedings of IEEE Globecom*, December 1990.

[19] M. Sidi, W. Z. Liu, I. Cidon, and I. Gopal, "Congestion control through input rate regulation," in *Proceedings of IEEE Globecom*, December 1989.

[20] H. J. Chao, "Architecture design for regulating and scheduling user's traffic in ATM networks," in *Proc. of ACM SIGCOMM 92*, pp. 77–87, 1992.

[21] H. T. Kung and A. Chapman, "The FCVC (flow-controlled virtual channels) proposal for ATM networks," Version 2.0, 1994.

[22] H. T. Kung, R. Morris, T. Charuhas, and D. Lin, "Use of link-by-link flow control in maximizing atm network performance: Simulation results," in *Proceedings of IEEE Hot Interconnects Symposium, '93*, August 1993.

[23] N. Yin and M. G. Hluchyj, "On closed-loop rate control for ATM cell relay netorks," in *Proceedings of IEEE INFOCOM'94*, June 1994.

[24] P. Newman, "Backward explicit congestion notification for ATM local area networks," in *Proceedings of IEEE Globecom'93*, December 1993.

[25] H. J. Chao and N. Uzun, "A VLSI sequencer chip for ATM traffic shaper and queue manager," *IEEE Journal of Solid-State Circuits*, vol. 27, no. 11, pp. 1634–1643, November 1992.

[26] *Fiber Distributed Data Interface (FDDI) – Token Ring Media Access Control (MAC).* American National Standard, ANSI X3.139, 1987.

[27] F. E. Ross, "An overview of FDDI: The fiber distributed data interface," *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 7, pp. 1043 – 1051, September 1989.

[28] R. Osborne, “A direct deposit model for low latency messaging,” Technical Report MN94-04, Mitsubishi Electric Research Laboratories, Cambridge Research Center, February 1994.

[29] J. Howard Mitsubishi Electric Research Laboratories, Personal communication, March 1994.

[30] H. Yamanaka Mitsubishi Electric Corporation, Personal communication, December 1993.