REAL-TIME COMMUNICATION IN ATM NETWORKS

Qin Zheng[†], Kang G. Shin[‡], and Chia Shen[†]

[†]Mitsubishi Electric Research Labs., Inc.
Cambridge Research Center
201 Broadway
Cambridge, MA 02178

[‡]Real-time Computing Laboratory
EECS Department
The University of Michigan
Ann Arbor, Michigan 48109

Abstract

The asynchronous transfer mode (ATM) is expected to be the multiplexing and switching technique for the future broadband integrated service digital networks (BISDNs) which can transport almost all types of traffic including bursty data traffic and continuous voice/video frames. However, this capability cannot be realized without a proper congestion-control scheme. This paper discusses the suitability of the real-time channel approach [1] for congestion control in ATM networks. A main contribution of the paper is the development of a new message-transmission scheme for real-time channels which does not require each cell to carry deadline information which is difficult to do with the standard ATM cell format. This makes our scheme compatible with, and implementable in, ATM networks. The new scheme also has a number of advantages over existing ones. The achievable quality of service with the proposed approach would make ATM networks capable of supporting both real-time services and dynamic bandwidth sharing which have been promised but not yet fully realized.

1 Introduction

ATM networks are expected to become the next generation of digital communication networks, and provide high-quality and flexible communication services both for wide-area information transportation over BISDNs and for the basic infrastructure of computer networks and intra-device connections [2–4]. High-quality communication will enable the network to support real-time services such as voice/video transmissions, and flexible communication will improve efficiency in supporting heterogeneous services that require different transmission bandwidths and qualities by statistically sharing network resources.

Two key techniques are used for ATM networks to achieve the above goal: packet-switched transmission and fast packet switching. Packet-switched transmission makes it efficient for a network to transport varying bit-rate traffic since many users can dynamically share a transmission link. Fast packet switching reduces the message-transmission delay by implementing switching functions in hardware. However, packet-switched transmission introduces ATM networks a congestion-control problem. Unlike with circuit-switched transmission where each user is guaranteed to have a certain bandwidth, the transmission bandwidth available to each user in a packet-switched network depends on the extant network traffic. When the traffic load is high, a network congestion may occur and packets may experience unacceptably long delays or may even be lost.

Two types of congestion-control schemes have been proposed: *reactive* control and *preventive* control. With reactive control, when a traffic congestion occurs the source nodes are instructed to throttle their traffic flow by giving feedback to them. A major problem with reactive control in high-speed networks is slow feedback. By the time that feedback reaches the source nodes and the correponding control is triggered, it may already be too late to react effectively to the congestion.

Preventive control, on the other hand, does not wait until a congestion actually occurs, but rather strives to prevent the network from reaching an unacceptable level of congestion. A common approach is to con-

*The work reported in this paper was supported in part by the Office of Naval Research under Grant No. N00014-92-J-1080 and the National Science Foundation under Grant No. MIP-9203895. Any opinions, findings, and recommendations expressed in this publication are those of the authors, and do not necessarily reflect the views of the funding agencies.

0-8186-6680-3/94 $04.00 © 1994 IEEE

trol traffic flow at entry points of a network, which is realized with *admission control* and *bandwidth enforcement*. Admission control determines whether or not to accept a new connection request at the time of call setup. This decision is based on the traffic characteristics of the new connection and the current network load. Bandwidth enforcement monitors individual connections to ensure that the actual traffic flow conforms to that specified at the time of connection establishment.

Considerable attention has been paid to the issue of congestion control in packet-switched high-speed networks [1,5–11]. The congestion-control strategies proposed in [1,5,8,9] aim at providing the transmission-delay bounds for applications, such as video and audio transmissions, that have stringent timing requirements in the face of other types of traffic. These congestion-control schemes can be categorized into two classes. One class of schemes deals with diverse real-time traffic mixture, and different bit-rate requirements of applications [1,9], while the other guarantees only one (or a small fixed number of) real-time traffic-delay requirement(s) [5,8]. The latter can use fixed-priority schemes with only two or three priority levels for scheduling packet transmissions at the switch, while the former must deal with potentially a very large set of delay bound requirements.

In [5,8], the framing strategy is employed. Together with the Stop-and-Go transmission scheme, this strategy provides bounded delays for packet transmission. Time is divided into *frames* which consist of a fixed number of *slots*, where a slot is the time required to transmit one packet. Delay-constrained packets from different incoming links during an arriving frame will be transmitted during a single departing frame, but not during any other time intervals. If a packet fails to find an empty slot during the output frame, it will be dropped. In this way, the delays of future packets are bounded by the maximum separation between the arriving frame and the departing frame. Non-real-time packets are serviced as best-effort traffic using any idle slots left in each departing frame. One major problem associated with the framing scheme is that it cannot prevent ill-behaving channels with real-time packets to flood the network, and thus causes other channels to drop packets. Another problem of the framing and Stop-and-Go strategy is the coupling between the delay bound that can be guaranteed and the bandwidth allocation granularity [5,11]. Specifically, since the delay bounds are a function of the frame size, a smaller frame size would provide lower delay bounds, but entails a larger granularity in bandwidth reservation. Lastly, with one frame size, only one delay bound is provided. The problems mentioned above make this congestion-control strategy unsuitable for networks where real-time communication requires a variety of delay bounds and bandwidth-allocation granularity.

The support for statistical multiplexing with the ATM to accommodate a wide variety of applications requires suitable congestion-control mechanisms. Schemes for admission control and bandwidth enforcement are not specified in the ATM standards. In spite of the considerable research efforts over the last several years, such schemes are still not well understood [2]. To alleviate this problem, we propose a congestion-control scheme based on the concept of real-time channel [1]. With this scheme, a network guarantees the timely transmission of real-time messages and handles non real-time messages on a best-effort basis. Another advantage of the scheme is that it integrates admission control and bandwidth enforcement so as to eliminate the need of a separate mechanism for bandwidth enforcement. Moreover, with the proposed scheme, messages need not carry any timing information, neither in the form of timestamps nor deadlines. Specifically, channel parameters can be recorded in the ATM switch VCI/VPI table and cell deadlines are calculated at cell arrival time at a switch. Although the schemes proposed in [1,9,12] also support real-time channels, it is different from the one described in this paper. First of all, the scheme of [1] assumes bandwidth enforcement by the source nodes, and that of [9] uses bandwidth enforcement by both the source nodes and intermediate node switches. Second, the schemes proposed in [1,9,12] require that each packet carry extra timing information, which causes ATM networks due to their standardized short cell format.

The paper is organized as follows. For completeness Section 2 reviews the basic concept and characteristics of real-time channels. Implementation of real-time channels in an ATM network is discussed in Section 3 where we propose a new message-transmission protocol. The protocol is suitable for ATM networks and has a number of advantages over existing ones. The paper concludes with Section 4.

2 Real-time Channels

Messages in a digital communication network can be categorized into two classes: real-time and non real-time. Real-time messages are required to be delivered to their destinations within some pre-specified delay

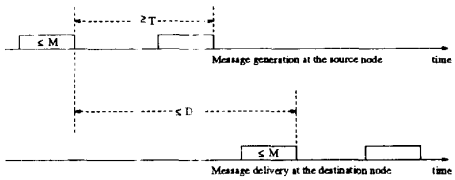

Figure 1: Parameters of a real-time channel.

bounds. Those messages not delivered within these delay bounds will be discarded, thus deteriorating quality of service. Non real-time messages, on the other hand, do not have tight-delay requirement. Loss of a certain percentage of packets due to network congestion can be tolerated by employing a proper packet-retransmission policy.

One way to guarantee the timely delivery of real-time messages over packet-switched networks is to use the concept of real-time channel [1, 13, 14]. A *real-time channel* is defined as follows [14].

Definition 2.1 (Real-time channel) .
A real-time channel, described by a three tuple (T, M, D), is a unidirectional connection between a source and a destination which guarantees that every message generated at the source will be delivered to the destination sequentially and in a time period no greater than D if the following two conditions are satisfied: (C1) the message inter-generation time is not smaller than T, and (C2) the message size does not exceed M.

The parameters of a real-time channel are graphically shown in Fig. 1. The real-time channel service is especially useful for real-time applications where the traffic characteristics are known in advance, e.g., voice/video transmissions. Consider the full motion video transmission as an example. A source node generates 30 frames per second, so the message inter-generation time $T = 33$ ms. The maximum message size is also known in advance, which is a function of the pixel size of a video frame. The third parameter of a real-time channel, D, represents the user-specified quality of service, i.e., the maximum acceptable delay that a video frame may experience over the channel. A small D, say $D = T = 33$ ms, is desirable for interactive video communications and a larger D may be acceptable for non interactive communications if a sufficient buffer space is provided at the destination.

For real-time applications where the traffic characteristics cannot be pre-determined exactly, estimated values of T and M can be used. During a transmission process, the source node is allowed to generate messages faster than T or larger than M at the risk that these messages may not be delivered within a delay bound D or may even be lost. However, the channel protection feature of real-time channels will guarantee that the violation of the traffic specification of one channel will not affect the normal operations of the other channels.

Non real-time messages are given lower transmission priority than real-time messages, so their presence will not affect the quality of service guaranteed for real-time channels. Usually, a network does not guarantee timely and/or reliable delivery of non real-time messages, thus needing a sophisticated transport layer protocol like TCP to retransmit packets which are lost or delayed for a period longer than a certain threshold. It should be noted, however, that some traditional non real-time services, such as TELNET and FTP, can also be accomplished with real-time channels. Using a real-time channel, the performance of TELNET could be improved by bounding the response time. Implementing FTP with real-time channels can facilitate the bandwidth management of connections.

As compared to the circuit-switched transmission, real-time channels have the advantages of flexibility and efficiency. In a network which provides real-time channel services, users can set up channels with bandwidth, specified by T and M, and delay bound D tailored to their applications. This flexibility in establishing different types of channels allows for the better management of network resources and reduces the users' costs if they are charged for their connections. It is in sharp contrast to the conventional circuit-switched transmission where users do not have many options on the bandwidth and quality (e.g., the delay bound D) of the circuits. The packet-switched transmission of a real-time channel also makes it efficient in supporting non real-time traffic. Unlike in a circuit-switched network where the bandwidth of a circuit is reserved exclusively for one user only, the bandwidth of a real-time channel can be shared by other users when the source of the channel is not using up all the reserved bandwidth.

Implementation of the real-time channel concept relies on (i) admission control via channel establishment and (ii) deadline scheduling of packet transmissions. Admission control requires the client to establish a real-time channel before starting message transmission. A real-time channel request may be accepted or rejected, depending on the current network-load condition. Admission control is necessary because message-delay bounds cannot be guaranteed without

controlling the traffic of the network.

Messages are split into packets which are then transmitted individually. Packet transmissions are scheduled as follows. Real-time packets are given priority over non real-time packets. Each real-time packet is assigned a deadline over each link it traverses which is calculated from the user-requested end-to-end delay bound and the packet's generation time. When several real-time packets contend for use of the same link, the packet with the earliest deadline is transmitted first. As will be discussed further in Section 3.1, deadline scheduling can minimize queueing delays and provide protection between channels.

Two protocols are needed to implement the above two techniques: a channel establishment protocol and a message transmission protocol [12, 14]. The channel establishment protocol handles requests for establishing real-time channels. It checks whether or not the requested end-to-end message delivery-delay bound can be guaranteed for a real-time channel under the current network-load condition. A channel-establishment request is granted only if the requested delay bound can be guaranteed. The message transmission protocol implements the deadline scheduling of message transmissions. It specifies how a message is divided into packets, and how deadlines of a packet over the links it traverses are calculated. Interested readers are referred to [12, 14, 15] for more details.

3 Real-time Communication in ATM Networks

ATM bears many similarities to real-time channels. Both of them are connection-oriented to provide a proper quality of service and use statistical packet multiplexing to achieve high transmission efficiency. However, the procedures used by real-time channels for admission control and message transmission are more sophisticated than those for the ATM in order to achieve *guaranteed* end-to-end delay bounds. In this section we will discuss how the approach of real-time channel can be applied for congestion control in ATM networks to achieve a guaranteed quality of service.

3.1 Transmission Scheduling

Messages are split into cells of 53 bytes each in an ATM network. Though the approach of real-time channel does not require cells to have the same size, a 53 bytes constant-size cell assumption will be used in the paper to make our results consistent with the ATM standards.

One of the reasons for ATM networks not being able to *guarantee* the timely delivery of messages and *completely* avoid cell losses is due to the First-In-First-Out (FIFO) (possibly with a few static priority levels) scheduling policy used in ATM switches for cell transmissions. With FIFO scheduling, all cells are treated equally. Thus, the quality of service provided to one user depends on the amount of traffic generated by others. A few malicious users could easily saturate a large number of transmission links and affect the normal transmission of all other users using these links. Priority scheduling cannot solve the problem because the quality of service provided to one user is affected by all other users with the same or higher priorities.

As discussed in the Introduction, one way to solve this problem is to use bandwidth enforcement at the network entry points. Users declare their traffic characteristics at the connection setup time, and bandwidth enforcement mechanisms ensure that they are honored. However, this scheme has some disadvantages as follows.

1. Bandwidth enforcement increases the complexity of a system. Basically, what bandwidth enforcement does is to hold early messages at the source node and release them when they become eligible for transmission. This is difficult to implement on high-speed networks, especially for nodes from which many traffic streams originate.

2. Bandwidth enforcement could become very inefficient. Violation of traffic specification by one user does not necessarily implies a network congestion. So, holding the extra traffic outside a network may be unnecessarily conservative and waste network resources. In other words, bandwidth enforcement reduces the ATM's ability of accommodating varying bit-rate traffic which has been claimed to be one of the main advantages of the ATM over the traditional Synchronous Transfer Mode (STM).

3. The underlying FIFO scheduling of cell transmissions is not suitable for heterogeneous real-time communications. With FIFO scheduling, it is difficult and inefficient to accommodate real-time services which have different bandwidth and delay requirements.

We propose to use the deadline scheduling for cell transmissions at ATM switches. The advantages of using a deadline scheduling policy are as follows.

1. Minimal effects of queueing delays: The deadline scheduling policy can minimize the effects of

queueing delays in the sense that given a set of cells with deadlines, if they are schedulable under any scheduling policy (i.e., every cell can be transmitted before its deadline), so can they under the deadline scheduling policy [16]. Thus, the deadline scheduling policy gives a communication network more capacity to accommodate real-time traffic than that with other scheduling policies. In other words, the deadline scheduling reduces the probability of unnecessarily rejecting connection setup requests.

2. Available admission-control schemes: A rich deadline scheduling theory has been established from which the admission-control schemes for real-time channels can be derived [14, 15]. This theory can be used directly for ATM networks from which the *guaranteed* quality of service can be provided if the deadline scheduling of cell transmissions is implemented in ATM switches.

3. Elimination of bandwidth enforcement: With the deadline scheduling of cell transmissions, cells from different sources interfere with each other only through their deadlines. Thus, with a proper deadline assignment and buffering scheme, it is easy to remove the adverse effect of a user's generation of more traffic than that specified at the connection setup time. This eliminates the need for bandwidth enforcement at the network entry points.

4. High transmission efficiency: Another major advantage of using the deadline scheduling is that the messages which violate the pre-specified message generation pattern can be transmitted over the network on a best-effort basis. Early messages will be assigned later deadlines but they can still be eligible for transmission if the network is lightly-loaded. This is in sharp contrast to bandwidth enforcement where early messages are held outside the network until they become on time. This feature is very important for applications where the traffic characteristics cannot be pre-determined accurately, and when the source generates more traffic than expected, it is still desirable that the extra traffic should be transmitted as soon as possible and as long as it does not interfere with the timeliness guarantees of other "law-abiding" channels.

Implementation of deadline scheduling requires ATM switches to have extra processing power. Unlike the conventional FIFO scheduling for which a newly-arrived cell is simply put at the end of a waiting queue and a transmitter always picks up the first cell in the waiting queue to transmit, the deadline scheduling requires that either the transmitter searches a waiting queue for a cell with the earliest deadline to transmit, or a newly-arrived cell is inserted at a proper position such that the cell with the earliest deadline is always at the head of the queue. In either case, the whole waiting queue must be searched once for every cell transmitted.

A key problem to the implementation of deadline scheduling is to make this scheduling process very fast. If the cell-scheduling time cannot be controlled to be less than the cell-transmission time, a queue will be formed at the scheduler. One should *not* use deadline scheduling for the queue at the scheduler since this would need another deadline scheduler. So, only the conventional FIFO or round-robin scheduling can be used, which could result in a larger queueing delay at the scheduler than that at a transmission link. In such a case it is meaningless to use deadline scheduling at a transmission link.

To solve this problem, a hardware implementation of the deadline scheduler was proposed in [14, 17] which can schedule an incoming cell in at most 12 clock cycles. With this architecture, a scheduler implemented with a 30 MHz clock frequency can easily accommodate ATM networks with the link transmission speed up to 1 Gbps. Chao and Uzun [18] also reported an implementation of a sequencer chip which does fast sorting. This chip can also be used for the implementation of deadline scheduling.

3.2 Real-time Channels over ATM Networks

As mentioned in Section 2, a real-time channel is implemented with two protocols: a channel establishment protocol and a message transmission protocol. The channel establishment protocol handles requests for establishing real-time channels. Its main function is to check if the requested end-to-end delay bound and buffer space can be guaranteed under the current traffic load. The message-transmission protocol specifies how cells are constructed from messages and how the deadline of each cell is calculated in order to guarantee the message link delay bound.

For the convenience of presentation, we assume that message size is measured in number of ATM cells and the time is measured in the unit of cell-transmission time. Suppose a real-time channel with a maximum message size M runs through n links with guaranteed

worst-case link delays $d_1, \cdots, d_n$, respectively. Then, the end-to-end message-delivery delay bound can be calculated as [12, 14],

$$D = \sum_{k=1}^{n} d_k - (n-1)(M-1). \quad (1)$$

With the above equation, the end-to-end message-delay bound D for a real-time channel can be guaranteed with one of the following methods.

M1: Divide the end-to-end delay bound D into link delay bounds d_k's such that Eq. (1) is satisfied. Check if each link can guarantee the corresponding link-delay bound under the current traffic load. If all the checks are positive, the requested real-time channel can be established. Otherwise, the channel request is rejected.

M2: Calculate the *minimum* link-delay bound over each link of the channel. If the end-to-end delay bound calculated from Eq. (1) using these minimum link-delay bounds is not larger than D, the requested channel can be established. Otherwise, the channel request is rejected.

A set of real-time channels $\tau_i = (T_i, C_i, d_i)$, $i = 1, 2, ..., n$, is said to be *schedulable* over a link if for all $1 \leq i \leq n$, the maximum delay experienced by channel i's messages over the link is not larger than the requested delay bound d_i. Then, establishment of a real-time channel with **M1** needs a solution to the following channel schedulability problem.

Problem 3.1 (Channel schedulability) .
Given a set of real-time channels $\tau_i = (T_i, M_i, d_i)$, $i = 1, 2, ..., n$, are they schedulable over a link?

Establishment of a real-time channel with **M2** needs a solution to the following minimum delay bound problem.

Problem 3.2 (Minimum delay bound) .
Suppose $n-1$ channels, $\tau_i = (T_i, M_i, d_i)$, $i = 1, 2, ..., n-1$, are schedulable over a link. Given a new channel τ_n with the minimum message inter-arrival time T_n and the maximum message size M_n, what is the minimum value of d_n such that all n channels are still schedulable over the link?

The deadline scheduling theory developed in [12, 15] solves the above two problems if the messages of all real-time channels abide by their pre-specified traffic generation pattern (i.e., the message inter-arrival time is not smaller than T and the message size is not larger than M). However, this assumption does not always hold in practice for the following two reasons.

1. A source node may generate messages faster and larger than the specified values T and M due to the use of estimated (instead of true) channel parameters or malfunctioning of the node. Since no bandwidth-enforcement mechanisms will be used at the source node, all these messages will be sent over the channel. This will affect the delay bounds guaranteed to all other channels over the link.

2. Even if a source node does not violate its traffic specification, the message inter-arrival time at an intermediate link can still be smaller than T due to uneven queueing delays at upstream links.

The message-transmission protocol must solve the above problem by using a proper cell-deadline assignment scheme. With the deadline scheduling of cell transmissions, cells of different channels interfere with each other through their deadlines only. Thus, the adverse effect of early-arriving cells can be eliminated by assigning them some later deadlines. This idea introduces the concept of *logical arrival time* [13].

If a cell is not early, its logical arrival time equals its actual arrival time and its deadline is calculated as its arrival time plus the link-delay bound. If a cell is early, its logical arrival time is set to be the time when it should have arrived and its deadline is calculated as its logical arrival time plus the link-delay bound. Notice that a cell could arrive early at a node in one of the following three ways:

1. A message of a size larger than M is generated at the source node. In this case, all the extra cells are said to be early and their logical arrival times should be set to the generation time of the next message.

2. A message is generated at a source node with inter-generation time smaller than T. In this case, all the cells of the message should be assigned a logical arrival time which is T units of time after the generation of the previous message.

3. Due to the uneven queueing delays at upstream links, message inter-arrival times at downstream links could be smaller than T. In this case, the cells of some messages would have to be assigned logical arrival times later than their actual arrival times to maintain the minimum message inter-arrival time T.

After the channel establishment calculationtep the channel specification (T, M, d) is stored in the switch, and the logical arrival time of each cell is calculated from this per-channel specification, upon each cell arrival, inside the switch. This eliminates the need for cells to carry deadlines. We now present a new message-transmission protocol which implements the above idea. We assume that one bit in the header of an ATM cell, denoted by F, is used to mark the first cell of a message. This bit can be, say, in the Generic Flow Control (GFC) field whose use has not yet been defined in the ATM standards.

The proposed message-transmission protocol is to be executed by the source nodes of real-time channels and ATM switches. The source node of a real-time channel is responsible for splitting messages into cells and forwarding them to an ATM switch. Its main function is to eliminate the adverse effect of over-sized messages by setting the F bits of appropriate cells to mark them as the first cells of the messages. In other words, it reorganizes messages into *logical* messages. If no over-sized messages are generated, the logical messages are the same as the actually messages. When a message of size larger than M is generated, the first M cells are viewed as one message, and the $(M+1)$-th cell is marked as the first cell of the next message. In this way, the ATM switches will see no over-sized logical messages. The detailed operations of a source node is described in the following protocol.

Protocol 3.1 (Message decomposition into cells) .

In this protocol, an integer K is used to count the number of cells of a message, and an integer O is used to detect oversize messages. $O > 0$ means that the current cell is viewed as a cell of the $(n+O)$-th message, where n is the current message. Initialize $O := 0$.

Step 1: *Wait until a message is generated. If $O = 0$, set $K := 0$. Otherwise, update $O := O - 1$.*

Step 2: *Assemble an ATM cell from the message and forward it to the ATM switch. If $K = 0$, set the F bit of the cell to 1 which marks it as the first cell of a message.*

Step 3: *Update $K := K + 1$. If $K = M$, set $K := K - M$ and $O := O + 1$.*

Step 4: *If no more bits left from the message, goto Step 1. Otherwise goto Step 2.*

After messages are split into cells and forwarded to an ATM switch, the switch needs to check whether or not these cells arrive early, and calculate logical arrival times and deadlines for them. The following protocol deals with deadline assignment for a real-time channel $\tau = (T, M, d)$ over a link. Recall that T and M are the user-specified minimum message inter-generation time and maximum message size, respectively. Let d be the assigned link-delay bound.

Protocol 3.2 (Cell deadline assignment) .
Let t_c record the arrival time of the current cell, t_m record the logical arrival time of the current message, and an integer K count the number of cells of the current message. Initialize $t_m := -T$.

Step 1: *Wait until a cell arrives. If it is the first cell of a message, goto Step 2. Otherwise, goto Step 3.*

Step 2: *Set $K := 1$. If $t_c - t_m < T$, set $t_m := t_m + T$. Otherwise, set $t_m := t_c$. Goto Step 4.*

Step 3: *Update $K := K + 1$ and $t_m := \max\{t_m, t_c - K\}$.*

Step 4: *Set the deadline of the cell as $t_m + d$. Goto Step 1.*

Step 2 of the above protocol ensures the logical inter-arrival time of two messages not to be smaller than T. Step 3 updates the logical arrival time of a message according to the actual cell arrival times. This update of logical message arrival time is necessary to ensure that the K-th cell of a message is always available K units of time after its logical arrival time.

Protocols 3.1 and 3.2 are illustrated in Fig. 2 with an example of $T = 3$ and $M = 2$. The upper part of the figure shows the generation of messages at the source node. The source node splits messages into cells, marks boundaries of logical messages, and forward the cells to an ATM switch. The cells that the ATM switch sees are shown in the middle of the figure. According to Protocol 3.2 and the actual cell-arrival times, the logical cell-arrival times are shown at the bottom of the figure. The deadline of a cell is calculated as the cell logical arrival time plus the assigned link-delay bound.

Comparing with the message-transmission protocols proposed in [12, 14], Protocols 3.1 and 3.2 have the following advantages:

1. Cells need not carry the deadline information in their headers. With Protocol 3.2, the deadline of a cell is calculated from the logical arrival time of the current message and the arrival

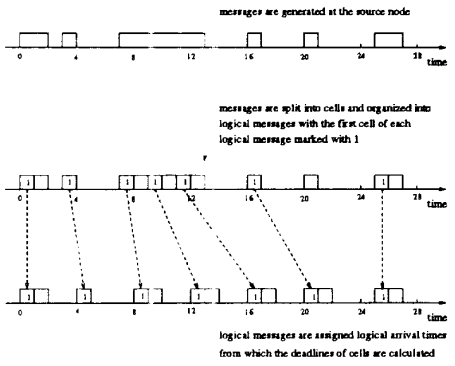

Figure 2: Logical messages and logical arrival times.

time of the cell, while the protocols of [12, 14] require the information of the cell deadline at the previous node. This feature is especially useful for ATM networks since the short, standardized ATM cell header does not have room to hold the cell-deadline information.

2. Clocks at different nodes need not be synchronized and no timestamp of cell transmission is necessary. The message-transmission protocols of [12, 14] assume either the existence of a global clock or the timestamping of a cell when it is transmitted. This increases the complexity of network hardware.

3. The functions of source nodes and ATM switches are clearly separated in Protocols 3.1 and 3.2. All ATM switches run the same protocol (Protocol 3.2) for all real-time cells, regardless whether they are connected to the source node of a real-time channel or not. This simplifies the switch design. Protocols of [12, 14] require a switch to run a different protocol for real-time packets originating from a node to which the switch is directly connected.

Another important problem which must be addressed to implement real-time channels is buffer management in ATM switches. It must satisfy the following criteria:

C1: Enough buffer space must be reserved for each channel such that no cells will be dropped due to buffer overflows when all real-time channels abide by their pre-specified message-generation patterns.

C2: Channel protection must be provided such that if some channels violate their traffic-generation specification, only those cells belonging to these "renegade" channels will be discarded when a buffer overflow occurs.

The buffer space needed for a real-time channel to satisfy **C1** is easy to calculate. Suppose a channel $\tau = (T, M, D)$ runs through n links $\ell_i, \cdots, \ell_n$, all of which guarantee link-delay bounds $d_1, \cdots, d_n$, respectively. Then the buffer space the channel needs at link ℓ_k is bounded by:

$$B = \lceil (\sum_{i=1}^{k} d_i)/T) \rceil M, \tag{2}$$

which corresponds to the worst case that the first message of a channel experiences the largest link delay and all the following messages experience no delay at all.

From Eq. (2), we can see that if the requested end-to-end delay bound $D < T$ then a buffer of size equal to M ATM cells is enough for each switch.

When some channels generate more messages than specified at the channel setup time, buffer overflows may occur. One way to avoid this is to incorporate a link-by-link feedback flow control scheme as the ones proposed by Kung and Chapman [19, 20]. With a link-by-link feedback flow control scheme, one node can send early or extra cells only if it is sure that buffer space for these cells are available at a downstream node, and hence, cell losses can be avoided completely. Feedback control may also reduce a real-time channel's requirement of buffer space. We are currently working on the details of this along with the implemention of real-time channels in ATM networks and will report these results in a forthcoming paper.

4 Conclusion

We have addressed the problem of providing real-time communication services in ATM networks. Using the concept of real-time channel, we can solve the fundamental congestion-control problem and can also provide the guaranteed quality of service in ATM networks. Our main contributions are to show the suitability of the real-time channel approach for ATM networks and develop a new message-transmission scheme which is suitable for ATM networks and has a number of significant advantages over other schemes.

Acknowledgement

The authors would like to thank Dr. Hugh Lauer of Mitsubishi Electric Research Laboratories for his valuable comments and helpful discussions on this paper.

References

[1] 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.

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

[3] A. Eckberg, "B-ISDN/ATM Traffic and Congestion Control," *IEEE Network*, Sept. 1992.

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

[5] T. E. Anderson, S. S. Owicki, and J. B. Saxe, "High speed switch scheduling for local area networks," *Proc. Int'l Conf. on Architectural Support for Programming Languages and Operating Systems*, pp. 98–110, October 1992.

[6] 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.

[7] A. Demers, S. Keshav, and S. ShenKer, "Analysis and Simulation of a Fair Queueing Algorithm," *Proceedings of ACM SIGCOMM 89*, pp. 3–12, 1989.

[8] L. Trajkovic and S. J. Golestani, "Congestion Control for Multimedia Services," *IEEE Network*, Sept. 1992.

[9] D. C. Verma, H. Zhang, and D. Ferrari, "Delay Jitter Control for Real-time Communication in a Packet Switching Network," *Proceedings of TriComm 91, Chapel Hill, NC*, pp. 35–43, April 1991.

[10] L. Zhang, *A New Architecture for Packet Switched Network Protocols*, PhD thesis, MIT, 1989.

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

[12] Q. Zheng, K. G. Shin, and E. Abram-Profeta, "Transmission of compressed digital motion video over computer networks," in *Digest of COMPCON Spring'93*, pp. 37–46, February 1993.

[13] D. D. Kandlur, *Networking in Distributed Real-Time Systems*, PhD thesis, University of Michigan, 1991.

[14] Q. Zheng, *Real-time Fault-tolerant Communication in Computer Networks*, PhD thesis, University of Michigan, 1993. PostScript version of the thesis is available via anonymous FTP from ftp.eecs.umich.edu in directory outgoing/zheng.

[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*, vol. 42, no. 2/3/4, pp. 1096–1105, February/March/April 1994.

[16] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard real-time environment," *Journal of the ACM*, vol. 20, no. 1, pp. 46–61, January 1973.

[17] Q. Zheng and K. G. Shin, "Real–time communication in local area ring networks," in *Conference on Local Computer Networks*, pp. 416–425, September 1992.

[18] 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.

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

[20] 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.