16

Analytic Models for Separable Statistical Multiplexing

Keith W. Ross[1] *and Véronique Vèque*[2]
(1) University of Pennsylvania and (2) Université de Paris-Sud
(1) Department of Systems, University of Pennsylvania, Philadelphia, PA 19104, USA. Telephone: 215-898-6069. Fax: 215-573-2065.
email: ross@eniac.seas.upenn.edu
(2) Laboratoire de Recherche en Informatique - CNRS, Université Paris-Sud, 91405 ORSAY CEDEX, FRANCE.
Telephone: 33-1-69416702. Fax: 33-1-69416586. email: vv@lri.fr

Abstract

We investigate a multiplexing scheme for ATM that statistically multiplexes VCs of the same service, but does not statistically multiplex across services. The scheme is implemented by allocating bandwidth to each service. In the static version, the allocations are fixed ; in the dynamic version, the allocations depend on the numbers of VCs in progress. Under minimal assumptions, we show that the distribution of the VC configuration has a product form. We use the product-form result to construct an efficient convolution algorithm to calculate VC blocking probabilities. We give a numerical example that demonstrates the rapidity of the algorithm and the potential efficiency of separable statistical multiplexing.

Keywords

Admission control, loss networks, performance evaluation, statistical multiplexing.

1 INTRODUCTION

It has long been known that statistical multiplexing of cell streams of the same service type can be highly cost efficient. This is true for delay-sensitive as well as delay-insensitive services. For example, statistical multiplexing of packet streams emanating from voice sources has long been used by telephone companies to increase efficiency, particularly on overseas links (Sriram, 1993).

S. Fdida et al. (eds.), *Data Communications and their Performance*

On the otherhand, statistical multiplexing of VCs across services rarely gives significant gains in performance when services have greatly different QoS (Quality of Service) requirements or greatly different cell generation properties (Gallassi,1990), (Takagi, 1991), (Bonomi, 1993). Indeed, if services with greatly different QoS requirements are statistically multiplexed, then an overall QoS must realize the most stringent QoS requirement ; thus some services enjoy an overly generous QoS, leading to inefficient use of resources. Similarly, if services with substantially different traffic characteristics are multiplexed, then the cell loss probabilities for the various sources can differ by more than one order of magnitude ; thus the network has to be engineered for a QoS requirement that may be overly stringent for a large fraction of the traffic.

A more serious problem is that with statistical multiplexing across services it is difficult to determine the acceptance region for admission control. The analytic models of cell loss for multiplexers which integrate multiservice VCs are not always accurate, and they typically rely on dubious assumptions. Determining the acceptance region with discrete-event simulation is also difficult, since the QoS requirements must be verified at each boundary point of the multidimensional acceptance region, and because the cell loss probabilities are minuscule.

In this paper we investigate a multiplexing scheme for ATM that statistically multiplexes VCs of the same service, but does not statistically multiplex across services. We refer to this scheme as separable statistical multiplexing. In many scenarios this scheme is almost as efficient as statistical multiplexing across and within services. Moreover, determining the acceptance region for separable statistical multiplexing is substantially easier, whether by analytic models or by discrete-event simulation.

Although separable statistical multiplexing has been proposed by many authors, under different names, analytic models to evaluate its VC-level performance are not available in the literature to the best of our knowledge. Explicitly taking into account cell-level QoS requirements of the heterogeneous services, we develop an analytic model for estimating VC blocking probability for separable statistical multiplexing. We make only two assumptions in our model: (1) VC establishment requests arrive according to Poisson processes ; (2) If a VC establishment request finds insufficient resources available, it is blocked and lost. We make no assumptions about the distribution of VC holding times, nor about the cell generation processes of the heterogeneous sources. Our analytic model leads to an efficient convolution algorithm to calculate VC blocking probabilities.

In Section 2 we define separable statistical multiplexing. In Section 3 we develop an efficient convolution algorithm to calculate VC blocking probability. In Section 4 we present some examples and numerical results.

2 SEPARABLE STATISTICAL MULTIPLEXING

Types of services

Going by different names, separable statistical multiplexing has been proposed for ATM by many authors (for example, Gallassi et al (1990), Sriram (1993), Bonomi et al(1993)). We describe this scheme with the aid of Figure 1. In Figure 1 there is a multiplexer that schedules for transmission on the link the cells that are queued in the buffers. Each buffer

aggregates the cell streams from one or more VCs. In Figure 1, the first buffer collects cells from VCs emanating from voice sources ; the second from Continuous Bit Rate (CBR) video sources ; the third from Variable Bit Rate (VBR) video sources ; the fourth from LAN-LAN interconnection sources ; and the fifth from delay insensitive sources, such as low-speed data, bulk data, and video delivery. Thus we have classified the VCs into four real-time services and one non-real-time service. The analytic model that we describe in the next section is independent of this classification, however. The number n_k next to the kth real-time service denotes the number of VCs of this service that are currently in progress. Except for the delay-insensitive services, we assume that the VCs belonging to the same service have identical cell generation statistics. Thus, if the multiplexer were to support two different types of video VBR – say, VHS and HDTV quality – then two services would have to be distinguished for video VBR.

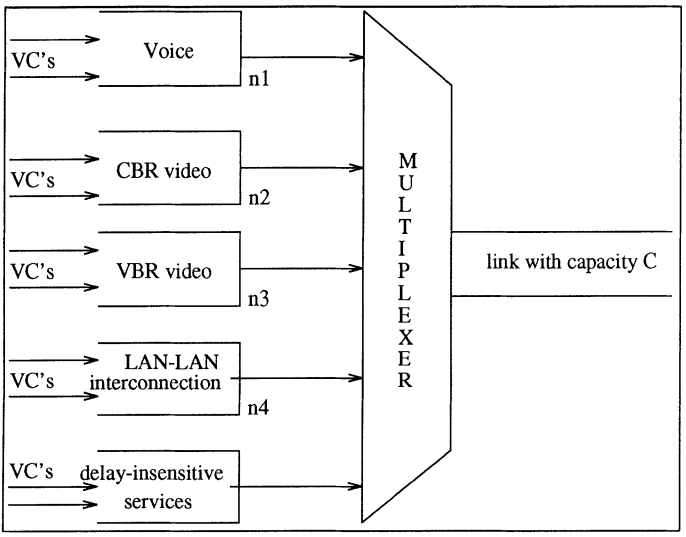

Figure 1 An ATM multiplexer integrating multiple services.

We assume throughout this paper that the buffer capacity allocated to each service is fixed. If a cell of a specific service arrives to find its buffer full, it is lost. Priority schemes for which a high-priority cell pushes out of the buffer a low-priority cell from the same service can also modeled. We neglect such priorities, however, in order not to obscure our main points about admission control.

Each service has a QoS requirement, which might be defined in terms of cell loss, cell delay, cell jitter, or a combination of these measures. The multiplexer must serve each buffer with sufficient frequency in order for the QoS requirements to be met for all VCs in progress. Obviously, the frequency with which the kth buffer must be serviced increases with n_k, the number of service-k VCs in progress.

Equivalent capacity

Before defining separable multiplexing, we digress and consider a link multiplexing n permanent service-k VCs, but no VCs from services other than k. This multiplexer's buffer is the kth buffer in the original multiplexer. Denote $\beta_k(n)$ for the minimum amount of link capacity needed in order for the QoS requirements to be met for the n service-k VCs. We call this function the **service-k capacity function**. Since $\beta_k(\cdot)$ is a function of a single parameter n, it should not be difficult to determine. For a CBR service and for peak-rate multiplexing the capacity function takes the form $\beta_k(n) = b_k n$, where b_k is the bit rate of a single VC. For bursty sources, the capacity function will reflect the economies of scale associated with statistical multiplexing : as n increases, the capacity function will increase, but its slope will decrease.

In particular, since this system involves only one service, it is substantially easier to analyze with discrete-event simulation than analytically a multiplexer which integrates multiple service types and allows statistical multiplexing across services. Furthermore, numerous simple, analytic models are available in the literature for approximating the capacity function for multiplexers with homogeneous sources. These models determine the **equivalent capacity** needed by n bursty homogeneous sources for a given QoS. (There are also in the literature some analytic models for heterogeneous sources, but they are not always accurate and depend on questionable assumptions.) Throughout the remainder of this paper we assume that the capacity functions are known. In Section 4, as an example we shall use one of the popular anaytical models for homogeneous sources to construct capacity functions.

Static Partitions

We now define separable statistical multiplexing. There are two versions: static partitions and dynamic partitions.

Consider again the multiplexer in Figure 1 with link capacity C. It is convenient to generalize the model so that there are K buffers for K delay-sensitive services and another buffer (labeled 0) for all the delay-insensitive services. Partition the capacity C into allocations $C_0, \ldots, C_K$ such that $C_0 + \cdots + C_K = C$. The $K+1$ buffers are served by the link in a weighted round-robin fashion, with the weights being proportional to the capacity allocations. For example, if $K = 2$, $C = 150$, $C_0 = 10$, $C_1 = 40$, $C_2 = 100$, then in a cycle of fifteen cells, the first buffer is served one time, the second four times, and the third ten times. If during the cycle the multiplexer finds one of the buffers empty, it instead serves the 0th buffer (delay-insensitive services). There are several specific algorithms in the literature for weighted round-robin scheduling ; for example see the fluid algorithm of Parekh and Gallager (1992,1993-1) or the dynamic-time-slice algorithm of Sriram (1993). Instead of this schedule, we could also use the Generalized Processor Sharing (GPS) scheduling (Parekh,1993-2) which is generally implemented in ATM multiplexers. In fact, the scheduling scheme has no consequence on our call admission technique.

Separable statistical multiplexing with a static partition admits a newly arriving delay-sensitive service-k VC if and only if $\beta_k(n_k + 1) \leq C_k$ when n_k service-k VCs are already in progress. Thus this scheme statistically multiplexes VCs within the same service k, but does not allow service-k VCs to interfere with service-j VCs for all $j \neq k$. Note that this scheme coupled with the round-robin service mechanism essentially guarantees that

the QoS requirements are met for all VC configurations. We write "essentially" because the cells from the kth service are not served at a constant rate of C_k, as is required in the definition of $\beta_k(n)$. Instead, due to the round-robin discipline, these cells are served at rate C in batches ; but the average service rate is C_k and the fluctuation should be negligible if the granularity of the round robin discipline is sufficient.

Dynamic Partitions

Since VC arrivals are random, there will be time periods when the number of VC establishment requests for a particular service are unusually large. With static partitions, the VC blocking for this service might be excessive during these periods. The following multiplexing scheme alleviates this problem by dynamically allocating bandwidth to the services. It is similar to the scheme proposed by Gallassi et al (1990) and to the scheme proposed by Sriram (1993). Let β_0 be a number less than C.

We again assume that the buffers are served by the link in a weighted round-robin fashion, but now with the weights being proportional to $\beta_0, \beta_1(n_1), \ldots, \beta_K(n_K)$. For example, suppose $K = 2$, $n_1 = 4$, $n_2 = 6$, $\beta_1(4) = 50$ Mbps, $\beta_2(6) = 80$ Mbps, and $\beta_0 = 10Mbps$. Then in a cycle of 15 cells, the first buffer is served 5 times, the second eight times, and the third (for time-insensitive services) is served two times (once for its allocation and once because there is a free slot in the cycle). Again, if during a cycle the multiplexer finds one of the buffers empty, then it instead serves the buffer for delay-insenstive services. Thus the round-robin weights dynamically change, but on the relatively slow time scale of VC arrivals and departures.

Separable statistical multiplexing with dynamic partitions admits a newly arriving service-k VC, $k = 1, \ldots, K$, if and only if

$$\beta_1(n_1) + \cdots + \beta_k(n_k + 1) + \cdots + \beta_k(n_K) \leq C - \beta_0. \tag{1}$$

This scheme again statistical multiplexes the VCs of the same service, but it does not limit a service to a fixed bandwidth allocation. Indeed, any one delay-sensitive service can consume up to $C - \beta_0$ of the bandwidth over a period of time. This scheme coupled with a dynamic round-robin service mechanism essentially guarantees that the QoS requirements are met for all VCs.

3 PERFORMANCE EVALUATION

In order to simplify the discussion, we henceforth assume that all services are delay-sensitive. Thus there is no longer a buffer delay-insensitive traffic in our model. We also assume that service-k VC establishment requests arrive according to a Poisson process with rate λ_k. The holding time of a service-k may have an arbitrary distribution ; denote $1/\mu_k$ for its mean. Also let $\rho_k := \lambda_k/\mu_k$.

We can easily analyze VC blocking for static partitions. The maximum number of service-k VCs that can be present in this system is $\lfloor \beta_k^{-1}(C_k) \rfloor$. Since there is no interaction between services, the probability of blocking a service-k VC is given by the Erlang loss formula with offered load ρ_k and capacity $\lfloor \beta_k^{-1}(C_k) \rfloor$.

Each partition $(C_1, \ldots, C_K)$ defines one static partition policy. If we define a revenue

rate r_k for each service k, we can employ dynamic programming to find the optimal separable multiplexing policy with static partitions ; see Ross (1995).

For the remainder of this paper we focus on separable statistical multiplexing for dynamic partitions. The set of all possible VC configurations for this scheme is

$$\Lambda^s := \{\mathbf{n} : \beta_1(n_1) + \cdots + \beta_K(n_K) \leq C\}$$

where $\mathbf{n} := (n_1, \ldots, n_K)$ is a VC configuration. Of course Λ^s is a subset of Λ, the set of all possible VC configurations that meet the QoS requirements (including those resulting from statistical multiplexing across services). Nevertheless, Λ^s may closely approximate Λ for certain scenarios, in which case little is lost by disallowing statistical multiplexing across services.

We now present a methodology for calculating VC blocking probabilities for separable statistical multiplexing with dynamic partitions. Let $\pi(\mathbf{n})$, $\mathbf{n} \in \Lambda^s$, be the equilibrium probability of being in VC configuration $\mathbf{n}$.

Theorem 1 *The equilibrium probability that the VC configuration is* $\mathbf{n}$ *has the following product form:*

$$\pi(\mathbf{n}) = \frac{1}{G} \prod_{k=1}^{K} \frac{\rho_k^{n_k}}{n_k!}, \quad \mathbf{n} \in \Lambda^s \tag{2}$$

where

$$G := \sum_{\mathbf{n} \in \Lambda^s} \prod_{k=1}^{K} \frac{\rho_k^{n_k}}{n_k!}. \tag{3}$$

Proof. First assume that the holding times are exponentially distributed and that $C = \infty$. Then the stochastic process corresponding to n_k is a birth-death process with equilibrium probability

$$\pi(n_k) = \frac{\rho_k^{n_k}}{n_k!} e^{-\rho_k}. \tag{4}$$

Furthermore, the K birth-death processes are independent, and hence the joint stochastic process corresponding to n is reversible. Imposing a finite value for C corresponds to truncating the state space of the joint stochastic process. The resulting truncated process has the equilibrium probabilities given above (Kelly, 1979). Finally, it follows from standard arguments that this result is insensitive to the holding time distributions (Kelly, 1979). □

The set of VC configurations for which a newly arriving service-l VC is accepted is

$$\Lambda_l^s := \{\mathbf{n} : \beta_1(n_1) + \cdots + \beta_l(n_l + 1) + \cdots + \beta_K(n_K) \leq C\}.$$

Therefore, from Theorem 1, the probability of blocking a newly arriving service-l VC is

$$B_l = 1 - \sum_{\mathbf{n}\in\Lambda_l^s} \pi(\mathbf{n}) = 1 - \frac{\sum_{\mathbf{n}\in\Lambda_l^s} \prod_{k=1}^{K} \rho_k^{n_k}/n_k!}{\sum_{\mathbf{n}\in\Lambda^s} \prod_{k=1}^{K} \rho_k^{n_k}/n_k!}. \tag{5}$$

Thus, to obtain the probability that a service-l VC is blocked, it suffices to calculate the sums in (**??**). One possible approach is to use Monte Carlo summation for loss networks (Ross and al, 1992) (Ross, 1995). Another way for calculating blocking probabilities is to use a recursive algorithm as developed by Kaufman (1981) but it only works when the $\beta_k(n)$ functions are linear ; it not the case here (see figures 2 and 3 as examples of $\beta_k(n)$ functions). Below we give alternative approach based on a convolution algorithm.

Henceforth assume that $\beta_k(n)$ is integer valued. Consider calculating the sum in denominator of (**??**):

$$G := \sum_{\mathbf{n}\in\Lambda^s} \prod_{k=1}^{K} \frac{\rho_k^{n_k}}{n_k!}. \tag{6}$$

Note that

$$\begin{aligned} G &= a \sum_{\mathbf{n}\in\Lambda^s} \prod_{k=1}^{K} e^{-\rho_k} \frac{\rho_k^{n_k}}{n_k!} \\ &= a \sum_{\mathbf{n}\in\Lambda^s} P(Y_1 = n_1, \ldots, Y_K = n_K) \\ &= aP(\beta_1(Y_1) + \cdots + \beta_K(Y_K) \le C) \\ &= a \sum_{c=0}^{C} P(\beta_1(Y_1) + \cdots + \beta_K(Y_K) = c) \end{aligned}$$

where

$$a = e^{\rho_1 + \cdots + \rho_K}. \tag{7}$$

and the Y_k's are independent random variables, with Y_k having the Poisson density

$$P(Y_k = n) = \frac{e^{-\rho_k}\rho_k^n}{n!} \qquad n = 0, 1, 2, \ldots \tag{8}$$

Let

$$g_k(c) = P(\beta_k(Y_k) = c), \quad c = 0, 1, \ldots, C \tag{9}$$

and

$$\mathbf{g}_k = [g_k(0), g_k(1), \ldots, g_k(C)]. \tag{10}$$

Then

$$G = a\sum_{c=0}^{C}(\mathbf{g}_1 \otimes \cdots \otimes \mathbf{g}_K)(c), \tag{11}$$

where $\otimes$ denotes the convolution operator, that is,

$$(\mathbf{g}_1 \otimes \mathbf{g}_2)(c) = \sum_{d=0}^{c} g_1(d)g_2(c-d). \tag{12}$$

Since $\beta_k(\cdot)$ is (almost certainly) an increasing function, it should not be difficult to obtain the $\mathbf{g}_k$'s. The $K-1$ convolutions in (**??**) can be done in a total of $O(KC^2)$ time. (This complexity depends on the granularity of the units for C.) Calculating the numerator in (**??**) can be done in the same manner by replacing $\beta_l(n)$ by $\beta_l(n+1)$ for all n. The techniques in Section 3.5 of Ross (1995) can accelerate the calculation of the K blocking probabilities, $B_1, \ldots, B_K$.

We conclude this section by mentioning some generalizations and extentions. First, the assumption of Poisson arrivals can be relaxed — the same convolution algorithm can be used for arrival rates of the form $\lambda_k(n_k)$ and, in particular, for finite-population arrivals. Second, since derivatives of blocking probabilities can also be represented in terms of normalization constants, the above convolution algorithm can also be used to obtain these performance measures. Third, our model for separable multiplexing can be used to obtain the optimal admission control policy subject to the constraint that the statistical multiplexing is separable ; see Ross (1995).

4 NUMERICAL EXAMPLE

As we mentioned earlier, the capacity functions, $\beta_k(\cdot)$'s can be obtained with discrete-event simulation or approximated analytically. We now outline one analytical approach, due to Guérin et al (1991). For $k = 1, \ldots, K$, assume the following QoS requirement for a service-k VC: No more than the fraction ϵ_k of the VC's cells may be lost.

Digress again and consider a multiplexer supporting n permanent service-k VCs. Assume that each VC alternates between *On Periods* and *Off Periods*. The VC generates cells at the peak rate during an On Period ; it generates no cells during an Off Period. Let b denote the peak rate (in the same units as C) during an On Period. Assume that the lengths of these periods are independent and exponentially distributed. Denote Δ for the average On Period (in seconds). Denote u for the utilization of a VC, that is, the average On Period divided by the sum of the average On Period and the average Off period. Let Q be the capacity of the input buffer and ϵ be the QoS requirement. Guérin et al approximate the capacity function as follows :

$$\beta_k(n) = \min\{\beta_k^{(1)}(n), \beta_k^{(2)}(n)\}, \tag{13}$$

where

$$\beta_k^{(1)}(n) = n[\frac{\ln(\epsilon)\Delta(u-1)b - Q + \sqrt{[\ln(\epsilon)\Delta(u-1)b - Q]^2 + 4Q\ln(\epsilon)\Delta u(u-1)b}}{2\ln(\epsilon)\Delta(u-1)}] \quad (14)$$

and

$$\beta_k^{(2)}(n) = nbu + b\sqrt{nu(1-u)}\sqrt{-2\ln(\epsilon) - \ln(2\pi)}. \quad (15)$$

Clearly this method for estimating $\beta_k(n)$ is quite simple. Note that b, ϵ, Δ, and u are different for different services.

Our numerical example is for a multiplexer of capacity $C = 150$ Mbps, integrating three delay-insensitive services ($K = 3$). We set the buffer capacity, Q, equal to 6 Mbits for each service. We have used the following parameters for the three services as defined in Table 1.

Table 1 Parameters for Multiplexer with Three Services

Class k	Peak Rate b_k	Burst Length Δ_k	Utilization u_k	QoS ϵ_k
1	1Mbps	100msec	0.4	10^{-5}
2	10Mbps	100msec	0.2	10^{-4}
3	5Mbps	100msec	0.5	10^{-6}

We use the above procedure to determine the capacity functions for the three services. We have rounded up all the $\beta_k(n)$'s to the nearest integer. Figure 2 compares the used capacity for service 3 and three allocation schemes : mean rate, peak rate and equivalent capacity. We see that curves for mean rate and peak rate are linear because each time a new connection arrives the capacity increases by 2.5 Mbit for mean rate and by 5 Mbit for peak rate. At the opposite, the $\beta_3(n)$ is not linear. The mean rate function gives always the minimal capacity but does not guarantee the QoS. At the opposite, the peak rate function is maximal.

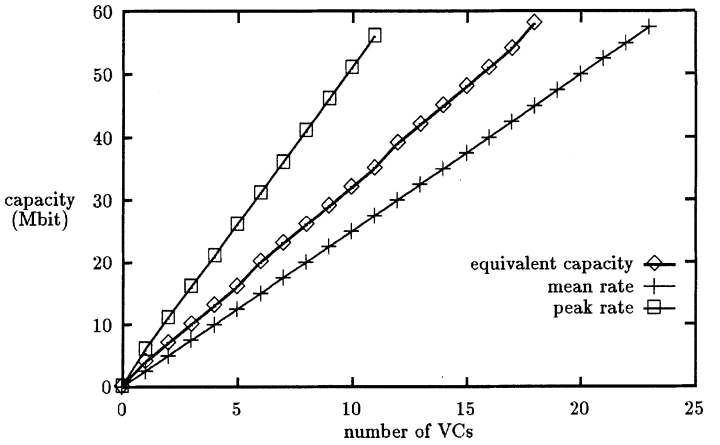

Figure 2 Capacity for peak rate, mean rate and equivalent capacity allocation schemes versus number of VCs (service 3).

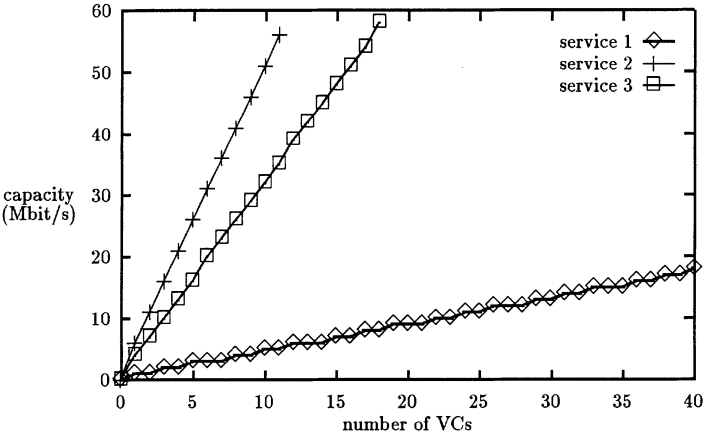

Figure 3 Equivalent capacity for the three types of services depending on the number of VCs.

Figure 3 presents the $\beta_k(n)$'s for the three services versus n. They logically increases with n and depends on their peak rate.

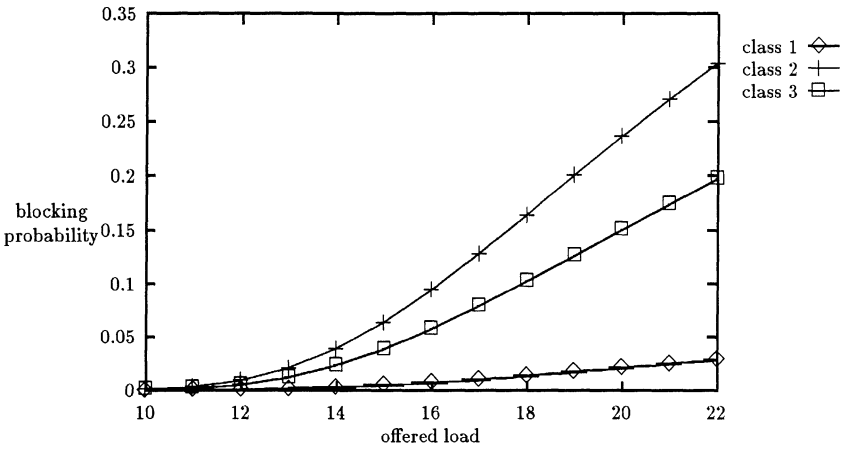

Figure 4 Blocking probabilities for the three classes versus offered load.

Figure 4 presents the blocking probabilities for separable statistical multiplexing obtained from the convolution algorithm. In this and the subsequent figures we set $\rho_1 = \rho_2 = \rho_3$ and plot blocking probabilities as a function of ρ_1. The amount of time required by the convolution algorithm for a specific value of ρ_1 is less than a second on a SPARC 2 workstation.

Figure 4 shows that blocking probabilities depend mainly on the peak rate b_k as b_1 is upper to b_3 which is upper to b_2. They depend also on mean rate $b_k u_k$ of the service, and to a lesser extent on the QoS parameter because probabilities for class 3 are close to those of class 2. As expected, service-1 VCs have the lowest blocking probability because of their low peak and average cell generation rates. It is interesting to note that although class-2 has a lower average rate and a less stringent QoS requirement than class-3, it has a higher VC blocking probability. This is due to its high peak rate, which renders its cell stream very bursty. We also note that VC blocking probabilities greatly vary from service to service.

Figures 5 to 7 compare the performance of separable statistical multiplexing to peak-rate multiplexing. There is one figure for each service. The curves for peak rates are obtained by setting $\beta_k(n) = b_k n$ for all services.

As expected, these figures show that the blocking probabilities for separable statistical multiplexing is less than that for peak-rate multiplexing. What may be surprising is how dramatic this difference in performance can be. For example, with $\rho_1 = 12$, the blocking probabilities for all three services with separable multiplexing is less than 1% ; this blocking probability is roughly 4%, 33%, and 19% for the three services with peak-rate multiplexing. The curves for statistical multiplexing *with* statistical multiplexing across

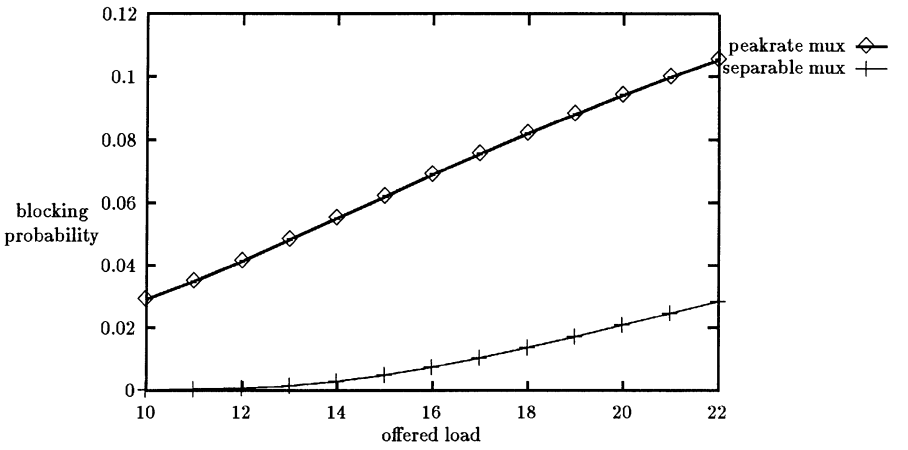

Figure 5 Blocking probabilities versus offered load for service-1 VCs.

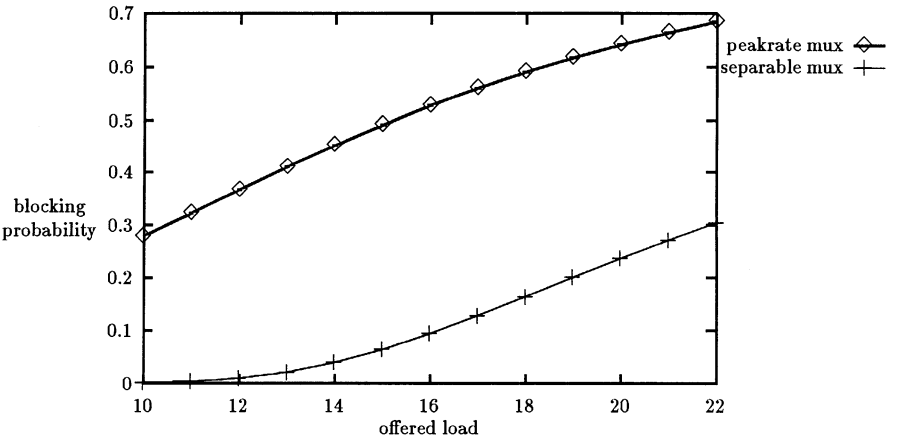

Figure 6 Blocking probabilities versus offered load for service-2 VCs.

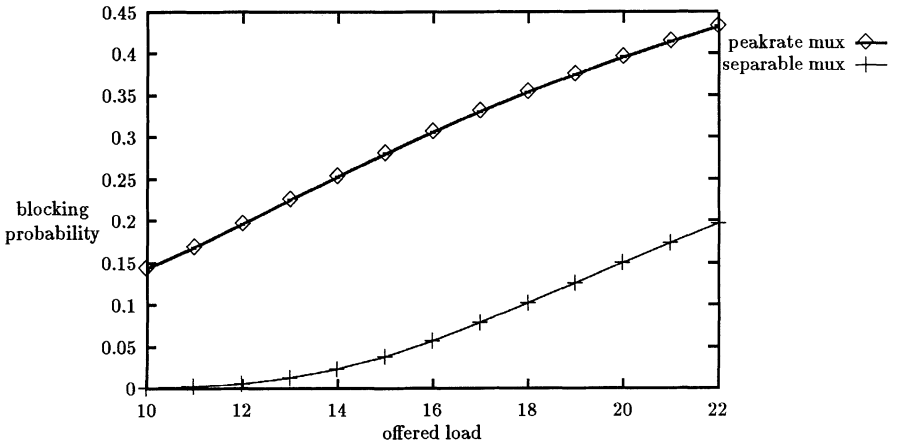

Figure 7 Blocking probabilities versus offered load for service-3 VCs.

classes would lie somewhere below the curves for separable statistical multiplexing. We conjecture that they would be not far below.

5 CONCLUSION

We have developed an efficient convolution algorithm to estimate VC blocking probabilities for separable statistical multiplexing. The numerical results show that separable statistical multiplexing can give substantial gains in performance over peak-rate multiplexing.

There are several related problems that merit attention. (1) A detailed study comparing the blocking probabilities for separable statistical multiplexing with "maximal multiplexing", that is, multiplexing across and within services. Estimating blocking with maximal multiplexing would require discrete-event simulation at the cell. (2) For separable statistical multiplexing, a cell-layer simulation should verify that the QoS requirements are indeed met with the weighted round-robin scheduling disciplines. (3) A theory for separable statistical multiplexing for *networks* should be developed (see Ross, 1995).

REFERENCES

F. Bonomi, S. Montagna, and P. Paglino.(1993) A further look at statistical multiplexing in ATM networks. *Computer Networks and ISDN Systems*, **26**, 119–38.

G. Gallassi, G. Rigolio, and L. Verri.(1990) Resource management and dimensioning in ATM networks. *IEEE Network Magazine*, **05**, 8–17.

R. Guérin, H. Ahmadi, and M. Naghshineh. (1991) Equivalent capacity and its application to bandwidth allocation in high-speed networks. *IEEE Journal on Selected Areas in Communications*, **9**, 968–81.

J.S. Kaufman. (1981) Blocking in a shared resource environment. *IEEE Trans. on Comm.*,**COM-29**, 1474–81.

F.P. Kelly. (1979) *Reversibility and Stochastic Networks.* Wiley, Chichester.

A.K. Parekh and R.G. Gallager. (1992) A generalized processor sharing approach to flow control in integrated services networks. In *Proceedings of IEEE INFOCOM'92.*

A.K. Parekh and R.G. Gallager. (1993) A generalized processor sharing approach to flow control in integrated services networks. In *Proceedings of IEEE INFOCOM'93.*

A.K. Parekh and R.G. Gallager. (1993) A generalized processor sharing approach to flow control in integrated services network: the single node case. *IEEE/ACM Transaction on Networking*, **1**, 344–57.

K. W. Ross. (1995) *Multiservice Loss Models for Broadband Telecommunication Networks.* Springer-Verlag, London.

K.W. Ross and J. Wang. (1992) Monte Carlo summation applied to product-form loss networks. *Probability in the Engineering and Informational Sciences*, 323–48.

K. Sriram. (1993) Methodologies for bandwidth allocation, transmission scheduling, and congestion avoidance in broadband ATM networks. *Computer Networks and ISDN Systems*, **26**, 43–60.

Y. Takagi, S. Hino, and T. Takahashi. (1991) Priority assignment control of ATM line buffers with multiple QOS classes. *IEEE Journal on Selected Areas in Communications*, **9**, 1078–92.