Traffic-Controlled Rate-Monotonic Priority Scheduling of ATM Cells

Seok-Kyu Kweon and Kang G. Shin

Real-Time Computing Laboratory
Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, Michigan 48109-2122.
email: {skkweon, kgshin}@eecs.umich.edu

Abstract

Future integrated services networks are expected to carry various types of traffic including real-time packet video for which stringent quality of service (QoS) requirements must be met. In this paper, we propose a cell multiplexing scheme for providing real-time communication services in point-to-point ATM networks. Our scheme is simple enough to be used in high-speed ATM networks, yet reasonably efficient in terms of accommodating channel requests. The scheme employs the leaky bucket model as the input traffic description model and regulates the input traffic at User Network Interface (UNI) to comply with the input traffic specification. Inside the network ATM switches provide bounded local delays to individual cells using a traffic controller and a non-preemptive rate-monotonic priority scheduler.

1 Introduction

ATM networks have been drawing significant attention as the main technology for implementing B-ISDN due mainly to its potential for efficiency and flexibility. In order to realize the potential of B-ISDN, ATM networks must support a wide variety of traffic and meet diverse service and performance requirements. Among B-ISDN services, providing performance (delay and/or delay jitter bound) guarantees is essential to such real-time applications as video & audio conferencing and video on-demand. Unlike the datagram service, real-time applications must meet very stringent performance requirements in terms of delay, delay jitter, throughput, and packet loss rate.

The work reported in this paper was supported in part by the National Science Foundation under Grant MIP-9203895 and the Office of Naval Research under Grant N00014-94-1-0229. The opinions, findings and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the view of the funding agencies.

Among these performance parameters, delay is considered as the most important parameter in real-time communication services [1], since the cell error rate can be kept very low with the use of advanced networking technologies. Especially, real-time communication services require *each* cell delivery delay, not the average delay, as a QoS requirement. If a cell arrives at the destination after its deadline has expired, its value to the application may be greatly reduced or even worthless. In some circumstances, a cell missing its deadline is considered lost. Thus, the cell delivery delay must be bounded and predictable for real-time applications. It is impossible to meet different QoS requirements for different real-time applications using the best-effort delivery service of conventional packet-switched networks, because their packet multiplexers do not differentiate between real-time and non-real-time traffic, nor among real-time messages themselves. Recently, several packet multiplexing techniques have been proposed to provide different QoS for different real-time applications [2–9]. Interested readers are referred to [10] and [11] for a detailed comparison of these techniques. Although many of them can provide bounded end-to-end delays in a point-to-point network, they differ in input traffic specification, channel admissibility and implementation complexity. Among them, Packet-by-packet Generalized Processor Sharing (PGPS)[12, 13] is the most efficient in channel admissibility for two reasons: one is the optimal deadline scheduling and the other is the efficiency of its input traffic specification. However, it may not be a good candidate for realizing real-time communication in ATM networks because of the complexity of its deadline scheduling. Deadline scheduling requires the incoming cells to be sorted based on their deadlines (virtual finish times in case of PGPS), which is computationally expensive. In this paper, we propose a new traffic control scheme called the *Traffic-Controlled Rate-Monotonic Priority Scheduling* (TCRM) which provides user-requested delay guarantees in point-to-point ATM networks. It satisfies both the simplicity and efficiency requirements of an ATM switch. This scheme requires traffic regulation at UNI and scheduling at each link along the path. We divide the multiplexer of each link into two components: traffic regulator and scheduler. Using this mechanism, TCRM (1) has efficiency close to PGPS in terms of channel admissibility; (2) is simple enough to operate in a high-speed switching environment like ATM networks; (3) requires only a very small buffer space for each real-time channel.

2 The Proposed Scheme

We assume that the leaky bucket model [14,15] is given as the input traffic description. The leaky bucket model, simply denoted by (σ_i,ρ_i), is to place a smoothing buffer (leaky bucket regulator) of size σ_i and token generation rate ρ_i at the network entrance so that the burstiness of input traffic into the network may be reduced, thus lowering the network resource reservation requirement. This (σ_i,ρ_i) model is determined based on the performance requirements as well as the characteristics of input traffic. In Section 3, we will discuss the efficiency of the leaky bucket model.

Our scheme requires User Network Interface (UNI) and each ATM switch along the path to cooperate in order to provide real-time communication services. UNI regulates each channel i's traffic so that the cell arrival rate at the network entrance is bounded by ρ_i. UNI must have buffer space of σ_i bits to avoid cell loss. The network service provider ensures that the requested channel i gets its (minimum) service rate ρ_i at every switch along the path through appropriate admission control and run-time processing. For this purpose, the switch needs to enforce a special cell scheduling policy.

2.1 Traffic Shaping at UNI

Given the traffic model (σ_i,ρ_i), the user requests the cell transmission rate ρ_i from the network. After establishing the channel based on an appropriate admission test, the user begins to transmit its traffic according to the (σ_i,ρ_i) model. At the network entrance, the UNI regulates the incoming traffic specified by the (σ_i,ρ_i) model. The UNI regulates the maximum cell transmission rate into the network entrance below ρ_i by keeping the minimum cell inter-transmission time larger than L/ρ_i, where L denotes the length of one cell (53 bytes). That is, when the k^{th} cell of channel i has arrived at the UNI at time A_k, its transmission

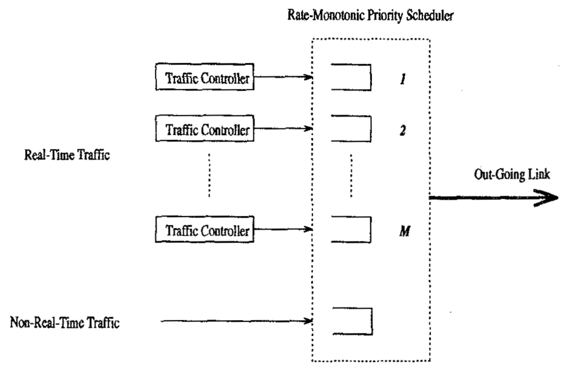

Figure 1: Structure of TCRM

time, X_k, is calculated as:

$$X_k = \begin{cases} A_1 & k=1 \\ \max(X_{k-1}+\frac{L}{\rho_i},\ A_k) & k \geq 2. \end{cases} \qquad (2.1)$$

Until X_k, the UNI holds the cell in its buffer. Since one cell is permitted to be transmitted every time interval of length L/ρ_i, the minimum and maximum guaranteed service rates of the queue are both ρ_i over an interval L/ρ_i.

2.2 Traffic Regulation and Scheduling inside the Network

We model an ATM switch as an output-buffered multiple-input-multiple-output switch [16]. In this model, no cells are lost due to contention within the switch fabric and contention exists only among those cells sharing the same outgoing link. Assuming that the switching delay is negligible as compared to the queueing delay at an output buffer, we concentrate on controlling the queueing delay at the output buffer in order to achieve a bounded end-to-end delay. TCRM is a cell multiplexer which decides the service order of cells waiting in the output buffer in order to provide different link delay bounds for different channels.

Figure 1 shows the structure of TCRM consisting of *traffic controllers* and *rate-monotonic priority scheduler*. A traffic controller is assigned to each individual real-time channel and the rate-monotonic priority scheduler is shared by all the channels which have already been established.

2.2.1 Traffic Controller

The function of the traffic controller is to keep the cell arrival rate at the scheduler below the token generation rate ρ_i. It holds the incoming cells until their supposed arrival times, and then transfers them into the scheduler. This supposed arrival time is called the *logical arrival time* in [6, 9]. The logical arrival time of an incoming cell is calculated based on that of the previous cell of the same channel. Thus, the logical arrival time of the k^{th} cell at the n^{th} node, $X_{k,n}$, is calculated as:

$$X_{k,n} = \begin{cases} A_{1,n} & k = 1 \\ \max(X_{k-1,n} + \frac{L}{\rho_i},\ A_{k,n}) & k \geq 2, \end{cases} \quad (2.2)$$

where $A_{k,n}$ is the actual arrival time of the k^{th} cell at node n. Note that the inter-logical arrival time of the incoming cells is at least L/ρ_i. Assuming that the cell arrival rate at the traffic controller is under ρ_i, we can assure that at most one cell for channel i can exist in the traffic controller of channel i, since the traffic controller is permitted to transfer a cell every L/ρ_i seconds. Hence, the traffic controller requires buffer space for storing only one cell.

2.2.2 Non-Preemptive Rate-Monotonic Priority Scheduling

After the cell stream of real-time channel i passes through the traffic controller, the cell arrival rate at the scheduler of every switch is bounded by ρ_i. In order to prevent the unbounded accumulation of cells in the scheduler, we must provide the minimum throughput ρ_i to this channel. The minimum cell inter-arrival time and cell delay bound are therefore given by L/ρ_i. Liu and Layland [17] proved that the rate-monotonic priority scheduling is optimal among all fixed-priority scheduling policies when the deadline of each task is the same as the period of each task. A cell is treated as a "task", and the rate-monotonic priority cell scheduling is optimal among fixed-priority scheduling policies[1] that achieve the guaranteed throughput, because the cell inter-arrival period is the same as the cell delivery deadline ($=L/\rho_i$). So, we employ the rate-monotonic priority scheduler for transmitting cells. This scheduling policy assigns higher priority to channels with higher request rates, i.e., higher ρ_i.

Liu and Layland's analysis [17] is based on a preemptive scheduling policy. In our scheme we use the non-preemptive rate-monotonic priority scheduling policy as the cell scheduler for the efficiency of cell transmission. (Employing a non-preemptive policy doesn't affect the optimality of the rate-monotonic priority scheduling, since the non-preemptive rate-monotonic priority scheduling policy is optimal among non-preemptive fixed-priority policies, which can be proved using the same arguments in the proof of Theorem 2 in [17].) The non-preemptive rate monotonic priority scheduler assigns a priority level to each real-time channel according to its required throughput and in-progress cell transmission will not be preempted. As a result, the scheduler provides the minimum throughput ρ_i to each channel i.

Since the cell inter-arrival time is larger than, or equal to, L/ρ_i and one cell is permitted to be transmitted every L/ρ_i, at most one cell can stay in the scheduler at any time. Hence, the scheduler needs a buffer of one cell for each real-time channel.

2.3 Admission Control

In order to provide throughput guarantees with the non-preemptive rate-monotonic priority scheduler, we need appropriate admission control for real-time channels. The admission control test involves every node along the path of the real-time channel. If any node along the path fails this test, the channel request must be denied.

At the scheduler, the throughput guarantee is made not for bit-by-bit but for cell-by-cell. That is, when each cell arrives at the scheduler with the minimum cell inter-arrival time, L/ρ_i, guaranteed by the traffic controller, the scheduler must finish the transmission of the cell before the next cell's earliest arrival time, which is the current cell's arrival time plus L/ρ_i. We need a schedulability test to verify whether or not the worst-case delivery time is smaller than the local delay bound L/ρ_i. In a way similar to Kandlur's [9], we derive the schedulability test for the proposed scheme. Consider a set of real-time channels $\{i,\ i = 1, \ldots, M\}$ which share a common link l, where M is the number of existing real-time channels on link l. Denote the throughput of channel i by ρ_i, and assume that channels are indexed in the descending order of priority so that $\rho_i \geq \rho_j$ if $i < j$. Then the schedulability test is given as:

$$\sum_{j=1}^{i-1} C \lceil \frac{L/\rho_i}{L/\rho_j} \rceil + 2C \leq \frac{L}{\rho_i} \quad \text{for } i = 1, \ldots, M, \quad (2.3)$$

where C is one cell transmission time, $\frac{L}{\rho_i}$ is the link delay bound of channel i's cell, and all channels j, $1 \leq j < i$, have higher priority than channel i. Note that the first term of Eq. (2.3) denotes the sum of all the transmission times of cells belonging to the channels of higher priority than channel i in the worst case.[2]

[1] Although fixed-priority scheduling policies are less efficient than deadline-scheduling policies in terms of network utilization [17], we prefer the implementation simplicity of the rate-monotonic priority scheduling.

[2] Here, the worst case means that all the channels of higher

The one C of the second term denotes the time to complete in-progress cell transmission. The other C denotes the transmission time of a cell belonging to channel i. Conceptually, the schedulability condition implies that the transmission of a cell of channel i must be finished within its link delay bound even in the worst case. If the schedulability condition fails, the cell of channel i cannot be transmitted in time in the worst case. Therefore, the schedulability condition is also the necessary condition.

Using this argument, we can show that TCRM emulates circuit-switching in the cell level. Let us define $T_i(t,s)$ as the channel i's traffic (or number of cells) transmitted over a link during a time interval $[t,s)$ for any t,s such that $t \leq s$. Then,

$$L\lfloor\frac{s-t}{L/\rho_i}\rfloor \leq T_i(t,s) \leq L\lceil\frac{s-t}{L/\rho_i}\rceil \qquad (2.4)$$

The lower bound is derived from the fact that a local delay bound is guaranteed by the scheduler and the upper bound comes from traffic regulation by the traffic regulator. Therefore, the average traffic service rate of channel i, $R_i(t,s)$, during the interval $[t,s)$ is given as:

$$R_i(t,s) = \rho_i, \qquad (2.5)$$

where $s-t = k(L/\rho_i)$ and $k = 1,2,\ldots$ In other words, the throughput of channel i is guaranteed to be ρ_i during any time interval of length L/ρ_i, implying that TCRM emulates circuit-switching in the cell level. This feature allows us to provide the CBR services *in addition to* real-time communication services without losing the statistical multiplexing gain of ATM networks.

In [18], a simple schedulability test algorithm is derived based on Eq. (2.3).

2.4 Bounding End-to-End Delays in Multi-hop Connections

Using the fact that TCRM guarantees the minimum throughput ρ_i for channel i, we can derive the end-to-end delay bound of channel i. Given the input traffic specification (σ_i, ρ_i) of channel i, during any time interval of length t, the amount of traffic generated by the user may not exceed $\sigma_i + t \cdot \rho_i$. For the sake of convenience, we assume that σ_i is the multiple of the length of one cell L. By UNI's traffic regulation, the cell arrival rate at the network entrance is limited by ρ_i and the burst is held at the buffer of the UNI. We assign a buffer space of σ_i for channel i.

priority than channel i generate their cells concurrently with channel i.

We now show the boundedness of end-to-end delivery delays. First, we show that the queue size at the input buffer of the UNI cannot be larger than σ_i. During a time interval $[s,t)$ for any s,t such that $s \leq t$, the maximum amount of traffic that has arrived at the UNI, $x_i(s,t)$, is given by:

$$x_i(s,t) = \sigma_i + \rho_i(t-s) \qquad (2.6)$$

under the leaky bucket model. However, since the traffic is transmitted cell-by-cell, the maximum number of cells that have arrived at the UNI during $[s,t)$ is given by

$$x_i(s,t) = \sigma_i + L\lfloor\frac{s-t}{L/\rho_i}\rfloor. \qquad (2.7)$$

During the same interval, the minimum amount of traffic transmitted into the outgoing link at the UNI, $y_i(s,t)$, is given by

$$y_i(s,t) = L\lfloor\frac{s-t}{L/\rho_i}\rfloor, \qquad (2.8)$$

which comes from the fact that channel i is guaranteed to have the minimum throughput ρ_i. Therefore, the maximum backlog at the input buffer during the interval $[s,t)$ is given by $B_{max} = x_i(s,t) - y_i(s,t) = \sigma_i$, and the maximum number of cells that can exist at the buffer is σ_i/L.

Using this fact, we can derive the following theorem on the end-to-end delivery delay bound in ATM networks.

Theorem 2.1: If the input traffic of a real-time channel i is specified by (σ_i, ρ_i) and its guaranteed throughput is ρ_i, then the end-to-end delivery delay of any cell belonging to channel i is bounded by

$$D_i = \frac{\sigma_i}{\rho_i} + N\frac{L}{\rho_i} + \sum_{k=1}^{N} e_k, \qquad (2.9)$$

where N is the number of hops that channel i must take and e_k is the propagation delay at the k^{th} link.

The proof of this theorem is omitted due to space limit. Informally, the first term in Eq. (2.9) indicates the maximum queueing delay at the UNI. The second and third terms are the sum of the maximum queueing delays and the sum of propagation delays, respectively.

Our approach shares the same idea of splitting the traffic controllers and the scheduler to provide the bounded link delays in Rate Controlled Static Priority

Queueing (RCSP) [6]. Since TCRM allocate a different priority level to each real-time channel, they can provide arbitrary link delay bounds, whereas RCSP provides a preset finite number of link delay bounds. This enables the allocation of a wide range of bandwidths in TCRM at the expense of more complex channel management.

3 Comparative Evaluation of Channel Admissibility

To show the efficiency of TCRM, we present a simple numerical example which compares the channel admissibility of our scheme and three other schemes, i.e., Real-Time Channel (RTC) [9], RCSP [6] and PGPS with real compressed video sequences. The reason why we choose the three schemes is that TCRM shows a similarity to RTC and RCSP in terms of structure and that PGPS shows the best efficiency in terms of channel admissibility [11].

RCSP, RTC and PGPS adopted (X_{min}, X_{ave}, I, P), $(S_{max}, R_{max}, B_{max})$ and (σ_i,ρ_i) models, respectively [6,8,9]. Here X_{min} is the minimum cell inter-arrival time, X_{ave} is the average cell inter-arrival time over an averaging interval of length I, P and S_{max} are the cell size, R_{max} is the maximum cell generation rate, B_{max} is the maximum burst size, σ_i is the burst size, and ρ_i is the token generation rate.

Our network model is a homogeneous ATM network which has 11 serially-connected nodes. The reason for using a multi-hop network is that in RTC and RCSP, end-to-end delay bounds are given as the sum of link delay bounds, but in PGPS and TCRM, the multi-hop affects the end-to-end delay bounds differently (see Eq. (2.9)). In the network, the first node is the sender and the 11^{th} is the receiver. Thus, the path from the sender to the receiver has 10 intermediate links. Each link has the transmission bandwidth of 100 Mbits/sec. The traffic data used in our calculation are obtained from two MPEG-coded movie clips: *Starwars* (Sequence 1) and *Honey, I Blew Up the Kids* (Sequence 2). These sequences consist of frames generated once every 1/30 second. The frame sizes of two sequences are plotted in Figure 2, where the x-axis is the frame number and the y-axis is the frame size in cells. In this example, we consider two cases: one is multiplexing homogeneous traffic, and the other is multiplexing heterogeneous traffic. In multiplexing homogeneous traffic, we consider either sequence alone, and calculate the end-to-end delay bound against the number of real-time channels established. In multiplexing heterogeneous traffic, we attempt to establish real-time

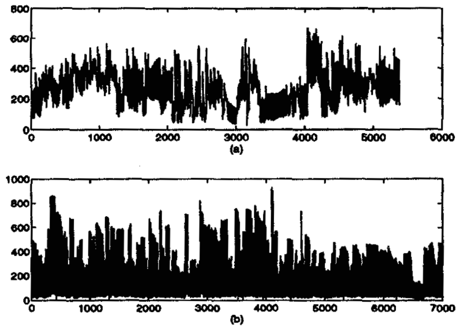

Figure 2: Frame sizes of compressed video sequences

(a) *Starwars* (b) *Honey, I Blew Up the Kids*

channels for both sequences together, and calculate the number of real-time channels of Sequence 1 establishable while varying the number of real-time channels of Sequence 2.

3.1 Multiplexing Homogeneous Traffic

First, consider Sequence 1. The maximum packet size is one cell long (53 bytes $\times$ 8 bits/byte = 424 bits), the maximum number of cells in one frame is 670, and the average number of cells is 227.69. Thus, the peak traffic generation rate is 8.52 Mbits/sec, and the average traffic generation rate is 2.89 Mbits/sec. The user's end-to-end delay requirement is given as $10 \times 1/30$ sec and thus, the local delay bound is 1/30 sec. We now want to compute how many real-time channels described above can be established.

To calculate the end-to-end delay bounds under TCRM, we first determine the burst size σ_i and the required throughput ρ_i from the frame size. The throughput ρ_i is obtained by dividing the link speed by the number of channels plus one, which is needed to pass the schedulability test. Note that this service rate must be larger than the average traffic generation rate (here 2.89 Mbits/sec) to provide bounded end-to-end delays. A burst is given as the accumulated traffic arrived minus the traffic being transmitted at the guaranteed throughput. The maximum size of burst is chosen as the bucket size σ_i. By plugging these parameters into Eq. (2.9), we calculate the end-to-end delay bounds for Sequence 1. In this example, we assume that the propagation delay is negligible. The results are plotted in Figure 3: one can establish up to 21 channels given the user-requested end-to-end delay bound is 1/3 sec.

For PGPS, we derive (σ_i, ρ_i) in the same way as

TCRM, except that ρ_i is obtained by dividing the link speed by the number of channels. This is because PGPS has only the link utilization test for its admission control. Using the formula in [13], we calculate the end-to-end delay bounds. Given the user-requested bound 1/3 sec, up to 22 channels can be established.

With RTC, we can establish 11 channels based on the schedulability test in [9]. This number is the same as that of a circuit-switched network which reserves the peak traffic generation rate. Considering the fact that the schedulability condition is based on the maximum traffic generation rate, this is an expected result. The calculation of the end-to-end delay bounds is based on the worst-case response time in [9].

For RCSP, we assume here only one priority level with a local delay bound of 1/30 sec, because the characteristics of all the channels are assumed to be homogeneous. Zhang [19] derived the local delay bound for the case when the average link utilization does not exceed 1. It is given differently depending on whether the peak utilization exceeds 1 or not. We calculated the parameters X_{min}, X_{ave} and I in order to use them in the calculation of delay bounds. First, X_{min} is simply given by the reciprocal of the peak cell generation rate. X_{ave} is chosen as the maximum value under the constraint that the average link utilization may not exceed 1. Thus, X_{ave} is given by one cell size times the number of channels established divided by the link capacity. I is determined so that the average cell inter-arrival time during any time interval over I is larger than X_{ave}. In our calculation, I shows very large values when the peak link utilization exceeds 1. This is why the delay bounds are so large when the peak link utilization exceeds 1. Given the user requested end-to-end delay bound 1/3 sec, up to 15 channels can be established.

In Figure 3, until the 11^{th} channel is established, all four schemes show reasonable end-to-end delay bounds. However, when the peak utilization exceeds 1 (i.e., the number of channels is greater than 11), RCSP is shown to exhibit rapidly-increasing delay bounds. RTC does not guarantee bounded delays when the peak utilization exceeds 1. TCRM and PGPS show very reasonable end-to-end delays even when the number of real-time channels is fairly large. The reason why TCRM is slightly less efficient than PGPS is that TCRM employs a fixed-priority scheduling, while PGPS adopts an optimal dynamic priority scheduling which requires a complex sorting process.

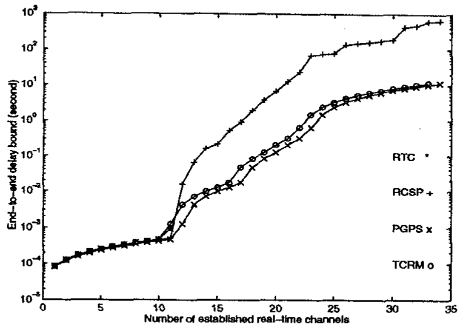

Figure 3: Achievable end-to-end delay bounds: *Starwars*

For Sequence 2, we have obtained a result similar to those of Sequence 1. The result is omitted due to space constraint.

3.2 Multiplexing Heterogeneous Traffic

In order to compare the channel admissibility in a heterogeneous environment, we calculate the number of establishable real-time channels of Sequence 2 with a fixed number of real-time channels of Sequence 1. For this, we need to fix the traffic description parameters. Based on the user requirements and traffic characteristics, we derive the traffic description parameters for TCRM, PGPS, and RCSP from the case when a maximum number of real-time channels are establishable while meeting the user end-to-end delay requirement, 1/3 sec. We consider the same network model and user requirements as the homogeneous case, and we omit the analysis of RTC since its channel admissibility differs little from the case of multiplexing homogeneous traffic.

In this analysis, we first establish a fixed number of real-time channels of Sequence 1, then determine the maximum number of establishable real-time channels of Sequence 2. By increasing the number of channels of Sequence 1, one can decrease that of Sequence 2. Figure 4 shows the results for RCSP, PGPS and TCRM. Note that the channel admissibility of TCRM is very close to that of PGPS while that of RCSP is not very good.

In this example, we have verified that all four schemes show different degrees of channel admissibility. The difference among RTC, RCSP[3] and PGPS

[3]Delay-EDD [5] is omitted in this comparison, since it shares many similarities with RTC and RCSP. In particular, it shares the same input traffic specification with RCSP. But its admission control test for deterministic performance guarantees is

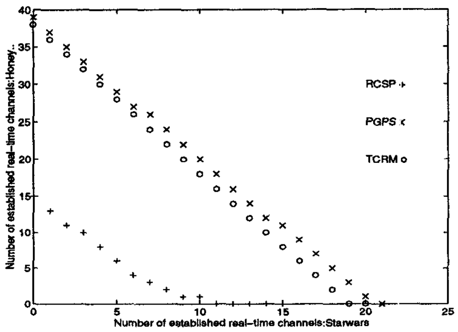

Figure 4: Channel accommodability for the heterogeneous case

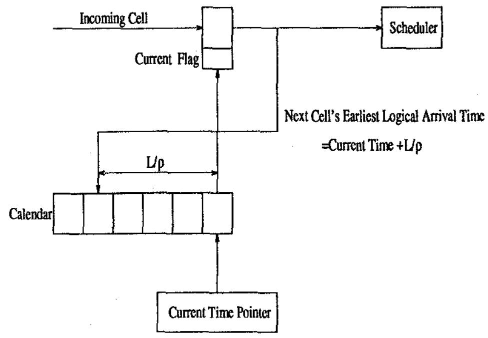

Figure 5: Implementation of TCRM

in channel admissibility is due to their input traffic specifications rather than the cell scheduling mechanisms themselves. In particular, the leaky bucket model seems the most efficient. The main reason for this is that the reserved traffic generation rate ρ_i is not based on the average traffic generation rate and the traffic peak generation rate as in RCSP or solely the peak traffic generation rate as in RTC, but instead the smallest ρ_i is chosen as long as user delay requirements are satisfied. By employing the leaky bucket model as the input traffic specification, TCRM provides efficiency close to PGPS in terms of channel admissibility.

4 Implementation

In this section, we propose a simple structure of the traffic controller of TCRM, requiring operations simple enough to operate in high-speed ATM networks.

TCRM consists of two components: traffic controller and scheduler. The scheduler employs a fixed-priority scheduler. The simplicity of the traffic controller is very important, considering that the priority scheduler is not a complex component. Zhang and Ferrari [6] proposed use of a modified version of calendar queue as the rate-controller whose function is very similar to the traffic controller. The calendar queue can be used to implement the traffic controller of TCRM.

We propose a simpler structure of the traffic controller by modifying Zhang's rate-controller, since TCRM has very simple characteristics as compared to RCSP as follows: 1) calculation of logical arrival time is easier as one can see in Eq. (2.2) and 2) at any instant, there is at most one cell per real-time channel at the traffic controller at each link because of traffic regulation at the previous switch. Thus, we need buffer space of only one cell at the traffic controller.

based solely on the peak traffic generation rate, and thus, its channel admissibility is very poor.

Using these properties, we propose a simple structure for TCRM as shown in Figure 5. There is a buffer for an incoming cell and this buffer is tagged with Current Flag which indicates the cell in the buffer is current or early. If the cell is current and so Current Flag is on, the cell is transferred into the priority scheduler. Otherwise, the cell must wait until Current Flag will be set at its logical arrival time by the action of Calendar and Current Time Pointer. Each real-time channel is assigned its own Calendar. When the cell in the buffer is transferred into the scheduler, the next cell's earliest logical arrival time (=Current Time + L/ρ) is recorded in Calendar. Note that the calculation of the earliest logical arrival time is unnecessary since it is a constant term for each real-time channel. Current Time Pointer gets incremented at an appropriate time interval. When an entry in Calendar is enabled by Current Time Pointer, the real-time channel identifiers stored in the entry are used to activate Current Flags of the corresponding real-time channels.

The design of Calendar is very simple since it is based on a bitmap in which one bit is assigned to each real-time channel, and thus, its storage requirement for each entry is very small. The length of Calendar increases when the network must support low-bandwidth channels with large link delay bounds. This problem can be solved by adopting a hierarchical structure for Calendar. That is, one can achieve a long Calendar by managing several sub-calendars which have different time-grains. Consider the simplest case with two sub-calendars: High-Calendar and Low-Calendar. High-Calendar has a larger time grain than Low-Calendar. By letting High-Calendar activate the Low-Calendar instead of activating Current Flag directly, one can achieve a very long Calendar with a small amount of memory.

5 Conclusion

In this paper, we have proposed an ATM cell multiplexer called the Traffic-Controlled Rate-Monotonic Priority Scheduling (TCRM) to realize performance-guaranteed real-time communication on ATM networks. TCRM (i) provides bounded end-to-end delays which are essential for real-time communication, and (ii) is simple enough to operate in high-speed ATM networks. Using this property one can provide CBR services in ATM networks while keeping the statistical multiplexing gain. We have proposed a simple structure of traffic controller by modifying the calendar queue which requires only a very small memory space and is simple enough to operate at a very high speed. In a comparative evaluation, we have shown that TCRM is similar to RCSP and RTC in terms of implementation complexity but can achieve high channel admissibility similar to PGPS which is more complex to implement.

In this paper, we have dealt only with hard real-time communication using a non-work-conserving cell service discipline like TCRM. However, it is important to provide *statistical* performance guarantees in ATM networks since future multimedia communication is likely to have many statistical real-time requirements. We are currently investigating how to provide statistical real-time communication services using TCRM.

References

[1] S. Shenker, D. Clark, and L. Zhang, "A service model for an integrated services internet", IETF working draft, Oct. 1993.

[2] L. Zhang, "Virtual clock: a new traffic control algorithm for packet switching networks", *in Proc. of ACM SIGCOMM 90*, pp. 19–29, 1990.

[3] C.R. Kalmanek, H. Kanakia, and S. Keshav, "Rate controlled servers for very high-speed networks", *in Proc. of IEEE GLOBECOM 90*, pp. 12–20, 1990.

[4] S. J. Golestani, "A stop-and-go queueing framework for congestion management", *in Proc. of ACM SIGCOMM 90*, pp. 8–18, 1990.

[5] D. Ferrari and D. C. Verma, "A scheme for real-time channel establishment in wide-area networks", *IEEE J. Select. Areas Commun.*, vol. SAC-8, pp. 368–379, Apr. 1990.

[6] H. Zhang and D. Ferrari, "Rate-controlled static-priority queueing", *in Proc. of IEEE INFOCOM 93*, pp. 227–236, 1993.

[7] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm", *in Proc. of ACM SIGCOMM 89*, pp. 1–12, 1989.

[8] A. K. Parekh, *A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks*, PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Feb. 1992.

[9] D. D. Kandlur, K. G. Shin, and D. Ferrari, "Real-time communication in multi-hop networks", *in Proc. 11-th Int'l Conf. Distributed Comput. Systems*, pp. 300–307, May 1991, (An extended version appeared in *IEEE Trans. on Parallel and Distributed Systems*, vol. 5, no. 10, pp. 1044–1056, October 1994).

[10] H. Zhang and S. Keshav, "Comparison of rate-based service disciplines", *in Proc. of ACM SIGCOMM 91*, pp. 113–121, 1991.

[11] S. Kweon and K. G. Shin, "Comparison of three performance-guaranteed packet scheduling algorithms", Real-Time Computing Laboratory Technical Report, May 1994.

[12] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single node case", *IEEE/ACM Transaction on Networking*, vol. 1, pp. 344–357, June 1993.

[13] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The multiple node case", *in Proceedings of IEEE INFOCOM 93*, pp. 521–530, 1993.

[14] R. L. Cruz, "A calculus for network delay, part I: network elements in isolation", *IEEE Trans. on Information Theory*, vol. 37, pp. 114–131, Jan. 1991.

[15] R. L. Cruz, "A calculus for network delay, part II: network analysis", *IEEE Trans. on Information Theory*, vol. 37, pp. 132–142, Jan. 1991.

[16] Martin de Prycker, *Asynchronous Transfer Mode: solution for broadband ISDN*, Ellis Horwood, London, 1991.

[17] C.L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard real-time environment", *Journal of ACM*, vol. 20, pp. 44–61, Jan. 1973.

[18] S. Kweon and K. G. Shin, "Traffic-controlled rate-monotonic priority scheduling", Real-Time Computing Laboratory Technical Report, May 1995.

[19] Hui Zhang, *Service Disciplines For Packet-Switching Integrated-Services Networks*, PhD thesis, Department of Computer Science, The University of California, Berkeley, CA, 1993.