TRAFFIC SHAPING FOR ATM NETWORKS: ASYMPTOTIC ANALYSIS AND SIMULATIONS

by

G. De Veciana and J. Walrand

Memorandum No. UCB/ERL M92/135

8 December 1992

COVER PAGE

TRAFFIC SHAPING FOR ATM NETWORKS: ASYMPTOTIC ANALYSIS AND SIMULATIONS

by

G. De Veciana and J. Walrand

Memorandum No. UCB/ERL M92/135

8 December 1992

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

TITLE PAGE

TRAFFIC SHAPING FOR ATM NETWORKS:
Asymptotic Analysis and Simulations

G. DE VECIANA AND J. WALRAND
Department of Electrical Engineering and Computer Sciences
University of California at Berkeley
Berkeley CA 94720

December 8, 1992

Abstract

We study a simple flow control method for ATM networks called periodic averaging of rate (PARing). This method attempts to reduce the burstiness of traffic by buffering. We present asymptotic results, obtained via the theory of large deviations, on memory requirements, the *multiplexing gain* of PARed sources, and tradeoffs between buffering inside and outside of the network. We discuss some results on the analysis of networks (beyond single queues) when this method is used. These results exhibit the potential for network level interaction between streams. We then propose an improved version of PARing which is akin to using a moving average filter. We present preliminary simulations for video sources.

1 Introduction

Overflows in ATM networks are caused by fluctuations that occasionally make the input rate to a buffer larger than its output capacity. Such fluctuations can be reduced by shaping the traffic to make it more regular. In order to shape the traffic flow, sources temporarily buffer their traffic before releasing it smoothly into the network.

In effect, traffic shaping moves some of the queueing outside or to the edge of the network. By smoothing out its traffic, a source reduces the burstiness of traffic within the network and thereby reduces the need for queueing. The rationale for moving the queueing outside of the network is that source buffers need only accommodate the fluctuations of the traffic of single sources whereas the network buffers must accommodate the fluctuations of the superposition of the traffic streams flowing through them. While the superposition of many traffic streams is statistically more regular than individual streams, it remains nevertheless true that distributing the buffers to the different sources simplifies the network design and allocates more cost to bursty users rather than imposing that cost on the network itself, and therefore, on all the users. Moreover, by individually smoothing out traffic sources, one can avoid situations in which bursty sources penalize other traffic sharing the same resources.

A number of traffic shaping methods have been proposed and studied[14, 1, 7, 12]. We propose *periodic averaging of rate* (PARing) as an alternative to methods such as leaky bucket. PARing has the distinct advantage that its performance can be reasonably estimated and its impact on the network performance and design can be studied.

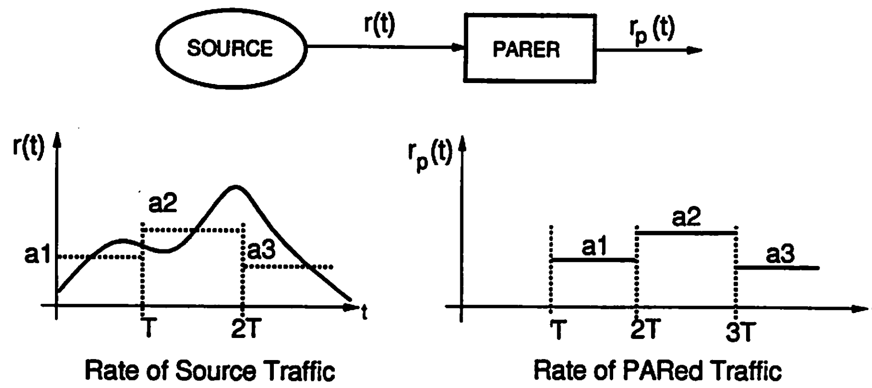

Figure 1: PARing a variable rate source.

Before proceeding, consider the manner in which a traffic source is PARed, see Fig. 1. The figure shows a source producing a time-varying rate $r(t)$. PARing operates as follows :

1) Store output of source during time interval $[0, T)$, $n := 1$;

2) Compute average rate, a_n, of source during $[(n-1)T, nT)$;

3) Release data stored during $[(n-1)T, nT)$ at a rate a_n during $[nT, (n+1)T)$ while simultaneously storing data produced during the epoch, $n := n + 1$;

4) Repeat steps 2 and 3.

The output of the PARer has a time-varying rate $r_p(t)$. During each time interval $[nT, (n+1)T)$, the output rate is constant $r_p(t) = a_n$, the average rate of the source during $[(n-1)T, nT)$.

PARing makes the source as regular as possible within the given interval, while introducing a known delay. This regularity is achieved by averaging the source rate over time. When a buffer is used by a large number of independent sources, they will typically average each other out, e.g. 1000 voice calls sharing a common buffer. In this case PARing will be of little use. However, if each source uses a substantial fraction of the buffer bandwidth, then PARing the traffic will have a significant effect. For instance, if the sources are variable bit rate codecs with a peak rate of 40 Mbps, sharing a transmitter at 622 Mbps, then PARing should greatly reduce the amount of memory required in the switch.

The value of the averaging duration T is a design parameter. As T increases, the traffic becomes smoother, but the delay and the required memory increase. Thus for traffic that is delay-critical, such as video and interactive services, a small value for T should be selected. A sensible choice, would be to take T of the order of the propagation time through the network (i.e. $\sim$ 25 msec). Such a small delay corresponds to a reasonable memory requirement which could be inexpensively accommodated at the network interface. When the traffic is not delay-critical, such as non-interactive file transfers, the value of T could be much larger. In this case memory storage would in principle not be required, as the files are typically stored on disk. Thus the hardware and software requirements for PARing will depend on the nature of the service. We believe that with decreasing memory costs an implementation of PARing would be quite inexpensive, in most cases.

In our PARing framework, a user can select the parameter T of the PARer and the memory size to meet a desired quality of service. The network designer can scale buffers and design call acceptance and routing strategies with known performance criteria. This paper will explore some aspects of performance analysis for networks with PARed sources.

The paper is organized as follows. In §2 we investigate the size of the memory buffers required at the edge of the network to PAR different types of sources. In §3 we discuss the capacity of single links, supporting PARed traffic. In §4, we introduce the notion of *effective bandwidth* for PARed sources, and discuss its properties. We also define the notion of *multiplexing gain* as a means to quantify the benefits of ATM, with PARed sources. In §5 we discuss the basic dilemma of whether to buffer sources inside or outside of a network. We investigate the capacity of networks supporting PARed traffic in §6. In §7 we analyze an improved version of our PARing scheme which is akin to using a moving average filter. In §8 we present some preliminary simulations exhibiting the benefits of PARing. In particular we investigate how successful this traffic shaping technique is for a given video source. Finally, §9 summarizes the main ideas that have been presented and the tradeoffs underlying the PARing idea.

2 Memory Requirements

We first estimate the amount of memory required at the edge of the network where PARing is to take place. An equivalent problem is to evaluate the probability that an unexpectedly large traffic rate is obtained during a given averaging period T. The calculation of such probabilities corresponds to the basic large deviation asymptotic[2]. We use the theory of large deviations to obtain estimates for relatively large averaging periods.

Below we denote the mean arrival rate of a source by m and compute the probability that the empirical average rate exceeds this mean by a margin $\alpha - m$. Under weak assumptions, this probability is approximated by,

$$\mathsf{P}\left(\frac{1}{T}\sum_{i=1}^{T} X_i \geq \alpha\right) \approx \exp[-Th(\alpha)]$$

where X_i is interpreted as the arrivals during the i^{th} time interval and $h(\alpha)$ is the large deviation rate function associated with the source. If the arrivals are i.i.d., the rate function is given by, $h(\alpha) = \sup_\lambda[\lambda\alpha - \Lambda(\lambda)]$, the Fenchel-Legendre transform of the log-moment generating function of a typical increment $\Lambda(\lambda) = \log \mathsf{E} \exp(\lambda X_0)$. When the traffic does not have independent arrivals the log-moment generating function is taken to be $\Lambda(\lambda) = \lim_{t\to\infty} \frac{1}{t} \log \mathsf{E} \exp(\lambda \sum_{i=1}^{t} X_i)$ if it exists, and subject to some technical conditions the same result holds, see the Gärtner-Ellis theorem[6].

In order to find the memory needed for PARing we fix T and determine the memory size B_{par} required to guarantee that the probability that a given cell is lost, P(cell loss), in a PARing interval is less than $\exp(-T\delta)$, i.e., we guarantee that P(cell loss) $\leq \exp(-T\delta)$. Statistical constraints of this nature are useful when some loss can be tolerated. Typically the constraints are quite stringent though, e.g., smaller than 10^{-6}.

Note that B_{par}/T is the maximum traffic rate that can be absorbed by the PARing buffer without loss. Thus the simple calculation that follows gives the required constraint :

$$\mathsf{P}(\text{loss}) = \mathsf{P}(\text{mean rate} \geq B_{par}/T) \approx \exp[-Th(B_{par}/T)] \leq \exp[-T\delta]$$

where P(loss) is the probability that there is any loss in a given PARing interval. This is to be contrasted with the probability that any given cell is lost P(cell loss). However, one can argue that on an exponential scale P(loss) $\approx$ P(cell loss). Indeed, the number of arrivals during a PARing

Type of Source	Exponent $h(\alpha)$
Discrete Time	
Normal: $N(m,\sigma^2)$ i.i.d.	$\frac{(\alpha-m)^2}{2\sigma^2}$
General Gaussian Process: Mean m, Dispersion d	$\frac{(\alpha-m)^2}{2d^2}$
Continuous Time	
Poisson: Rate m	$\alpha\log(\frac{\alpha}{m})+m-\alpha$
Markov Modulated Fluid: $m=\frac{\lambda r_1+\mu r_0}{\lambda+\mu}$	$\frac{\left[\sqrt{(\alpha-r_0)\mu}-\sqrt{(r_1-\alpha)\lambda}\right]^2}{r_1-r_0}$

Table 1: Rate functions for some sources of interest.

interval is $O(T)$ and when loss occurs, the number of lost cells, L, will be small and independent of T. So the asymptotic probability of cell loss, is $\frac{L}{O(T)}\mathrm{P}(\text{loss})$ which is dominated by the exponential term.

This large deviation bound suggests that we should select B_{par} and T such that,

$$h(B_{par}/T) > \delta.$$

Table 1 summarizes some results for sources of interest in modeling ATM traffic where explicit expressions can be obtained. The interested reader is referred to Bucklew[2] for an explanation of the generic large deviation bound for i.i.d. sources such as Gaussian and Poisson. Similar results can be obtained for arrival processes that satisfy strong mixing conditions or the Gärtner-Ellis theorem[6]. An interesting treatment of general Gaussian processes can be found in Courcoubetis and Walrand[3]. In order to analyze the case of Markov fluids, the large deviations of the empirical distribution for the underlying Markov chain may be considered. In our simple example, the source turns off (on) with intensity μ (λ), and generates traffic rate r_0 (r_1) when off (on). More general results for Markov fluids and Markov modulated Poisson processes have been found by Kesidis[11] and deVeciana et al[5].

It is possible to obtain analytical expressions for many other types of sources. The generality of this principle leads us to suspect that any sufficiently well behaved (mixing) stationary ergodic source will probably have an associated rate function $h(\alpha)$, although it may be hard to compute, e.g., video sources. Some analytic models for video sources have been proposed [13, 10, 15], but it is difficult to evaluate if they are appropriate. This depends to a great extent on the phenomena one wishes to study. In particular we are interested in the queueing properties of these sources. An alternative approach is to use a veritable simulation of compressed video, to obtain the rate function directly. Such a study will be discussed in §8.

3 Buffer Asymptotics

Next we determine the effect of PARing within the network. Consider a single buffer,B, shared by N virtual circuits, each having PARed traffic streams A^i_n, with averaging periods T which are synchronized. The evolution of the queue length is given by

$$X_{n+1} = (X_n + A_n T - cT)^+$$

where $A_n = \sum_{i=1}^N A^i_n$, c is the service rate of the buffer and X_n is the buffer occupancy at time nT. Below we consider two regimes. On one hand, for large B and relatively small T, overflows are due to long term *accumulation*. On the other hand, for large T and small B, overflows occur as T time-scale *fluctuations*. The latter is the regime in which we expect PARing to be effective, whence it is important to make precise what are the relevant overflow time-scales, and what is a large T relative to B.

3.1 Overflow Time-scale

First consider large B asymptotics. In this case the queue evolution corresponds to a reflected random walk, whose large deviations have been studied in some detail[16, 11, 6]. We give a brief heuristic derivation of the probability that the free buffer occupancy exceeds a high level B during a busy cycle. As seen in §2 nice sources will have an associated large deviation rate function $h(\alpha)$, characterizing the probability that the empirical mean of the source is α for a relatively long period of time. Because the rate function is convex, one finds that typically paths leading to overflows follow a straight line. Thus, in order to induce an overflow, a source will fire at an approximately constant rate, α, for $B/(\alpha - c)$ seconds. The probability of overflow is then $\exp[-Bh(\alpha)/(\alpha - c)]$, where $\alpha > c$. The most likely slope for this path is that minimizing the cost (rate$\times$time) subject to an overflow :

$$\mathbf{P}(\text{overflow in a busy cycle}) \approx \exp\left[-B \inf_{\alpha > c} \frac{h(\alpha)}{\alpha - c}\right].$$

Let α^* denote the slope of typical overflowing paths, i.e., the minimizer of $h(\alpha)/(\alpha - c)$, the time to overflow is then $B/(\alpha^* - c)$. Thus when $T \ll B/(\alpha^* - c)$, averaging the input rate will not significantly affect overflowing paths since accumulation occurs over a period of time significantly larger than T, and overflowing paths are straight lines. We conclude that PARing will not effect the overflow probabilities in this regime. This result can be extended to the case of multiple sources and will be used in §5 to compare PARing with buffering in the network.

Now consider averaging intervals T which are large relative to the time constant of a typical buffer overflow (without PARing) i.e., $T > B/(\alpha^* - c)$. In this case, overflows in switch buffers occur on the *time-scale* of T rather than as slow accumulations in the evolution equation proposed above. For example, often one finds that $\alpha^* \approx (c - m) + c$ where m is the mean traffic rate.[1] The overflow time-scale is then $B/(c(1 - \rho))$. Thus PARing will be effective when the network buffers B are relatively small, the utilization ρ is small and/or the bandwidth c is large. For other sources similar qualitative results should hold.

Consider for example an M/D/1 queue, with bandwidth $c = 622$Mbps, utilization $\rho = 0.8$, and buffer size $B = 30$ cells. This corresponds to a probability of cell loss of approximately 10^{-6}. The time-scale of overflows in this system will be approximately 0.1 msec. Indicating that PARing intervals on the order of 1 msec, should have a significant impact on the queue dynamics.

[1] Intuitively $c - m$ is the mean rate at which one would expect a large queue buildup to settle down. This indicates that traffic builds up as if relaxations were reversed in time, and holds in particular for M/M/1 and M/D/1 queues[8].

3.2 Loss asymptotics for PARed sources

In computing the loss asymptotics for PARed sources, we will assume that there is no carry-over from one PARing interval to the next. The probability of loss is then simply the probability that the aggregate input traffic rate exceeds the service rate, c, at the given node. The motivation for neglecting carry over is not solely one of convenience. In practice packets of real time sources that are held over one or two PARing intervals will suffer excessive delays. The network should be scaled so that such delays are very unlikely or provide a packet dropping mechanism to relieve congestion.

We compute the probability that the aggregate traffic of N PARed sources, A_n, exceeds the service rate of the buffer over the n^{th} PARing interval,

$$\mathbf{P}(\text{cell loss}) \approx \mathbf{P}(A_n > c) = \mathbf{P}\left(\sum_{i=1}^{N} \frac{1}{T} \sum_{j=nT}^{(n+1)T} X_j^i > c\right).$$

Once again we use large deviations to obtain a suitable estimate. When the sources have independent increments one can argue,

$$\mathbf{P}(\text{cell loss}) \approx \mathbf{P}\left(\frac{1}{T} \sum_{j=nT}^{(n+1)T} \sum_{i=1}^{N} X_j^i > c\right) = \mathbf{P}\left(\frac{1}{T} \sum_{j=nT}^{(n+1)T} Y_j > c\right)$$

where we have interchanged summations to consider the empirical mean of T random variables, Y_j, each being the sum of the arrivals for the N sources over the j^th time interval. Let $h(\alpha)$ denote the rate function associated with the random variables Y_j. The large deviation approximation is,

$$\mathbf{P}(\text{cell loss}) \approx \exp\left[-T \inf_{\alpha > c} h(\alpha)\right] = \exp\left[-Th(c)\right].$$

However, using the independence of the sources we have that,

$$\Lambda(\lambda) = \log \mathbf{E} \exp(\lambda \sum_{i=1}^{N} X_0^i) = \sum_{i=1}^{N} \log \mathbf{E} \exp(\lambda X_0^i) = \sum_{i=1}^{N} \Lambda_i(\lambda).$$

Finally we obtain a useful form for the exponent,

$$\mathbf{P}(\text{cell loss}) \approx \exp\left[-T \inf_{\sum_{i=1}^{N} \alpha_i = c} \sum_{i=1}^{N} h_i(\alpha_i)\right].$$

The intuition is hopefully clear: N sources make a joint effort to supply the buffer a traffic load c, such that the total cost (rate) is minimized. Such heuristic arguments are transparent and can typically be substantiated by detailed calculations; in the sequel we will use this type of reasoning when possible with the understanding that they can be proven. In the case where the sources do not have independent increments the log-moment generating function is taken to be $\Lambda_i(\lambda) = \lim_{t\to\infty} \frac{1}{t} \log \mathbf{E} \exp(\lambda \sum_{j=1}^{t} X_j^i)$ if it exists, and subject to some technical conditions the same results will hold, see the Gärtner-Ellis theorem [6].

We require that $\mathbf{P}(\text{cell loss}) < \exp[-T\delta]$, and obtain the following constraint :

$$\inf_{\sum_{i=1}^{N} \alpha_i = c} \sum_{i=1}^{N} h_i(\alpha_i) \geq \delta. \tag{1}$$

Thus in order to check that a buffer is not overloaded we should solve this optimization problem and check that this constraint is satisfied. Simple ways of dealing with this problem are explored in the next section. We have tacitly assumed throughout that the PARing intervals are synchronized, and we have ignored the effect of PARing losses on network losses. This is in fact the worst case scenario, since averaging across PARing intervals can only help.

4 "Effective Bandwidths" for PARed sources

Given the service rate of a buffer, and the types of sources currently loading it, we need a reasonable scheme for call acceptance. We wish to guarantee that within a given PARing interval, $\mathbb{P}(\text{cell loss}) \leq \exp[-T\delta]$. For clarity, we present a sequence of problems leading to a conservative notion of effective bandwidth which we use in the sequel.

4.1 Single source

Consider a single source of type i, with rate function $h_i(\cdot)$. Define the effective bandwidth $\alpha_i(\delta)$ as the minimum service rate required to maintain our statistical guarantee, i.e., by Eq.1,

$$\alpha_i(\delta) \triangleq h^{-1}(\delta) \triangleq \inf\{\alpha : \alpha > m, h_i(\alpha) > \delta\}.$$

Note that,

$$\alpha_i(\delta) < c \quad \Leftrightarrow \quad \mathbb{P}(\text{cell loss}) \leq \exp[-T\delta],$$

since $h(\cdot)$ is nondecreasing. Loosely speaking we could define the effective bandwidth in a more intuitive fashion as $\alpha_i(\delta) \geq m$ such that $h_i(\alpha_i(\delta)) = \delta$, however for sources with bounded traffic rates, e.g., the on/off Markov fluids discussed in §4.4, not all values of δ are attained. When this is the case we assign an effective bandwidth equal to the maximum source rate.

4.2 Multiple sources of the same type

Now suppose N_i sources of type i, are loading the buffer. Our previous notion of effective bandwidth leads to the following *conservative* bound :

$$N_i\alpha_i(\delta) < c \quad \Rightarrow \quad \mathbb{P}(\text{cell loss}) \leq \exp[-T\delta].$$

One can verify this fact by considering the exponent and using the convexity of $h_i(\cdot)$:

$$\inf_{N_i m_i = c} N_i h_i(m_i) = N_i h_i(c/N_i).$$

Now since $N_i\alpha_i(\delta) < c$, we must have $\alpha_i(\delta) < c/N_i$ so $\delta = h(\alpha_i(\delta)) < h(c/N_i)$ and the exponent of sources satisfying this constraint will actually be larger than $N\delta$, which is clearly conservative.

Alternatively we can define the *joint* effective bandwidth of N_i sources by,

$$\alpha(N_i, \delta) \triangleq N h^{-1}(\delta/N) \triangleq \inf\{\alpha : \alpha > Nm, Nh(\alpha/N) > \delta\}.$$

Note this is not significantly harder to compute, as we are assuming we know $h_i(\cdot)$ or an approximation of it. This choice makes the notion of effective bandwidth tight, i.e., it is indeed the minimum service rate we can tolerate while maintaining the loss requirement.

4.3 Multiple sources and multiple types

In general a buffer may service N types of sources, N_i of each type i. In order to deal with the multiplicity of sources in a *simple* fashion we must resort to a conservative strategy.

We associate with the collection of sources of each type their effective bandwidth $\alpha(N_i, \delta)$, in the sense of the previous section. The acceptance condition becomes,

$$\sum_{i=1}^{N} \alpha_i(N_i, \delta) < c \Rightarrow \mathbf{P}(\text{cell loss}) \leq \exp[-T\delta].$$

This condition can easily be verified by an argument similar to those above, but this strategy is once again somewhat conservative. Indeed the condition is sufficient but not necessary. The proposed call acceptance strategy corresponds to checking if the cumulative effective bandwidth, in the sense above, does not exceed the service rate.

Of course, an alternate strategy would be to compute and generate tables of the acceptable loads for a buffer, by solving the optimization problem associated with the exponent.

One may interpret this scheme as one in which averaging over time and over calls of the same type is permitted, ignoring averaging across types. We observed, in §4.2 that the joint effective bandwidth of N_i sources of the same type, is not N_i times that of a single source. This phenomenon is exactly the idea behind *statistical multiplexing* which motivated the asynchronous transfer mode.

4.4 Statistical multiplexing of PARed sources

If the sources are all of the same type, we can define the *multiplexing gain* $G(N, \delta)$ as the ratio of N times the effective bandwidth of a single source over the joint effective bandwidth of N sources,

$$G(N, \delta) \triangleq \frac{N\alpha(1, \delta)}{\alpha(N, \delta)} = \frac{h^{-1}(\delta)}{h^{-1}(\delta/N)}.$$

Were $h^{-1}(\cdot)$ is the inverse of $h(\cdot)$ defined in §4.1. One can easily convince oneself that the multiplexing gain increases with the number sources N.

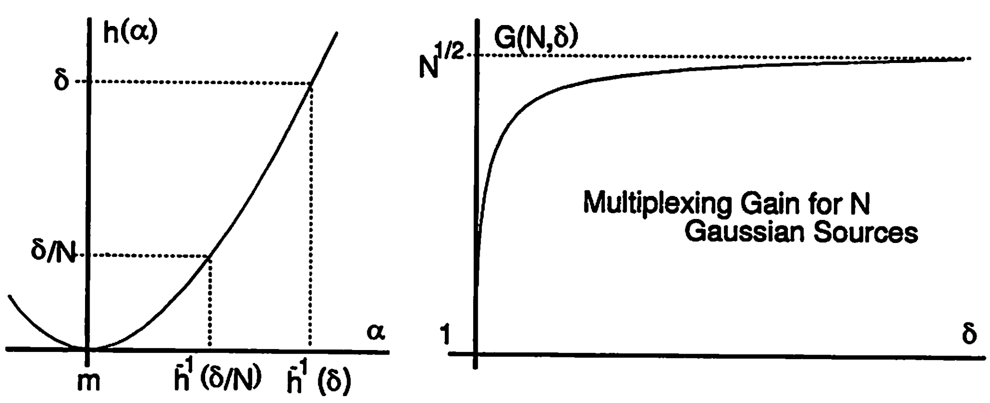

Figure 2: Rate function and statistical multiplexing for Gaussians.

Example 1: Consider the case of Gaussian random variables where $h(\alpha) = (\alpha - m)^2/2\sigma^2$. In this case we have,

$$G(N,\delta) = \frac{m + \sigma\sqrt{2\delta}}{m + \sigma\sqrt{2\delta/N}}.$$

Fig.2 shows a plot of this function. On one hand when $\delta = 0$, the gain is one, since the effective bandwidth for both cases will equal the mean arrival rate. On the other hand, we have $\lim_{\delta\to\infty} G(N,\delta) = \sqrt{N}$, which one might expect since this is scaling of the standard deviation for N Gaussian random variables. For fixed δ the plot of $G(N,\delta)$ has a similar qualitative behavior; for $N = 1$, $G(N,\delta) = 1$ and there is once again an asymptote,

$$\lim_{N\to\infty} G(N,\delta) = 1 + \frac{\sigma\sqrt{2\delta}}{m}.$$

Thus once we decide on a loss constraint, there is a bound on the multiplexing gains that can be obtained.

Example 2: Consider the case of on/off Markov fluids introduced in §2. Their behavior is qualitatively quite different from the Gaussian case. The "rate" is concentrated in the interval $[r_0, r_1]$, i.e., the rate function is infinite outside this interval. A typical example is shown in Fig.3. Note that $h(r_1) = \mu$ as one would expect since in order to obtain an empirical mean of r_1 we need to stay *on* for a prolonged period of time. Thus the exponent should correspond to the tail of an exponential waiting time with rate μ. The inverse function is,

$$h^{-1}(\delta) = r_1 + \frac{(r_1 - r_0)(\delta(\mu - \lambda) - \mu(\mu + \lambda) + 2\sqrt{\delta\lambda\mu(\lambda + \mu - \delta)}}{(\lambda + \mu)^2},$$

when $\delta \leq \mu$ and is r_1 for $\delta > \mu$. Since the traffic rate associated with this source is bounded by r_1, when $c \geq r_1$ no loss occurs. This fact leads to the surprising behavior of the multiplexing gain shown in Fig.3. Unlike the case of Gaussian sources, $G(N,\delta)$ will decrease once δ becomes too large. This occurs when the loss constraint δ is approximately μ, beyond this level the effective bandwidth for a single source saturates at r_1, while that of the N sources continues to increase. As shown in the figure, $G(N,\mu)$ can be bounded by,

$$\lim_{N\to\infty} G(N,\mu) = \frac{r_1(\lambda + \mu)}{r_1\lambda + r_0\mu}.$$

We can interpret this expression as follows : increases in the peak firing rate r_1 or decreases in the average burst size μ^{-1} will improve the maximum multiplexing gain, i.e., "bursty" sources are the candidates of choice for PARing. Further increases in the loss constraint reduce the benefits of multiplexing, in fact for $\delta \geq N\mu$, $G(N,\delta) = 1$. For fixed δ the gain is concave and increases with N to an asymptotic value.

To summarize, we find somewhat surprisingly that for some types of sources stringent statistical constraints can lead to decreases in the multiplexing gain. Sharing of resources can lead to gains in bandwidth as well as buffering requirements. The latter is discussed in the context of traffic shaping in the next section.

5 Comparison: Where to buffer?

We wish to address the following question: Can one characterize the tradeoff between buffering within the network and buffering at the sources (PARing) subject to a statistical loss guarantee?

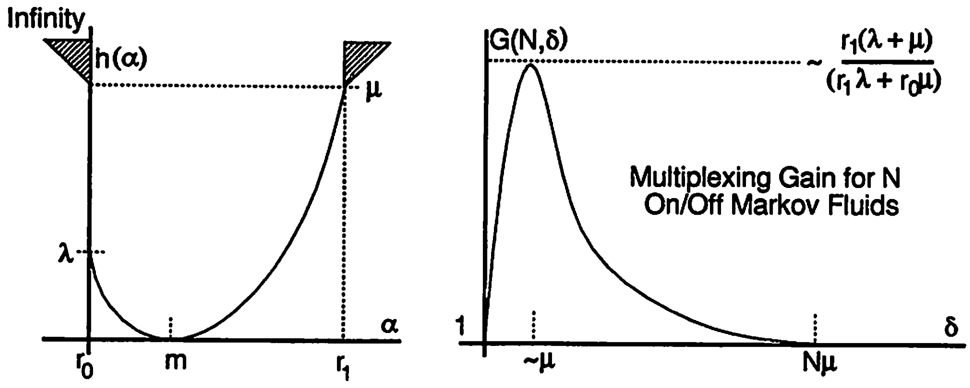

Figure 3: Rate function and statistical multiplexing for Markov fluids.

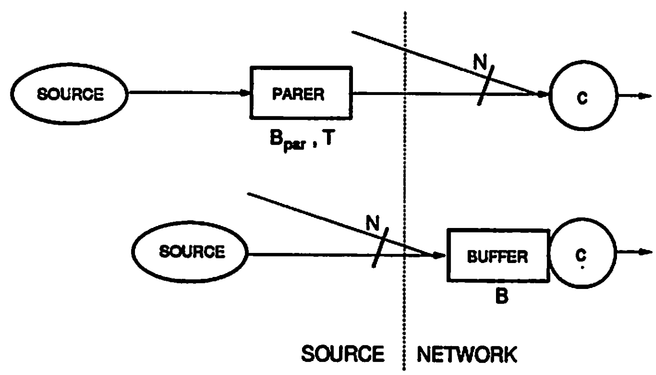

Figure 4: Comparing buffering strategies.

The tradeoff that we propose to explore is shown in Fig.4. On one hand we consider N sources flowing through a network buffer B with service rate c, where B is quite large in order to guarantee that losses are small. On the other hand, we PAR N sources using separate buffers of size B_{par}, and an averaging interval T, which is large to ensure that the losses in the network are small for every PARing interval even when carryover is neglected. We assume that all the sources are the same, and have the required properties for the existence of a large deviations principle with rate function $h(\cdot)$. We impose a loss constraint, $\mathbf{P}(\text{loss}) < \exp[-\delta]$, where the exponent is not scaled by B or T, to facilitate comparisons.

Network Buffering In §3 we briefly introduced large deviation asymptotics that can be obtained for the buffer occupancy of deterministic queues. In the above setup we have that,

$$\mathbf{P}(\text{network overflow}) \approx \exp[-BK_N]$$

where

$$K_N = \inf_{\alpha > c/N} \frac{h(\alpha)}{\alpha - c/N}.$$

One can then argue that the loss probability has the same asymptotics as the probability of overflow for busy cycles, modulo a small correction term[11]. In order to satisfy our loss constraint we require that, $BK_N \geq \delta$, so we take $B = \delta / K_N$.

Source Buffering In order to guarantee few losses with PARing we must select both B_{par}, and the averaging interval T. The appropriate choice of T follows from §3 and §4 :

$$\mathsf{P}(\text{network loss}) \approx \exp[-TK_S]$$

where,

$$K_S = Nh(c/N).$$

We need $TK_S \geq \delta$ so it suffices to take $T = \delta/K_S$. In order to control losses at the PARing buffer we further require $Th(B_{par}/T) > \delta$, for details see §2 and recall our new convention in specifying the loss constraint. Thus the smallest permissible PARing buffer is :

$$B_{par} = Th^{-1}(\delta/T) = \frac{\delta h^{-1}(K_S)}{K_S},$$

by substituting the value of K_S we obtain,

$$B_{par} = \frac{\delta h^{-1}(Nh(c/N))}{Nh(c/N)}.$$

To answer our question we must compare B with NB_{par}. We can define the network *buffering advantage* by

$$B(N,c) \triangleq \frac{NB_{par}}{B} = \left[\inf_{\alpha>c/N} \frac{h(\alpha)}{\alpha - c/N}\right] \frac{h^{-1}(Nh(c/N))}{h(c/N)}.$$

Thus depending on N, c and the type of source, $B(N,c)$ will determine the savings obtained by buffering within the network. Below we consider the case of Gaussian arrivals, as well as on/off Markov fluids.

Example 1: For Gaussian arrivals, we can easily solve the optimization problem leading to the network overflow asymptotics :

$$\inf_{\alpha>c/N} \frac{h(\alpha)}{\alpha - c/N} = \frac{2(c/N - m)}{\sigma^2},$$

where we continue with the notation in §4.4, and assume that the queue is stable i.e., $Nm < c$. The buffering gain is given by :

$$B(N,c) = 4\left(\frac{Nm}{c - Nm} + \sqrt{N}\right).$$

As shown in Fig. 5, $B(N,c) > 4$. It is always more economical to buffer within the network. This benefit increases sharply as Nm approaches c, i.e., we push the system to service a maximum number of sources. This result might have been expected, it corresponds to the idea behind multiplexing thousands of voice calls in common buffers.

Example 2: The case of Markov fluids is somewhat more complex. Recall that in §2 we introduced the rate function for two state Markov fluids, and in §4.4 we defined the equivalent bandwidth or inverse of this function. The optimization problem associated with network buffer asymptotics has been solved, see for example deVeciana et al.[5], it gives the following result :

$$\inf_{\alpha>c/N} \frac{h(\alpha)}{\alpha - c/N} = N\frac{(\lambda + \mu)(c - Nm)}{(c - Nr_0)(Nr_1 - c)},$$

where m corresponds to the mean traffic rate offered by one such source, and we require stability, $Nm < c$ as well as the possibility of overflows $Nr_1 > c$.

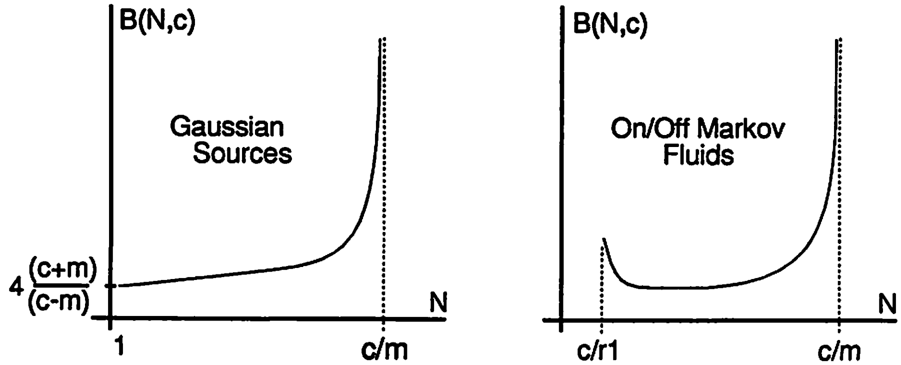

Figure 5: Buffering advantage for Gaussian sources and on/off Markov fluid.

Using these results the buffering gain can be written as :

$$B(N,c) = N\frac{(\lambda+\mu)(c-Nm)}{(c-Nr_0)(Nr_1-c)}\frac{h^{-1}(Nh(c/N))}{h(c/N)}.$$

A typical graph of $B(N,c)$ versus N is shown in Fig. 5, it exhibits the trends of interest. Note that for large N network buffering is very advantageous. Unlike the Gaussian case, as N becomes small, i.e., of order c/r_1, $B(N,c)$, increases, albeit not much. Intuitively, when there are very few sources, network overflows become very unlikely, and PARing is inefficient. However when a moderate number of sources are involved the advantage of buffering within the network is not as great, in this regime PARing becomes a viable option. This corresponds to a regime in which the peak rate of the sources may take a substantial amount of the total bandwidth, and there is a potential for much loss.

Our asymptotic analysis partially demonstrates some of the buffering tradeoffs in a network design. Indeed, within the network, traffic streams share several buffers before reaching their destinations. Thus, in order to make a fair comparison, we should include the resources consumed at each node for each path. Moreover, buffers within the network feed into fast output lines, hence the memory required within the network will be expensive relative to single source PARing buffers.

Network design involves many other criteria and tradeoffs. The most important in the setup we have proposed is delay. The PARing scheme has a worst case delay of $2T$, and an average of about T. A precise comparison of delays incurred is difficult to carry out. The sharing of resources within the network will most probably reduce average delays relative to the PARed setup. However, PARing distributes delays among the various sources. Moreover, PARing will guard bursty sources from interfering with other traffic in the network, whence one can conceive of simplified network buffering policies, and improved fairness in access to resources. The smoothing out of traffic entering the network, should in turn reduce packet jitter, which can in turn reduce buffering requirements at the destination. A balanced design needs to envisage many criteria, to this end this framework is only a beginning[4].

The two cases we have considered represent extremes in a spectrum of more realistic configurations where buffering in the network as well as traffic shaping are used. Having studied the interaction of sources within buffers, we move to problems at the network level, with multiple buffers and routing paths.

6 Networks and PARing

In this section we extend the ideas used for single buffers to obtain interesting results characterizing the behavior of interacting streams in a network of deterministic buffers. Sources are assumed to be PARed over relatively large intervals T, thus over an interval $[nT, (n+1)T)$ the aggregate arrival rate to the network, A_n^v, for each virtual path v, will be constant, and have the deviations studied in the previous section. The rate on the output link of a buffer can be computed by a max(min(), 0) operation. Indeed the output rate will be $\max(\min(A_n, c), 0)$ where c is the service rate and hence the largest rate one can observe at the output and A_n is the aggregate arrival to that node. The purpose of the $\max(\cdot, 0)$ function, is to guarantee that the output rate is non-negative, thus keeping our model honest. Using this setup we can determine the rate along every link in the network for a given PARing interval, which enables us to calculate the large deviations of the entire network. We propose this simple model as a first approximation, to one that should include carry-over as well as propagation delays. To our knowledge this is the first such model.

6.1 Asymptotics of a PARed network

We will consider the network shown in Fig. 6. There are three virtual paths, $v = 1, 2, 3$ along each path will flow N_v, $v = 1, 2, 3$, PARed sources. The service rates for the switching nodes are

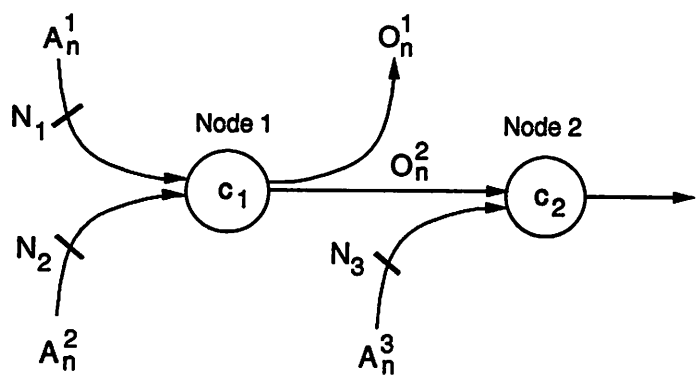

Figure 6: Network Model.

c_1 and c_2. For simplicity we assume that all the sources are of the same type. This setup should exhibit representative behavior of interacting homogeneous streams. We assume the network is stable. Thus the net traffic into each node is less than or equal to the service rate; $(N_a + N_b)m < c_1$ and $(N_b + N_c)m < c_2$, where m denotes the mean arrival rate of one of the PARed sources. Finally we will denote the output traffic of node 1 by O_n^1, O_n^2. When no overflow occurs these will simply correspond to the arrivals on the respective paths. However when an overflow does occur at node 1, we will set $o_n^1 + o_n^2 = c$ such that their proportion of the bandwidth corresponds to their contribution to the aggregate arrival rate. We analyze the network, by computing the overflow asymptotics at the two nodes.

Node 1:

$$\begin{aligned} \mathsf{P}(\text{loss at node 1}) &= \mathsf{P}(A_n^1 + A_n^2 > c_1) \\ &\approx \exp[-TK_0] \end{aligned}$$

where

$$K_0 = (N_1 + N_2)h(\frac{c_1}{N_1 + N_2}).$$

The reader may wish to refer back to §4 where this case was considered.

Node 2:

$$\begin{aligned} \mathsf{P}(\text{loss at node 2}) &= \mathsf{P}(O_n^2 + A_n^3 > c_2) \\ &= \underbrace{\mathsf{P}(O_n^2 + A_n^3 > c_2 | A_n^1 + A_n^2 > c_1)\ \mathsf{P}(A_n^1 + A_n^2 > c_1)}_{\text{U1}} + \\ &\quad + \underbrace{P(O_n^2 + A_n^3 > c_2, A_n^1 + A_n^2 \le c_1)}_{\text{U2}} \end{aligned}$$

Before estimating U1 note that if an overflow occurs at node 1 we expect that the output O_n^2 will be $\frac{N_2}{N_1+N_2}c_1$, since this corresponds to the most likely way for an overflow to occur. Let $\tilde{c}_2 = c_2 - \frac{N_2}{N_1+N_2}c_1$ Our estimate for U1 should be by now familiar.

$$\begin{aligned} \text{U1} &\approx P(A_n^3 > \tilde{c}_2)\exp[-T(N_1 + N_2)h(\frac{c_1}{N_1 + N_2})] \\ &\approx \exp[-TK_1] \end{aligned}$$

where

$$K_1 = (N_1 + N_2)h(\frac{c_1}{N_1 + N_2}) + N_3 h(\frac{\tilde{c}_2}{N_3})$$

We can estimate U2 as follows. The goal is to find the large deviation exponent of the event that node 2 overflows but not necessarily as a consequence of an overflow in node 1. Clearly this event is a continuous function of the empirical means of the arrival rates along each of the virtual paths. Each of these empirical means has an associated large deviation principle. One can compute the large deviations of a continuous function of the latter by way of the contraction principle[2]. This gives the following characterization of the exponent.

$$\begin{aligned} \text{U2} &\approx \exp[-TK] \\ K &= \min \quad N_1 h(x) + N_2 h(y) + N_3 h(z) \\ &\quad \text{s.t. } N_1 x + N_2 y \le c_1 \\ &\qquad\quad N_2 y + N_3 z \ge c_2 \end{aligned}$$

This constrained minimization problem can be solved via Lagrange multipliers. One finds that only two cases are possible corresponding to different sets of active constraints. Each case has a relatively nice interpretation.

Case 1 Node 2 overflows without overflowing node 1. This occurs when,

$$\frac{N_2}{N_2 + N_3} \times c_2 < c_1 - N_1 m. \qquad (2)$$

Thus, the traffic rate required from the calls on path #2 in order to overflow node 2, does not exceed the average capacity available at node 1 for those sources. The asymptotics correspond to a network without node 1.

$$K_2 = (N_2 + N_3)h(\frac{c_2}{N_2 + N_3})$$

Case 2 Overflows at node 2 induce an overflow at node 1. This corresponds to the case in which both constraints are active. Somewhat surprisingly this case corresponds to a situation in which the traffic along path 1, transmits *below* its expected mean rate. The interpretation is that in order to overflow node 2 we require so much excess flow from path #2, that the traffic along path #1 must be decreased so that it may get through. We find that when constraint Eq. 2 does not hold i.e.

$$\frac{N_2}{N_2+N_3} \times c_2 \geq c_1 - N_1 m,$$

then

$$K_3 = N_1 h(\frac{c_1 - N_2 y}{N_1}) + N_2 h(y) + N_3 h(\frac{c_2 - N_2 y}{N_3}),$$

where y, the deviant traffic rate for path #2, satisfies

$$h'(y) = h'(\frac{c_1 - N_2 y}{N_1}) + h'(\frac{c_2 - N_2 y}{N_3}),$$

and in fact $\frac{c_1 - N_2 y}{N_1} < m$.

One can show that $K_1 \geq K_3 \geq K_0$. Indeed, K_0 is the exponent for the probability that node 1 overflows, while K_1, K_3 correspond to overflows in both nodes 1 and 2, thus $K_0 \leq K_1, K_3$. Next, $K_2, K_3 \leq K_1$, since K_2 or K_3, whichever holds, represents the most *likely* way to get an overflow in node 2, while K_1 corresponds to a specific setup in which we consider first an overflow at node 1, and then conditioning on this event, an overflow at node 2. These relationships can be established directly by considering the respective optimization problems.

We can summarize our results for this network by taking the dominant exponents, i.e. dropping K_1 and hence U1,

$$\begin{aligned}
\mathbf{P}(\text{loss at node 1}) &\approx \exp[-TK_0] \\
\mathbf{P}(\text{loss at node 2}) &\approx \exp[-TK_2] \quad \text{if Eq. 2 holds} \\
&\approx \exp[-TK_3] \quad \text{otherwise}
\end{aligned}$$

6.2 Capacity of a network

In this subsection we will consider the *capacity* of the network. This will be the natural extension of effective bandwidth, and corresponds to the practical application we had in mind. We wish to determine the number of calls that may be routed through the network while guaranteeing a given probability of loss, i.e.

$$\begin{aligned}
\mathbf{P}(\text{loss at node 1}) &\leq \exp[-T\delta_1] \\
\mathbf{P}(\text{loss at node 2}) &\leq \exp[-T\delta_2].
\end{aligned}$$

Referring to the summary above, we require,

$$\begin{array}{lll}
\text{Node 1} & K_0 \geq \delta_1 & \\
\text{Node 2} & K_2 \geq \delta_2 & \text{if Eq. 2 holds} \\
& K_3 \geq \delta_2 & \text{otherwise.}
\end{array}$$

Finally, if $\delta_1 = \delta_2 = \delta$ we have $K_0 \geq \delta \Rightarrow K_3 \geq \delta$. Thus using our notion of effective bandwidth, in §4 we obtain two independent constraints,

$$\begin{array}{ll}
\text{Node 1} & \alpha(N_1 + N_2, \delta) < c_1 \\
\text{Node 2} & \alpha(N_2 + N_3, \delta) < c_2
\end{array}$$

A satisfactory call acceptance scheme need only check that each buffer along a given virtual path can tolerate another call.

A word of caution however, if for some reason the network designer decides that it is appropriate to require $\delta_1 \leq \delta_2$ then the above decoupling will not hold, and the following situation may arise: in order to route a new call along path #3, we must check the number of sources flowing through path #1, see Fig. 6. To be specific suppose the above situation arises, i.e. Eq. 2 does not hold, then we must check whether $K_3 \geq \delta_2$, and in order to do so we need to know N_1, N_2, N_3, c_1, c_2 which may not be readily available. On a larger scale, when such a situation occurs, it will be very difficult to allocate resources appropriately.

These results should extend to more complex situations, with multitype sources, and networks, though the calculations will no doubt be cumbersome.

7 Sliding Window PAR (SPAR)

Herein, we propose a modified version of our PARing scheme, which overcomes some of its drawbacks. The idea is to consider a traffic shaping scheme in which the the source is buffered in a queue with a

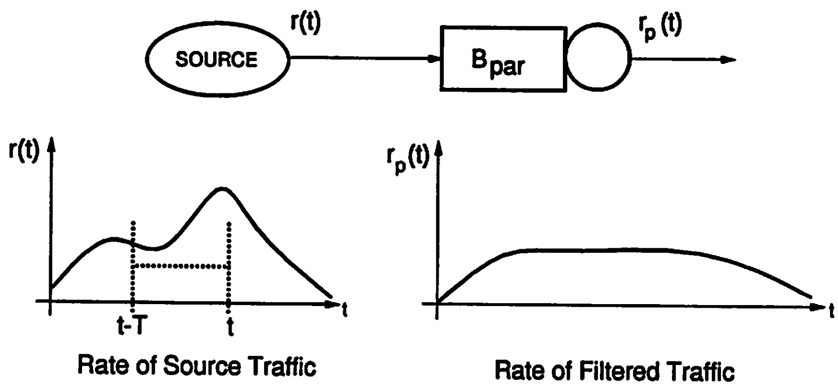

Figure 7: SPARing a variable rate source.

time-varying service rate corresponding to the average traffic rate within the current time window, i.e., $[t-T, t)$, see Fig. 7. Thus a source with time-varying traffic rate $r(t)$ is released into the network as,

$$r_p(t) = \frac{1}{T}\int_{t-T}^{t} r(\tau)d\tau,$$

which is simply the moving average of the original process. This will not only reduce delays, but also the size of the PARing and network buffers. We still allow a design parameter, namely the size of the averaging window T, representing the extent of smoothing which is desired. The asymptotic behavior of this scheme can be analyzed in a similar fashion to PARing.

7.1 Memory Requirements

The evolution of the queue, X_t in a SPARing buffer is given by the following equation,

$$X_t = X_{t-1} + A_t - \frac{1}{T}\sum_{i=t-T}^{t-1} A_i$$

where we have discretized time and consider an averaging window of length T time units. A_i denotes the the arrivals for the given source during the i^{th} time interval. A simple manipulation gives,

$$X_t = \frac{1}{T}\sum_{i=1}^{T} iA_{t-T+i} \stackrel{d}{=} X$$

Thus the *maximum* delay for this scheme is T, since no arrival will be in queue T time units after it enters. SPAR buffers should be scaled such that $\mathsf{P}(X > B_{par}) < \exp[-T\delta]$ where we have adopted a notation for the constraint which is consistent with §2.

A large deviation bound can be obtained in the case where the arrivals are i.i.d. :

$$\mathsf{P}\left(\frac{1}{T}\sum_{i=1}^{T} iA_i \geq B_{par}\right) = \mathsf{P}\left(\frac{1}{T^2}\sum_{i=1}^{T} iA_i \geq B_{par}/T\right) \approx \exp[-Th_p(B_{par}/T)].$$

The rate function $h_p(\cdot)$ follows from the the Gärtner-Ellis Theorem[6], it is the Fenchel-Legendre transform of following limit if and when it exists :

$$\begin{aligned}
\Lambda_p(\lambda) &= \lim_{T\to\infty}\frac{1}{T}\log\mathsf{E}\exp(\lambda\sum_{i=1}^{T}\frac{i}{T}A_i) = \lim_{T\to\infty}\frac{1}{T}\log\prod_{i=1}^{T}\mathsf{E}\exp(\frac{i\lambda}{T}A_i) \\
&= \lim_{T\to\infty}\frac{1}{T}\sum_{i=1}^{T}\Lambda(\frac{i\lambda}{T}) = \frac{1}{\lambda}\int_0^{\lambda}\Lambda(x)dx
\end{aligned}$$

where $\Lambda(x) = \log\mathsf{E}\exp[xA_0]$. This is the average of the log-moment generating function over $[0, \lambda)$. The rate function is given by,

$$h_p(\alpha) = \sup_{\lambda}[\lambda\alpha - \frac{1}{\lambda}\int_0^{\lambda}\Lambda(x)dx]$$

$\Lambda_p(\lambda) \leq \Lambda(x)$ by Jensen's inequality on the convex function $\Lambda(\cdot)$[6] so clearly $h_p(\alpha) \geq h(\alpha)$. For Gaussian and Poisson sources we compute $h_p(\cdot)$ explicitly :

$$\begin{aligned}
h_p(\alpha) &= 3\frac{(\alpha - m/2)^2}{2\sigma^2} \quad \text{Gaussian} \\
h_p(\alpha) &= \alpha\log\frac{2\alpha}{m} + m/2 - \alpha \quad \text{Poisson}
\end{aligned}$$

These results should be compared with their counterparts in Table 1. SPARing reduces the mean (by a half) as well as the variance of potential accumulation in the averaging buffer.

The behavior of sources with dependencies is somewhat harder to determine. For example, suppose $\{A_i\}_{i=1}^{\infty}$ is a stationary Gaussian process. The distribution of X can be obtained explicitly in terms the mean m, and covariance terms up to T^{th} order $(c_i,\ 0 < i \leq T)$ of the source :

$$X \stackrel{d}{=} N(\mu, \sigma^2)$$

$$\mu = \frac{m(T+1)}{2}$$

$$\sigma^2 = \frac{1}{T^2}[c_0 \sum_{i=1}^{T} i^2 + 2\sum_{i=1}^{T}\sum_{j<i} c_{i-j} ij].$$

The tail of this Gaussian gives an idea of the probability of having a queue length exceeding B_{par}. Sources with positively correlated arrivals are more likely to overflow as the latter lead to larger fluctuations. Note however, that only correlations on the T time-scale are relevant.

7.2 Network Buffers

Loss will occur if the moving average rate $r_p(t)$ during a given time interval exceeds the service rate c. Thus assuming T is large the asymptotics should be roughly those obtained in §3. However, a more detailed calculation of the network overflow asymptotics for sources with independent increments, shows that there may even be some improvements. Loss will occur during an interval $[0, T)$ if the traffic released into the network exceeds the capacity during that period :

$$\mathbf{P}(\text{cell loss}) \approx \mathbf{P}\left(\sum_{j=1}^{T} \frac{1}{T} \sum_{i=j-T}^{j-1} A_i \geq cT\right) = \mathbf{P}\left(\frac{1}{T}\sum_{j=1-T}^{T-1}(1 - |\frac{j}{T}|)A_j \geq c\right) = \exp[-Th_n(c)],$$

where $h_n(\cdot)$ can be obtained by calculations similar to those above :

$$h_n(\alpha) = \sup_{\lambda}[\lambda\alpha - \Lambda_n(\lambda)] \text{ where } \Lambda_n(\lambda) = \frac{2}{\lambda}\int_0^{\lambda} \Lambda(x)dx.$$

Note that in fact $\Lambda_n(\lambda) = 2\Lambda_p(\lambda) \leq \Lambda(\lambda)$ this follows from the convexity of the log-moment generating function, and the fact that $\Lambda(0) = 0$. Thus $h_n(\alpha) \geq h(\alpha)$, so the sliding window scheme will decrease losses in the network beyond PARing. Similar improvements are expected for general sources.

In order to obtain this improvement in performance some overhead is required. To compute the target rate of packet release a simple FIR filter is needed. The scheduling of departures of the packets could be done via a reasonable clocking method. The simulations in the next section exhibit the effectiveness of traffic shaping.

8 Simulations

In this section we present preliminary simulations of PAR traffic shaping for somewhat bursty traffic. The traffic corresponds to the variable bit rate at the output of a simulated video coder. The data set used for this purpose was generated by a simulated coder including DCT and Huffman coding but no motion compensation which is available from Bellcore. The data represents the bit rate trace on a 1.4 msec scale (slice) for 2 hours of the movie "Star Wars"; this time-scale was deemed reasonable for studies of ATM traffic. 53 byte packets with 48 bytes of data were used as the standard unit (cell) of traffic. The average arrival rate for the source is then approximately 13,000 cells/sec corresponding to 6 Mbps. The source traffic was modeled by a piecewise linear or fluid model. We note at the outset that the trace appears to be highly nonstationary. The characteristics of such sources will depend on the chosen compression schemes. The performance of PARing will in turn depend on the dynamics of the source. For more information on this data set we refer the reader to the work of Garrett and Vetterli[9].

Since video traffic is delay-critical, we take a relatively small PARing interval T of approximately 25 msec (approximate propagation time in a network).

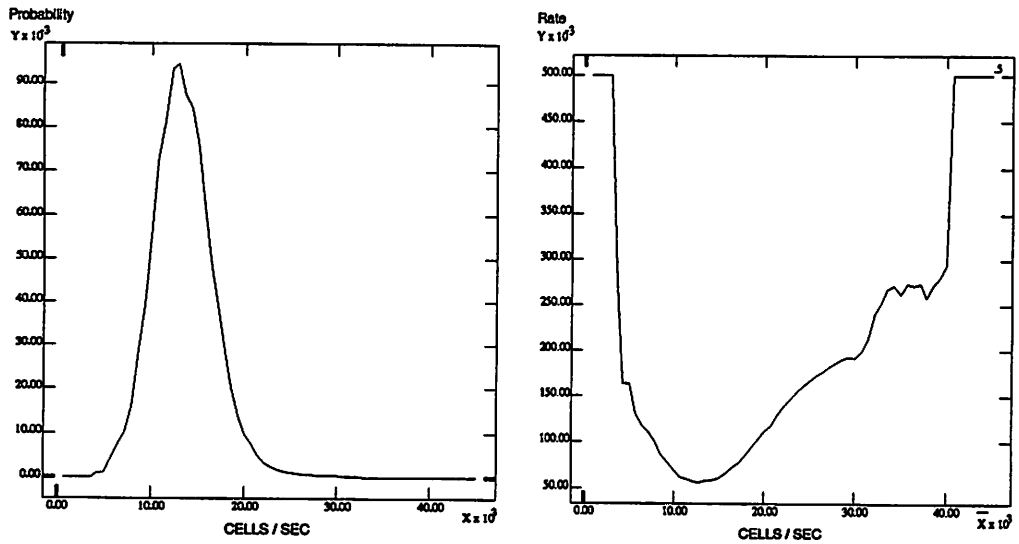

Figure 8: Histogram and rate function for a T-PARed video source(T=25ms).

Rate Function $h(\alpha)$: We collected an empirical distribution for the average rate (in cells/sec) over typical T sections of this source, Fig 8. This data was then transformed to obtain a ***rough*** idea of rate function for a video source by the transformation $h(\cdot) = -\frac{\log 10 p(\cdot)}{T}$, which is shown in Fig. 8. In the region were events become rare the rate function is not smooth because estimates have large variance, however it appears to be approximately convex. We have tacitly assumed that the source has a typical behavior over a T time-scale, this may however not be the case.

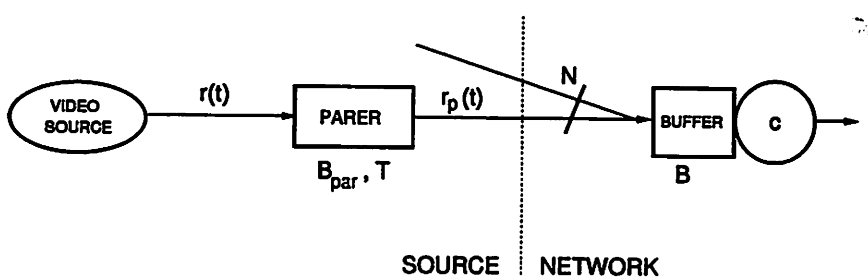

Figure 9: Simulation of PARed Video.

Averaging Interval versus Network Loss: Here, our primary goal was to gauge the reduction in loss within the network when sources are PARed, so we take $B_{par} = \infty$. The PARing setup is shown in Fig. 9 where $N = 5, 10$. The graphs in Fig. 10 show the loss rate of PARed versus regular sources, for various averaging intervals, $T = 25, 50, 125$ msec, and relatively small network buffer sizes B in cells, where this method is expected to be effective. The service rate c was selected such that the utilization of the network buffer was approximately 80% or 90%.

These results exhibit some of the properties one would expect. The loss rate decreases when we enlarge the averaging interval, but the effective gain one can obtain tapers off. Recall, that ideally one should average on the typical time-scale of accumulation in the network buffers. Beyond this point we expect long term accumulation to dominate the overflow asymptotics and thus the loss rate, see §3. These two regimes are exhibited quite dramatically in the plot of the log of the probability of overflow for busy cycles versus the buffer size. Fig. 11 exhibits the transition between the two asymptotic regimes. For small buffers, PARing is advantageous; performance is sensitive to the the size of T and B. While for large buffers, long term buildup and carry-over cause overflows. In this case, our asymptotics predict that the log of the probability of overflow should become approximately linear in B, see Fig. 11. As the utilization decreases, the advantage of PARing improves since losses are likely to correspond to short term buildups rather than extended periods of high traffic rates.

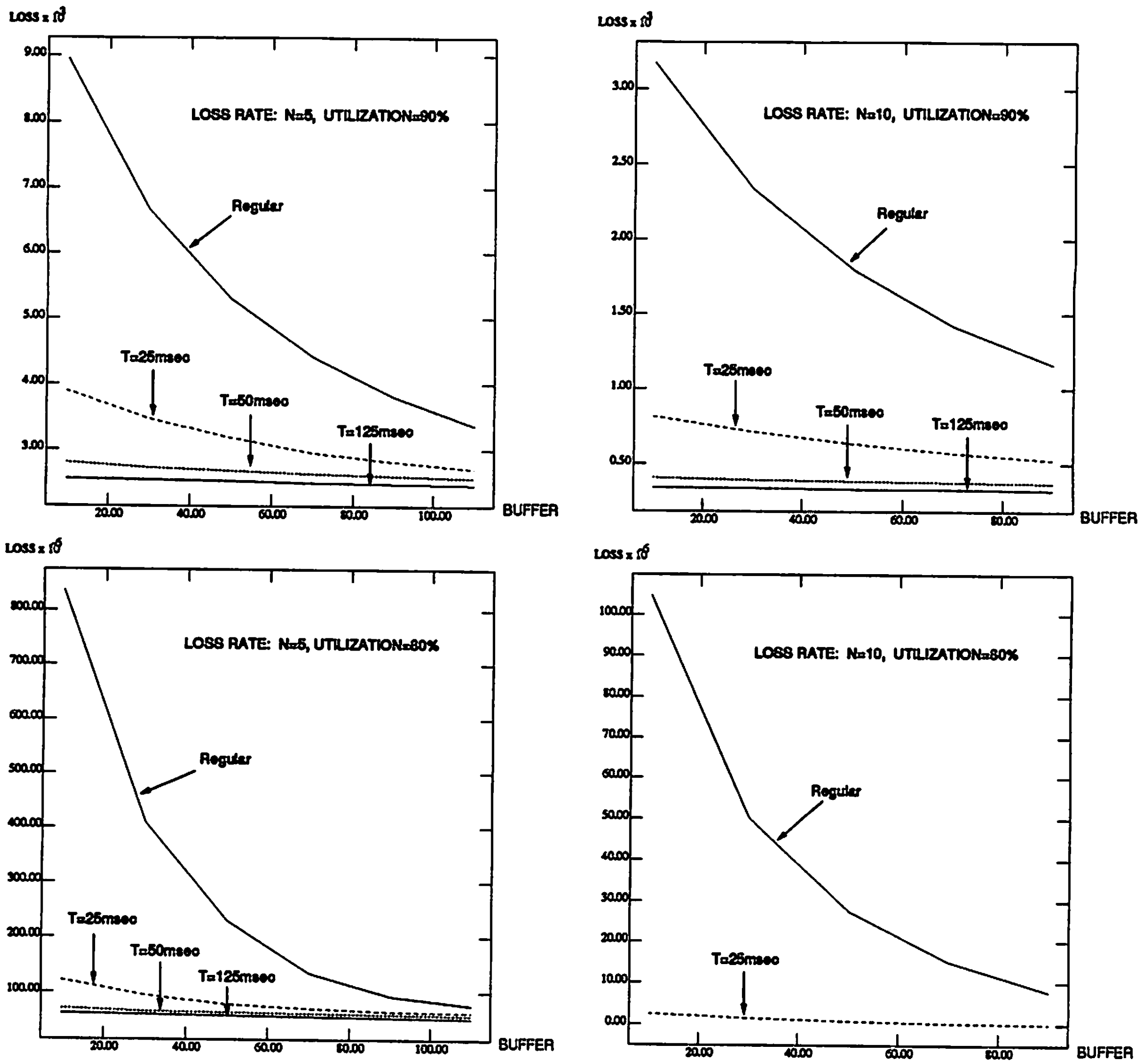

Figure 10: Loss rate for PARed versus regular sources.

PARing and Loss Rates: A more realistic simulation requires taking finite PARing buffers, B_{par}, in addition to finite network buffers B. For fixed $T = 25$msec, we explore the tradeoff of the loss rates as a function of these two parameters, see Fig. 12. One can consider three loss rates : in the PARing buffers, in the network buffer, and the total loss incurred. As B_{par} increases loss in the PARing buffers decreases dramatically while losses in the network increase by a moderate amount. For large enough B_{par} the total loss rate is very close to that we obtained for infinite PARing buffer, see Fig. 10. Consider for example the case $N = 5$, $\rho = .8$; in order to maintain the a total loss rate

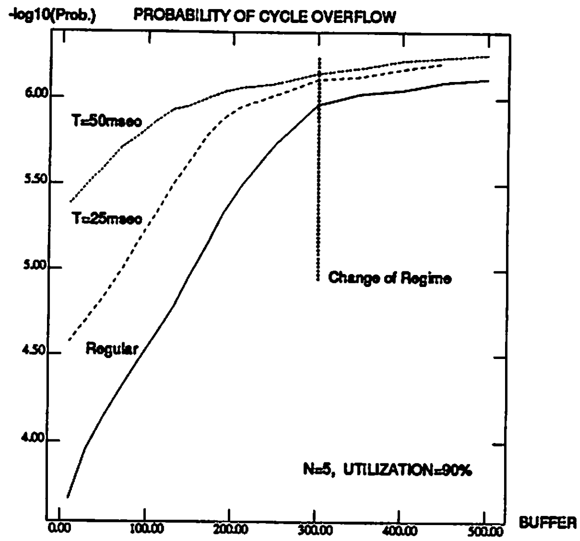

Figure 11: Overflow probability for busy cycles.

of 2. 10^{-4} without traffic shaping we require $B = 60$ however when PARing the network buffer can be reduced to $B = 10$, but for each source a PARing buffer of size $B_{par} = 750$ is required. So the simulated buffering advantage is 65. Clearly the PARing buffer B_{par} is the main cause of losses, the sliding window scheme improves performance significantly.

SPARing and Total Loss Rate: We simulated the scheme described in §7. Fig. 13 shows the total loss rate, with a sliding window $T = 25$msec, and a variety of B_{par}, B, N and utilizations. The loss rates corresponding to a setup with no traffic shaping have been superimposed. The traffic shaping buffers are approximately halved. Consider for case $N = 5, \rho = .8$ again, in order maintain a loss rate of 2. 10^{-4} the network buffer $B = 60$ required can be reduced to 10 while using $B_{par} = 350$. The simulated buffering advantage in this case is 30. The sliding window idea halves the required averaging buffers.

We believe that the performance of this traffic shaping scheme will improve as sources become more "bursty", on a T time-scale. More detailed simulations were not carried out since ultimately one should investigate the performance tradeoffs for a standard such as MPEG but designed explicitly for ATM networks where inter and intra-frame coding would result in efficient but variable bit rates which in turn could be smoothened by PARing on the correct time-scale. The development of these standards though, will depend on ATM network design, this is where coding and network design meet, and tradeoffs such as subjective quality, priority schemes and other network design and control schemes need to be considered.

The objective of these simulations was to exhibit the qualitative behavior of rate averaging. A more detailed comparative investigation of traffic shaping is required to study the effectiveness of this idea.

9 Conclusion

We proposed a simple flow control method corresponding to periodically averaging the rate of a source (PARing). Our primary goal was to consider alternatives for limiting cell loss in the presence of

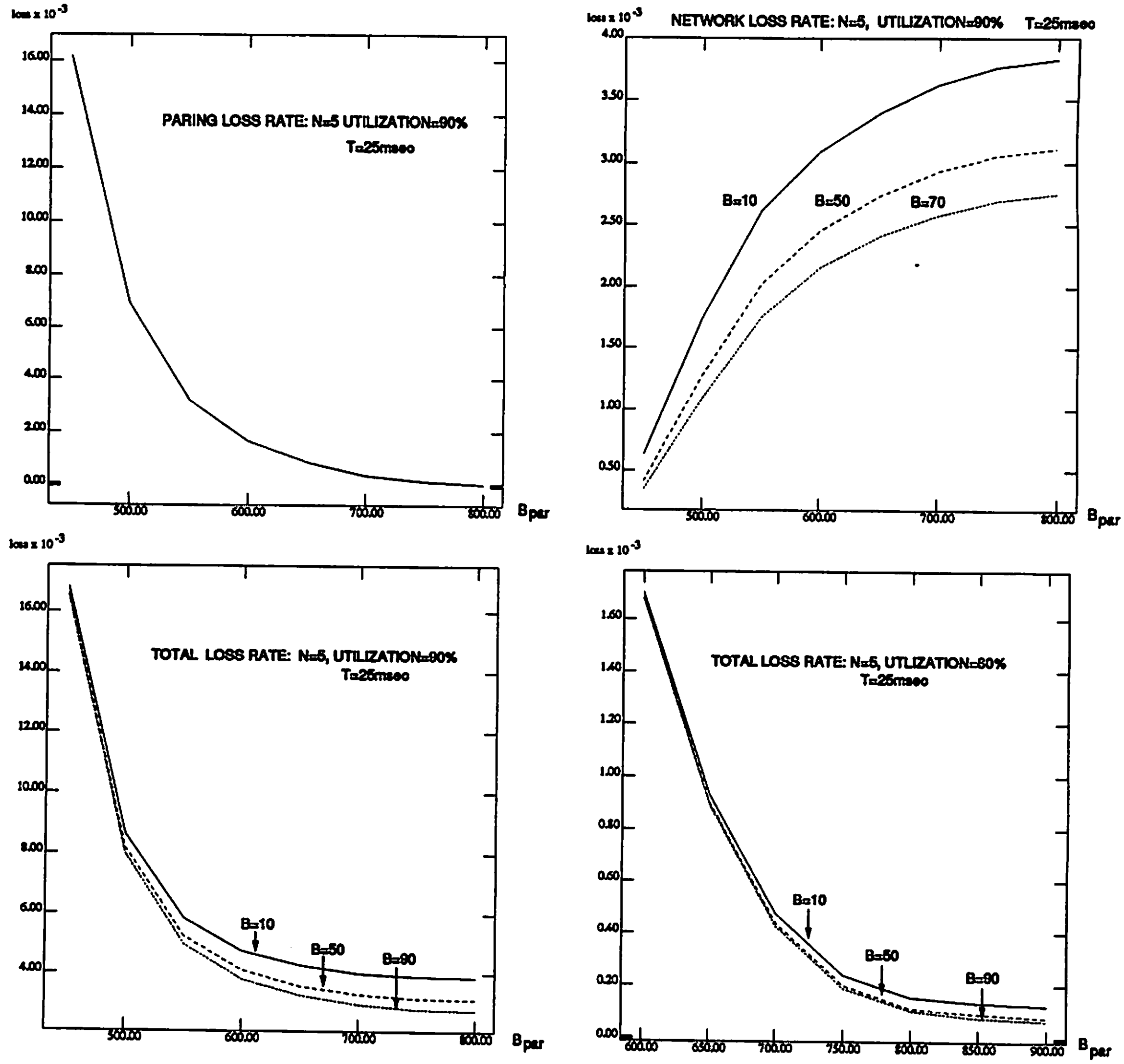

Figure 12: Loss rates vs B and B_{par} (T = 25msec).

bursty traffic. In the context of traffic shaping, we investigated the benefits of sharing resources using large deviation asymptotics. The sharing of bandwidth was explored via the concept of *multiplexing gain*, while the sharing of network buffers was characterized by the *buffering advantage*. In particular we used statistical cell loss constraints to identify bandwidth requirements, and found, somewhat remarkably, that in the case of on/off Markov fluid sources very stringent constraints may decrease the multiplexing gain of sharing bandwidth. Our asymptotic results confirmed the notion that sharing buffers in the network is advantageous. However, in order to fully understand the tradeoffs of traffic shaping many other performance criteria need to be investigated, such as delay and jitter. PARed sources are delayed by T, but this may in turn be compensated by subsequent reductions in network delays due to the reduced burstiness (and jitter) of traffic. Reductions in jitter, will also decrease the buffering required at the destination.

The cost of PARing lies mainly in increased memory requirements at the sources but this may not be consequential given the current reduction in price and the relatively fast memory required

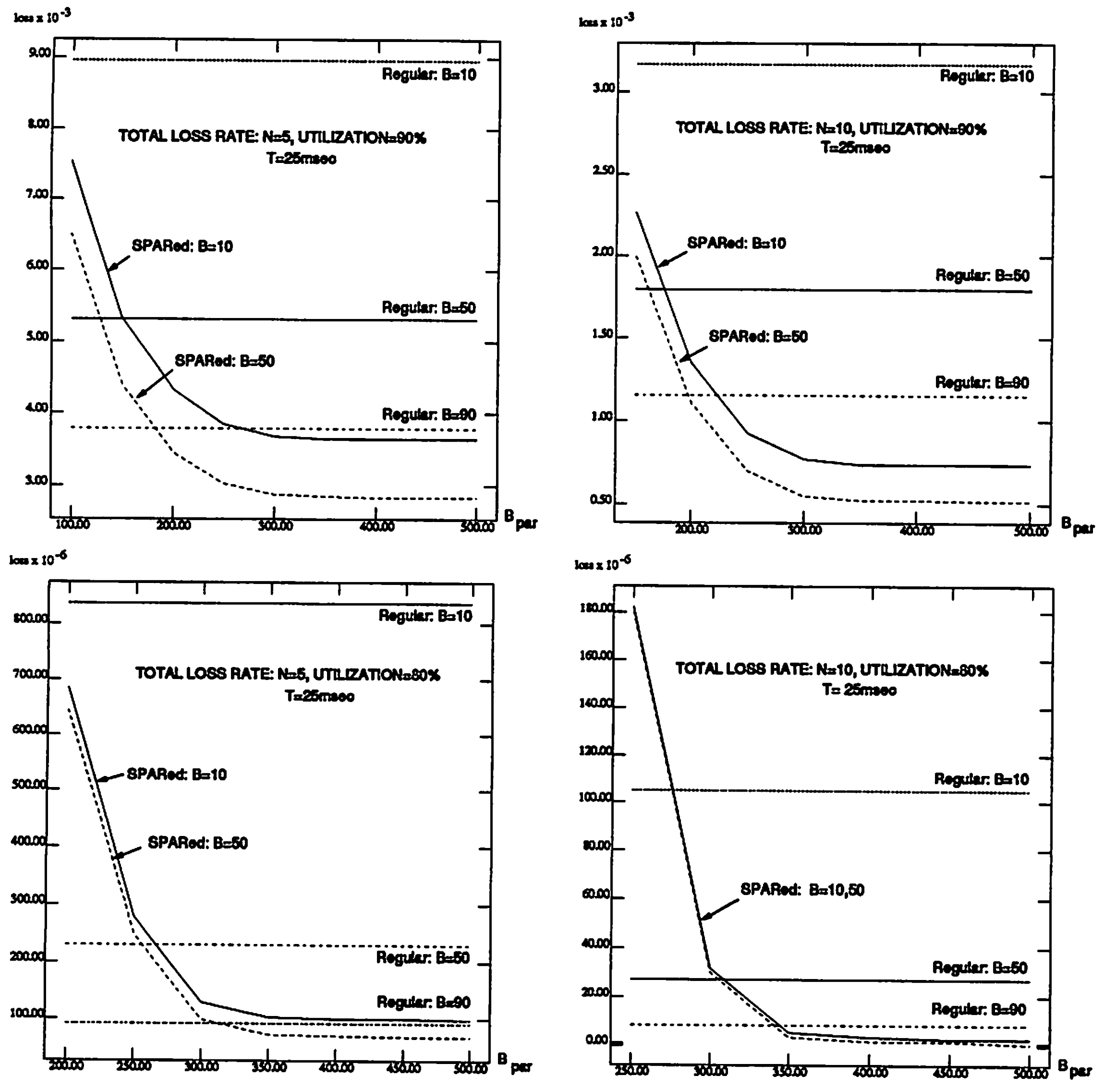

Figure 13: SPARing Loss rates vs B and B_{par} (T = 25msec).

within the network. Moreover costs will be distributed in an equitable fashion, i.e. bursty users will pay more to shape their traffic at the network interface. Another compelling reason to buffer at the network edge is that buffering within the network will make the network service dependent, thus future services may require the redesign and installation of new switches.

An important advantage of PARing over other shaping techniques is the potential for analyzing performance. In this paper we have considered the asymptotic memory, bandwidth, and shaping requirements as well as call acceptance for a simple network model. Our network model demonstrates the alarming possibility of interaction among streams which are not be directly sharing same resource.

A modified version of PARing which improves overall performance was also considered; it corresponds to using a moving average filter to smooth the traffic rate. Preliminary simulations for a particular compressed video source exhibit the type of improvement one might expect. In a real network, performance will depend on a variety of unknowns, the most important being the switches and their service disciplines, and the characteristics of the traffic. We hope in this study to have emphasized the need to focus on time-scales in the design and analysis of networks. Source fluctuations should be considered relative to the size of buffers and/or the delays that can be tolerated. Traffic shaping may be inconsequential if overflows occur as long term accumulations. The typical time-scale of network overflows determines the usefulness of traffic shaping as well as other control strategies.

References

[1] K. Bala, I. Cidon, and K. Sohraby. Congestion control for high speed packet switched networks. In *INFOCOM*, 1990.

[2] J.A. Bucklew. *Large Deviation Techniques in Decision, Simulation and Estimation.* John Wiley and Sons, New York, NY, 1990.

[3] C. Courcoubetis and J. Walrand. Note on effective bandwidth of atm traffic. *preprint.*

[4] G. de Veciana. *Design Issues in ATM Networks: Traffic Shaping and Congestion Control.* PhD thesis, Dept. of EECS, University of California, Berkeley, in preparation, 1993.

[5] G. de Veciana, C. Olivier, and J. Walrand. Large deviations of birth death Markov fluids. *to appear, Probability in the Engineering and Informational Sciences*, 1992.

[6] A. Dembo and O. Zeitouni. *Large Deviations Techniques and Applications.* Jones & Bartlett, Boston, 1992.

[7] A. I. Elwalid and D. Mitra. Analysis and design of rate-based congestion control of high speed networks, i: stochastic fluid models, access regulation. *Queueing Systems*, 9:29–64, 1991.

[8] M. R. Frater. *Estimation of the Statistics of Rare Events in Data Communications Systems.* PhD thesis, Dept. of Systems Engineering Research School of Physical Sciences, The Australian National University, 1990.

[9] M.W. Garrett and M. Vetterli. Congestion control strategies for packet video. In *Fourth International Workshop on Packet Video*, Kyoto, Japan, August 1991.

[10] S. Huang. Source modelling for packet video. In *ICC Proceedings*, 1988.

[11] G. Kesidis. *Cell Loss Estimation in High-Speed Digital Networks.* PhD thesis, Dept. of EECS, University of California, Berkeley, 1992.

[12] S. Low. *Traffic Management of ATM Networks: Service Provisioning, Routing, and Traffic Shaping.* PhD thesis, Dept. of EECS, University of California, Berkeley, 1992.

[13] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J.D. Robbins. Performance models of statistical multiplexing in packet video communications. *IEEE Trans. on Comm.*, 36:834–844, 1988.

[14] H. Ohnishi, T. Okada, and K. Noguchi. Flow control schemes and delay/loss tradeoff in atm networks. *IEEE Journal on Selected Areas in Communications*, 6, 1988.

[15] P. Sen, B. Maglaris, N.E. Rikli, and D. Anastassiou. Models for packet switching of variable-bit-rate video sources. *IEEE JSAC*, 7:865–869, 1989.

[16] J. Walrand. *An Introduction to Queueing Networks.* Prentice-Hall, 1988.