Performance Analysis of Dynamic Flow Setup in ATM Networks

Kohei Shiomoto† Qiyong Bian‡ Jonathan S. Turner‡
†NTT Network Service Systems Laboratories
3-9-11 Midori, Musashino, Tokyo 180-8585, Japan
E-mail: Shiomoto.Kohei@nslab.ntt.co.jp
‡Washington Univeristy in St. Louis
One Brookings Drive, St. Louis, MO 63130-4899, U.S.A.
E-mail: {bian,jst}@cs.wustl.edu

Abstract

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 queueing performance of the dynaflow switch and quantify the relationship between the loss ratio and the buffer size.

1 Introduction

1.1 Virtual circuit v.s. datagram

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.

From the viewpoint of carring WWW-like traffic, we compare virtual circuit and datagram services in Fig. 1. Virtual circuit service like asynchronous transfer mode (ATM) is recognized as a platform for future wide-area B-ISDNs. Combination of self-routing switching fabric and labels attached to cells to identify a circuit enable high speed cell switching. 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 in domains of time and space when processing the many identically addressed packets in a large web transfer. In addition, conventional datagram services cannot allocate link bandwidth to a data transfer or make intelligent routing decisions based on bandwidth requirements.

1.2 Dynaflow service

The *Dynaflow* service, introduced in [2], has characteristics of both virtual circuits and datagrams. Like a datagram service, it requires no prior end-to-end session establishment,

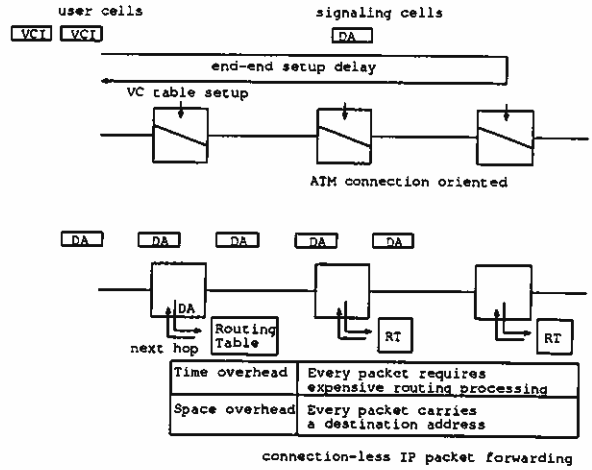

Figure 1: Comparison of virtual circuit and datagram services.

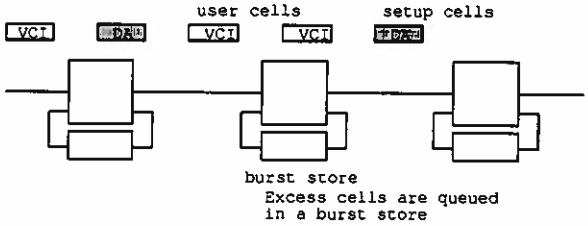

Figure 2: Dynaflow service.

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 [3, 4, 5, 6], 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. When a switching node receives setup cells, it creates a state information regarding association of incoming and outgoing VCIs attached to cells, routing tag, required bandwidth. Subsequent cells are identified by VCI. Data cells are switched by VC table lookup and VC swapping, both of which are performed in $O(1)$. Routing table lookup is performed only for setup cells carring a destination address. 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

0-7803-4788-9/98/$10.00 © 1998 IEEE.

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 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 [7] 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.

1.3 Organization of this paper

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 Section 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 Section 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.

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).

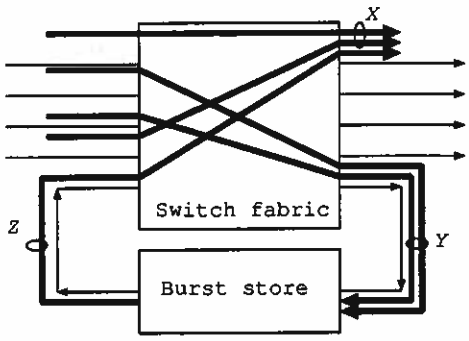

Figure 3: Definitions of X, Y, and Z.

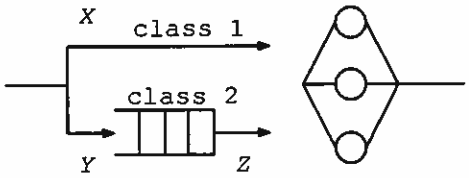

Figure 4: Equivalent queueing model.

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 λ. Each flow has an exponentially distributed burst duration with mean of μ^{-1}. Let us define the probability distribution for X and Y at time t.

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

This system can be regarded as a quasi birth and death process [8], 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

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

The state transition diagram is shown in Fig. 5. The infinitesimal generator of this quasi birth and death process is expressed in terms of $(C+1) \times (C+1)$ sub-matrices, $\boldsymbol{B}_0$, $\boldsymbol{B}_1$, $\boldsymbol{A}_0$, $\boldsymbol{A}_1$, and $\boldsymbol{A}_2$ as in Eq. (1).

$$\boldsymbol{Q} = \begin{pmatrix} \boldsymbol{B}_0 & \boldsymbol{A}_0 & \mathbf{0} & \cdots & \cdots & \cdots \\ \boldsymbol{B}_1 & \boldsymbol{A}_1 & \boldsymbol{A}_0 & \mathbf{0} & \cdots & \cdots \\ \mathbf{0} & 2\boldsymbol{A}_2 & \boldsymbol{A}_1 & \boldsymbol{A}_0 & \mathbf{0} & \cdots \\ \vdots & \mathbf{0} & 3\boldsymbol{A}_2 & \boldsymbol{A}_1 & \boldsymbol{A}_0 & \mathbf{0} \\ \vdots & \vdots & \mathbf{0} & \ddots & \ddots & \ddots \end{pmatrix}. \quad (1)$$

The contents of the sub-matrices 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

$$\boldsymbol{A}_0 = \begin{pmatrix} 0 & 0 \\ 0 & \lambda \end{pmatrix}. \tag{2}$$

The submatrices governing the downward transition of Y are given by

$$\boldsymbol{B}_1 = \boldsymbol{A}_2 = \mu \boldsymbol{I}. \tag{3}$$

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

$$\boldsymbol{B}_0 = \begin{pmatrix} d_0 & \lambda & 0 & \cdots & \cdots & 0 \\ \mu & d_1 & \lambda & 0 & \cdots & 0 \\ 0 & 2\mu & d_2 & \lambda & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & 0 & (C-1)\mu & d_{C-1} & \lambda \\ 0 & 0 & 0 & 0 & C\mu & d_C \end{pmatrix}, \tag{4}$$

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

$$\boldsymbol{A}_1 = \boldsymbol{B}_0 - \boldsymbol{B}_1. \tag{5}$$

Lettin $\boldsymbol{\pi}_k = (\pi_{k(C+1)}, \ldots, \pi_{k(C+1)+C})$, we have the relationship:

$$\boldsymbol{\pi}_{k-1} = \boldsymbol{\pi}_k \boldsymbol{C}_k, \tag{6}$$

where $\boldsymbol{C}_k = -k\boldsymbol{A}_2(\boldsymbol{C}_{k-1}\boldsymbol{A}_0 + \boldsymbol{A}_1)^{-1}$ and $\boldsymbol{C}_1 = -\boldsymbol{B}_1\boldsymbol{B}_0{}^{-1}$. For the derivation of Eq. (6), see appendix A. Letting $\boldsymbol{D}_i = \boldsymbol{C}_i{}^{-1}$ and $\boldsymbol{R}_k = \prod_{i=1}^{k} \boldsymbol{D}_i$, we have

$$\boldsymbol{\pi}_k = \boldsymbol{\pi}_0 \boldsymbol{R}_k. \tag{7}$$

The normalization condition is given by

$$\sum_{k=0}^{\infty} \boldsymbol{\pi}_k \boldsymbol{e} = 1, \tag{8}$$

where $\boldsymbol{e}$ is the unit column vector whose elements are all 1 (the dimension of $\boldsymbol{e}$ should be defined appropriately in the context in what follows). Eq. (8) can be rewritten as

$$\boldsymbol{\pi}_0 \left(\boldsymbol{I} + \sum_{k=1}^{\infty} \boldsymbol{R}_k \right) \boldsymbol{e} = 1. \tag{9}$$

In actual calculation, we truncate the infinite summation in Eq. (9). Define $\boldsymbol{S}_n = \boldsymbol{I} + \sum_{k=1}^{n} \boldsymbol{R}_k$. We calculate $\boldsymbol{\pi}_k$ up to $k = n$ such that $\boldsymbol{S}_n \approx \boldsymbol{S}_{n-1}$, where $\approx$ indicates the element-wise equality of two matrices. Once n has been decided, $\boldsymbol{\pi}_k$ 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

$$\boldsymbol{\pi}_{n-1}\boldsymbol{A}_0 + \boldsymbol{\pi}_n \boldsymbol{A}'_1 = \boldsymbol{0}, \tag{10}$$

where $\boldsymbol{A}'_1 = \boldsymbol{A}_1 + \boldsymbol{A}_0$. Substituting the relationship of Eq. (6) into Eq. (10), we have

$$\boldsymbol{\pi}_n(\boldsymbol{C}_n\boldsymbol{A}_0 + \boldsymbol{A}'_1) = \boldsymbol{0}. \tag{11}$$

Solving Eq. (11) with the nominal normalization condition $\boldsymbol{\pi}_n \boldsymbol{e} = 1$, we have $\boldsymbol{\pi}_n$. Once we have $\boldsymbol{\pi}_n$, we get $\boldsymbol{\pi}_k$ downward from $k = n-1$ to 0 by using Eq. (6).

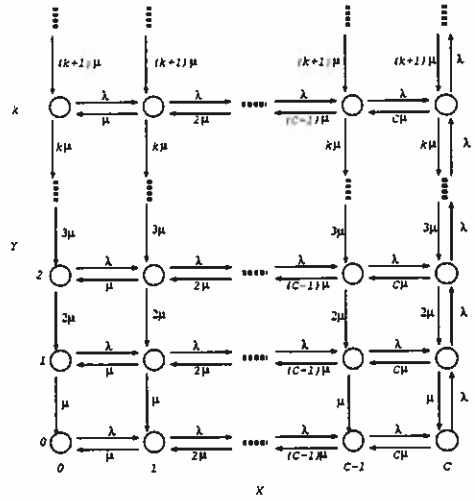

Figure 5: State transition diagram for (X,Y).

2.3 Cell loss ratio

From the steady-state distribution, we have the distribution of the number of flows diverted from a single output link, $f_Y(y) \equiv Prob\{Y = y\}$ as follows.

$$f_Y(y) = \sum_{x=0}^{C} \pi(x,y) \qquad \text{for } y = 0\ldots. \tag{12}$$

Let W denote the total number of flows diverted from all the output links, i.e., $W = \sum_{i=1}^{M} Y_i$, where M denotes the number of output links and Y_i denotes the number of diverted flows from i-th output link. $f_W(w) \equiv Prob\{W = w\}$ is the M-fold convolution of $f_Y(w)$.

$$f_W(w) = f_Y \underbrace{*\ldots*}_{M \text{ times}} f_Y(w). \tag{13}$$

We assume that the link capacity between the burst store and the ATM switch is equivalent to the input and/or output link of ATM switch, i.e., C and there are N_b burst stores are connected to the ATM switch. We have the cell loss ratio (CLR).

$$CLR = \frac{1}{A}\int_0^{\infty} (w - N_b C)^+ f_W(w)dw, \tag{14}$$

where $A = M\lambda/C\mu$ denotes the average arrival rate offered to the entire switch and

$$(x)^+ = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Figure 6 shows that the relationship between the CLR and the number of the burst stores. We set the number of switch ports to 256 and the offered load $\rho = \lambda/C\mu$ is 0.6. VC peak rate is changed from 1/16 to 1 of the link capacity. In Fig. 6 we observe that the CLR drops sharply after a certain number of burst store ports. This knee point corresponds to the average number of diverted flows. For example, if the VC peak rate is 1/8, the average number of diverted flows from one output link $\bar{Y}$ is found to be 0.33. It follows that the total number of diverted flows from all the output links $\bar{W}$ becomes 256×0.33 = 84 flows = 10.6 (=84/8) burst stores.

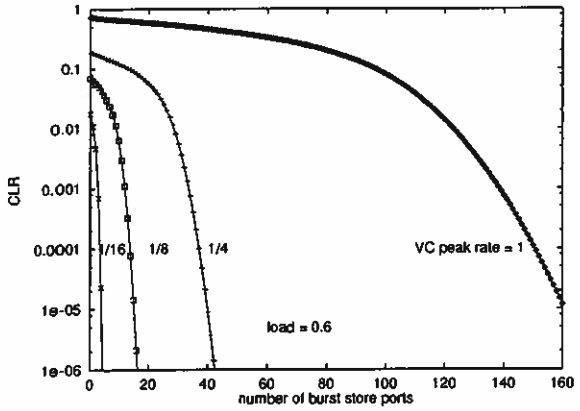

Figure 6: Cell loss ratio versus the number of burst store ports.

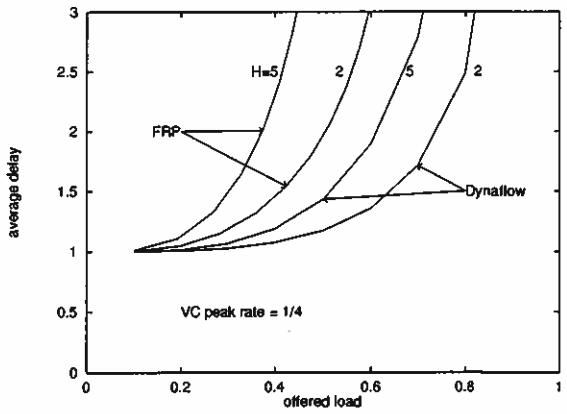

Figure 7: Comparison of end-end delay of FRP and Dynaflow.

The CLR curves drops sharply after 10.6 burst store ports in Fig. 6 for the VC peak rate = 1/8. If the VC peak rate is 1/8, then 17 burst store ports are sufficient to support a CLR of less than 1.0e-6. As the switch size grows, we can expect the number of burst stores needed to maintain the CLR below 1.0e-6 grows more slowly.

2.4 End-to-end delay

If both class 1 and 2 traffic flows are treated equally in the queueing system in Fig. 4, this system is regarded as an $M/M/m$ queueing system. We derive the mean system time (queueing delay plus burst transmission time) of class 1 and 2 in terms of the mean system time of the $M/M/m$ queueing system.

This system is a work conserving system with the assumption of the Poisson arrival and the exponential service time. Applying the work conserving law [9], we have

$$\rho_1 T_1 + \rho_2 T_2 = \rho T, \tag{15}$$

where ρ_i and T_i denote the offered load and mean system time of class i, respectively and ρ and T denote the offered load and mean system time of the $M/M/m$ queueing system, where $\rho = \lambda/C\mu$. Since class 1 traffic is not queued, the mean system of class 1 is given by the mean burst duration μ^{-1}. The offered loads for classes 1 and 2 are given by splitting the total offered load ρ by probability $g_X(C)$, where

$$g_X(x) \equiv Prob\{X = x\} = \sum_{y=0}^{\infty} \pi(x,y) \qquad \text{for } x = 1 \ldots \text{C}. \tag{16}$$

Thus $\rho_1 = \rho(1 - g_X(C))$ and $\rho_2 = \rho$-ρ_1.

Consider a tandem network composed of H nodes. The average queueing delay for class 2 at each node is given by $T_2 - 1/\mu$. Assuming independence among the nodes, the average end-end delay is approximated by

$$D_H = \sum_{k=0}^{H} \binom{H}{k} g_X(C)^k (1-g_X(C))^{H-k} (k(T_2 - 1/\mu) + 1/\mu). \tag{17}$$

Here, we compare the end-end delay of the Dynaflow and the FRP services. To calculate the end-end delay of FRP we employed the method in [10]. Figure 7 shows the end-end delay time as a function of the offered load when H=2 and 5. The delay time is normalized by burst duration in Fig. 7. We assume that a propagation delay time is zero. We observe that while the maximum throughput is limited for FRP as the number of hops increases, the maximum throughput of the Dynaflow is not limited even if the number of hops grows. In FRP, the bandwidth of each hop link all the way to the destination needs to be reserved in advance before the source starts sending burst. As the number of hops grows, this is less likely to be successful on all the links between the source and the destination. On the other hand, Dynaflow allows a burst to be sent to an intermediate node and stored in that node's burst store until the downstream link bandwidth becomes available. Therefore the Dynaflow service outperforms the FRP service with respect to the end-end delay and the maximum throughput.

3 Naive design of burst store

3.1 The burst store selection policy

As the switch size increases, extra burst stores are attached to a Dynaflow switch. In a switch with multiple burst stores, it is crucial to select an appropriate burst store, when a flow is to be diverted. The burst store that is most unlikely to overflow after the flow has been diverted is considered as a candidate. In what follows, we develop a formula for calculating the time for the burst store to overflow.

The burst store is composed of a collection of queues assigned to individual flows (henceforth we call it a *flow queue*). At a certain instant, some flow queues may increase and the others may decrease. Once a decreasing flow queue becomes empty, the flow queue does not contribute to the total queue length evolution. Therefore the total queue length evolves in a complicated manner.

Let $a_i(t)$, $b_i(t)$, $d_i(t)$, and $q_i(t)$ denote input, output, net rate and length of queue i at time t. The net rate is defined as the input rate minus the output rate of the queue, $d_i(t)$=$a_i(t) - b_i(t)$. For brevity, we omit the time t of notations when $t = 0$. We consider a new flow is going to be diverted to the burst store at $t = 0$. Let the new flow's input, output, net rate, and queue length be denoted by $a(t)$, $b(t)$, $d(t)$, and $q(t)$.

Suppose there are m queues (queue 1,...m) in the burst store and n out of them are monotonically decreasing at $t = 0$. Let τ_i denote the time for the queue of flow i to become empty, i.e., $\tau_i = -q_i/d_i$ for $1 \le i \le n$. Rearrange the suffix so that $i < j$ if $\tau_i < \tau_j$. The increasing and/or decreasing rate of the total queue length changes at τ_i $(i = 1, \ldots, n)$.

The total queue length $Q(\tau_k)$ at τ_k is given by

$$Q(\tau_k) = Q + \tau_k(d + D) - \sum_{j=1}^{k}(\tau_k d_j + q_j). \qquad (18)$$

For the derivation of Eq. (18), see Ref [11]. Note that to calculate $Q(\tau_k)$ in Eq. (18), we only have to know about the state of flows 1 through k.

In order to keep track of the total queue length evolution, we sort all the decreasing queues with respect to τ_k. Then we calculate $Q(\tau_k)$ incrementally using τ_i, where $i \leq k$ and check if it is larger than the maximum queue size Q_{max}. If it is larger than Q_{max}, the time of overflow is given by

$$\tau = \frac{(Q(\tau_k) - Q_{max})\tau_{k-1} + (Q_{max} - Q(\tau_{k-1}))\tau_k}{Q(\tau_k) + Q(\tau_{k-1})}. \qquad (19)$$

When we calculate all the $Q(\tau_k)$, $k = 1, \ldots, n$ and find that $Q(\tau_n) \leq Q_{max}$, the time to overflow is given by

$$\tau = \frac{Q_{max} - Q(\tau_n)}{d + D - \sum_{i=1}^{n} d_i}. \qquad (20)$$

We cannot estimate the exact time to overflow since another new flow might arrive or depart before the buffer overflows, causing τ_k to change. Despite this, this selection policy is the most reasonable one which we could take at the decision time. For the time being, we assume that this selection policy is used. We investigate alternative selection policies to relax the computational complexity in Section 3.3.

3.2 Buffer size requirements

We conducted computer simulation to investigate the CLR due to burst store buffer overflow. We assume that offered load = 0.6, switch size = 256 ports excluding burst stores ports, and the VC peak rate = 1, 1/4, 1/8, 1/16 and the number of burst stores is 180, 45, 20, 9, respectively. Figure 8 shows the CLR as a function of the burst store buffer size. The buffer size is normalized by the number of output links and the average burst length. The buffer size is distributed among all the burst stores evenly. From Fig. 8 we can dimension the buffer size required to achieve the target CLR. For example a buffer size of about 1.1 bursts per output link is required for VC peak rate = 1/4 to keep the CLR less than 1e-5. Since this buffer is distributed over 45 burst stores and there are 256 ports, each burst store requires 5.8 (=1.1×256/45) bursts in this case. So a burst store of 4 MBytes could handle an average burst size up to 700 kBytes and a burst store of 16 Mbytes could handle an average burst size up to 2.8 MBytes.

3.3 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 Section 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 three approximate burst store selection methods: (1) least input rate, (2) least net rate, and (3) longest approximate time to overflow. Regarding the method (3), 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

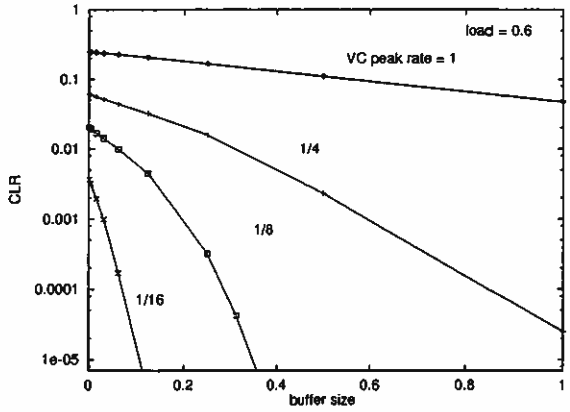

Figure 8: CLR as a function of burst store buffer size.

Section 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 $-\frac{Q}{D+d} + \frac{Q_{max}}{d}$.

We conducted computer simulation to investigate the effect of these schemes. The simulation results for these approximations are shown in Fig. 9. We used the same assumption as in Fig. 8. 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 Section 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 τ_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

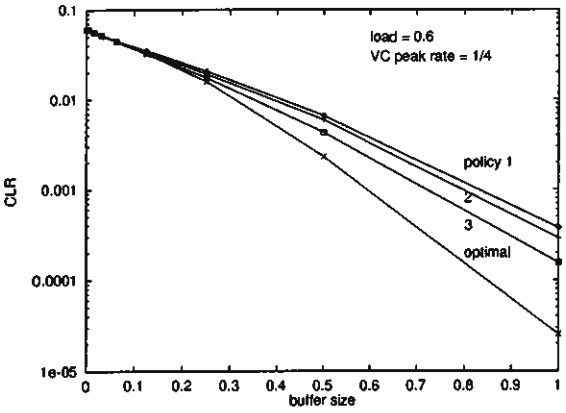

Figure 9: Comparison of the burst store selection policies.

Dynaflow to the Fast reservation protocol (FRP) and demonstrated that Dynaflow 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

This work was performed at Washington University while Kohei Shiomoto was on leave from NTT Network Service Systems Laboratories, Japan. 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.

[3] D. P. Tranchier, P. E. Boyer, Y. M. Rouaud, and J. Y. Mazeas, "Fast bandwidth allocation in ATM networks," in *Proc. of ISS'92*, pp. 7–11, 1992.

[4] J. Turner, "Managing bandwidth in ATM networks with bursty traffic," *IEEE Network*, vol. 6, no. 5, pp. 50–58, Sept. 1992.

[5] 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, pp. 1609–1616, Dec. 1988.

[6] ATM Forum, "Traffic management 4.0," Apr. 1996.

[7] T. Chaney, J. A. Fingerhut, M. Flucke, and J. S. Turner, "Design of a gigabit ATM switch," in *Proc. of IEEE Infocom '97*, 1997.

[8] M. F. Neuts, *Matrix Geometric Solutions in Stochastic Models: An Algorithmic Approach*, Baltimore, MD: John Hopkins Univ. Press, 1981.

[9] E. Gelenbe and I. Mitrani. *Analysis and Synthesis of Computer Systems*, Academic Press, 1980.

[10] H. Suzuki and F. A. Tobagi. "Fast bandwidth reservation scheme with multi-link & multi-path routing in ATM networks," in *Proc. of IEEE INFOCOM'92*, pp. 10A.2.1–10A.2.8, 1992.

[11] K. Shiomoto, Q. Bian, and J. Turner, "Loss and delay analysis of dynamic flow setup in ATM networks ," to appear in IEICE Trans. Commun., Sep. 1997.

Appendix

A Derivation of Eq. (6)

Here, we derive the relationship between π_{k-1} and π_k in Eq. (6). From Eq. (1), the relationship among the first three probability vectors can be written as

$$\pi_0 B_0 + \pi_1 B_1 = \mathbf{o} \quad (21)$$

$$\pi_0 A_0 + \pi_1 A_1 + 2\pi_2 A_2 = \mathbf{o} \quad (22)$$

$$\pi_1 A_0 + \pi_2 A_1 + 3\pi_3 A_2 = \mathbf{o} \quad (23)$$

From Eq. (21), we have

$$\begin{aligned} \pi_0 B_0 &= -\pi_1 B_1 \\ \pi_0 &= -\pi_1 B_1 B_0^{-1} \\ &= \pi_1 C_1, \end{aligned}$$

where $C_1 = -B_1 B_0^{-1}$. From Eq. (22), we have

$$\begin{aligned} 2\pi_2 A_2 &= -\pi_0 A_0 - \pi_1 A_1 \\ &= -\pi_1 C_1 A_0 - \pi_1 A_1 \\ &= -\pi_1 (C_1 A_0 + A_1) \\ \pi_1 &= -2\pi_2 A_2 (C_1 A_0 + A_1)^{-1} \\ &= \pi_2 C_2, \end{aligned}$$

where $C_2 = -2A_2(C_1 A_0 + A_1)^{-1}$. From Eq. (23), we have

$$\begin{aligned} 3\pi_3 A_2 &= -\pi_1 A_0 - \pi_2 A_1 \\ &= -\pi_2 C_2 A_0 - \pi_2 A_1 \\ &= -\pi_2 (C_2 A_0 + A_1) \\ \pi_2 &= -3\pi_3 A_2 (C_2 A_0 + A_1)^{-1} \\ &= \pi_3 C_3, \end{aligned}$$

where $C_3 = -3A_2(C_2 A_0 + A_1)^{-1}$. By induction, we have the relationship:

$$\pi_{k-1} = \pi_k C_k, \quad (24)$$

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