RCBR: A Simple and Efficient Service for Multiple Time-Scale Traffic*

M.Grossglauser	S.Keshav	D.Tse
INRIA	Cornell University	U.C. Berkeley
BP 93	4130 Upson Hall	261M Cory Hall
06902 Sophia Antipolis Cedex	Ithaca NY 14853	Berkeley CA 94720
France	USA	USA
matthias.grossglauser@inria.fr	skeshav@cs.cornell.edu	dtse@eecs.berkeley.edu

Abstract

Variable bit rate compressed video traffic is expected to be a significant component of the traffic mix in integrated services networks. This traffic is hard to manage because it has strict delay and loss requirements while simultaneously exhibiting burstiness at multiple time-scales. We show that burstiness over long time-scales, in conjunction with resource reservation using one-shot traffic descriptors, can substantially degrade the loss rate, end-to-end delay and statistical multiplexing gain of a connection. We use large-deviation theory to model the performance of multiple time-scale traffic and to motivate the design of Renegotiated Constant Bit Rate (RCBR) Service.

Sources using RCBR service are presented with an abstraction of a fixed-size buffer which is drained at a constant rate. They may renegotiate the drain rate to match their workload. Because all traffic entering the network is CBR, RCBR requires minimal buffering and scheduling support in switches. We show that the service is suitable for both stored and online video sources.

An RCBR source must decide when to renegotiate its service rate, and what the new service rate should be. We present a) an algorithm to compute the optimal renegotiation schedule for stored (off-line) traffic, and b) a heuristic to approximate the optimal schedule for online traffic. We also discuss measurement-based admission control for RCBR traffic.

Simulation experiments show that RCBR is able to extract almost all of the statistical multiplexing gain available by exploiting slow time-scale variations in traffic. Moreover, simple admission control schemes are sufficient to keep the renegotiation failure probability below a small threshold, while still offering high

*This work has been presented in part at NOSSDAV '95 [19] and at ACM SIGCOMM '95 [18].

link utilization. Thus, we believe that RCBR is a simple, practical, and effective service for carrying multiple time-scale traffic.[1]

1 Introduction

Video traffic is expected to be a significant component of the traffic mix in integrated services networks. Video is invariably compressed with either constant quality (and variable bit rate) or constant bit rate (and variable quality). Constant bit rate compressed video, which is predominant in current networks, may exhibit visual glitches in information-rich scenes. To minimize these glitches, the coding rate has to be large enough to encode all but a few of the scenes in the video stream, leading to a reduction in the available statistical multiplexing gain [44]. This has led to great interest in variable bit rate video compression, and techniques for carrying such traffic in computer networks [6, 7, 32, 33, 29, 1]. [2]

A key characteristic of a compressed video source is its *burstiness*. A bursty source occasionally transmits at a peak rate significantly larger than its long term average rate. Recent research has determined another key characteristic: the presence of traffic variations over multiple time-scales [34, 35, 12, 13]. Intuitively, there is a variation in source rate not only over a period of milliseconds to seconds, corresponding to variations within a scene, but also over a period of tens of seconds to minutes, corresponding to scenes with differing information content. Taken together, these facts imply that a compressed video source can transmit at its peak rate over multiple time scales.

We discuss in Section 2 how burstiness at multiple time scales, in conjunction with a traditional one-shot traffic descriptor (such as a leaky bucket), leads to performance problems. Instead, we argue that a *renegotiated service* best addresses the presence of burstiness over multiple time-scales. This motivates the design of Renegotiated Constant Bit Rate (RCBR) service for carrying compressed video traffic. Sources using RCBR service are presented with an abstraction of a fixed-size buffer drained at a constant bit rate called the *drain rate*. Sources choose the drain rate to match their current short-term average rate and renegotiate this rate in response to changes in their workload. Because all traffic entering the network is CBR, RCBR requires minimal buffering and scheduling support in switches. We show in Section 3 that the service is suitable for both stored and online video sources and that its signaling overhead is likely to be manageable with current technology. Note that RCBR is the simplest possible renegotiated service.

An RCBR source must determine *when* to renegotiate and *what* rate to choose during a renego-

[1] An earlier version of this paper appeared in ACM SIGCOMM '95 [18]. This version has fewer details on large-deviation modeling, but a more detailed study of admission control.

[2] For notational convenience, we will refer to a variable bit rate compressed video source simply as a 'compressed video source'.

tiation. These decisions constitute a per-source *renegotiation schedule*. We present algorithms for choosing renegotiation schedules in Section 4.

We evaluate RCBR both analytically and through simulation. Our results in Section 5 indicate that RCBR allows a network operator to extract almost all of the statistical multiplexing gain achievable by multiplexing large numbers of compressed video sources. For example, if an MPEG-1 compressed version of the "Star Wars" movie is transferred through our service, and if the average service rate over the lifetime of the connection is 5% above the average source rate of 374kbps, then 300kbit worth of buffering at the end-system and an average renegotiation interval of about 12s are sufficient for RCBR. In contrast, a non-renegotiated service with the same service rate would require about 100Mbit of buffering at the end-system (cf. Fig. 5).

A natural question is how to admit RCBR sources into the network while still allowing network operators to stochastically bound performance metrics such as the renegotiation failure probability and link utilization. We discuss simple measurement-based admission control schemes suitable for RCBR sources in Section 6. We show that a memoryless scheme is not robust. We advocate the use of memory, i.e., history about the past bandwidth of calls, to achieve satisfactory robustness.

While our focus is on compressed video traffic, our results are applicable to multiple time-scale traffic in general. Sections 7 and 8 present related work and place our work in context of other services for carrying variable bit rate traffic.

2 Performance Problems for Multiple Time-Scale Sources

It has been observed by several researchers [34, 35, 12, 13] that compressed video traffic typically exhibits burstiness over multiple time-scales. While the short term burstiness of MPEG sources due to the I, B, and P frame structure is well known, they have found fairly long duration, as long as 30 seconds, when the data rate of the video source is continuously near its peak rate. This is due to scenes with considerable motion or rapid chrominance and luminance changes such as those caused by flashing lights, where, independent of the coding algorithm, the coder generates traffic near its peak rate. Unfortunately, these peak rates are much higher than the long term average rate. For example, we find that for an MPEG-1 compressed version of the "Star Wars" movie, there are episodes where a sustained peak of five times the long term average rate lasts over 10 seconds.

Compressed video traffic is expected to be carried in ATM networks using either CBR or VBR service, and in the Integrated Services Internet using Guaranteed Service [5] (The Integrated Services Internet also allows for a VBR-like Controlled Load service, but this service is too imprecisely defined at the time of this writing to map to any of the ATM classes. Therefore, our remarks apply only to the Internet's Guaranteed Service class.) With CBR service, a source is restricted to a bit rate that it chooses at the time of connection setup. With VBR or Guaranteed Service, the source chooses both a *token bucket size* and a *token rate*. These correspond roughly to the largest size burst allowed from the source into the network and its long-term average rate.

We model all three services as follows. Traffic from a source is queued at a buffer at the end-system, and the network drains the buffer at a given *drain rate*. The drain rate for a CBR source is the connection rate, and for a VBR or Guaranteed Service source is the token rate. With VBR service, data may leave the buffer at a rate greater than the drain rate if the token bucket is non-empty. The key fact is that with a non-renegotiated service, as is the case in both ATM and Integrated Services Internet proposals, a source chooses the drain rate exactly once, at the time of connection establishment. (The Integrated Services Internet proposal– specifically the RSVP resource reservation protocol– does require a source to periodically refresh its reserved rate, and renegotiation could be piggybacked with a refresh. However, refreshes are currently viewed primarily as a mechanism for state management, rather than for rate adaptation. Sources are therefore expected to choose a token rate once and to merely repeat this request when refreshing their reservation [49]).

If sources exhibiting bursts at multiple time scales are allowed only a single drain rate to describe their behavior, they are faced with a series of poor choices. Assume, for the moment, that the drain rate is chosen close to the long term average rate in order to maximize the statistical multiplexing gain in the network. Then, during sustained peaks, the source buffer fills up at the peak rate and is drained at the drain rate. If the peak rate is much higher than the average rate, either the data buffer has to be very large, or the loss rate will be unacceptably high. If the loss rate is made small by provisioning large data buffers, this leads to expensive buffers at end-systems and long delays for the sources. Even if the data buffering costs are not excessive, the ensuing delays may not be tolerable for interactive applications.

With VBR or Guaranteed service, we can deal with sustained bursts by choosing a large token bucket, thus admitting part or all of the burst into the network. We call this the *unrestricted sharing* approach to dealing with bursts. The problem with this approach is that unless intermediate switches and the receiver have large data buffers (which, in some cases, may need to be on the order of tens of megabytes), sources have no assurance that their data will not be lost if bursts coincide. We call this loss of *protection*. Providing protection with unrestricted sharing is expensive and can potentially lead to excessive queueing delays. Note that there is a tradeoff between the drain rate and the largest-size burst that may enter the network. A source can minimize delay and reduce the probability of cell loss, but only at the expense of a reduced statistical multiplexing gain.

Thus, burstiness at slow time-scales with a non-renegotiated drain rate leads either to a) loss of statistical multiplexing gain, b) large data loss rate, c) large buffers in end systems or switches, leading to delays and expensive line cards or d) loss of protection. Current (non-renegotiated) services cannot simultaneously avoid all four problems because sustained peaks in workload are not adequately captured by a static descriptor, such as a leaky bucket. We argue that these peaks are better captured by renegotiation of the drain rate at a slower time-scale. A more detailed discussion of the effectiveness of renegotiation in solving these problems can be found in Section 8.

3 The RCBR Scheme

3.1 RCBR Service Description

With static CBR service, during call setup, a source requests a constant bandwidth from the network [24, 25]. Because a source is described by a single number, the admission control test is trivial. Moreover, because traffic entering the network is smooth, internal buffers can be small and packet scheduling need only be FIFO [17]. With RCBR, a source can renegotiate its service rate. Renegotiation consists of sending a signaling message requesting an increase or decrease in the current service rate. If the request is feasible, the network allows the renegotiation. Upon successful completion of the request, the source is free to send data at the new CBR rate. RCBR therefore retains the simplicity and small buffer requirements of CBR service.

Renegotiation failure What happens if a renegotiation fails? A trivial solution is to try again. Of course, data will build up in the source's data buffer while the second request proceeds and there is the possibility of excessive delay, and even data loss. This may not be acceptable for some sources. Such sources might reserve resources at or close to the peak rate, so that the frequency of renegotiation is highly reduced and so is the possibility of renegotiation failure. There is a three-way tradeoff between buffer size (and delay), requested rate and the frequency of renegotiation. In any case, note that even if the renegotiation fails, *the source can keep whatever bandwidth it already has.*

Second, during admission control, a switch controller might reject an incoming call even if there is available capacity, if the resources used by the new call will make future renegotiations more likely to fail. This allows the network operator to trade off call blocking probability and renegotiation failure probability. We consider admission control in more detail in Section 6.

Finally, the signaling system could ask the user or application (perhaps out of band) to reduce its data rate. Since the network interface (i.e. the session layer or NIU) is expected to be no more than a few milliseconds away from the end point, the control loop between the network interface and the user will be tight, so that responding to such signals should be easy, particularly for adaptive codecs [27]. Recent work suggests that even stored video can be dynamically requantized in order to respond to these signals [38, 10].

Thus, there are several viable alternatives for dealing with renegotiation failures. With an appropriate combination, some users can choose to see few or no renegotiation failures, while others might trade off a non-zero renegotiation failure rate for a lower cost of service.

Stored and Interactive Sources Stored video (off-line) and interactive (on-line) applications use RCBR services differently. Off-line sources can compute the *renegotiation schedule* in advance and can initiate renegotiations in anticipation of changes in the source rate. Moreover, if all systems in the network share a common time base, advance reservations could be done for some or all of the data stream [47]. Interactive applications must compute the renegotiation schedule on-the-fly. For such applications, we propose that an active component monitor the buffer between the application and the network and initiate renegotiations based on the buffer occupancy. This monitor could be part of the session layer in an ISO protocol stack, or reside in the Network Interface Unit (NIU) for "dumb" endpoints. It would need to be activated only when data is written to or drained from the buffer. Note that in both the on-line and off-line cases, renegotiation signaling and data transfer occur in parallel. Algorithms for computing the renegotiation schedule for off-line and on-line applications are presented in Section 4.

3.2 Implementation

During renegotiation, a switch controller need *not* recompute routing, allocate a connection identifier or acquire housekeeping records. Thus, signaling for renegotiation is much less expensive than signaling for call setup and need not use the same protocol. This allows us to exploit lightweight signaling mechanisms for renegotiation. A hardware implementation of signaling for renegotiation is described in [3].

In an ATM network, sources can reuse the Resource Management (RM) cell mechanism, originally proposed for ABR service, for lightweight signaling. An RCBR source sets the Explicit Rate (ER) field in the RM cell to the difference between its old and new rates [3]. On receiving an RM cell, a switch-controller (or a dedicated hardware module, as in ABR) determines the output port of the VCI in one lookup, and the utilization and capacity of the output port in a second lookup. With this information, it checks if the current port utilization plus the rate difference is less than the port capacity. If this is true, then the renegotiation request succeeds, and the VCI and port statistics are updated. Otherwise, the controller modifies the ER field to deny the request. Note that the logic to modify the ER field with RCBR is simpler than that required for fair-share computation in ABR. Thus, the deployment of ATM switches with ABR support is an existence proof that RCBR support in ATM switches is feasible.

[3]We use a difference because this simplifies the computation at the switch controller, which need not keep track of the source's rate. This has the problem of parameter drift in case of RM cell loss. To overcome this, we can resynchronize rates by periodically sending an RM cell with the true explicit rate, instead of a difference.

In the Integrated Services Internet, sources and receivers periodically refresh their network reservation state using the RSVP signaling protocol [49]. A source periodically emits a PATH message describing its characteristics, and each receiver periodically emits a RESV message requesting a reservation. To renegotiate its service rate, a source should change its traffic description (flowspec) in the PATH message, and the receivers should correspondingly change their reservation in the RESV message. We do not have enough experience with RSVP to determine whether this mechanism is sufficiently lightweight for renegotiation. If this is not the case, we may need to augment RSVP with a lightweight renegotiation protocol. In any case, we anticipate that renegotiations will happen only around every ten seconds or so (see Section 5.2), so the overhead for RCBR at each source is inherently small.

Note that in order to limit the renegotiation rate, it is likely that a user will be charged for each renegotiation, just as users are now charged per call setup. This affects the choice of renegotiation schedule, as discussed in Section 4.

3.3 Scaling

We now consider how well RCBR scales with latency in the path, number of sources, and number of hops. Scaling with path latency is different for on-line and off-line applications. Off-line applications are insensitive to path latency because they can compensate for an increased latency by initiating renegotiation earlier. However, the performance of applications with on-line RCBR decreases with an increase in latency, because these applications must predict their future data rate, and prediction accuracy decreases with increased latency. This can be compensated for by increasing the end-system buffer or by asking for more bandwidth than needed, thus reducing the statistical multiplexing gain. We do not yet have analytical expressions or simulation results studying the effect of renegotiation delay on RCBR performance.

Signaling load increases linearly with the number of RCBR sources in the network. With hardware support, we believe that an ATM switch can support several tens of thousands of simultaneous RCBR sources. The bottleneck in RM cell processing is the time taken to lookup per-VCI state. Since RCBR support does not require per-VCI state, we do not anticipate difficulties in scaling ATM switches to handle more renegotiating sources. Scaling of Integrated Service routers is still a matter of speculation.

As the mean number of hops in the network increases, the probability of renegotiation failure is likely to increase, since each hop is a possible point of failure. Moreover, the net renegotiation signaling load on the network also increases. However, if there is a simultaneous increase in the number of alternate routes in the network, then load balancing at the call level might reduce the load at each hop, thus compensating for this increase. This is still an open area for research.

4 Computation of Renegotiation Schedules

In this section, we address the problem of deciding when to request a bandwidth renegotiation from the network, and how much bandwidth to ask for. We present two algorithms that transform a given data rate function into a stepwise CBR data rate function. The first algorithm determines an optimal schedule for a playback application based on total knowledge of the user's data rate function and a pricing model discussed below. The second algorithm is a causal heuristic that could be used for interactive users, where the rate function is not known in advance.

4.1 Optimal Renegotiation Schedule

We model the problem with a slotted time queue. For video, a time slot would typically be the duration of a frame. Renegotiations occur on the boundary between slots. Let $r_i, i = 0, 1, \ldots, N-1$ denote the amount of data entering the queue during time slot i, and let s_i denote the service rate during time slot i. The session duration is N time slots. We assume the service rate during any time slot is in a given set $\mathcal{C} = \{c_0, c_1, \ldots, c_{K-1}\}$.

We have assumed a constant cost per renegotiation ϕ and a cost γ per *allocated* bandwidth and time unit. Therefore, the total cost is given by

$$\phi \cdot \sum_{i=1}^{N-1} (1 - \delta(s_{i-1}, s_i)) + \gamma \cdot \sum_{i=0}^{N-1} s_i \tag{1}$$

with

$$\delta(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases}$$

For a given r_i, the optimal allocation minimizing the total cost, has to be found, subject to the buffer constraint

$$0 \leq b_i \leq B \text{ for } i = 0, 1, \ldots, N-1 \tag{2}$$

where b_i is the queue size at the end of time slot i, with

$$b_i = \begin{cases} 0 & \text{if } i < 0 \\ \max\{b_{i-1} + r_i - s_i, 0\} & i = 0, 1, \ldots, N-1 \end{cases} \tag{3}$$

We solve this optimization problem with a Viterbi-like algorithm [45]. Let us first introduce some notation (cf. Fig. 1). A *node* is a 4-tuple (i, k, b, w), where i denotes (discrete) time, $k \in \{0, \ldots, K-1\}$ denotes a bandwidth allocation $c_k \in \mathcal{C}$, $b \in \{0, \ldots, B\}$ denotes a buffer occupancy, and w denotes the weight, which equals the partial cost of the best path to this node. A *branch* connects a node (i, k, b, w) to another node $(i+1, k', b', w')$ if $b' = \max\{b_i + r_{i+1} - c_{k'}, 0\}$. It has an associated weight of $\gamma \cdot s_{i+1} + \phi \cdot (1 - \delta(s_i, s_{i+1}))$. A branch represents one step in the evolution of the system state, given a choice of the new rate allocation $c_{k'}$. A *path* is a sequence of branches. The cost of a path is the sum of the cost of its branches. All possible paths form the *trellis*. A *full path* is a path connecting a node with $i = 0$ with a node with $i = N - 1$, and corresponds to a feasible renegotiation schedule.

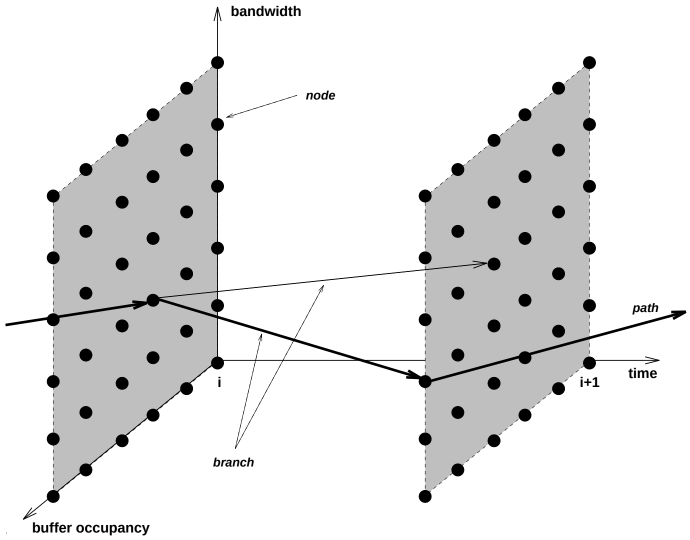

Figure 1: An illustration of the trellis to be used for the Viterbi-like algorithm.

Now we can formulate the optimization problem as follows: find the shortest path from some node at time zero to some node at time $N - 1$. The algorithm to do this is presented below.

1. Set $i = 0$. Create the initial set of nodes $(0, k, 0, 0)$ for $k \in \{0, \ldots, K - 1\}$.

2. Create all the branches between nodes of slot i and nodes of slot $i+1$. Set the weight according to (1) for the nodes of slot $i + 1$.

3. Prune paths according to Lemma 1 given below.

4. Increment i and repeat steps 2 and 3 as long as $i < N$.

5. Choose one of the paths with the minimum weight as the solution.

We now present a lemma that governs the pruning of paths.

Lemma 1 *A path X going through a node $x = (i, k_x, b_x, w_x)$ is not optimal if there exists a path Y*

through a node $y = (i, k_y, b_y, w_y)$ *such that* [4]

$$b_y \leq b_x \text{ and } w_y \leq w_x - \begin{cases} \phi & \text{if } k_y \neq k_x \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Proof: Assume the condition is true. First, if $k_x = k_y$, then path Y has smaller or equal buffer occupancy and smaller or equal weight than path X. Due to the buffer constraint, for all future time slots, the best full path containing X must have a bandwidth allocation that is at least the bandwidth allocation of the best full path containing path Y. Therefore, it cannot have a lower weight than the best full path containing Y. Second, if $k_x \neq k_y$ then for any $k \in \{0, \ldots, K-1\}$ such that a branch from x to a node $x' = (i+1, k, b_{x'}, w_{x'})$ exists, there exists a branch from y to a node $y' = (i+1, k, b_{y'}, w_{y'})$ such that $b_{y'} \leq b_{x'}$, as the service rate in interval $i+1$ is the same and by assumption, $b_y \leq b_x$. As the difference in cost of the branch connecting y to y' and the branch connecting x to x' cannot be larger than ϕ, the first part of the proof applies to x' and y'. □

Instead of the buffer bound (2), it is also possible to enforce a delay bound. This might be desirable in real-time applications, if sufficient buffer space is available, but the Quality of Service still requires to keep delays low. The condition for all data entering during time slot $i - D$ to have left at the end of time slot i is

$$b_{i-D} \leq \sum_{j=-D+1}^{0} s_{i+j} \quad i = D, \ldots, N-1 \tag{5}$$

The runtime complexity of the optimization algorithm very much depends on the cost ratio ϕ/γ, the buffer size B and above all the number of bandwidth levels K. Also, the user rate function $\{r_i\}$ has has an impact on how many candidates remain valid at each time slot. We have found that if we restrict K to about 20, optimizations can be done in reasonable time, even for long traces like the Star Wars movie (approx. 174000 samples) [12]. For larger K, e.g. 100, it quickly becomes impracticable, because of an explosion in the number of paths that have to be considered in the algorithm. For example, for $K = 20$ (with the bandwidth levels chosen uniformly within $c_0 = 48$kbps and $c_{K-1} = 2.4$Mbps), the computation took about 20 minutes on an Sun UltraSparc 1, while with $K = 100$, the computation took more than a day.

We call *bandwidth efficiency* the ratio of the original stream's average rate to the average of the piecewise constant service rate, i.e.

$$\frac{\sum_{i=0}^{N-1} r_i}{\sum_{i=0}^{N-1} s_i}.$$

The graph "OPT" in Figure 2 shows the mean renegotiation interval and the bandwidth efficiency for various choices of the cost ratio ϕ/γ, for a buffer size $B = 300$kbit, which represents a buffering

[4]Note that this allows us to do more than the "standard Viterbi" pruning, i.e. among paths terminating in a common node, keep only the one with the lowest weight. We can also prune across nodes.

delay of slightly less than 1 second (recall that the average rate of the trace is 374 kbps). It is clear from Fig. 2 that there exists a tradeoff between bandwidth efficiency and renegotiation frequency. This tradeoff depends on the cost ratio ϕ/γ: raising the price for renegotiation results in a lower renegotiation frequency, but also in a lower bandwidth efficiency, and vice versa. The network operator can announce these prices to the user, and the user optimizes his network usage accordingly. Note how close the bandwidth efficiency gets to one with very reasonable renegotiation frequencies; for example, with one renegotiation every 7 seconds, we achieve over 99% of bandwidth efficiency. This is a clear manifestation of the slow time-scale behavior of compressed video streams.

4.2 Causal Renegotiation Schedule

For interactive (online) sources, the optimization algorithm described above cannot be used to determine optimal renegotiation points. For such sources, causal heuristics have to be used to make decisions about requesting new rates. Such heuristics predict the future bandwidth requirement based on some statistics collected in the past. The goal of this section is to show that heuristics resulting in satisfactory performance do indeed exist, although their derivation is somewhat *ad hoc*.

The heuristic we present is based on a AR(1) bandwidth estimator and on buffer thresholds. Three parameters have to be tuned: a high and a low buffer threshold B_h and B_l, respectively, and a time constant T, which should reflect the long-term rate of change of the rate function. The rate predictor we have used is

$$\widehat{r}_{i+1} = (1 - T^{-1})\widehat{r}_i + T^{-1}(r_i + \max\{b_i - B_h, 0\}) \tag{6}$$

where r_i is the actual incoming rate during slot i, and b_i is the buffer size at the end of slot i. The additional term $T^{-1} \cdot \max\{b_i - B_h, 0\}$ in the estimator adds the bandwidth necessary to flush the current buffer content within T. This is necessary to have a sufficiently fast reaction to sudden large buffer buildups.

The algorithm is very simple. Let

$$s_{new} = \left\lceil \frac{\widehat{r}_{i+1}}{\Delta} \right\rceil \Delta \tag{7}$$

with Δ the bandwidth allocation granularity. A new bandwidth s_{new} is then requested if

$$(b_i > B_h \text{ and } s_{new} > s) \text{ or } (b_i < B_l \text{ and } s_{new} < s) \tag{8}$$

It can be seen in Fig. 2 that using the heuristic, we need about one renegotiation a second to achieve 95% of bandwidth efficiency (with B_l = 10kbit, B_h = 150kbit, T = 5 frames, and Δ = 150kbps). Although this is considerably less than what can be achieved with the optimal allocation, it still represents a relatively small load on the signaling system. However, this gap

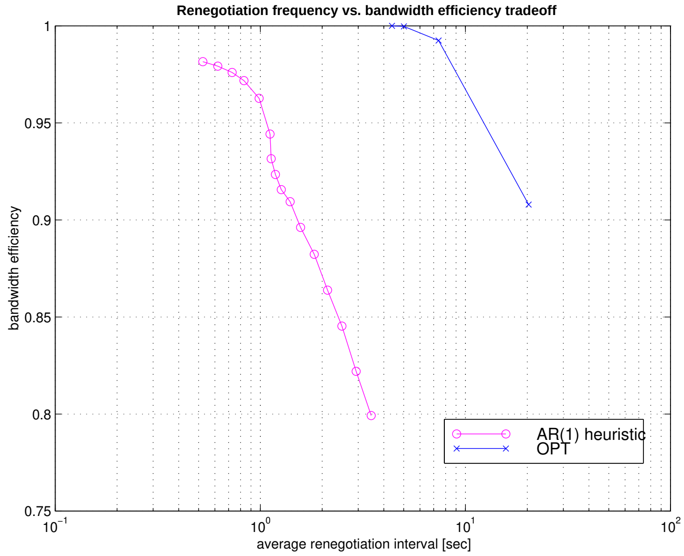

Figure 2: The tradeoff between bandwidth efficiency and renegotiation frequency for the AR(1)-based heuristic, compared to the optimum, for the "Star Wars" trace. The parameters for the AR(1) heuristic are: $B_l = 10$kbit, $B_h = 150$kbit, $T = 5$ frames, and Δ is varied from 25kbps (left) to 400kbps (right). In this example, the buffer occupancy never exceeds $B = 300$kbit.

suggests a potential for better heuristics, and we hope to address this problem in future research. For example, the prediction quality could be improved by taking into account the inherent frame structure of MPEG encoded video.

5 Performance of RCBR

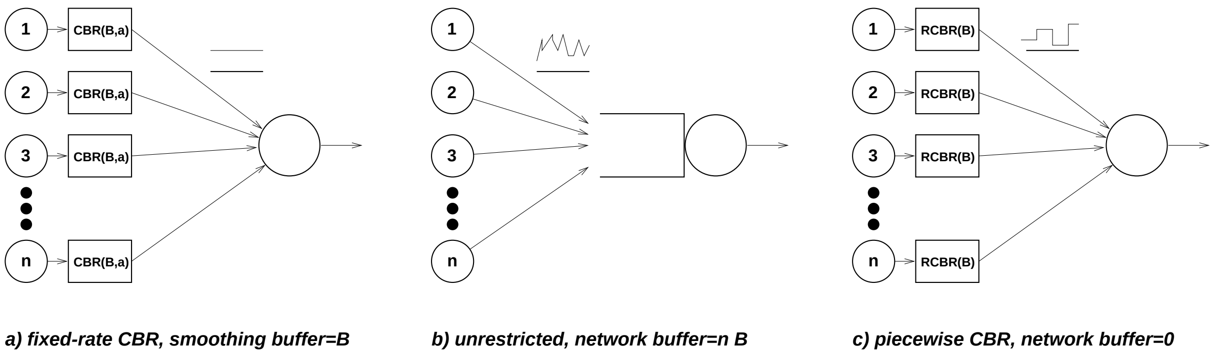

Figure 3: The three scenarios to assess statistical multiplexing gain (SMG) of our proposed service.

In this section, we would like to get a better understanding of the statistical multiplexing gain (SMG) achievable using the RCBR scheme, by means of both a theoretical analysis of a multiple time-scale source model as well as simulation experiments on real traffic. More specifically, we compare the SMG of RCBR with that of two other scenarios (cf. Fig. 3). The first scenario (a) represents traditional CBR service, with a smoothing buffer of size B at the network entry and a fixed CBR rate $\bar{c}$ for each source. Here, there is no multiplexing between traffic of different sources. The second scenario (b) multiplexes n streams without any restriction on a server with rate c and buffer size nB. This gives the maximum achievable SMG for the given sources. The third scenario (c) represents the RCBR approach. Each source is smoothed by a dedicated buffer of size B and transformed into a stepwise CBR stream, which is then transported without further buffering in the network (except some cell level buffering). The total service rate is c and the total amount of buffering is fixed at nB in all three scenarios. While the theoretical analysis gives insights as to the nature of the SMG captured by the RCBR, the experimental results quantify the amount of gain for video traffic.

5.1 Analysis of a Multiple Time-Scale Model

We consider the following discrete-time traffic model for an individual video source. Let X_t be the amount of data (measured in bits, bytes, cells etc.) generated per time-slot (duration of a frame,etc.). The process $\{X_t\}$ is *modulated* by an irreducible finite state Markov chain such that the value of X_t is a function of the current state. The Markov structure models the correlation in the data generation rate over time. The state space $\mathcal{S}$ is decomposed into a union of disjoint subsets $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_K$; each $\mathcal{S}_k$ can be interpreted as the state space of a *fast time-scale sub-chain*. The dynamics within each sub-chain model fast time-scale behavior (such as correlations between adjacent frames). Transitions between various sub-chains, on the other hand, happen very rarely compared with the transitions inside each sub-chain; these transitions model the slow time-scale dynamics of the traffic stream (such as scene change). Let $\alpha_1, \alpha_2, \ldots, \alpha_m$ be the probabilities of these rare transitions; these are very small parameters. Thus, the source would typically spend a long time in a sub-chain, and then occasionally jump to a different sub-chain. In the analysis below, we are interested in the regime when the buffer time-scale is large enough to smooth out the fast time-scale fluctuations of the traffic, but is small compared to the slow transition time-scale.

This multiple time-scale Markov-modulated model has been used in several video traffic studies [40, 31]. The sustained peak observed by several researchers corresponds to remaining in a high-rate sub-chain for a long time in this multiple time-scale model. See Figure 4 for an example of a source with three sub-chains.

We shall now characterize the resource requirements for multiple time-scale sources under the

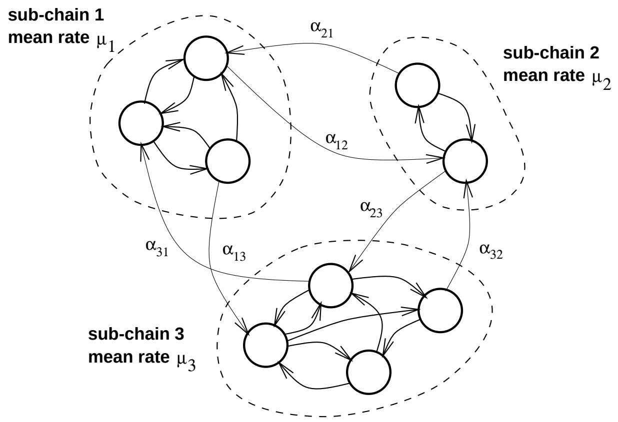

Figure 4: A multiple time-scale source with 3 sub-chains.

three scenarios in Fig. 3, for given loss probability requirements. Consider first scenario (a), when each individual stream is smoothed by a buffer B and allocated a fixed CBR rate of $\bar{c}$. The minimum drain rate $\bar{c}$ required to achieve a target QoS buffer overflow probability p_{qos} is known as the *equivalent bandwidth* $e(p_{qos})$ of the source, and for *single time-scale* Markov sources, this has been explicitly computed in terms of the statistics of the source [14, 11, 28]. This equivalent bandwidth is based on a large deviations estimate of the buffer overflow probability, in the regime of large buffer size B. It can be shown that the equivalent bandwidth is between the mean and peak rates of the stream, and it measures the amount of smoothing of the stream by buffering. A large buffer B in this context means that the buffer is sufficiently large to smooth out the fluctuation of the traffic stream.

Analogous results have been obtained for multiple time-scale Markov traffic [41]. For multiple time-scale sources, one now has to look at the *joint* asymptotic regime when simultaneously the rare transition probabilities α_i's are close to zero and the buffer size B is large enough to absorb the fast time-scale fluctuations of the stream. It is shown in [41] that the equivalent bandwidth $e(p_{qos})$ of the multiple time-scale stream is given by

$$e(p_{qos}) = \max_{1 \le k \le K} e_k(p_{qos}), \tag{9}$$

where $e_k(p_{qos})$ is the equivalent bandwidth of the kth fast time-scale sub-chain when considered in isolation. The intuition is that buffer overflows are due mainly to the effects of the most bursty sub-chain, and thus the drain rate needed for the entire stream is the drain rate of that particular sub-chain. In particular, the drain rate needed is greater than $\hat{\mu}$, the maximum of the average rates of the sub-chains. This implies that the gain due to buffering alone is rather limited for multiple time-scale traffic, as the CBR rate needed for the stream is determined by the worst-case sub-chain. The theoretical result also makes precise the intuition we presented in Section 2, that static traffic descriptor (in this case, the CBR rate) leads to a wasteful allocation of resources for multiple time-scale traffic.

To get significant multiplexing gain beyond that obtained by buffer smoothing, the limitation imposed by the slow time-scale dynamics can be overcome by multiplexing many independent streams. By a law of large number effect, the probability that many streams are simultaneously in a bursty sub-chain is small, so that a small loss probability can be guaranteed even if the capacity allocated per stream is less than $\hat{\mu}$. This is shown in scenario (b) in Fig. 3, where n independent and statistically identical streams are multiplexed. If we scale the total link rate $c := n\bar{c}$ and the total buffer as nB (i.e. the link capacity and buffer space *per stream* is fixed in this scaling), an estimate of the buffer overflow probability, in the regime of large n, can be obtained in terms of *only* slow time-scale statistics of the individual stream (with the fast time-scale dynamics averaged out.) [41]. Specifically, consider a random variable which takes on the value μ_k with probability π_k, where π_k is the steady-state probability that the stream is in sub-chain k and μ_k is the mean rate of sub-chain k. Let L be the log moment generating function of this random variable:

$$L(r) \equiv \log \sum_{k=1}^{K} \pi_k \exp(\mu_k r).$$

and define L^* by:

$$L^*(\mu) = \max_{r>0}[\mu r - L(r)],$$

the *Legendre* transform of L. Then the asymptotic estimate of the loss probability when there are many sources and the buffering B per source large is given by:

$$p \approx \exp(-L^*(\bar{c}) \cdot n) \tag{10}$$

Note that (10) is simply the Chernoff's estimate of the probability that the streams are in a combination of sub-chains whose total mean rate exceeds the channel capacity [46, 23]. Note that this estimate does not depend on the fast time-scale statistics of the streams nor the specific value of buffer size B, provided that it is large enough to absorb the fast time-scale variations of the streams. This result can be interpreted as a decomposition of the gain from multiplexing large number of multiple time-scale streams in a *buffered* node into two components. The first component is the gain obtained from *buffering*: its effect is essentially to remove the time-scale fluctuations of the sources. The second component is the gain from *averaging* between sources: it only depends on the slow time-scale statistics, and is the same as that obtained in a *bufferless* system with the fast time-scale fluctuations removed from At the slow time-scale, the buffer is too small to have any significant effect. Note also that for a target overflow probability p_{qos}, the total link rate needed can be computed from eqn. (10).

Finally, we consider the RCBR scenario, (c) in Fig. 3, where the multiplexing node is bufferless and users have dedicated buffer. We characterize how much of the multiplexing gain in the shared buffer case (scenario (b)) our proposed scheme can capture. Assume that the scheme does an ideal job in separating the slow and fast time-scales, such that it renegotiates a new CBR rate whenever the source jumps from a fast time-scale sub-chain to another. For a buffer overflow probability requirement p_{qos}, the new CBR rate it should renegotiate for is the equivalent bandwidth $e_k(p_{qos})$ of the sub-chain k the source enters. Since π_k is the steady-state probability that the stream is in sub-chain k, the stream will demand a CBR rate of $e_k(p_{qos})$ for π_k long-term fraction of time. The probability of renegotiation failure is roughly the probability that the total CBR bandwidth demand exceeds the available capacity; for large n, we can use Chernoff's estimate to approximate this as

$$\exp(-L_e^*(\bar{c}) \cdot n), \tag{11}$$

where

$$L_e(r) = \log \sum_{k=1}^{K} \pi_k \exp(e_k(p_{qos}, B) \cdot r) \qquad L_e^*(\mu) = \max_{r>0}[\mu r - L_e(r)]$$

and $\bar{c}$ is the link capacity per source. Comparing this to the loss probability (10) when there is a shared buffer of size nB, we see that this renegotiation failure probability is larger since the equivalent bandwidth $e_k(p_{qos})$ of every sub-chain is greater than its mean rate μ_k. Viewed in another way, the capacity per stream needed for the same level of performance is greater in our scheme. This discrepancy in bandwidth requirement is due to the fact that our scheme does not take advantage of a large shared buffer to effectively absorb all fast time-scale variations through statistical multiplexing. Thus, out of the two components of the SMG in the shared buffer case, RCBR extracts the component obtained from averaging between sources. Our scheme essentially focuses on the gain in the averaging of the slow time-scale dynamics rather than the smoothing of the fast time-scale dynamics. However, for sources with small fast time-scale fluctuations superimposed on larger slow time-scale variations, the equivalent bandwidths of the sub-chains will be close to the mean-rates for reasonably sized buffers, and the discrepancy will be small. This is further substantiated by the experimental results presented next.

5.2 Experimental results

We shall now present simulations results comparing the performance in the three scenarios in Fig. 3. The stream we have used is the MPEG-1 encoded trace of the Star Wars movie [12]. The n sources are randomly shifted versions of this trace. The buffer size B was chosen as 300kbit, slightly more than the maximum size of three consecutive frames in the trace. This approximately corresponds to the buffering of current video codecs. The renegotiation schedule used in the experiments is computed using the off-line optimization algorithm described in Section 4.1, with a bandwidth granularity of 1 kbps and an average of 1 renegotiation every 12 seconds.

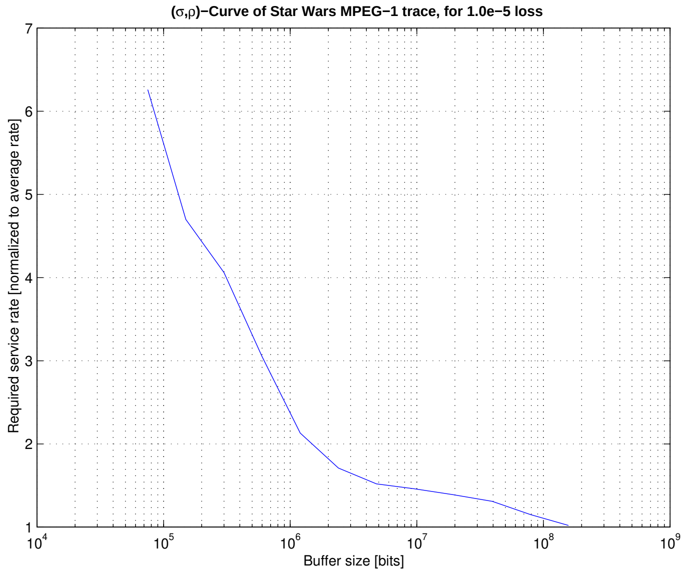

Figure 5: The (σ, ρ)-curve of the video trace for 10^{-5} loss.

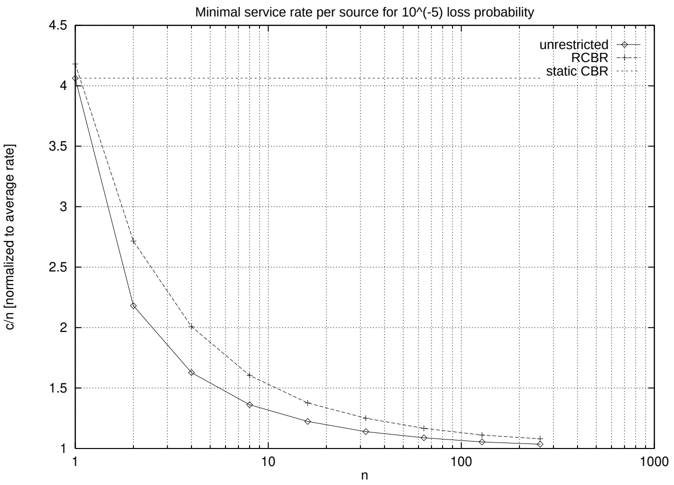

Figure 6: Statistical multiplexing gain (SMG) achievable for 10^{-5} loss probability.

To assess the SMG for all three scenarios, we have determined the channel service rate per stream c/n, as a function of n, needed to guarantee a desired bit loss probability. In scenario (a) and (b), bits are lost due to buffer overflow. In scenario (c), bits are lost due to failure in renegotiating for a higher CBR rate (in which case we assume the source has to temporarily settle for whatever bandwidth remaining in the link until more bandwidth becomes available). Determining c is straightforward for scenario (b). For scenarios (a) and (c), we find for each n the *minimum* c that guarantees the desired loss probability: for each n, we do a binary search on c; for each step in the search, we do many simulations, where each simulation has a randomized phasing of the sources, and compute the average fraction of bits lost as an estimate of the loss probability. At each step, we repeat the simulations until the sample standard deviation of the estimate is less than 20% of the estimate. Results for 10^{-5} loss probability requirement are depicted in Fig. 6.

In the CBR case (a), the bandwidth per stream is $\bar{c}$, of course, regardless of the number of streams n. Note that $\bar{c}$ can be determined from the corresponding (σ, ρ) curve of this trace in Fig. 5 (For a given buffer size σ, this curve gives the minimum service rate ρ such that the fraction of bits lost is less than 10^{-5}.) As has been previously observed in the literature, this is close to the peak rate [35, 13]. For the given buffer size and loss ratio, $\bar{c}$ is 4.06 times the trace's average rate of 374kbps.

Our scheme achieves slightly less SMG than the unrestricted case because buffers are not shared and the fast time-scale multiplexing gain is not exploited, as explained in the theoretical analysis. Nevertheless, we are able to extract most of the SMG, especially for a large number of multiplexed streams. For example, for $n = 100$ streams, we require less than a third of the bandwidth of the static CBR approach. Asymptotically, the value for c/n for the stepwise CBR function approaches the inverse of the bandwidth efficiency obtained in the optimization algorithm.

6 Admission Control

In this section we present some analytical and experimental results on admission control schemes suitable for RCBR. RCBR belongs to the class of *statistical services*. Statistical services are based on a stochastic traffic model, and the QoS guarantee to the user, in this case the renegotiation failure probability, is stochastic in nature. The advantage of a statistical service over a deterministic service is the higher statistical multiplexing gain that can be achieved, as we have noted in Section 2. A statistical service has the disadvantage of being hard to police. Also, it is cumbersome or impossible for the user to come up with a tight a-priori traffic descriptor. Therefore, we propose to use *measurement-based admission control (MBAC)* in conjunction with RCBR [26, 15, 42, 20]. Measurement-based admission control shifts the burden of traffic parameter specification from the user to the network. Instead of the user giving an explicit traffic specification, the network attempts to "learn" the statistics of existing calls by making on-line measurements. This approach has several advantages. First, the user-specified traffic descriptor can be trivially simple (e.g. peak rate). Second, an overly conservative specification does not result in an overallocation of resources for the entire duration of the call. Third, policing is reduced to enforcing peak rate. The goal of this section is to illustrate some of the problems of MBAC, as well as possible approaches to devise robust schemes.

Let us first discuss the admission control problem assuming the traffic specification is known. More specifically, given a renegotiation schedule, we can compute the empirical distribution (histogram) of bandwidth requirements throughout the lifetime of a call, i.e. the fraction of time π_k that a bandwidth level c_k is needed during the call, $k = 1 \ldots K$. This distribution can be viewed as the traffic descriptor of the call. When there are n such calls sharing a link of total capacity c, the renegotiation failure probability p_f can be estimated by Chernoff's approximation as in (11):

$$p_f \approx \exp\left(-L^*\left(\frac{c}{n}\right) \cdot n\right) \tag{12}$$

where

$$L(r) = \log \sum_{k=1}^{K} \pi_k \exp(c_k r) \qquad L^*(\mu) = \max_{r>0}[\mu r - L(r)]$$

Using this formula, the maximum number of calls the system can carry for a given threshold p_{qos} on the renegotiation failure probability can be computed, and new calls will be rejected when this number is exceeded. Note that the system will deny new calls even when there is available capacity, so as to safeguard against fluctuations of bandwidth requirements of the calls already admitted. Thus, Chernoff's approximation quantifies the amount of slack needed in the available capacity. The accuracy of this approximation is quite good. We refer the reader to [18] for an experimental verification of the Chernoff bound.

In practice, we often do not have a reliable traffic descriptor. Even for stored video, where the empirical bandwidth distribution could be computed in advance, user interactivity (fast forward, pause etc.) reduces the accuracy of this descriptor. However, we can estimate the traffic descriptor in the following way.The idea is simply to estimate the distribution by measuring the current state of the network, and use the estimate as the proxy for the true distribution. More specifically, the scheme determines the number of calls $n_k(t)$ that is currently reserving bandwidth level c_k, for each k ($k = 1, \ldots, K$). This yields an empirical distribution $\{\hat{\pi}_k\}$ of bandwidth requirements for a typical call, where $\hat{\pi}_k = \frac{n_k(t)}{n(t)}$ and $n(t)$ is the number of calls currently in the system at time t. The empirical distribution $\{\hat{\pi}_k\}$ is then used in place of the actual distribution $\{\pi_k\}$ in (12) to estimate the renegotiation failure probability, based on which an admission control decision is made. In control theory, this controller is said to be *certainty equivalent*: the controller assumes that the measured values are the true parameters and acts like the optimal controller having perfect knowledge of the values of those parameters. Moreover, this scheme is also *memoryless*, i.e. every time a new call arrives, the scheme uses only information about the *current* state of the network in making the decision of accepting or rejecting the call. This memoryless certainty-equivalent scheme has also been studied by Gibbens et. al. [16] in the context of admission control of on-off sources.

Note that the error associated with any estimation procedure can translate into erroneous call admission decisions, which in turn can compromise the QoS provided to users. Furthermore, a measurement-based call admission controller is a dynamical system, with call arrivals and departures, and parameter estimates that vary with time. The dynamics of this system have a large impact on the performance of the MBAC. In particular, we now show that a memoryless admission controller as described above is *not robust*.

We discuss our simulation results obtained for the memoryless MBAC. We compare its performance with the scheme having perfect knowledge, in the dynamic scenario where calls arrive and depart from the system. In particular, we are interested in two performance measures: the steady-state renegotiation failure probability and the average fraction of the total bandwidth utilized. The success of the measurement-based admission control scheme is evaluated by how well it meets the QoS-requirement (in terms of renegotiation failure probability) and how close its bandwidth utilization is to that of the optimal scheme with perfect *a priori* knowledge of call statistics.

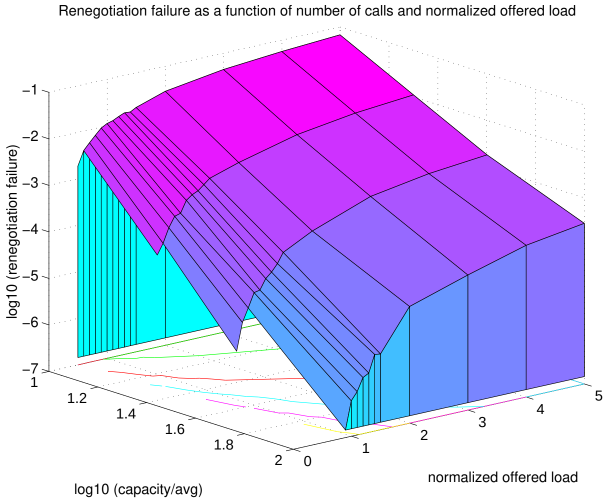

Figure 7: The memoryless scheme: renegotiation failure probability.

The simulation set-up is as follows. Each call is a randomly shifted version of a Star Wars RCBR schedule. Calls arrive according to a Poisson process of rate λ. [5] We measure both the average utilization and the renegotiation failure probability. Each interval of the length of the trace (approximately two hours) provides us with one sample for these probabilities. We collect samples until the 95%-confidence interval for both probabilities is sufficiently small with respect to the estimated value (within +/- 20% of the estimated value). For the renegotiation failure probability, we also stop if the target failure probability of 10^{-5} lies to the right of the confidence interval, i.e., if we are confident that the actual failure probability is lower than the target. This is necessary in order to terminate simulations within reasonable time when the observed renegotiation failure is very low (e.g. 10^{-8}).

[5] Note that as a by-product of using RCBR schedules instead of full per-frame traces as input, the simulation efficiency is greatly improved, as we only need to simulate the renegotiation events instead of each frame.

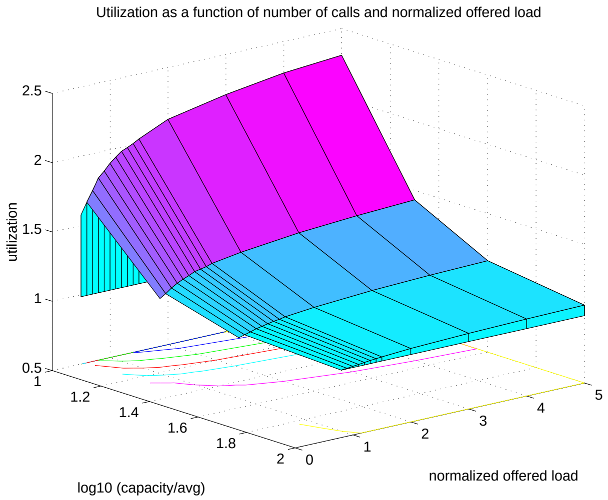

Figure 8: The memoryless scheme: normalized utilization.

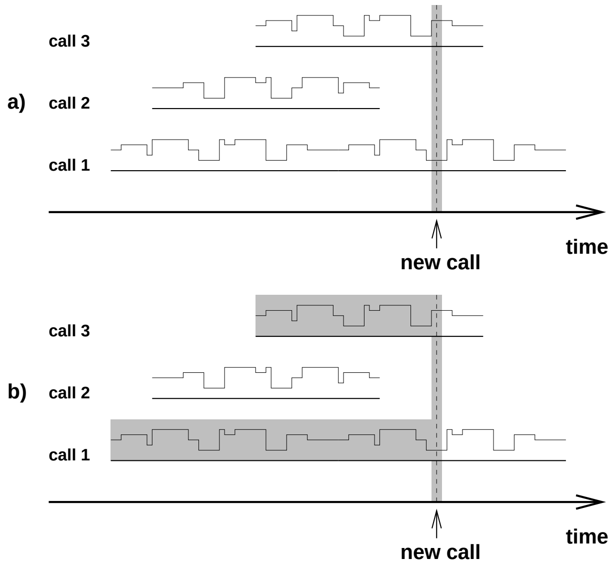

Figure 9: Two ways to estimate the call's marginal bandwidth distribution: a) memoryless; b) with memory, by collecting per-call history.

Fig. 7 and 8 show the renegotiation failure probability and the utilization for the memoryless scheme, respectively. The link capacity is expressed as a multiple of the call average rate. The normalized offered load is the offered load normalized by the link capacity. The utilization is normalized to the utilization that is achieved when call admission is performed based on the Chernoff approximation (12) and perfect knowledge of the call's marginal distribution.

It can be seen from Fig. 7 that the memoryless scheme performs very poorly for small link capacities. The renegotiation failure probability is three to four orders of magnitude larger than the target. From Fig. 8, it is clear that the memoryless scheme admits too many calls for small link capacity, as the utilization is much greater than the utilization under the scheme with perfect knowledge, which matches the target QoS precisely. We see that in this regime, the estimation error severely degrades the performance of the system. For larger systems, e.g. $n = 100$, the performance improves, meeting the target QoS for low offered loads. Also note that the renegotiation failure probability increases with the offered load. This is because a higher call arrival rate results in more "opportunities" to go wrong, i.e., to admit a call that should not have been admitted.

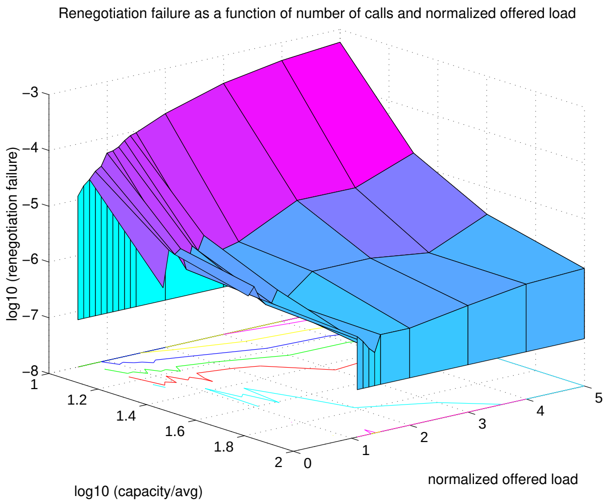

Figure 10: Scheme with memory: renegotiation failure probability.

We now discuss an approach for obtaining more robust schemes. We propose a scheme that relies on more memory about the system's past bandwidth reservations to come up with more accurate estimate of the marginal distribution. In this scheme, we keep track of how often each bandwidth level c_k has been reserved by any of the calls currently in the system. In other words, we accumulate information about the entire history of each call present in the system, and use this information to construct the empirical distribution $\{\pi_k\}$ of bandwidth requirements for a typical call, and make admission decisions based on the test (12) (cf. Fig. 9). This scheme is a considerable improvement over the memoryless scheme. Its renegotiation failure probability is about two orders of magnitude below that of the memoryless scheme over the whole range of link capacities and offered loads we have simulated (cf. Fig. 10). Figure 11 shows that like the memoryless scheme, this scheme is too optimistic in admitting calls for small link capacities. For larger link capacities, the utilization converges to the one obtained with perfect a-priori knowledge of the call statistics.

In summary, we see that the memoryless scheme is not robust over the range of parameters

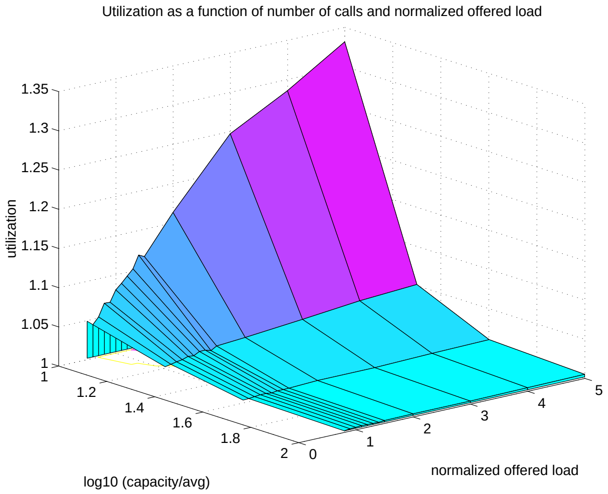

Figure 11: Scheme with memory: normalized utilization.

we have considered. The performance of call admission can be enhanced by using more history about the past behavior of calls. In practice, however, this gain will have to be traded off with the slower responsiveness to non-stationarities in the bandwidth requirement statistics. A better understanding of this tradeoff from both a theoretical and an experimental standpoint is needed. In particular, it is of interest to identify a memory size which can reap the bulk of the benefit of using more memory. Also, both our theoretical and experimental results focus on the homogeneous situation, where all calls have similar statistics. It is important to look at the heterogeneous case as well. These questions are outside the scope of this paper. We refer the reader to some of our recent work on measurement-based admission control [20, 42].

7 Related Work

The key contributions of our paper are: (a) to note that compressed video traffic has significant burstiness in the slow time-scale, (b) showing that renegotiation allows us to extract almost all the SMG available from exploiting this variation, and (c) admission control for loosely-constrained traffic sources. Recently Chong et al [4] and Zhang and Knightly [48] have independently published work that comes to the same conclusions. Zhang and Knightly present a renegotiated VBR service. Chong et al have concentrated on the online prediction problem using artificial neural networks. Our work differs from theirs in some important aspects. First, our work is based on the theoretical foundation of large deviation analysis of multiple time-scale sources, which gives us deeper insight into the nature of the multiplexing gain and allows us to formally study the renegotiation failure probability for ensembles of renegotiating sources, which is asymptotically correct in the regime stated in the theorems. In contrast, Chong et al [4] based their analysis solely on the power spectral density of the traffic and moreover does not consider the statistical multiplexing issues. As pointed out by Hajek and He [21], second-order statistics alone do not uniquely specify loss probabilities and thus it is important to understand the regime in which these approximations are valid. Second, we have obtained an optimal off-line renegotiation algorithm [18]. Third, we have considered admission control for renegotiating sources. Finally, we have considered the system aspects of the problem in more detail. Nevertheless, we feel that their work complements ours, in that it reinforces the importance of renegotiation for multiple time-scale sources.

The two core mechanisms for RCBR are renegotiation and rate prediction. In-call renegotiation has been proposed for bursty data traffic by Hui [23], Turner [43], Doshi and Dravida [9], and Boyer and Tranchier [3]. In their work, a traffic source sets up a burst level reservation before sending, or in some cases during a burst. However, since data traffic bursts can occur every tens of milliseconds the reservation process has to be fast. This speed is not essential for RCBR, where renegotiations happen once every tens of seconds. In addition, we believe that renegotiation is effective mainly as a mechanism to extract SMG from slow time-scale variations in source traffic. Data traffic exhibits burstiness in the fast time-scale, and thus renegotiation for data traffic is not likely to be economical in practice. Nevertheless, the mechanisms for renegotiation proposed in the literature can be used for RCBR with minor changes.

De Veciana and Walrand have proposed a *periodic averaging of rate* scheme to smooth traffic at the network edge [8]. Like RCBR, the output of their traffic shaper is also a piecewise CBR stream. The basic difference, however, is that they do not model the multiple time-scale nature of the traffic stream, and their scheme is not designed to capture the SMG from multiplexing many sources with slow time-scale dynamics.

The off-line schedule computation problem has also been addressed in Salehi et al's recent work [39]. They propose to use a client buffer and *work-ahead* smoothing, i.e., sending data ahead of schedule, in order to achieve an additional reduction in the flow's bandwidth fluctuation. They present an *optimal smoothing* algorithm that transforms an arbitrary data stream into a piecewise-CBR stream that minimizes both the peak rate and the rate variance, and show that this approach allows to considerably reduce the renegotiation frequency under RCBR service. Their work provides an interesting alternative for computing an optimal renegotiation schedule. Rexford et al [37] study the smoothing problem under the assumption that only limited knowledge about the future frame sizes is available. McManus and Ross discuss heuristics for the same problem setting [30]. They show the condition on the bandwidth b, the number of prefetched frames d, and the client-side memory size B, under which a sequence of VBR video can be transmitted at a constant rate without overflowing or underflowing the client buffer. Based on these conditions, heuristics are developed that yield a piecewise constant-rate transmission schedule. The client-side memory size and the number of prefetched frames are shown to decrease with the number of intervals, i.e., the number of renegotiations.

The on-line rate prediction problem has been extensively studied from several different perspectives in the past. Adas has addressed it using adaptive linear filtering [2], and he reports good prediction performance over a range of compressed video sequences. Other promising methods are described in [22]. Chong et al [4] have proposed an Artificial Neural Network based approach for prediction, and have shown that it compares well with more traditional alternatives. Reininger et al [36] have investigated methods to renegotiate Variable Bit Rate (VBR) parameters, including the peak rate, the sustained rate, and the burst size. The focus of their work is on the on-line prediction problem. A drawback of their scheme is the large number of parameters to be tuned (sliding window size $\mathrm{UPC}_{\mathrm{window}}$, aggressiveness factor k, four buffer thresholds D_1, D_2, I_1 and I_2, target quantization Q_{target}, and renegotiation delay parameters S and L).

Current proposals in the ATM Forum for dealing with ABR traffic are similar in spirit to RCBR in that a source obtains a stepwise-CBR rate allocation from the network. However, in the ABR framework, there is an assumption that the source has an intrinsically infinite data rate that is modulated by the fair share of the available network capacity. Thus, the data rate from a source is dynamically adapted to the available capacity in the network. This is the opposite of our situation, where the source has an intrinsic data rate that the network tries to accommodate. In other words, in the ABR case the rate information flows from the network to user; in the RCBR case, the information flows from the user to the network.

Our work on admission control is related to that described in Gibbens et al [16]. They advocate an approach based on using the current traffic load measurement in making admission control decisions. However, their focus is on how the measurement information can be combined with *a priori* knowledge of the traffic sources, while we investigate the improvement in performance through the use of more memory of the past network state. Moreover, our schemes are evaluated on real traffic sources, while theirs are on synthetic on-off sources. Recent work on measurement-based admission control has also been reported in Jamin et al [26]. However, their scheme has several parameters that have to be tuned in order to compensate for the sources of error discussed in the previous section (estimation error, dynamics); clear insight into how to set these parameters is lacking.

8 Discussion and Conclusions

We believe that the performance tradeoff space for traffic management policies looks something like Figure 12. Starting from the right and moving to the left, we have the Synchronous Digital Hierarchy (SDH) for telephony, Static CBR, Static VBR, Renegotiated CBR, Renegotiated VBR (RVBR),

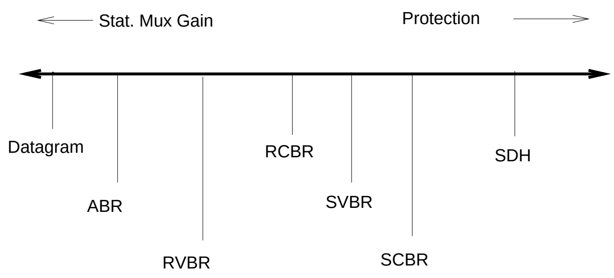

Figure 12: Design space for traffic management policies.

ABR, Controlled Load, and finally, unrestricted datagram service. In SDH, each call is associated with a time slot, and thus a corresponding bandwidth, that cannot be shared with any other call. Static CBR and static VBR are described in Section 2 and have one-shot traffic descriptors. RCBR and its corresponding service, RVBR, add renegotiation to CBR and VBR respectively. ABR service in ATM networks guarantees a connection zero loss, but its service rate changes as a function of other traffic in the network. With this service, the network agrees to perform admission control such that a source's performance does not substantially degrade, but the degree of degradation is not quantified. Finally, "datagram" refers to unrestricted sharing of all network resources.

As we move from right to left, the statistical multiplexing gain (SMG) achievable increases, but if the network resources allocated to a stream are kept the same, the protection between streams decreases. That is, one stream can more adversely affect another's performance in terms of its service rate and loss rate. For example, as one moves from Static CBR to Static VBR, more SMG is possible, but there is a greater loss of protection. This is because with a fixed amount of buffering a VBR source could experience packet loss due to a coincident burst from another source. Note that in moving from Static CBR to Static VBR similar protection can be obtained, but only at the cost of increased buffering, or by describing source traffic with more parameters [29]. Similarly, as we move from Static VBR to RCBR, we incur renegotiation overheads, but can potentially exploit slow time-scale variations in the source rate to get increased SMG. RVBR allows more SMG, since both slow and fast time-scale variations are exploited. However, there is more overhead for renegotiation, per-stream regulation, and larger buffers at each switch. The next step along the spectrum is to ABR, where there is much less protection between streams, because each user's bandwidth depends on the demand of the others. However, even more SMG is possible, since SMG is extracted at the burst level. Controlled Load service offers potentially even more SMG than ABR service, but at the expense of a non-zero loss rate. Finally, with datagram service, the most SMG is available, since call level, burst level, and cell level statistical multiplexing is possible. Unfortunately, datagram service also has the least protection - a single burst from a malicious or ill-behaved source can affect all the others.

The point is that RCBR is not a panacea. It is one choice in a spectrum of choices for carrying compressed video. We feel that RCBR is best suited to traffic whose variation is not confined to the fast time-scale. This seems to match at least the subset of the compressed video traffic workload that has been measured in the literature. Other services could also be used to carry compressed video traffic: ABR, Static VBR, RVBR, and Static CBR have all been proposed in the literature. Ultimately, a network provider and user must choose a service based on their relative costs, efficiencies and afforded qualities of service.

Nevertheless, we feel that RCBR has some clear benefits. First, it is relatively easy to implement, since we are adding a renegotiation component to the well-understood Static CBR service. While RCBR admission control is potentially complex, this is more than balanced by the fact that *neither complex scheduling disciplines nor large buffers are required in the network switches* [17]. RCBR allows us to keep the network core fast, cheap and dumb (at least in the data path), and put intelligence in the edges to extract the SMG from slow time-scale variations.

Second, an RCBR network is always stable, in the sense that the sum of arrival rates to a multiplexing point is always smaller than the corresponding service rate. Each admitted call or burst moves the system from a stable configuration to another stable configuration. Thus, the network operator can easily guarantee zero loss and small queueing delays within the network.

We have already shown that RCBR gets more SMG than a static service. There is another significant advantage. Users of a static service get only one chance to provide the network with a traffic descriptor. If they guess wrong, they either get poor SMG, or suffer from large delays, which might be unacceptable. With RCBR, a source has the option to modify its traffic descriptor over time. The danger is that the network might admit too many ill-described users, so that at some future time, the renegotiation failure rate may be too high. This is because there really is no free lunch. If a user is admitted into a network before its traffic is characterized, then there is always the possibility that mistakes will be made by admitting too many users. However, Section 6 indicates that we might be able exploit the law of large numbers to make this risk acceptably small.

It is instructive to compare RCBR with unrestricted sharing. With unrestricted sharing ("datagram service") we achieve the maximum SMG, but the least protection. In practical terms, with unrestricted sharing, a source must always be prepared to deal with data loss (for example, by using forward error correction or retransmitting data). Moreover, data loss is unpredictable. The analogue to loss in RCBR is renegotiation failure. With RCBR, however, a source retains its existing bandwidth even if renegotiation fails. Besides, a source is explicitly informed about renegotiation failure so that it can take corrective measures. This makes it easier to integrate RCBR with techniques such as dynamic requantization of stored video, adaptive coding and multilevel scalable coding.

To conclude, we have shown that a source with slow time-scale variations would suffer performance problems when carried over a static service. Large deviation analysis provides theoretical insight into this problem and motivates the design of the RCBR service. We have considered the system aspects of implementing RCBR and have carried out several experiments to measure its performance. The results in Section 5.2 show that RCBR obtains most of the slow time-scale SMG with a fairly small load on the signaling system. Further, it is possible to compute the optimal renegotiation schedule for a real traffic source in a reasonable amount of time. Finally, we have studied the call admission problem and come up with admission control tests based on a large deviation analysis. Thus, our analysis and experiments show that RCBR service is efficient and well suited for multiple time-scale traffic.

9 Acknowledgments

We are indebted to Mark Garrett for providing the traces of MPEG compressed Star Wars. We wish to thank the anonymous reviewers for very valuable and extensive feedback that has considerably improved the presentation of this paper.

References

[1] ATM Forum Traffic Management Specification Version 4.0. ATM Forum Specification /af-tm-0056.000, ATM Forum, April 1996.

[2] A. Adas. Supporting Real Time VBR Video Using Dynamic Reservation Based on Linear Prediction. In *Proc. IEEE INFOCOM '96*, San Fransisco, CA, March 1996.

[3] P.E. Boyer and D.P. Tranchier. A reservation principle with applications to the ATM traffic. *Computer Networks and ISDN Systems*, 24:321–334, 1992.

[4] S. Chong, S.Q. Li, and J. Ghosh. Predictive Dynamic Bandwidth Allocation for Efficient Transport of Real-Time VBR Video over ATM. *IEEE Journal on Selected Areas of Communications*, 13:12–23, January 1995.

[5] D. Clark, S. Shenker, and L. Zhang. Supporting real-time applications in an integrated services packet network: Architecture and mechanism. In *Proc. ACM SIGCOMM '92*, pages 14–26, 1992.

[6] Rene L. Cruz. A Calculus for Network Delay, Part I: Network Elements in Isolation. *IEEE Transactions on Information Theory*, 37(1):114–131, January 1991.

[7] Rene L. Cruz. A Calculus for Network Delay, Part II: Network Analysis. *IEEE Transactions on Information Theory*, 37(1):132–141, January 1991.

[8] G. de Veciana and J. Walrand. Traffic Shaping for ATM Networks: Asymptotic Analysis and Simulations. *submitted to IEEE/ACM Transactions on Networking*, 1992.

[9] B. Doshi and S. Dravida. Congestion Controls for Bursty Data Traffic in Wide Area High Speed Networks: In-Call Negotiations. *Proc. ITC Specialist Seminar 7, Morristown, NJ*, 1990.

[10] A. Eleftheriadis and D. Anastassiou. Meeting Arbitrary QoS Constraints Using Dynamic Rate Shaping of Coded Digital Video. *Proc. 5th Workshop on Networking and Operating System Support for Digital Audio and Video*, pages 95–106, April 1995.

[11] A. Elwalid and D. Mitra. Effective Bandwidth of General Markovian Traffic Sources and Admission Control of High-Speed Networks. *IEEE/ACM Transactions on Networking*, 1:329–343, June 1993.

[12] M. W. Garrett. *Contributions Toward Real-Time Services on Packet Switched Networks.* PhD thesis, Columbia University, 1993. Chapter IV.

[13] M. W. Garrett and Walter Willinger. Analysis, Modeling and Generation of Self-Similar VBR Video Traffic. In *Proc. ACM SIGCOMM '94*, pages 269–280, London, UK, August 1994.

[14] R.J. Gibbens and P.J. Hunt. Effective Bandwidths for the Multi-type UAS Channel. *Queueing Systems*, 9:17–27, 1991.

[15] R.J. Gibbens, F.P. Kelly, and P.B. Key. A Decision-theoretic Approach to Call Admission Control in ATM Networks. *IEEE Journal on Selected Areas of Communications*, pages 1101–1114, August 1995.

[16] R.J. Gibbens, F.P. Kelly, and P.B. Key. A Decision-theoretic Approach to Call Admission Control in ATM Networks. *IEEE JSAC, Special issue on Advances in the Fundamentals of Networking*, August 1995.

[17] M. Grossglauser and S. Keshav. On CBR Service. In *Proc. IEEE INFOCOM '96*, San Francisco, California, March 1996.

[18] M. Grossglauser, S. Keshav, and D. Tse. RCBR: A Simple and Efficient Service for Multiple Time-Scale Traffic. In *Proc. ACM SIGCOMM '95*, pages 219–230, Boston, Mass., August 1995.

[19] M. Grossglauser, S. Keshav, and D. Tse. The Case Against Variable Bit Rate Service. In *Proc. 5th Intl. Workshop on Network and Operating System Support (NOSSDAV '95)*, pages 307–310, Durham, New Hampshire, April 1995.

[20] M. Grossglauser and D. Tse. A Framework for Robust Measurement-Based Admission Control. In *Proc. ACM SIGCOMM '97*, Cannes, France, September 1997.

[21] B. Hajek and L. He. On variations of queue responses for inputs with identical mean and autocorrelation fucntions. In *Conference on Information Sciences and Systems*, Princeton University, 1996.

[22] I. Hsu and J. Walrand. Quick Detection of Changes in Traffic Statistics: Application to Variable Rate Compression. In *Proceedings of the 32nd Allerton Conference on Communications, Control and Computing, Monticello, IL*, 1993.

[23] J.Y. Hui. Resource Allocation for Broadband Networks. *IEEE Journal on Selected Areas in Communications*, 6(9), December 1988.

[24] ITU. *Q.2931 Digital Subscriber Signalling System No. 2 (DSS 2) User-Network Interface (UNI) layer 3 specification for basic call/connection control*, February 1995. available from http://www.itu.ch.

[25] ITU. *Q.2971 Broadband integrated services digital network (B-ISDN) Digital subscriber signalling system No.2 (DSS 2) User-network interface layer 3 specification for point-to-multipoint call/connection control*, October 1995. available from http://www.itu.ch.

[26] S. Jamin, P. B. Danzig, S. Shenker, and L. Zhang. A Measurement-Based Admission Control Algorithm for Integrated Services Packet Networks. In *Proc. ACM SIGCOMM '95*, 1995.

[27] H. Kanakia, P.P. Mishra, and A. Reibman. An Adaptive Congestion Control Scheme for Real-Time Packet Video Transport. *Proc. ACM SigComm*, 1993.

[28] G. Kesidis, J. Walrand, and C.S. Chang. Effective Bandwidths for Multiclass Markov Fluids and Other ATM Sources. *IEEE/ACM Transactions on Networking*, 1(4):424–428, August 1993.

[29] E. Knightly and H. Zhang. Traffic Characterization and Switch Utilization using a Deterministic Bounding Interval Dependent Traffic Model. In *Proc. IEEE INFOCOM '95*, Boston, Mass., April 1995.

[30] J. M. McManus and K. W. Ross. Video-on-Demand Over ATM: Constant-Rate Transmission and Transport. *IEEE Journal on Selected Areas of Communications*, pages 1087–1098, August 1996.

[31] M. Nomura, T. Fujii, and N. Ohta. Basic Characteristics of Variable Rate Video Coding in ATM Environment. *IEEE Journal on Selected Areas of Communications*, 7(5), June 1989.

[32] A. K. Parekh and R. G. Gallager. A Generalized Processor Sharing Approach to flow control in Integrated Services Networks - The Single Node Case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, June 1993.

[33] A. K. Parekh and R. G. Gallager. A Generalized Processor Sharing Approach to flow control in Integrated Services Networks - The Multiple Node Case. *IEEE/ACM Transactions on Networking*, 2(1):137–150, April 1994.

[34] E. P. Rathgeb. Modeling and Performance Comparison of Policing Mechanisms for ATM Network. *IEEE Journal on Selected Areas in Communications*, 9(3):325–334, April 1991.

[35] E. P. Rathgeb. Policing of Realistic VBR Video Traffic in an ATM Network. *International Journal of Digital and Analog Communications Systems*, 6:213–226, 1993.

[36] D. J. Reininger, D. Raychaudhuri, and J. Y. Hui. Bandwidth Renegotiation for VBR Video Over ATM Networks. *IEEE Journal on Selected Areas of Communications*, 14(6), August 1996.

[37] Jennifer Rexford, Subhabrata Sen, Jayanta Dey, Wu chi Feng, James Kurose, John Stankovic, and Don Towsley. Online Smoothing of Live, Variable-Bit-Rate Video. In *International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV '97)*, pages 249–257, May 1997.

[38] R. Safranek, C. Kalmanek, and R. Garg. Methods for Matching Compressed Video to ATM Networks. *Proc. of IEEE IT Workshop on Information Theory, Multiple Access and Queueing Theory, St. Louis*, page 6, April 1995.

[39] J. Salehi, Z. Zhang, J. Kurose, and D. Towsley. Supporting Stored Video: Reducing Rate Variability and End-to-End Resource Requirements through Optimal Smoothing. *Proc. ACM SIGMETRICS, Philadelphia, PA*, May 1996.

[40] P. Sen, B. Maglaris, N. Rikli, and D. Anastassiou. Models for Packet Switching of Variable-Bit-Rate Video Sources. *IEEE Journal on Selected Areas of Communications*, 7(5), June 1989.

[41] D. Tse, R. Gallager, and J. Tsitsiklis. Statistical Multiplexing of Multiple Time-Scale Markov Streams. *IEEE JSAC, Special issue on Advances in the Fundamentals of Networking*, August 1995.

[42] D. Tse and M. Grossglauser. Measurement-Based Call Admission Control: Analysis and Simulation. In *Proc. IEEE INFOCOM '97*, Kobe, Japan, April 1997.

[43] J.S. Turner. Managing Bandwidth in ATM Networks with Bursty Traffic. *IEEE Network Magazine*, September 1992.

[44] W. Verbiest and L. Pinnoo. A variable bit rate codec for ATM networks. *IEEE Journal on Selected Areas of Communications*, 7(5):761–770, June 1989.

[45] A.J. Viterbi and J.K. Omura. *Principles of Digital Communication and Coding*. McGraw-Hill, 1979.

[46] A. Weiss. A New Technique for Analyzing Large Traffic Systems. *Advances in Applied Probability*, 18:506–532, 1986.

[47] L.C. Wolf, L. Delgrossi, R. Steinmetz, S. Schaller, and H. Wuttig. Issues in Reserving Resources in Advance. *Proc. 5th Workshop on Networking and Operating System Support for Digital Audio and Video*, pages 27–37, April 1995.

[48] H. Zhang and E.W. Knightly. A New Approach to Support Delay-Sensitive VBR Video in Packet-Switched Networks. *Proc. 5th Workshop on Networking and Operating System Support for Digit al Audio and Video*, pages 275–286, April 1995.

[49] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. RSVP: A New Resource ReSerVation Protocol. *IEEE Network*, September 1993.