9. D. Tse, R. Gallager, and J. Tsitsiklis. Statistical multiplexing of multiple time-scale markov streams. *Preprint.*
10. J.S. Turner. Managing Bandwidth in ATM Networks with Bursty Traffic. *IEEE Network Magazine*, September 1992.

This article was processed using the LaTeX macro package with LLNCS style

guarantee zero loss and small delays. In contrast, a traditional VBR service requires the network core to have large buffers and make complex admission control decisions.

In future work, we plan to consider more sophisticated buffer monitoring algorithms, as well as a theoretical analysis of the call admission problem to quantify the call versus renegotiation failure tradeoff. Further, our solution does not invalidate the use of VBR at the fast time-scales. In fact, some additional gain might be extracted by admitting small bursts at the fast time-scale.

## 8 Related Work

It has recently come to our notice that Chong et al [?] have independently arrived at conclusions similar to others, that is, compressed VBR traffic has sustained peaks, and thus is best carried by a renegotiated service. They have concentrated in their work on coming up with good predictors for rate changes. In contrast, our work has used large deviation analysis for insight into the problem coupled with a systems perspective. In this sense, we feel that their work complements ours, even though they were the first to recognize the problems caused by sustained peaks.

The two core mechanisms in RCBR are fast renegotiation (or, equivalently, fast reservation) and rate prediction. A fast reservation mechanism has been independently proposed for bursty data traffic by Hui [5], Turner [10], and Boyer and Tranchier [1]. De Veciana and Walrand have proposed a *periodic averaging of rate* scheme to smooth traffic at the network edge [2].

## References

1. 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.
2. G. de Veciana and J. Walrand. Traffic shaping for ATM networks: Asymptotic analysis and simulations. *Preprint*, 1992.
3. Mark W. Garrett. *Statistical Analysis of a Long Trace of Variable Bit Rate Video Traffic*. PhD thesis, Columbia University, 1993. Chapter IV.
4. Mark W. Garrett and Walter Willinger. Analysis, Modeling and Generation of Self-Similar VBR Video Traffic. In *ACM Sigcomm '94*, pages 269–280, University College London, London, UK, August 1994.
5. J.Y. Hui. Resource Allocation for Broadband Networks. *IEEE Journal on Selected Areas in Communications*, 6(9), December 1988.
6. H. Kanakia, P.P. Mishra, and A. Reibman. An Adaptive Congestion Control Scheme for Real-Time Packet Video Transport. *Proc. ACM SigComm*, 1993.
7. Erwin 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.
8. Erwin P. Rathgeb. Policing of Realistic VBR Video Traffic in an ATM Network. *International Journal of Digital and Analog Communications Systems*, 6:213–226, 1993.

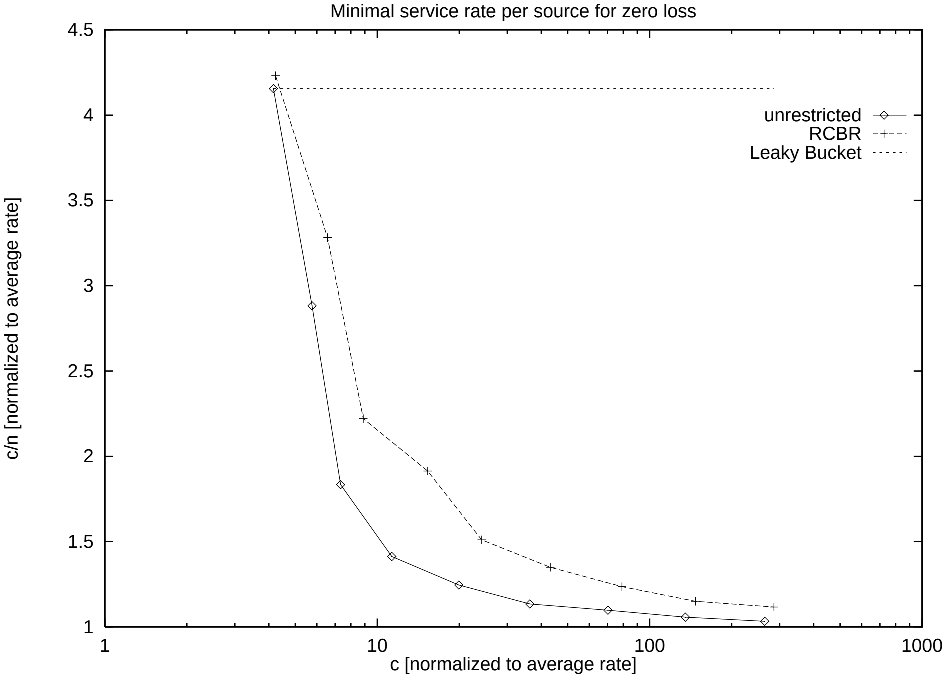

**Fig. 1.** Statistical Multiplexing Gain (SMG) achievable.

decreases exponentially with the number of multiplexed streams. If the cost of renegotiation is zero, then perfect SMG can be achieved. Perfect SMG can also be obtained by a leaky bucket with large buffers, but only at the cost of large buffers in the switches, and consequent loss of inter-stream protection. If there are no renegotiations, then the behavior is identical to a leaky bucket with zero buffer.

One reason an RCBR-like scheme has not been proposed in the past is that long traces of video traffic have been studied only recently. When examining short clips of video of about 10 to 20 seconds in length, the multiple time-scale behavior is less apparent. Thus, we are indebted to Rathgeb, and Garrett and Willinger for their work [7, 8, 3, 4]. RCBR develops naturally by coupling their observations to the theoretical analysis of Tse, Gallager and Tsitisklis [9].

We claim that RCBR solves the two major problems raised in Section 2 by a) doing away with one-shot traffic descriptors for VBR sources and b) dealing explicitly with sustained peaks in the source rate. Two additional advantages of RCBR are that it is easy to implement, since CBR service is well understood and we are merely adding a fast renegotiation component to it. By building on CBR service, we make the network design simple, since neither complex scheduling disciplines nor large buffers are required for CBR traffic. Second, the system is always stable. Each admitted call or burst moves the system from a stable configuration to another stable configuration. Thus, the network core can easily operator makes this tradeoff by picking a slackness threshold: if the current usage exceeds this threshold, new calls would be rejected. Thus, we believe that there are a number of techniques we can call upon to deal with renegotiation failures.

## 5 Analysis

Using a multiple time-scale Markov model for the traffic stream, we can characterize the multiplexing gain for our proposed scheme. We can also compute a Chernoff estimate for renegotiation failure probability for i.i.d sources. Finally, we can compute renegotiation schedules that can take advantage of the time-scale separation without explicit knowledge of which sub-chain the source is in at any given time. Considerations of space preclude us from describing this work here. The interested reader is referred to [?].

## 6 Experimental results

We have compared the statistical multiplexing gain (SMG) achievable through our scheme with two other scenarios. The first scenario (a) multiplexes $n$ streams without any restriction on a server with rate $c$ and buffer size $nB$. This is used to estimate the maximum achievable SMG for the given sources. The second scenario (b) has a leaky bucket regulator policing the stream entering the network. The average rate $a$ policed by the leaky bucket is chosen so that the maximum backlog is $B$. This can be determined from the corresponding $(\sigma, \rho)$-curve of this trace. The third scenario (c) represents RCBR.

The streams we have used are $n$ random shifted versions of an MPEG-1 encoded trace of the Star Wars movie [3]. The buffer size $B$ was chosen as 300 kbit, slightly more than the maximum size of three consecutive frames in the trace.

To assess the SMG for all three scenarios, we have determined the bandwidth per stream $c/n$ as a function of $c$, depicted in Fig. 6. We observe that case (a) represents a lower bound on the bandwidth per source for zero loss, and therefore an upper bound on the SMG. This is because the buffer is shared among all $n$ streams. In the leaky bucket regulated case (b), the bandwidth per stream is $a$, regardless of the service rate $c$. Our scheme achieves slighly less SMG than the unrestricted case, because buffers are not shared, and because there is a small difference between the source average bandwidth and the allocated average bandwidth. Nevertheless, we are able to extract most of the SMG, especially for high service rates. For example, at a trunk rate of 100 times the average rate, we require about three times less bandwidth than the leaky bucket approach.

## 7 Discussion

RCBR explicitly builds upon the presence of sustained bursts to achieve nearly optimal SMG. Large deviation analysis shows that the renegotiation failure rate sub-chain model fast time-scale behavior (such as correlations between adjacent frames) while the transitions between the sub-chains model slow time-scale behavior (such as scene changes). 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.

There are several key results of this work. First, when one computes the *equivalent bandwidth* of an individual multiple time-scale stream, It is found that one has to allocate the maximum of the *equivalent bandwidths* of the fast-sub-chains. This essentially means that one has to allocate a rate near the sustained peak, and it underscores the fact that the statistical multiplexing gain due to smoothing using buffers is of limited use for traffics such as compressed video, because the slow time-scale is significantly longer than the delay requirement. On the other hand, when a large number of independent multiple time-scale streams are multiplexed together, much more gain can be obtained beyond looking at the equivalent bandwidth of each stream in isolation. This gain is due to the fact that with high probability, not too many sources can be in a high-rate sub-chain at any one time. Thus, the bulk of the gain is obtained through averaging between sources with respect to the slow time-scale dynamics rather than through smoothing by the buffer.

## 4 A New Scheme for VBR: RCBR

RCBR augments standard CBR service with a fast renegotiation mechanism. A signaling mechanism for this purpose is described in [1]. For stored data stream applications, the series of CBR rates is known in advance, and so renegotiation to increase the service rate can be carried out before actually increasing the data rate. For interactive applications, the renegotiation schedule cannot be calculated in advance. Instead, we propose that an active component monitor the user-network buffer and initiate renegotiations based on the buffer occupancy level. While we cannot present details for lack of space, analysis shows that with careful coding and cleanly designed end-systems and switch controllers, renegotiation overheads are small.

Due to the inherent uncertainty in the bandwidth requirements of calls over their lifetime, there is always a possibility that a renegotiation for a higher bandwidth can fail due to unavailable capacity. This may not be acceptable for some users. Such users 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 clear tradeoff between buffer size, 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. Alternately, on a failure, the signaling system could ask the user or application to reduce its data rate. Responding to such signals should be straightforward, particularly for adaptive codecs [6]. Finally, during admission control, a switch controller might reject a call even if there is available capacity. This allows the network operator to trade off call blocking probability and renegotiation failure probability. The network traffic in terms of multiple time-scales, rather than as CBR with bursts, as in earlier work (Section 7).

## 2 Problems with Current VBR Service

There are two major problems with current service definitions. First, many researchers assume that users of VBR service can describe their traffic using a small number of descriptors, such as the token bucket size and token drain rate for the leaky bucket scheme. We have found that it is hard to determine these parameters even for a stored data stream, since there isn't a unique mapping from a data stream to a descriptor. The situation becomes even worse for interactive applications, where the data stream is not known in advance.

The single largest application using VBR service is expected to be compressed video. The second problem with current VBR service is that typical compressed video traffic does not match the moderately bursty traffic model envisaged by designers of VBR service. It has been observed by several researchers [7, 8, 3, 4] that, independent of the coding algorithm, there are fairly long durations, when the data rate of the video source is continuously near the peak rate. For such traffic, if a leaky-bucket like descriptor is used, one is faced with a series of poor choices. Assume for the moment that the token rate is chosen close to the average rate in order to maximize the statistical multiplexing gain. Now, if the token bucket is chosen to be small, then during sustained peaks, either the data buffer at the regulator has to be very large, or there will be many losses. If the loss rate is to be small, and data buffers are made large, this leads to expensive regulators and long delays for the sources.

On the other hand, if the token bucket is made large enough to rapidly drain bursts, then the network and receiver will need large data buffers to prevent cell loss during a burst. Further, even a compliant source has considerable freedom to disrupt the network core by sending in data in very large bursts (on the order of tens of Megabytes). We call this loss of *protection*, since other endpoints would be unprotected from bursts from a compliant but ill-conditioned source.

Thus, the phenomenon of sustained peaks leads either to a) loss of smoothing gain or b) large loss rate or c) large delays and expensive regulators or d) loss of protection. Given the current framework, there is no way to avoid all four problems simultaneously. This is a simple consequence of the fact that the sustained peaks in workload violate the design assumptions for VBR service.

## 3 Statistical Multiplexing of Multiple Time-Scale Sources

Recent work provides the theoretical basis for understanding the gain achievable by multiplexing traffic sources exhibiting sustained peaks [9]. In this work, each variable-rate stream is modeled as a process modulated by a multiple time-scale Markov chain: a chain which consists of several sub-chains between which the transitions have very small transition probability. The dynamics within each

# The Case Against Variable Bit Rate Service

M. Grossglauser, S. Keshav and D. Tse

AT&T Bell Laboratories, Murray Hill, NJ 07974, USA

**Abstract.** There are two major problems with current designs for VBR service a) it is inherently hard to compactly and completely describe VBR traffic sources and b) sustained peaks in the source rate substantially degrade the performance. We propose a new scheme, Renegotiated CBR (RCBR) that augments CBR service with fast renegotiation and buffer monitoring. Analysis shows that the scheme is easily implementable and solves the problems above. In more general terms, we believe that a clean system design must rely on traffic management policies that clearly separate call level, burst level, and cell level time-scales. RCBR works well because it makes intelligent use of this time-scale separation.

## 1 Introduction

Current designs for integrated services networks typically provide three types of service: Constant Bitrate (CBR), Variable Bitrate (VBR) and Available Bitrate (ABR) VBR service has been designed in *anticipation* of future traffic, particularly compressed video traffic. Such traffic is modeled as having an intrinsic long-term average rate, but with periods in which data is generated in a burst at some peak rate. If this traffic were to be carried by CBR service, either each burst would need to be smoothed out at the network entrance, leading to possibly intolerable delays, or the CBR rate has to be close to the peak rate, limiting the statistical multiplexing gain. If this traffic were to be carried by ABR service, there are no guarantees on delays and losses. Thus, current designs augment a standard CBR service with the ability to admit limited bursts into the network. The hope is that by admitting bursts users would get smaller delays. At the same time, if the burst size is limited by an intelligent admission control scheme and policing, limited buffering in the network would be sufficient to give strong guarantees on delays and losses. Note that it is important to be able to characterize both the worst case burst size and the long term average rate for the scheme to work well.

We argue that this design for VBR service suffers from two major flaws (Section 2). In fact, a different modeling of VBR traffic leads to some insight into a better service definition (Section 3) This motivates Renegotiated CBR (RCBR) service, where the network core is CBR, but rapid renegotiation of CBR rates allows the network operator to extract almost all of the statistical multiplexing gain inherent in the traffic (Section 4). Analysis and experimentes indicate that RCBR is stable, efficient and has low overhead (Sections 5 and 6). We believe that our approach is successful because it correctly models compressed video