

PAPER Special Issue on ATM Traffic Control and Performance Evaluation

# Loss and Delay Analysis of Dynamic Flow Setup in ATM Networks

Kohei SHIOMOTO<sup>†</sup>, Member, Qiyong BIAN<sup>††</sup>, and Jonathan S. TURNER<sup>††</sup>, Nonmembers

**SUMMARY** In recent years, there has been a rapid growth in applications such as World Wide Web browsing, which are characterized by fairly short sessions that transfer substantial amounts of data. Conventional connection-oriented and datagram services are not ideally engineered to handle this kind of traffic. We present a new ATM service, called *Dynaflow* service, in which virtual circuits are created on a burst-by-burst basis and we evaluate key aspects of its performance. We compare Dynaflow to the Fast reservation protocol (FRP) and show that Dynaflow can achieve higher overall throughput due to the elimination of reservation delays, and through the use of shared "burst-stores." We study the queuing performance of the dynaflow switch and quantify the relationship between the loss ratio and the buffer size.

**key words:** ATM, datagram, connection-oriented, burst transmission

## 1. Introduction

Among the applications on the Internet, the world wide web (WWW) has come to account for a dominant proportion of total traffic [1]. Through its linked hypertext structure, the WWW allows us to access documents located at sites distributed all over the world. Typical documents appearing on the web range in size from a few kilobytes to several megabytes. Because the linked hypertext structure can lead to rapid hopping from site to site, there has been a rapid growth in the number of interactive sessions that last for short periods of time, often less than a second.

Since asynchronous transfer mode (ATM) is recognized as a platform for future wide-area B-ISDNs, it should support web applications efficiently. Unfortunately current ATM services are not well suited to handle large numbers of short-lived sessions. Establishing virtual circuits in ATM networks introduces a substantial control overhead that is a significant source of inefficiency for short-lived sessions. Although a datagram service like IP packet forwarding can eliminate the overhead associated with connection establishment, datagram networks must perform a great deal of redundant work when processing the many identically addressed packets in a large web transfer (see Fig. 1). In

addition, conventional datagram services cannot allocate link bandwidth to a data transfer or make intelligent routing decisions based on bandwidth requirements.

The *Dynaflow* service, introduced in [2], [3], has characteristics of both virtual circuits and datagrams. Like a datagram service, it requires no prior end-to-end session establishment, but like virtual circuit, it forwards data by hardware switching, can assign bandwidth for the duration of a burst transmission, and makes routing decisions based on bandwidth needs. Although several schemes for burst-level bandwidth reservation in ATM have been studied previously [4]–[7], they operate over pre-established virtual circuits and hence lack the flexibility and low overhead that the dynaflow service seeks to provide.

In the *Dynaflow* service, virtual circuits are established on a burst-by-burst basis, by sending a burst setup cell at the start of a burst transmission. Burst setup cells, contain the sender's address, the destination address, and the rate at which the sender is transmitting cells. Setup cells must be sent periodically during the burst, in order to maintain the virtual circuit. At the end of a burst transmission, the sender forwards a *release* cell, causing all allocated resources to be freed. Resources are also released if no setup cells are received during a timeout interval. No end-to-end acknowledgment of

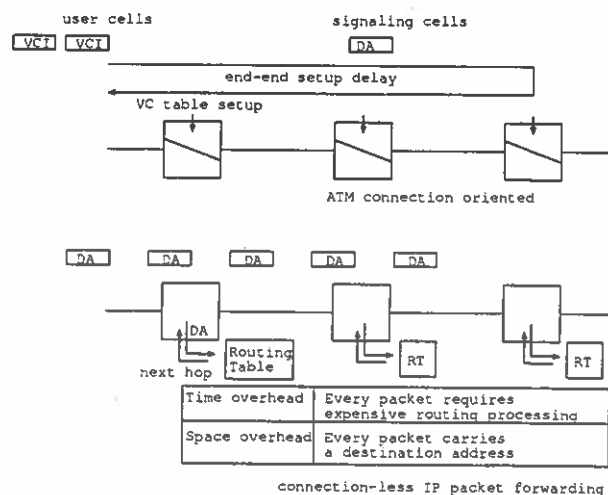


Fig. 1 Comparison of connection-oriented ATM and IP datagram forwarding.

Manuscript received September 22, 1997.

Manuscript revised December 10, 1997.

<sup>†</sup>The author is with NTT Network Service Systems Laboratories, Musashino-shi, 180-8585 Japan.

<sup>††</sup>The authors are with Washington University in St. Louis, USA.

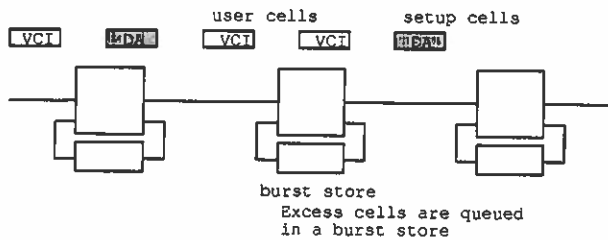


Fig. 2 Dynaflow service.

the burst setup cell is required before data transmission commences. Thus end-end setup delay is avoided and the control overhead is reduced as illustrated in Fig. 2.

The routing and admission control decisions made at a dynaflow switch are fairly simple. An IP lookup yields a list of outgoing links that are on paths to the desired destination. The first link in this list with sufficient unused bandwidth to handle the new flow is selected, and the flow is switched through to that link at the ATM level. If none of the outgoing links can currently accept the flow, it is diverted to a central shared buffer called a "burst store." Data is generally forwarded from the burst store on a best-effort basis, using whatever bandwidth is available to reach the destination.

The logical burst store is composed of a collection of physical burst stores, each of which is connected to a core ATM switch [8] via one pair of input and output ports. Each burst store is composed of a collection of queues assigned to individual flows, using a shared memory. As the core ATM switch size grows, the number of burst stores increases. If the network is engineered so that most bursts are switched directly to an output link, we can expect high statistical multiplexing gains, so as the switch size grows, the number of burst stores required will grow more slowly than the size of the switch. This allows us to obtain considerable economy of scale.

In this paper, our main concern is the delay and the cell loss performance of the burst store. For more details on the Dynaflow protocol, refer to [2]. We develop a mathematical model of the burst store and analyze it to assess the performance of the dynaflow service and to study how different control policies for the burst store affect its performance. We compare the performance of the Dynaflow service and the Fast Reservation Protocol (FRP) service for a tandem node configuration.

The organization of the rest of the paper is as follows. In Sect. 2, we develop the mathematical model and derive performance measures such as the delay, cell loss rate, and end-end delay. We compare the performance of the Dynaflow service and the FRP service using a tandem node configuration. In Sect. 3 we address the design trade-offs for the burst store and show the relationship between the cell loss rate and the buffer size. Section 4 concludes the paper.

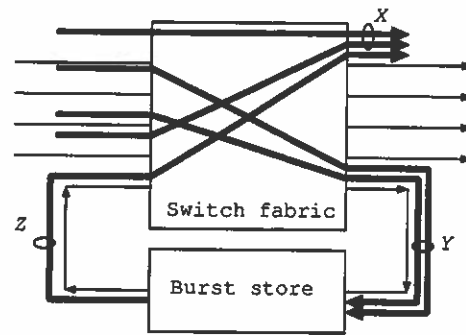


Fig. 3 Definitions of X, Y, and Z.

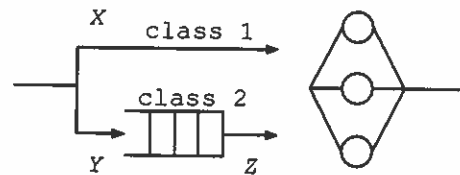


Fig. 4 Equivalent queueing model.

## 2. Performance Analysis

We developed a mathematical model to analyze the loss ratio and the delay time of the Dynaflow service. In this section, we assume that the buffer size of the burst store is large enough that the cell loss occurs only due to the lack of the bandwidth of the link between the burst store and the core switch fabric.

### 2.1 Modeling

We focus on a certain output link. If the output link bandwidth is available, a newly arriving burst is forwarded from input to output link directly. If the newly arriving burst is rejected due to the lack of output link bandwidth, it is diverted to the burst store. The burst store is connected to the switching fabric via input and output ports. Let  $X$ ,  $Y$ , and  $Z$  denote the numbers of flows directly forwarded from input to output link, flows diverted to the burst store, and flows served at the burst store (See Fig. 3).

Taking the output link capacity as unity, the VC peak rate is a  $1/C$  of the link capacity, where  $C$  is an integer. The output link is modeled by a queueing system with  $C$  servers. The equivalent queueing model is shown in Fig. 4. In the queueing model, there are two classes: class 1 flows are forwarded directly from input to output link and class 2 flows are diverted to the burst store. If the newly arriving burst finds that  $X < C$ , it is marked as class 1 and forwarded directly from input to output. Otherwise the burst is marked as class 2 and diverted to the burst store. Note that  $Z = \max(C - X, Y)$ .

2.2 Steady State Distribution

We assume that a flow arrives according to Poisson process with rate  $\lambda$ . Each flow has an exponentially distributed burst duration with mean of  $\mu^{-1}$ . Let us define the probability distribution for  $X$  and  $Y$  at time  $t$ .

$$\pi(x, y; t) \equiv \text{Prob}\{X = x, Y = y \text{ at time } t\}.$$

This system can be regarded as a quasi birth and death process [9], whose state is denoted by  $(x, y)$ , where  $x \geq 0, 0 \leq y \leq C$ . The states are ordered in the lexicographic order. Namely the steady state probability of the  $((C + 1)x + y)$ -th state is denoted by  $\pi_{(C+1)x+y}$  and it is defined as

$$\pi_{(C+1)x+y} \equiv \lim_{t \rightarrow \infty} \pi(x, y; t).$$

The state transition diagram is shown in Fig. 5. The infinitesimal generator of this quasi birth and death process is expressed in terms of  $(C + 1) \times (C + 1)$  submatrices,  $B_0, B_1, A_0, A_1,$  and  $A_2$  as in Eq.(1).

$$Q = \begin{pmatrix} B_0 & A_0 & 0 & \cdots & \cdots & \cdots \\ B_1 & A_1 & A_0 & 0 & \cdots & \cdots \\ 0 & 2A_2 & A_1 & A_0 & 0 & \cdots \\ \vdots & 0 & 3A_2 & A_1 & A_0 & 0 \\ \vdots & \vdots & 0 & \ddots & \ddots & \ddots \end{pmatrix}. \quad (1)$$

The contents of the sub-matrices are given in Appendix A. Letting  $\pi_k = (\pi_{k(C+1)}, \dots, \pi_{k(C+1)+C})$ , the stationary probability vector  $\pi_k$  can be obtained using the relationship in Eq. (2).

$$\pi_{k-1} = \pi_k C_k, \quad (2)$$

where  $C_k = -kA_2(C_{k-1}A_0 + A_1)^{-1}$  and  $C_1 =$

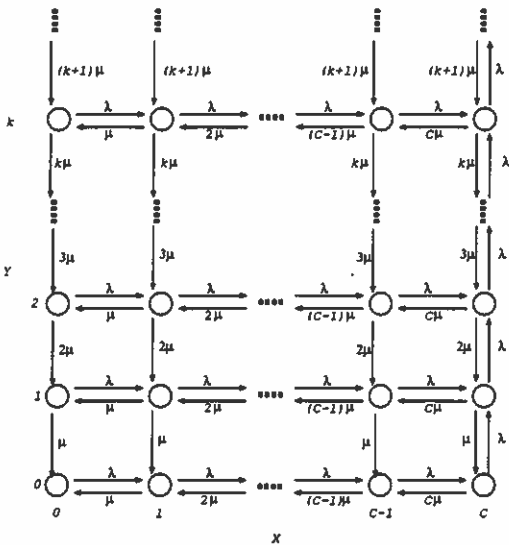


Fig. 5 State transition diagram for  $(X, Y)$ .

$-B_1 B_0^{-1}$ . For the derivation of Eq.(2), see Appendix B. Letting  $D_i = C_i^{-1}$  and  $R_k = \prod_{i=1}^k D_i$ , we have

$$\pi_k = \pi_0 R_k. \quad (3)$$

The normalization condition is given by

$$\sum_{k=0}^{\infty} \pi_k e = 1, \quad (4)$$

where  $e$  is the unit column vector whose elements are all 1 (the dimension of  $e$  should be defined appropriately in the context in what follows). Equation (4) can be rewritten as

$$\pi_0 \left( I + \sum_{k=1}^{\infty} R_k \right) e = 1. \quad (5)$$

In actual calculation, we truncate the infinite summation in Eq.(5). Define  $S_n = I + \sum_{k=1}^n R_k$ . We calculate  $\pi_k$  up to  $k = n$  such that  $S_n \approx S_{n-1}$ , where  $\approx$  indicates the element-wise equality of two matrices. Once  $n$  has been decided,  $\pi_k$  is derived downward from  $k = n$  to 1. We truncate Eq. (1) into  $n \times n$  blocks, each of which is a submatrix whose size is  $(C + 1) \times (C + 1)$ . The relationship of the  $n$ -th block of the truncated infinitesimal generator becomes

$$\pi_{n-1} A_0 + \pi_n A'_1 = 0, \quad (6)$$

where  $A'_1 = A_1 + A_0$ . Substituting the relationship of Eq. (2) into Eq. (6), we have

$$\pi_n (C_n A_0 + A'_1) = 0. \quad (7)$$

Solving Eq. (7) with the nominal normalization condition  $\pi_n e = 1$ , we have  $\pi_n$ . Once we have  $\pi_n$ , we get  $\pi_k$  downward from  $k = n - 1$  to 0 by using Eq. (2). The steady-state probability calculation algorithm is summarized as follows.

Step 1 Initial setting.

Set  $B_0, B_1, A_0, A_1,$  and  $A_2$ .

Step 2 Determine the truncation location  $n$ .

Step 2.1 Set  $S_1 = I + R_1$  and  $k = 2$ .

Step 2.2 Calculate  $C_k = -kA_2(C_{k-1}A_0 + A_1)^{-1}$ ,  $D_k = C_k^{-1}$ , and  $R_k = \prod_{i=1}^k D_i$ . Set  $S_k = S_{k-1} + R_k$ .

Step 2.3 Check the convergence. If  $S_k \approx S_{k-1}$ , then go to Step 2.4. Otherwise  $k = k + 1$  and goto Step 2.2.

Step 2.4 Determine  $n$  as the truncation location. Set  $n = k$ .

Step 3 Calculate  $\pi_n$ . Solve  $\pi_n (C_n^{-1} A_0 + A'_1) = 0$  with the nominal normalization condition  $\pi_n e = 1$ , where  $A'_1 = A_1 + A_0$ .

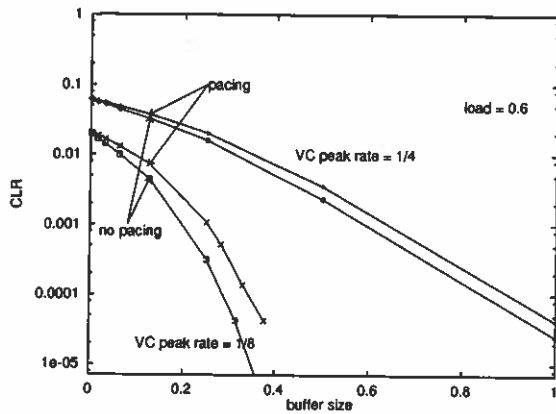


Fig. 16 Impact of pacing factor on CLR due to burst store buffer overflow.

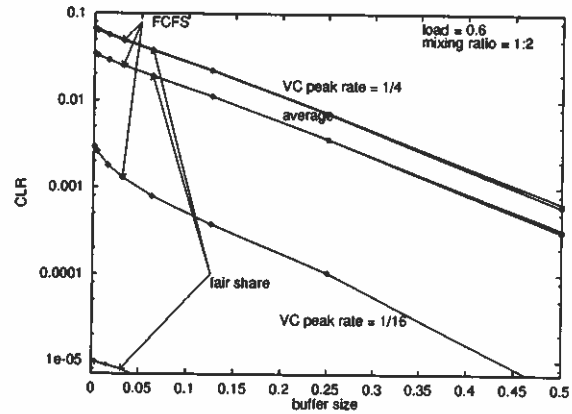


Fig. 18 CLR as a function of the burst store buffer size under heterogeneous traffic (Mixing ratio is 1:1).

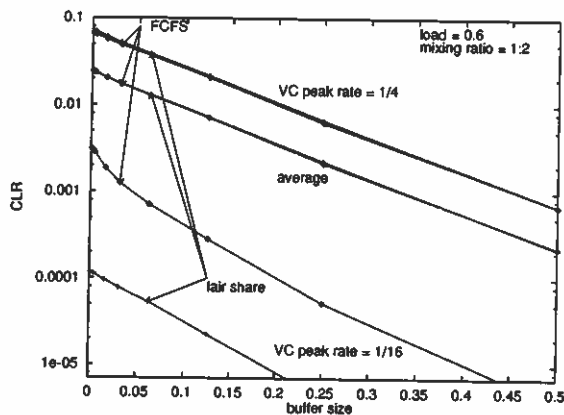


Fig. 17 CLR as a function of the burst store buffer size under heterogeneous traffic (Mixing ratio is 1:2).

### 3.4 Fair Share Bandwidth Allocation

If the residual bandwidth of the output link is smaller than the requested bandwidth of the new flow, the flow whose rate is the highest is selected to be diverted to the burst store. This mechanism allows the flows whose rates are lower than the fair share rate to obtain the requested rate.

We investigate the effect of this mechanism on the performance through computer simulation. We assume two classes of flow. Class 1 flow is high-speed flow, whose peak rate =  $1/4$  while class 2 flow is low-speed flow, whose peak rate =  $1/16$ . The mixing ratio of classes 1 and 2 is 1:2 with respect to the arrival rate of packets. Total offered load is kept at 0.6. The switch is composed of 256 ports and 45 burst stores.

Figure 17 shows the CLR as a function of buffer size. For comparison, we plotted the results by the first come first served (FCFS) discipline. Under FCFS discipline, when the bandwidth of the new flow exceeds the residual bandwidth, the flow is diverted to the burst store. If bandwidth becomes available later and a new flow arrives, the new flow is forwarded from input to

output link directly even if there are some flows in the burst store.

CLR 1, CLR 2 and CLR denotes the CLR experienced by the class 1, class 2 and both of classes, respectively. We observe that CLR 2 is reduced by the fair share discipline compared to the FCFS, while there is little difference between the average CLR and CLR 1. In other words, the CLR of class 2 is improved at the expense of a slight degradation in CLR of class 1 as a result of employing the fair share discipline. Another observation is that CLR 1 is higher than CLR 2 with both disciplines. This is because class 1 is more bursty than class 2 and when class 1 sends data it is likely to see congestion because class 1 itself contributes to the congestion.

Figure 18 shows the CLR as a function of buffer size when the mixing ratio of classes 1 and 2 is 1:1. We observe that the difference in CLR 2 between the FCFS and the fair share schemes is much greater than in Fig. 17. If a class 2 flow arrives at the output port and finds it congested, it is likely that there are class 1 flows currently being forwarded directly to the output link, one of which is selected as the diverted flow on behalf of the new arriving class 2 flow by the fair share scheme. Through these simulation results we found that the fair share discipline reduces the CLR of the less bursty traffic at the expense of a slight degradation in CLR of the bursty traffic.

### 3.5 Alternative Burst Store Selection Policies

Once we have decided to divert the flow to a burst store, we should choose the burst store that is most unlikely to overflow. We developed the burst store selection method in Sect. 3.1. Calculating the time to overflow, however, takes a non-trivial amount of time, which is proportional to the number of diverted flows. Below we consider approximate burst store selection methods.

**Policy 1: Least loaded input rate** At the selection

epoch, we take into account only the input rate of each burst store and select the burst store whose input rate is the smallest. We select burst store  $i$  such that

$$A^{(i)} = \min_j A^{(j)},$$

where  $A^{(i)}$  denotes the total input rate of the  $i$ -th burst store. The advantage of this policy is that the decision is simple. Its disadvantage is that the decision may be sub-optimal since we do not use information about output rate or queue length.

**Policy 2: Least loaded net rate** At the selection epoch, we take into account the input rate and output rate of the burst stores and calculate the net rate. We select the  $i$ -th burst store such that

$$D^{(i)} = \min_j D^{(j)},$$

where  $D^{(i)}$  denotes the total net rate defined as the total input rate minus the total output rate of the  $i$ -burst store. Policy 2 is as simple as policy 1 and more efficient than it.

**Policy 3: Longest time to overflow** We select the burst store that would take the longest time to overflow if we admit the new flow. While the queue length evolution we developed in Sect. 3.1 keeps track of the exact queue length evolution of each individual flow, the method we develop here neglects the evolution of the individual flow's queue. This reduce the computational complexity. We consider two cases.

**Case 1**  $D + d \geq 0$

When the total net rate including the newly diverted flow is positive, the queue is approximated to increase monotonically and eventually overflow after the time  $\frac{Q_{max}-Q}{D+d}$  elapses.

**Case 2**  $D + d < 0$

Since the queue length of the new flow grows monotonically, the burst store eventually overflows even though the total net rate is negative, i.e.,  $D + d < 0$ . This implies that there is a time instant when the total net rate  $D(t)$  becomes positive. The total increasing rate of the burst store queue becomes positive after all the decreasing queues are empty. We assume that all the queues do become empty and calculate the time to overflow as shown in Fig. 19, as  $-\frac{Q}{D+d} + \frac{Q_{max}}{d}$ .

We conducted computer simulation to investigate the effect of these schemes. The simulation results for these approximations are shown in Fig. 20. We used the

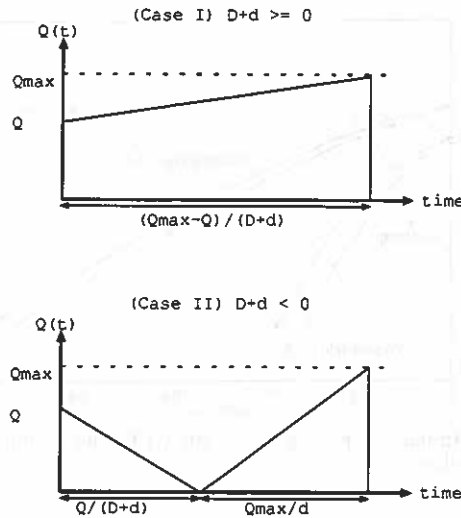


Fig. 19 Rough approximation of time to overflow in the burst store.

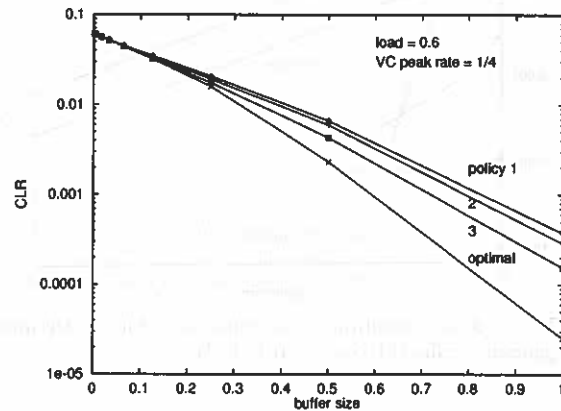


Fig. 20 Comparison of the burst store selection policies.

same assumption as in Fig. 14. We only plot the result for one VC peak rate: 1/4. We also plot the result obtained by an optimal scheme using employ the selection policy developed in Sect. 3.1. All the above policies are simple and their deviations from the optimal scheme are small. Since the exact method requires considerable computation time to sort the queues with respect to  $\tau_k$  and keep track of each individual queue behavior, we argue that these policies are effective.

**4. Closing Remarks**

We presented a new ATM service called *Dynaflow*, in which connections are established on-the-fly and maintained by periodically transmitted control cells. Since the Dynaflow service does not require pre-established connections, it can handle the growing number of short-lived connections such as WWW traffic with a minimum control overhead while guaranteeing the bandwidth needed to transfer the data for the session.

We developed a mathematical model to investigate the performance of a central shared buffer called a "burst store," to which a blocked flow is diverted if the output link is congested. We derived performance measures such as loss ratio, average delay time, and end-end delay time. We compared Dynaflo to the Fast reservation protocol (FRP) and demonstrated that Dynaflo can achieve higher overall throughput due to the elimination of reservation delays, and through the use of shared burst-stores. We also presented a naive design of the burst store architecture and showed the relationship between the CLR and the buffer size. Through computer simulation we demonstrated that the central shared buffer approach reduces the buffer size requirements by a factor of two as the switch size grows from 32 to 256 ports, allowing us to obtain economies of scale.

### Acknowledgments

The authors would like to thank Dr. Shigehiko Suzuki, Mr. Hirokazu Ohnishi, Mr. Zen-ichi Yashiro, and Dr. Naoaki Yamanaka for giving an opportunity to work on this research.

### References

- [1] "http://www.mit.edu/people/mkgray/net".
- [2] Q. Bian, K. Shiomoto, and J.S. Turner, "Dynamic flow switching—A new communication service for ATM networks," Tech. Rep. WUCS-97-26, Washington University, May 1997, <http://www.cs.wustl.edu/cs/techreports/1997/wucs-97-26.ps.z>.
- [3] Q. Bian, K. Shiomoto, and J.S. Turner, "Dynamic flow switching—A new communication service for ATM networks," Proc. of IEEE INFOCOM '98, pp.955–963, 1998.
- [4] D.P. Tranchier, P.E. Boyer, Y.M. Rouaud, and J.Y. Mazeas, "Fast bandwidth allocation in ATM networks," Proc. ISS '92, pp.7–11, 1992.
- [5] J. Turner, "Managing bandwidth in ATM networks with bursty traffic," IEEE Network, vol.6, no.5, pp.50–58, Sept. 1992.
- [6] H. Ohnishi, T. Okada, and K. Noguchi, "Flow control schemes and delay/loss tradeoff on ATM networks," IEEE J. Select. Areas Commun., vol.6, no.9, Dec. 1988.
- [7] ATM Forum, "Traffic management 4.0," April 1996.
- [8] T. Chaney, J.A. Fingerhut, M. Flucke, and J.S. Turner, "Design of a gigabit ATM switch," Proc. IEEE Infocom '97, 1997.
- [9] M.F. Neuts, "Matrix Geometric Solutions in Stochastic Models: An Algorithmic Approach," Johns Hopkins Univ. Press, Baltimore, MD, 1981.
- [10] E. Gelenbe and I. Mitrani, "Analysis and Synthesis of Computer Systems," Academic Press, 1980.
- [11] H. Suzuki and F.A. Tobagi, "Fast bandwidth reservation scheme with multi-link & multi-path routing in ATM networks," Proc. IEEE INFOCOM '92, pp.10A.2.1–10A.2.8, 1992.

### Appendix A: Sub-Matrices

Submatrices of Eq. (1) are given below. The dimensions of the following matrices are  $(C + 1) \times (C + 1)$ .

The submatrix governing the upward transition of  $Y$  is given by

$$A_0 = \begin{pmatrix} 0 & 0 \\ 0 & \lambda \end{pmatrix}. \quad (\text{A} \cdot 1)$$

The submatrices governing the downward transition of  $Y$  are given by

$$B_1 = A_2 = \mu I. \quad (\text{A} \cdot 2)$$

The submatrices governing the transition within the same level of  $y$  are given by

$$B_0 = \begin{pmatrix} d_0 & \lambda & 0 & \dots & \dots & 0 \\ \mu & d_1 & \lambda & 0 & \dots & 0 \\ 0 & 2\mu & d_2 & \lambda & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & 0 & (C-1)\mu & d_{C-1} & \lambda \\ 0 & 0 & 0 & 0 & C\mu & d_C \end{pmatrix}, \quad (\text{A} \cdot 3)$$

where  $d_i = -(\lambda + i\mu)$  for the level  $Y = 0$  and

$$A_1 = B_0 - B_1, \quad (\text{A} \cdot 4)$$

for the level  $Y \neq 0$ .

### Appendix B: Derivation of Eq. (2)

Here, we derive the relationship between  $\pi_{k-1}$  and  $\pi_k$  in Eq. (2).

From Eq. (1), the relationship among the first three probability vectors can be written as

$$\pi_0 B_0 + \pi_1 B_1 = 0, \quad (\text{A} \cdot 5)$$

$$\pi_0 A_0 + \pi_1 A_1 + 2\pi_2 A_2 = 0, \quad (\text{A} \cdot 6)$$

$$\pi_1 A_0 + \pi_2 A_1 + 3\pi_3 A_2 = 0. \quad (\text{A} \cdot 7)$$

From the Eq. (A.5), we have

$$\pi_0 B_0 = -\pi_1 B_1$$

$$\pi_0 = -\pi_1 B_1 B_0^{-1}$$

$$= \pi_1 C_1,$$

where  $C_1 = -B_1 B_0^{-1}$ . From Eq. (A.6), we have

$$2\pi_2 A_2 = -\pi_0 A_0 - \pi_1 A_1$$

$$= -\pi_1 C_1 A_0 - \pi_1 A_1$$

$$= -\pi_1 (C_1 A_0 + A_1)$$

$$\pi_1 = -2\pi_2 A_2 (C_1 A_0 + A_1)^{-1}$$

$$= \pi_2 C_2,$$

where  $C_2 = -2A_2 (C_1 A_0 + A_1)^{-1}$ . From Eq. (A.7), we have

$$3\pi_3 A_2 = -\pi_1 A_0 - \pi_2 A_1$$

$$= -\pi_2 C_2 A_0 - \pi_2 A_1$$

$$= -\pi_2 (C_2 A_0 + A_1)$$

$$\pi_2 = -3\pi_3 A_2 (C_2 A_0 + A_1)^{-1}$$

$$= \pi_3 C_3,$$

where  $C_3 = -3A_2(C_2A_0 + A_1)^{-1}$ . By induction, we have the relationship:

$$\pi_{k-1} = \pi_k C_k, \tag{A.8}$$

where  $C_k = -kA_2(C_{k-1}A_0 + A_1)^{-1}$ .

**Appendix C: Evolution of Queue Length in a Burst Store**

Here, we derive the formula to calculate the time for the burst store to overflow given by Eq. (16).

Figure A. 1 illustrates the evolution of the length of an individual queue and of the total queue. The total queue length at  $\tau_i$  is derived incrementally below. Since the queue of flow 1 becomes empty at time  $\tau_1$ , that is,  $\tau_1 d_1 = -q_1$ , we have

$$\tau_1 \sum_{j=1}^m d_j = \tau_1 \sum_{j=2}^m d_j - q_1. \tag{A.9}$$

Let  $Q(t) = \sum_{i=1}^m q_i(t)$ . Seeing the evolution of the individual queue length at  $\tau_1$  of Fig. A. 1, we have

$$\begin{aligned} Q(\tau_1) &= Q + \tau_1 d + \tau_1 \sum_{j=1}^m d_j \\ &= Q + \tau_1 d + \tau_1 \sum_{j=2}^m d_j - q_1 \end{aligned}$$

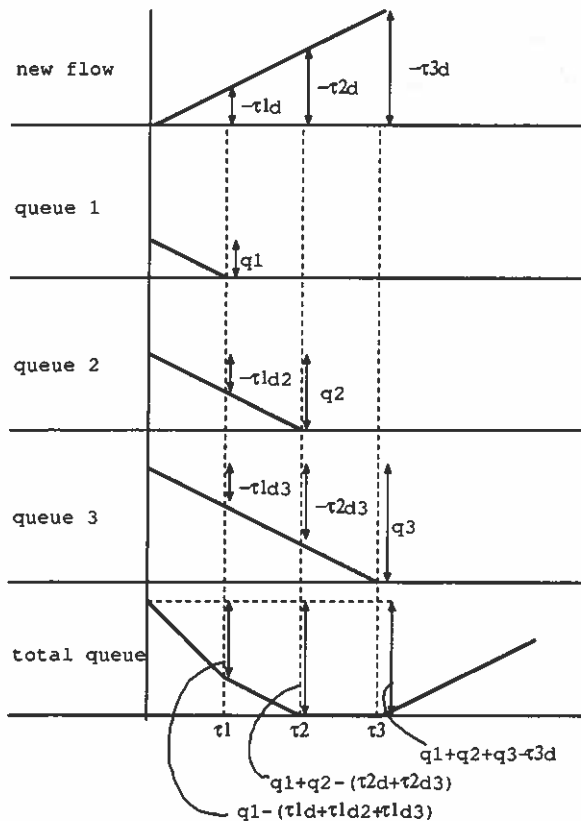


Fig. A. 1 Evolution of total queue length.

$$\begin{aligned} &= Q + \tau_1 d + \tau_1 \sum_{j=1}^m d_j - (\tau_1 d_1 + q_1) \\ &= Q + \tau_1 (d + D) - (\tau_1 d_1 + q_1), \tag{A.10} \end{aligned}$$

where  $D = \sum_{i=1}^m d_i$ . Similarly,

$$\begin{aligned} Q(\tau_2) &= Q + \tau_2 d + \tau_2 \sum_{j=3}^m d_j - (q_2 + q_1) \\ &= Q + \tau_2 (d + D) - \tau_2 (d_2 + d_1) - (q_2 + q_1) \\ &= Q + \tau_2 (d + D) - \sum_{j=1}^2 (\tau_2 d_j + q_j). \tag{A.11} \end{aligned}$$

By induction, we have the queue length at  $\tau_k$ .

$$Q(\tau_k) = Q + \tau_k (d + D) - \sum_{j=1}^k (\tau_k d_j + q_j). \tag{A.12}$$



**Kohei Shiomoto** received the B.E., M.E. and Dr.E. degrees from Osaka University, Osaka, Japan, in 1987, 1989, and 1998, respectively, all in information and computer sciences. He joined the Nippon Telegraph and Telephone corporation (NTT), Tokyo, Japan in 1989. From 1989 to 1994, he was engaged in research and development of ATM switching system in NTT Communication Switching Laboratories. From 1995 to August 1996,

he was engaged in research on traffic controls for B-ISDN at NTT Network Service Systems Laboratories. From August 1996 to September 1997, he was engaged in research on high speed networking as a Visiting Scholar at Washington University in St. Louis. Now he is engaged in research of high speed network architecture at NTT Network Service Systems Laboratories as a Senior Research Engineer. He received the Young Engineer Award of the Institute of Electronics, Information and Communication Engineers from the Institute of Electronics, Information and Communication Engineers (IEICE) in 1995. Dr. Shiomoto is a member of IEEE.



**Qiyong Bian** received the B.S. degree of Computer Science from Tsinghua University, Beijing, China in 1995. He is currently a graduate student at Washington University in St. Louis, U.S.A. His main interest is on the high speed networking and parallel processing.



**Jonathan S. Turner** is a Professor of Computer Science at Washington University and holds the Henry Edwin Sever Professorship. His primary research interest is the design and analysis of switching systems, with special interest in systems supporting multicast communication. In 1994, he received the Koji Kobayashi Computers and Communications Award from the IEEE for his work on the design of multicast switching systems. He

has been awarded about 20 patents for his work in this area and has many widely cited publications. His research interests also include the study of algorithms and computational complexity, with particular interest in the probable performance of heuristic algorithms for NP-complete problems.