To Appear in ACM/Springer-Verlag Multimedia Systems Journal

RED-VBR: A Renegotiation-Based Approach to Support Delay-Sensitive VBR Video

Hui Zhang
School of Computer Science
Carnegie Mellon University
hzhang@cs.cmu.edu

Edward W. Knightly
EECS Department
University of California at Berkeley
knightly@eecs.berkeley.edu

Abstract

Previous approaches to supporting video on packet-switched networks include deterministic service, statistical service, predicted service, and feedback-based schemes. These schemes represent different tradeoffs in quality of service (QOS), achievable network utilization, and method of dealing with overload. In this paper, we propose a new service called REnegotiated Deterministic Variable Bit Rate Service (RED-VBR) that attempts to strike an efficient balance with the above tradeoffs. The approach is based on deterministic guarantees with client controlled renegotiation of traffic and QOS parameters and graceful adaptation during overload periods. We introduce a connection admission control algorithm for RED-VBR which bounds the renegotiation failure probability. We evaluate the scheme using two traces of MPEG-compressed video and show that, even with simple renegotiation polices and relatively low renegotiation frequencies, high network utilization in the range of 50% to 80% can be achieved. For traffic that is bursty over long intervals, this represents a 100% to 150% improvement in network utilization compared to deterministic service. Compared to statistical and predicted service, our approach allows more graceful and client-controlled QOS degradation during overload period.

1 Introduction and Motivation

Future integrated services networks will have to support applications with diverse traffic characteristics and performance requirements. There are three important types of traffic for future integrated services networks: constant bit rate or CBR traffic, delay-sensitive variable bit rate or VBR traffic, and best-effort traffic. Among these, delay-sensitive VBR traffic poses a unique challenge. While resource reservation schemes work best for CBR traffic, and there are many congestion control algorithms based on feedback and re-transmission for best-effort traffic, there is no consensus on which strategy should be used for VBR traffic, in particular, compressed video. This is due mainly to two conflicting design goals: good quality of service and high network utilization.

Achieving both goals with bursty traffic is fundamentally difficult. Since a bursty source may generate various amounts of data during different time periods, the aggregate amount of traffic generated by many sources sharing the same network resources also varies over time. When the amount of aggregate incoming traffic is greater than the outgoing link speed, packets have to be buffered. If the situation persists, packets will be dropped due to buffer overflows, which will in turn cause the application's quality of service (QOS) to suffer. This problem is compounded by the nature of VBR video traffic: depending on the underlying information content of the video stream, bursts of high rate can persist for time scales on the order of many seconds over the duration of an entire complex, high-motion scene. Bursts of this time scale cannot be absorbed by network buffers or smoothed at the source because of the excessive delay that this would introduce and the excessive buffer sizes that it would require.

Thus, the fundamental problem is that when bursts from many sources collide inside the network, if the rate of the aggregated traffic is greater than the link speed and the situation persists for a certain period, the QOS of some or all connections will suffer. Various solutions have been proposed to address the problem, and they represent different ways of dealing with the tradeoff between QOS and network utilization. Previous solutions can be classified according to the following four categories: deterministic service with worst-case resource allocation [4, 10, 21, 22], statistical service with probabilistic allocation [8, 9, 16, 19, 23, 33], predicted service with observation-based admission control [6, 17], and feedback based scheme with no resource reservation [13, 18].

Previously proposed solutions for deterministic service *eliminate* the occurrence of overload situations by reserving resources according to a worst-case scenario. For example, in [22] a source is characterized by bounding rates over multiple interval lengths via the Deterministic Bounding Interval-Dependent (D-BIND) traffic model. While such an approach provides the best QOS, it does so at the expense of having low network utilization when traffic sources are very bursty, e.g., if their bounding rates decrease slowly with increasing interval length. In various ways, the other three approaches trade a higher network utilization for a potential degradation of QOS. However, they all suffer from some limitations. Statistical and predicted services try to *control* the frequency of the overload situation by exploiting statistical multiplexing (respectively using knowledge of source statistics and queue measurements). However, the overload situation may still happen, and at unexpected times. Additionally, during the overload period, QOS is likely to suffer significantly for all connections in an uncontrolled and difficult-to-predict way. As well, the QOS may drop significantly due to *consecutive* packet losses. This last problem is exacerbated for VBR *video* since VBR video may have very long burst lengths, on the order of scene lengths, possibly causing a persistent degradation in service when the bursts do collide. Feedback schemes with no reservations try to *adapt* and *react* to overload situations by using network congestion signals to reduce the rates of sources. Such schemes have the advantage that they can *gracefully* degrade QOS during an overload situation by exploring an important property of the compressed video: most of video compression algorithms have a quality control parameter that, when tuned, will output compressed video at different rates and qualities. The drawback of a feedback-based scheme is that, without some round robin type of scheduler at the switch, it won't work unless *all* sources cooperate. Even with switch support, it still has the fundamental problem that it is impossible to provide different types of QOS to different applications.

In this paper, we study a new approach to support VBR video called REnegotiated Deterministic Variable Bit Rate or RED-VBR service. We utilize two important properties of compressed video. First, compressed video traffic usually exhibits burstiness over multiple time-scales [12, 24, 27]. At least two levels of burstiness are important for a resource allocation algorithm: burstiness on a shorter time scale due to the coding algorithm and small-time-scale variations in picture information content, and burstiness on a longer time scale due to scene changes. Correspondingly, resource allocation algorithms should have mechanisms at different time-scales to achieve a high statistical multiplexing gain. Second, most of the video compression algorithms have some type of quality control factor (Q-factor) [11, 29]. By tuning this factor, a video source can tradeoff its bit rate for perceptual quality.

With RED-VBR, a source specifies its traffic using the recently proposed D-BIND traffic model [22], which captures the property that VBR video has different bounding rates over different interval lengths, and addresses short-term burstiness of the traffic streams. To address long-term burstiness, the application renegotiates its traffic parameters and QOS with the network when there is a *significant* change of long term traffic rate. Such a renegotiation scheme is possible for two reasons: (1) since renegotiations need to happen only when the traffic rate changes over long term, such renegotiation is not very frequent; (2) even if the renegotiation request for more resources cannot be satisfied, the application can *adjust* the Q-factor of its compression algorithm, and gracefully degrade its QOS based on the currently available resources.

While traditional reservation-less approaches do statistical multiplexing at the packet level (packets may be dropped), and traditional reservation-based service can be viewed as doing statistical multiplexing at the connection level (connection requests may be denied), our approach can be seen as doing statistical multiplexing at the segment level — resources are reserved on a per segment basis and reservation requests for a *segment* may be denied when the network is overloaded. An important feature of such an approach is that each individual application *determines for itself* the tradeoff between QOS and price-of-service by defining its own segmenting algorithm. The approach is statistical in that a renegotiation request for more network resources can be denied. However, compared to statistical service and predicted service, we avoid the uncontrollable and unpredictable packet drop behavior and the extended drop periods by introducing graceful degradation during overload situations.

Since the RED-VBR renegotiation scheme should be viewed as providing a statistical performance guarantee at the segment level rather than at the packet or connection level, we provide a new admission control scheme that can be used to determine the *segment-level* blocking probability. That is, given an arbitrary set of heterogeneous connections, we determine the probability that a connection's attempt to renegotiate for additional network resources will be denied or blocked by the network. A new connection is admitted only if all existing connections and the new connection will receive their required delay bounds and blocking probabilities.

The recent work on RCBR [14, 15], published in parallel with RED-VBR [34], is closely related to our RED-VBR service. While RCBR also proposes a renegotiated service to support VBR video, a difference is that RED-VBR builds the renegotiation service on top of a deterministic variable bit rate (D-VBR) service with the D-BIND traffic model, while RCBR build the renegotiation service on top of a constant bit rate (CBR) service. Since it explicitly models traffic burstiness, a D-VBR is a more efficient base service, i.e., it can achieve a higher network utilization than a CBR service for the same level of QOS. Consequently, a RED-VBR will require fewer renegotiations than RCBR for the same level of resource utilization. On the other hand, a CBR service is easier to implement than a D-VBR service. In another related work, the idea of using renegotiated service to support VBR video was independently proposed by Chong et. al. [5]. The focus of Chong's work is on renegotiation policies for live sources.

The remainder of this paper is organized as follows. In Section 2, we review a deterministic VBR service, which is the foundation for RED-VBR. Next, in Section 3, we outline the new RED-VBR service and discuss the implications of the introduction of the new control time-scale. We describe an admission control algorithm for RED-VBR in Section 4, and present renegotiation algorithms in Section 5. Finally, in Section 6, we empirically evaluate the RED-VBR scheme using traces of MPEG-compressed video.

2 Deterministic Variable Bit Rate Service

While the conventional wisdom has been that a deterministic service requires peak rate allocation and thus achieves the same network utilization as a CBR service, we will show in this section that this is not necessarily the case. A deterministic service will ensure that no packets are dropped or delayed beyond their reserved delay-bound, even in the *worst case*. However, for sources such as MPEG-compressed video, the largest local rate-variation is due to the alternation of inter-frame coded frames with intra-frame coded frames. That is, a larger intra-frame coded I-frame is immediately followed by a smaller inter-frame coded B-frame so that the micro-level burst does not persist for very long, even in the worst case. In order to characterize the property that VBR video has different bounding rates over different interval lengths, we use the Deterministic Bounding Interval-Dependent (D-BIND) traffic model which utilizes such properties in the connection admission control algorithm. With the D-BIND model, a Deterministic Variable Bit Rate or D-VBR service can achieve a considerably higher network utilization compared to a peak-rate-allocation scheme.

2.1 The Deterministic-BIND Model

As shown in [22], previous deterministic traffic models such as the (σ, ρ) model [7] and the $(Xmin,\ Xave,\ I,\ Smax)$ model [10] cannot capture the property that sources exhibit burstiness over a wide variety of interval lengths. The Deterministic Bounding Interval Dependent traffic model was introduced to address this issue. The key components of the D-BIND model are that it is *bounding*, required to provide deterministic QOS guarantees, and *interval-dependent*, needed to capture important burstiness properties of sources which in turn allows for a higher network utilization.

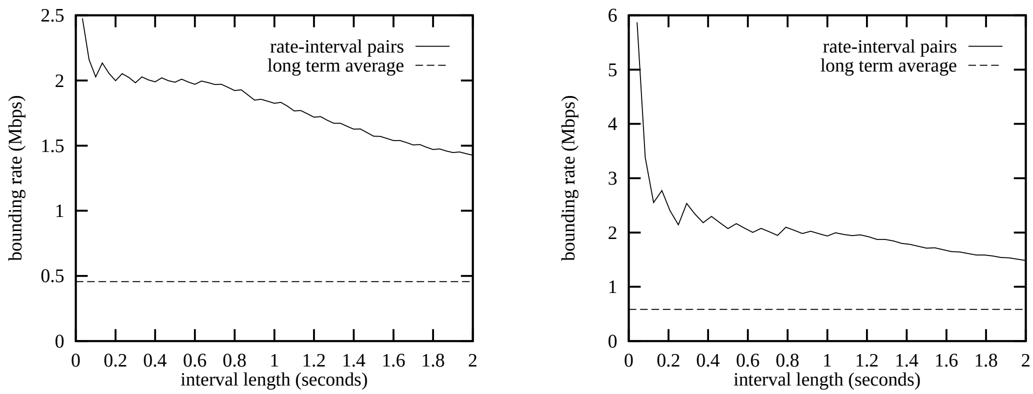
Each deterministic traffic model uses parameters to define a traffic constraint function $b(t)$ which constrains or bounds the source over every interval of length t. Denoting $A[t_1, t_2]$ a connection's arrivals in the interval $[t_1, t_2]$, the traffic constraint function $b(t)$ requires that $A[s, s+t] \leq b(t),\ \forall s, t > 0$. Note that $b(t)$ is a time-invariant deterministic bound since it constrains the traffic source over every interval of length t. For example, the (σ, ρ) model is defined such that $A[s, s+t] \leq \sigma + \rho t$ for all t.

The D-BIND model is defined via P rate-interval pairs $\{(R_k, I_k) | k = 1, 2, \cdots, P\}$ so that the constraint function is given by a piece-wise linear function

$$b(t) = \frac{R_k I_k - R_{k-1} I_{k-1}}{I_k - I_{k-1}}(t - I_k) + R_k I_k, \quad I_{k-1} \le t \le I_k \tag{1}$$

with $b(0) = 0$. Thus the rates R_k can be viewed as an upper bound on the rate over every interval of length I_k so that

$$A[t, t + I_k]/I_k \le R_k \quad \forall\, t > 0, k = 1, 2, \cdots, P. \tag{2}$$



(a) Advertisements

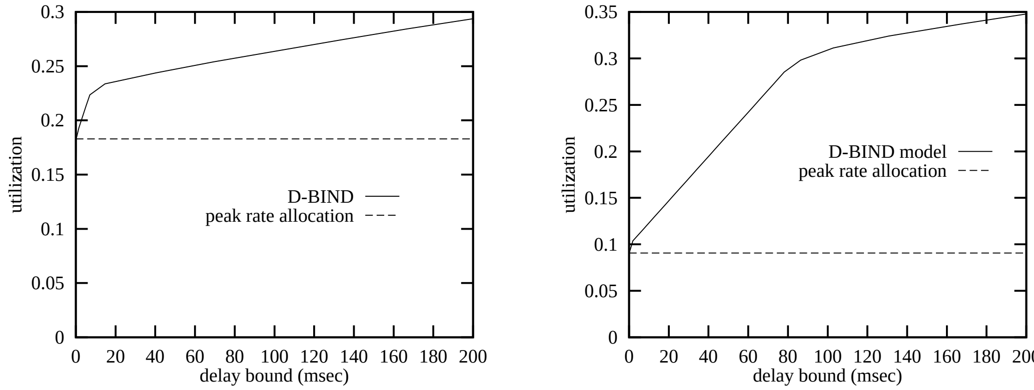
(b) Action Movie

Figure 1: D-BIND Characterization of Advertisements and Action Movie

Figure 1 shows a plot of the D-BIND rate-interval pairs for two traces of MPEG compressed video: a 10 minute series of advertisements and a 30 minute trace of an action movie (a "James Bond" film). The advertisement sequence is digitized to 160 by 120 pixels and compressed at 30 frames per second using the frame pattern **IBBPBB**. The movie is digitized to 384 by 288 pixels and compressed at 24 frames per second with pattern **IBBPBBPBBPBB**. Both sequences were compressed using constant-quality MPEG 1 compression performed in software (see [11] for details of the MPEG compression algorithm). Plotting the bounding rate R_k vs. interval length I_k, the figure shows that the D-BIND model captures the sources' burstiness over multiple interval lengths. For example, Figure 1(a) shows that for small interval lengths, R_k approaches the stream's peak rate of 2.48 Mbps, while for longer interval lengths, it approaches the long term average rate of 457 kbps, which is the total number of bits in the MPEG sequence divided by the duration of the sequence. For the action movie, Figure 1(b) shows that this stream's rate-interval pairs decrease from the peak rate of 5.87 Mbps to its average rate of 583 kbps. The general trend of the rate-interval curves is that the bounding rates approach the sources' peak rate for small interval lengths and the long-term average rate for longer interval lengths. The manner in which the bounding rates decrease with interval length largely determines the achievable utilization when multiplexing such sources. For example, when the bounding rate decreases very slowly from the peak rate to the long-term average rate, it indicates that it will be difficult to achieve high utilization when multiplexing such a source, since bursts of high rate *and* high duration cannot be effectively absorbed by network buffers.

Finally, we note that in practice, a small number of rate-interval pairs provides much of the utilization gain from the D-BIND model. Both [21] and [22] explore this tradeoff in different contexts, and these studies indicate that, under many conditions, more rate-interval pairs do not provide a significant utilization improvement.

2.2 Connection Admission Control for D-VBR



(a) Advertisements (b) Action Movie

Figure 2: Achievable Utilization for Advertisements and Action Movie

Deterministic admission control conditions rely on the delay analysis techniques of [4, 7, 25, 32]. These works show that the delay bound provided to a connection by a server is a function of the server speed and the traffic constraint functions of the all the connections sharing the server. This function varies with different service disciplines. For example, if a FCFS scheduler with link speed l is serving N connections, each bounded by its respective constraint functions $b_j(t), j = 1, \cdots, N$, then an upper bound on delay for all connections is given by:

$$d = \frac{1}{l} \max_{t \in I_k} \{ \sum_{j=1}^{N} b_j(t) - lt \}. \tag{3}$$

The proof starts with an expression for the work in the system at time t, $W(t) = \max_{s \leq t} (\sum_j A_j[s,t] - l(t-s))$ and uses bounds on individual sources $(b_j(t) \geq A_j[s, s+t] \; \forall s, t > 0)$ to bound the aggregate. Delay bounds for priority service disciplines that are more suitable for providing integrated services (e.g., Rate-Controlled Static Priority [31]) may be expressed in a similar manner [21].

For homogeneous connections and using the D-BIND model's constraint function, Equation (3) can be rewritten to express the maximum number of admissible connections as a function of the deterministic delay-bound:

$$N(d) = max\{n \mid nR_k I_k - lt \leq ld, k = 1, \cdots, P\}. \tag{4}$$

This number of connections then has a corresponding average utilization (that is also a function of delay bound) and is given by:

$$U(d) = \frac{N(d) R_\infty}{l} \tag{5}$$

where R_∞ is the source's long-term-average rate (total number of bits transmitted divided by the connections life-time) and l is the link speed.

Figure 2 illustrates the achievable multiplexer utilization for the aforementioned traces. Specifically, we use the D-BIND characterizations of Figure 1 together with the admission control condition of Equation (4) to calculate the maximum number of admissible connections as a function of delay bound for a link speed of $l = 45$ Mbps. We then use Equation (5) to convert this result to average utilization. Figure 2 then depicts average utilization versus deterministic delay-bound for the two video sequences. Note that increasing the delay-bound also corresponds to a directly-proportional increase in the buffer requirement in the multiplexer (see, for example, [30]). Thus, points on the curve represent the maximum average multiplexer utilization that can be achieved when multiplexing homogeneous connections so that no packets are dropped or violate their delay bounds, even in the worst case.

In [22], it was demonstrated that video sequences such as a lecture have rapidly decreasing rate-interval pairs, which in turn allow high network utilizations (e.g., above 60%), even for deterministic service. However, the utilizations are not as high for sequences such as advertisements and action movies. For example, for delay bounds below 100 msec, the advertisement sequence (Figure 2(a)) achieves an average utilization below 27%. The reason for this limit is that in order to provide a deterministic QOS guarantee, the admission control conditions must consider the *worst-case* so that, for example, the guaranteed delay bound is met even if the worst-case bursts of all sources are exactly synchronized. Such a severe QOS constraint directly limits the achievable network utilization.

Thus, if a source has long-duration bursts of high rate, i.e., (R_k, I_k) pairs that decrease slowly, it will be difficult to achieve high network utilization. Intuitively, since resource allocation for deterministic service is based on an upper bound of the source, a source's traffic specification is dominated by the worst-case segment, i.e., the segment with the highest rates over a longer interval. If the bounding rates in the worst-case segment are significantly above the long-term average rate (as for the sequences shown above), low utilization may occur. This measure of "burstiness" is formalized in [20].

In order to achieve higher utilization for such bursty streams, some statistical multiplexing has to be introduced. In the next section, we present an approach to support VBR video that is based on deterministic guarantees with client-controlled renegotiation of QOS parameters and graceful degradation during overload situations.

3 RED-VBR: REnegotiated D-VBR Service

As discussed in Section 1, there are at least two levels of burstiness of VBR video that are important for a resource allocation algorithm: burstiness on a shorter time scale due to the coding algorithm and small-time-scale variations in picture information content, and burstiness on a longer time scale due to scene changes. Burstiness on shorter time scales is effectively taken into account with the D-BIND model and the tighter admission control algorithms. It is burstiness on longer time scales that will result in a low network utilization for a deterministic service.

To increase the network utilization in this case, we propose that the application renegotiates its traffic specification and QOS with the network when its rate changes significantly, where "significantly" is defined by the individual application. We call the video sequence between any two adjacent renegotiation points a *segment*. For example, a session with duration T may have S segments $\{[0, t_1), [t_1, t_2), \cdots, [t_{S-1}, T]\}$, where t_s is the s^{th} renegotiation point. Within each segment or between each pair of negotiating points, a D-VBR service is provided. If a request for more resources is denied, the application will adjust the Q-factor of its compression algorithm and lower the transmission rate, which will gracefully lower the perceptual quality of the compressed video. Renegotiations are accomplished via the signaling mechanism such as the Dynamic Connection Management scheme in the Tenet Protocol Suite [28] or via an ATM signaling protocol in an ATM network. We name this scheme REnegotiated Deterministic Variable Bit Service or RED-VBR.

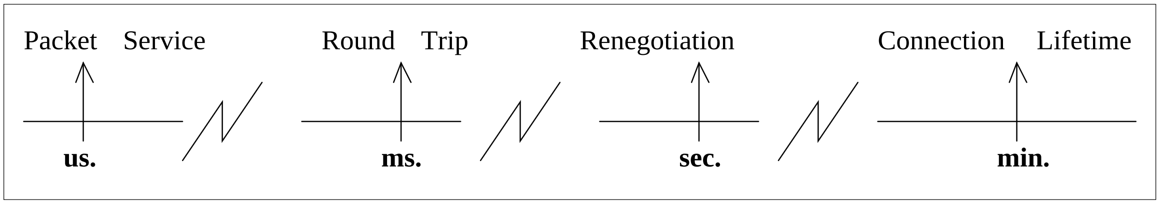


Figure 3: Important Control Time Scales

RED-VBR can be better understood by considering the time-scales that are important for network control as shown in Figure 3. Packet service disciplines at the switches operate at the timescale of a packet transmission time by determining which packet to service next when there is more than one packet in the queue. Connection admission control algorithms operate at the timescale of the connection life-time by deciding whether there are enough network resources to accept a new connection. While traditional resource reservation algorithms effect control at these two timescales, and feedback algorithms do control at the timescale of multiple round-trip times, our approach introduces a new control timescale that is between the round-trip time and connection life-time. It corresponds to the time scale over which the rate of compressed video changes significantly, where "significantly" is defined by the individual application.

An important feature of this approach is that each individual application *determines for itself* the tradeoff between QOS and price-of-service by defining its own segmenting algorithm. In one extreme, a video source which does not want to compromise its QOS at any time may have only one segment for the entire sequence. This is equivalent to the traditional deterministic service with no renegotiations. In the other extreme, a video source that wants to minimize reserved resources may want to renegotiate very frequently. Assuming that there is a pricing policy based on the amount of resources reserved, the first source will have the highest quality but more expensive service while the second source will have a cheaper service with the risk that it may have to degrade its QOS during periods of network overload if a renegotiation fails. If most applications are willing to pay for a more expensive service for better quality, the network may operate at a relatively low utilization. Alternatively, if most applications prefer a cheaper service but are willing to risk that they may have to gracefully degrade their QOS, the network will be able to operate at a relatively high utilization. In contrast, a deterministic service allows only the most expensive service with the best QOS. Additionally, the network will have a cost associated with a renegotiation that reflects the additional load on the network's signaling components. This cost can also be included in the price-of-service that may also affect an applications tradeoff between price and quality-of-service.

Thus, the approach provides a *statistical* service on the level of user-defined *segments* in that it is possible that at a transition from a low-bit-rate to a high-bit-rate video segment, the renegotiation request for more network resources will fail. However, unlike traditional statistical service, this approach gives a higher level of control to individual users and avoids uncontrollable packet drop behavior and extended drop periods by using a *deterministic* VBR service as its foundation.

	traffic known in advance	traffic unknown in advance
not delay-sensitive	video playback	live video recording
delay-sensitive	interactive video browsing	video conferencing

Figure 4: Classification of Video Transmission

To make the service practical, a number of issues need to be addressed. First, even though applications can gracefully adapt to overloading situations, it is still desirable to control the *frequency* of overloading. While overloading in a datagram network means that packets are dropped due to buffer overflow, in a RED-VBR service, it refers to the situation that a renegotiation request for more resources is rejected. To bound this blocking or renegotiation failure probability, we propose an admission control algorithm in Section 4 that calculates $Prob\{block\}$ at each connection establishment time, and accepts the new connection only when this probability is below a certain threshold.

A second issue relates to the renegotiation policy. That is: (a) when should a source renegotiate, or what is its segmenting algorithm? and (b) how can a source derive its D-BIND parameters for each segment *before* each renegotiation? Figure 4 shows how compressed video transmission can be classified according to whether the traffic is known in advance and whether the transmission is delay-sensitive. The degree of difficulty in solving the above problems varies according to which categories an application belongs to. For applications with traffic known in advance, these problems can be solved by off-line algorithms. We describe a a heuristic *off-line* algorithm in Section 5.1. The problems become harder when the traffic is delay-sensitive and unknown in advance. Specifying traffic parameters for live video is a difficult problem that is shared by most of the existing resource allocation schemes. In Section 5.2, we present a heuristic *on-line* algorithm that adaptively chooses traffic parameters for an unknown sequence.

A third issue with RED-VBR is that it places an additional load on the network's signaling components. Frequent renegotiations by many applications require network nodes to have a high throughput in processing renegotiation messages. Most existing signaling systems have relatively low signaling throughputs. For example, our measurements show that a commercial ATM switch can process only about 50 connection establishment requests per second. However, the low throughputs are more an artifact of poor implementation than the intrinsic complexity of signaling. In [3], it has been shown that by optimizing the software, a two-order of magnitude performance gain can be achieved for a portion of the signaling software. Our experience with the Tenet Real-Time Protocol Suite [1] indicates that the time to process a connection establishment message is around 2 msec on a DECstation 500/240 [2]. The bulk of this time is spent in overheads such as user-kernel crossing. The time to actually perform admission control tests and routing computation is relatively small (0.2 to 0.5 msec). Compared to connection establishment, renegotiation is relatively simple because many processing steps such as routing and allocation of data structures are not needed. The only computation to be performed is the admission control test. Our proposed admission control algorithm is much simpler than that used in the current implementation of the Tenet Protocol Suite. By processing the renegotiation request as a special case inside the kernel rather than passing the message to an application-level signaling-process, we believe that per node renegotiation processing can be easily achieved in less than 0.5 msec. In examples to be presented in the Section 6, we will show that the frequency of renegotiation for each application is in the order of tens of seconds. Operating at a 50% utilization, the signaling system can easily support more than 10,000 renegotiating sessions.

A final issue is the response time of the renegotiation request. Since a renegotiation takes one round-trip time to complete, the response time will be on the order of milliseconds for a LAN environment, tens of milliseconds for a WAN environment that spans continental U.S., and hundreds of milliseconds in a global network. Algorithms need to be developed to mask this delay so that the service provided to the application is not unnecessarily disrupted during this period. An off-line algorithm knows in advance when to renegotiate and can send the renegotiation message before the change of traffic characteristics actually happens. For an on-line algorithm, we propose to transmit, but mark, the extra packets above the current reservation as low priority during the renegotiation period. The network will drop low priority packets first when the extra packets cause network congestion.

4 Connection Admission Control for RED-VBR

As describe in Section 3, the RED-VBR renegotiation scheme should be viewed as providing a statistical performance guarantee at the segment level. That is, rather than providing a *cell-level* delay-bound-violation or loss probability, we provide a *segment-level* blocking probability. This blocking probability is the probability that a connection attempts to renegotiate for additional network resources, but has the request denied or blocked by the network. In this section, we present an admission control scheme for the RED-VBR service. Thus, we provide a method for calculating the blocking probability $Prob\{block\}$ for a new connection, given an arbitrary heterogeneous set of pre-existing connections.

4.1 Approach

In order to achieve a statistical multiplexing gain, an admission control algorithm must be able to exploit the statistical properties of individual sources or statistical independence between sources. Thus, previous approaches to providing a statistical performance guarantee are based on a stochastic model of a source (e.g., [8, 9, 16, 19, 23, 33]) such as a Markov-modulated fluid source.

However, such schemes require *a priori* knowledge of a source's statistical properties which may be difficult to obtain for "live" sources. Thus, while one purpose of RED-VBR is to achieve a statistical multiplexing gain in a more controlled manner than previous approaches, a second purpose is to provide a QOS guarantee to sources such as live-video that cannot obtain a traffic specification *a priori*. It will also support sources that have significant long-time-scale variations of their traffic specification over time that may be unknown or difficult to characterize at connection-setup time.

Therefore, the admission control algorithm that we provide is empirical at its foundation and does not require sources to specify how their traffic characterization will evolve over time. For established sources, we will infer information about the sources' "statistics" through simple computations on the history of the sources' reserved parameters over time.

Conceptually, the test can be viewed within the context of the deterministic test. As described in Section 2, a deterministic upper bound on delay for each connection is a function of the traffic constraint functions of all connections. In the case of RED-VBR service, the traffic constraint function of each connection is a piece-wise linear function given by the D-BIND model's $(R_{j,k}, I_{j,k})$ rate-interval pairs. For sources that are renegotiating, their D-BIND parameters and thus constraint function are being updated with time. Thus, for an interval of length $I_{j,k}$, a source will have different bounding rates $R^s_{j,k}$ depending on the traffic characteristics of segment s and thus a different value of the constraint function $b^s_j(I_k)$. We will use $\mathbf{B}_{j,k}$ to denote a random variable that represents the *distribution* of the constraint function across the source's segments. The distribution of $\mathbf{B}_{j,k}$ can be estimated from the source's previous renegotiated segments $R^1_{j,k}, \cdots R^{S_j}_{j,k}$, where these bounding rates are weighted by the duration of the segment τ^s_j. Or, as we propose in the next subsection, the first two moments of $\mathbf{B}_{j,k}$ can be used to approximate the distribution. Note that $\mathbf{B}_{j,k}/I_{j,k}$ is not the distribution of the source's rate per se. Rather, it is the distribution of the deterministic upper bound of the source in a random segment. It can then be used to determine the probability that, when each source wishes to transmit a random segment S_j, if this combination of *deterministic* segments can be scheduled with *deterministic* delay bound d.

Thus, for a FCFS service discipline and a deterministic delay bound d, the probability that a source has a renegotiation request denied by the network is given by:

$$Prob\{block\} = \max_{k=1,\cdots,P} Prob\{\sum_j \mathbf{B}_{j,k} - lI_k > ld\}. \tag{6}$$

4.2 Algorithm Specification

Since the segment blocking probability of Equation 6 is based on a sum of independent random variables, we utilize the Central Limit Theorem (CLT) ([26], p. 287) to approximate $Prob\{block\}$.

More precisely, if source j is currently established, it will have a history of rate-interval pairs $(R^s_{j,k}, I^s_{j,k})$ where $k = 1, \cdots, P$ indexes the rate-interval pair and $s = 1, \cdots, S_j$ indexes the segment number. Thus, the current segment has parameters $(R^{S_j}_{j,k}, I^{S_j}_{j,k})$. As described in Section 2, the deterministic admission control conditions transform source j's D-BIND rate-interval pairs into a piece-wise linear constraint function $b_j(\cdot)$ where $b_j(I_{j,k}) = R_{j,k}I_{j,k}$. The random variable $\mathbf{B}_{j,k}$ then characterizes the variation in $R^s_{j,k}I^s_{j,k}$ over the the different segments s. Thus, to utilize the CLT, the admission control algorithm keeps track of the mean and variance of the constraint function at each of the P interval lengths for each source. These quantities are respectively given by:

$$\mu_{j,k} = \frac{\sum_{s=1}^{S_j} R^s_{j,k} I^s_{j,k} \tau^s_j}{\sum_{s=1}^{S_j} \tau^s_j} \tag{7}$$

$$\sigma^2_{j,k} = \frac{\sum_{s=1}^{S_j} (R^s_{j,k} I^s_{j,k} - \mu^2_{j,k}) \tau^s_j}{\sum_{s=1}^{S_j} \tau^s_j}. \tag{8}$$

where τ^s_j is the length of time that source j had reserved resources with parameters of segment s. Thus, the algorithm does not need to store previous D-BIND characterizations. Rather, it updates the summation of Equations (7) and (8) with each new segment renegotiation.

The mean and variance of $\sum_{j=1}^N \mathbf{B}_{j,k}$ are given by $\mu_k = \sum_{j=1}^N \mu_{j,k}$ and $\sigma^2_k = \sum_{j=1}^N \sigma^2_{j,k}$ since $\mathbf{B}_{j,k}$ and $\mathbf{B}_{j',k}$ are independent random variables for $j \neq j'$. For the new connection N that is attempting establishment, the source specifies initial bounding rate-interval pairs (discussed further below) so that $\mu_{N,k} = R_{N,k} I_{N,k}$ and $\sigma^2_{N,k} = 0$. We utilize the CLT by approximating $\sum_{j=1}^{N} \mathbf{B}_{j,k}$ by the random variable $\mathbf{X}_k$ where $\mathbf{X}_k$ has a Normal distribution with mean μ_k and variance σ^2_k. Equation (6) is then approximated by

$$Prob\{block\} \approx \max_{k=1,\cdots,P} Prob\{\mathbf{X}_k - lI_k > ld\} \tag{9}$$

where this expression may be directly evaluated from the distribution of a Normal random variable.

The above test is the admission control test at the *session* or connection level that determines if the new video session can be admitted without causing excessive renegotiation blocks to any connections. At the *segment* level, when a source performs a renegotiation, the tests of Section 2 are used to determine if a deterministic guarantee can be provided to the *current* set of segments. If not, the request for increased resources for the segment is blocked, but the session is still established at its previously guaranteed rate.

4.3 Discussion

We note several points about the above admission control test. First, while RED-VBR does not require a statistical model per se, a statistical multiplexing gain is achieved through exploitation of statistical independence between sources and by using the previous renegotiation behavior of connections to model their future behavior. As sources go through further renegotiations, $Prob\{block\}$ can be further refined with use of the additional information.

For the new connection N that is attempting to be established, information about its future segments is not, in general, available. Thus, the source needs only to specify an estimate of its initial D-BIND parameters, which, through RED-VBR, can be adapted – immediately if the initial guess was too far off. Even for live video, we conjecture that such an initial parameter guess will not be a severe problem, since, knowing the uncompressed frame size and quality factor, one can always choose a conservative upper estimate based on characteristics of previous transmissions, even if the characterization utilizes only one rate-interval pair for the source's peak rate. Thus, we determine the new blocking probability $Prob\{block\}$ from the available information: 1) the current D-BIND parameters of the new source (possibly an upper estimate) and 2) the current and *past* D-BIND parameters of the segments of established sources along with the length of time that these segments was established.

Second, as discussed in Section 6, some traffic streams such as stored video may know the parameters of all of their segments *a priori*. In this "off-line" case, resources could be reserved in advance to decide *a priori* if everything is schedulable for all segments of all connections. In this case, we can provide a no-block renegotiated service (i.e., $Prob\{block\} = 0$) since we can calculate whether or not this combination of segments is schedulable. Even though such an off-line service that utilizes advance reservations is ultimately deterministic, there will still be a utilization or statistical-multiplexing-gain that comes from passing along knowledge of the "future" to the network.

Finally, we justify the use of the CLT in that, in the calculation of Equation (6), we are not dealing with a rare event or random variables with heavy tails as are often found in other contexts of performance evaluation of high-speed networks. First, we think of $Prob\{block\}$ as being in the range of 10^{-3}. Consider a 2-hour video that renegotiates every 10 seconds on average for a total of 720 segments. Ostensibly, if the source is

```
Procedure Segment_Offline(startIndex, endIndex)
  1. if (endIndex - startIndex ≤ MIN_RENEG_INTERVAL) {
         Output(startIndex, endIndex);
         return();
     }
  2. compute D-BIND curve R[1:P] for video sequence between frames startIndex and endIndex;
  3. identify the worst-case segment, mark its boundary frames using variables leftIndex and rightIndex;
  4. targetRate = ψ * R[P];
  5. extend the segment with the constraint that the average rate of the segment is above the targetRate
  6. Output (leftIndex, rightIndex);
  7.Segment_Offline(startIndex, leftIndex-1 );
  8. Segment_Offline(rightIndex + 1, endIndex);
```

Figure 5: A Off-line Algorithm to Segment a Video Sequence

utilizing the RED-VBR service it is willing to risk having at least one of its segments blocked and 10^{-3} is on the order of a reasonable value for $Prob\{block\}$. Alternatively, a much smaller $Prob\{block\}$ on the order of (say) 10^{-9} is not very meaningful to a source with hundreds of segments, and a source not willing to have any of its renegotiation requests blocked should utilize a deterministic service. Second, we do not expect the random variables $\mathbf{B}_{j,k}$ to possess properties such as heavy-tails that might invalidate the assumptions of the CLT. Recall that the rate-interval pair $(R_{j,k}, I_{j,k})$ is already a worst-case rate over the interval length $I_{j,k}$ and the random variable $\mathbf{B}_{j,k}$ is capturing the variation of this worst-case rate over time. Thus, "heavy-tails" that may be found in a stationary bit-rate distribution (e.g., [12]) are unlikely to occur in this scenario. Trace driven simulation will be used in Section 6 for an *empirical* justification of the use of the CLT.

5 Segmentation Algorithms for VBR Video

In this section, we present two video segmenting algorithms, an off-line algorithm, which knows the whole video sequence in advance, and an on-line algorithm, which knows only the sizes of the frames transmitted so far, but not the sizes of the frames to be transmitted in the future. Both algorithms are heuristic in nature. Our goal here is not to come up with the best segmenting algorithm, but rather to demonstrate that with RED-VBR, even simple segmenting algorithms can achieve substantial multiplexing gains. We leave the design of more advanced segmenting algorithms to future work.

5.1 Off-line Algorithm

The off-line algorithm segments the video sequence assuming the entire sequence is known in advance. Such an algorithm is interesting not only because it can be used for video playback applications, but also because it can serve as a benchmark for comparing the performance of on-line algorithms. A heuristic off-line algorithm is shown in Figure 5. The algorithm takes as its input a sequence of frame sizes (stored in the array video[ ]) and a parameter ψ $(0 \leq \psi \leq 1)$ that indicates how aggressively to segment. A higher ψ will generate

```
Procedure Segment_Online()
    initialize cur_reserve_R[ ];
    LastRenegIndex = 0;
    for (i = 0; i < MAX_FRAMES; i++) {
            compute R[1 : P] based on previous M frames ;
            if (R[k] > cur_reserve_R[k]) for any 0 ≤ k < P{
               for (k = 1; k ≤ P; k++)
                    cur_reserve_R[k] = α * max(cur_reserve_R[k], R[k]);
               LastRenegIndex = i;
               Renegotiate();
            } else if ((R[P] < β * cur_reserve_R[P]) AND
                      (i - LastRenegIndex ≥ MIN_RENEG_INTERVAL)) {
               for (k = 1; k ≤ P; k++)
                    cur_reserve_R[k] = cur_reserve_R[k] + R[k])/2;
               LastRenegIndex = i;
               Renegotiate();
            }
     }
```

Figure 6: A On-line Algorithm to Segment a Video Sequence

more segments and thus will potentially achieve a higher network utilization and less expensive service for the user. First, the algorithm calculates the D-BIND parameters for the entire sequence, i.e, P rate-interval pairs $(R_k, I_k)_{k=1}^{P}$. It then identifies the segment of length I_P that achieves the maximum number of worst-case rates where a worst-case rate is achieved if $A[t, t + I_k] = R_k I_k$. This segment is then extended to the left and right in time until the average rate of that segment has decreased to ψR_P, where R_P is the bounding rate over the longest parameterized interval length I_P. With this segment isolated, the procedure is recursively repeated over the remaining two segments until the sequence is completely segmented.

5.2 On-line Algorithm

In this section, we consider the more difficult case, such as live video transmission, where the traffic trace is not known in advance. We call the algorithm that dynamically computes a traffic specification and issues renegotiation requests the *on-line* algorithm.

Our heuristic on-line algorithm maintains the currently reserved D-BIND parameters and dynamically computes the D-BIND parameters of the previous P frames. The algorithm needs to make the following policy decisions based on the two sets of D-BIND parameters: (a) when to ask for more resources, and how much more? (b) when to ask for less resources, and how much less? In our algorithm, two parameters α and β ($\beta > \alpha; \alpha, \beta \geq 1$) are used to control the policies. If any rate in the measured D-BIND curve exceeds the corresponding rate in the reserved D-BIND curve (i.e., not enough resources are reserved), a renegotiation immediately takes place. The new traffic specification is chosen so that each bounding rate R_k is α times its currently measured value. Thus, in the case of increasing reserved resources, we reserve beyond the current requirements by a factor α so that numerous consecutive increases are not required. For downward renegotiation, the algorithm checks only the rate of the longest parameterized interval. If R_P has fallen below the $cur\_reserve\_R[P]$ by a factor of β, where $R[P]$ and $cur\_reserve\_R[P]$ are the measured and reserved bounding rate over the longest parameterized interval length I_P respectively, and there have been at least MIN_RENEG_INTERVAL frames since last renegotiation, where MIN_RENEG_INTERVAL is a constant, the algorithm will renegotiate to a lower reserved D-BIND parameterization. The lower D-BIND parameters are computed as the average of the currently reserved and currently measured D-BIND parameters. It is easy to see that $\beta > \alpha$ must hold in the algorithm to avoid oscillations between upward and downward renegotiations.

6 Empirical Evaluation of RED-VBR

In this section, we use trace-driven simulation based on the two traces of MPEG compressed video that were described in Section 2. We then do experiments with the admission control conditions of Section 4. These simulations and experiments will be referred to as "TDS" and "AC" below.

6.1 Experimental Setup

Figure 7 illustrates the scenario for the trace-driven simulation (TDS) experiments described below. In this setup, N video streams $j = 1, \cdots, N$ are multiplexed using the RED-VBR scheme. Each stream is transmitted starting with an independent random phase offset ϕ_j, with ϕ_j distributed uniformly between 0 and the length of the trace. Once the video source begins transmitting, the segmentation algorithm segments the video according to the algorithms in Section 5. Renegotiation requests are then submitted to the network accordingly, and some of these requests may be denied because the deterministic test described in Section 2 fails. When a stream reaches the end of the trace, it wraps around to the beginning. We then run 500 independent simulations, with each stream transmitting at least an entire trace during each simulation.

For the N multiplexed streams, we measure the fraction of time that a connection has its request for resources denied by the network. This fraction of time that a source is "blocked" is then averaged over all N streams and used as an empirical measure of the segment-level blocking probability $Prob\{block\}$. (An alternative empirical measure of $Prob\{block\}$ that yields similar results is the fraction of segments that are blocked.) When a renegotiation request for increased network resources is denied, the next renegotiation request is still sent according to only the segmentation algorithm, without being affected by the current denial. This is a policy that must be specified for the simulation, since, alternative policies may allow a source to more frequently attempt to renegotiate for more resources when it is in a blocked state.

The second empirical investigation evaluates the effectiveness of the admission control (AC). As described in Section 4, the RED-VBR admission control algorithm uses the history of established streams' renegotiation parameters together with the D-BIND parameters of the new source to provide a deterministic delay bound d together with a segment-level blocking probability $Prob\{block\}$. Thus, to determine the minimum d and $Prob\{block\}$ for a set of N video sessions, we use the renegotiation histories of the $(N-1)$ established sessions to determine $\mu_{j,k}$ and $\sigma^2_{j,k}$ as in Equations (7) and (8). For the new N^{th} connection, we use only its worst-case

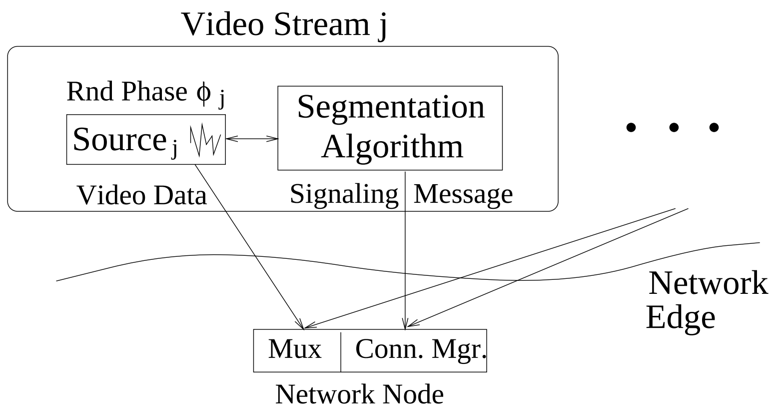


Figure 7: Trace-Driven Simulation Scenario

D-BIND parameters over the entire sequence, since its renegotiation "history" is not, in general, known at establishment time.

For both TDS and AC, we consider an output buffered switch with a link speed of $l = 45$ Mbps and the FCFS service discipline. We perform various experiments varying the following parameters:

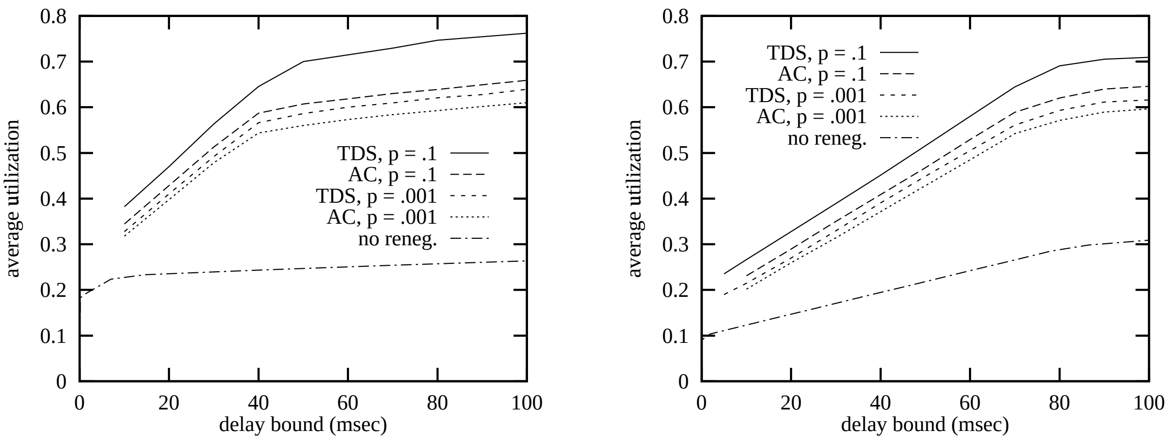
1. blocking probability $Prob\{block\}$.
2. number of connections, or average utilization $\frac{NR_\infty}{l}$
3. the trace: advertisements and action movie
4. delay bound d
5. segmentation algorithm (off-line vs. on-line)
6. renegotiation frequency

Section 6.2 explores items 1-4, considering only the off-line segmentation algorithm with $\psi = 0.7$. Section 6.3 explores items 2-6, fixing $Prob\{block\}$ to 0.01 and focusing primarily on items 5 and 6.

6.2 Evaluation of RED-VBR Service and AC Algorithm

Figure 8 illustrates the effectiveness of the RED-VBR service by showing the average network utilization achieved in various scenarios. For the two traces, advertisements in Figure 8(a) and the action movie in Figure 8(b), average utilization is shown as a function of delay bound for both TDS and AC, for blocking probabilities (p or $Prob\{block\}$ of .1, .001, and 0. $Prob\{block\} = 0$ represents the case of no renegotiation failures, or a D-VBR service. We use the off-line algorithm for segmenting the trace with segmentation parameter $\psi = 0.7$ for both streams. This resulted in average renegotiation intervals of 8.4 and 8.7 seconds for the respective streams.

There are several noteworthy points about Figures 8(a) and 8(b). First, the general trend of the curves is that as delay bound increases, more connections can be multiplexed so that a higher utilization is achievable. The lower curve of the figures (marked "no reneg.") depicts the achievable utilization for a static deterministic VBR service that does not use any renegotiations. This curve, also depicted in Figure 2, shows that because



(a) Advertisements

(b) Action Movie

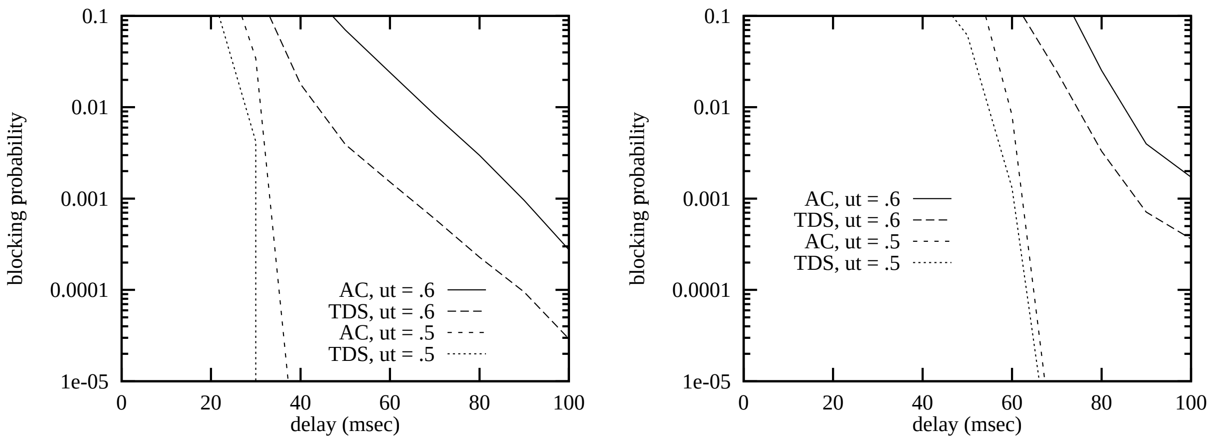
Figure 8: Comparison of AC Algorithm and TDS

of the worst-case nature of the D-VBR guarantee (no packets will be dropped or violate their delay bound) and the burstiness of these video sequences (especially with the "burstiness" definition in [20]), the achievable utilization for deterministic service is limited to approximately 27% for delay bounds under 100 msec. Alternatively, with RED-VBR, significant improvements in network utilization are possible. The upper four curves of Figure 8(a) and 8(b) show the TDS and AC performance of RED-VBR with $Prob\{block\}$ of .1 and .001.

Second, Figure 8 compares the average utilization achieved through trace-driven simulation with that achieved by the admission control conditions. For the trace driven simulation, N connections with random start times are multiplexed at the segment-level. Depending on the deterministic delay bound d (shown on the horizontal axis), a session will spend some fraction of the time in a blocked state. This is $Prob\{block\}$ for TDS. For the admission control, the algorithm in Section 4 calculates the minimum $Prob\{block\}$ for N multiplexed connections, given delay bound d and the renegotiation histories of the already-established $(N-1)$ connections. Figures 8(a) and 8(b) indicate that the AC algorithm is more conservative than the TDS, i.e., it may over-allocate, but not under-allocate resources. In addition, the AC algorithm achieves utilizations close to those indicated possible by the TDS.

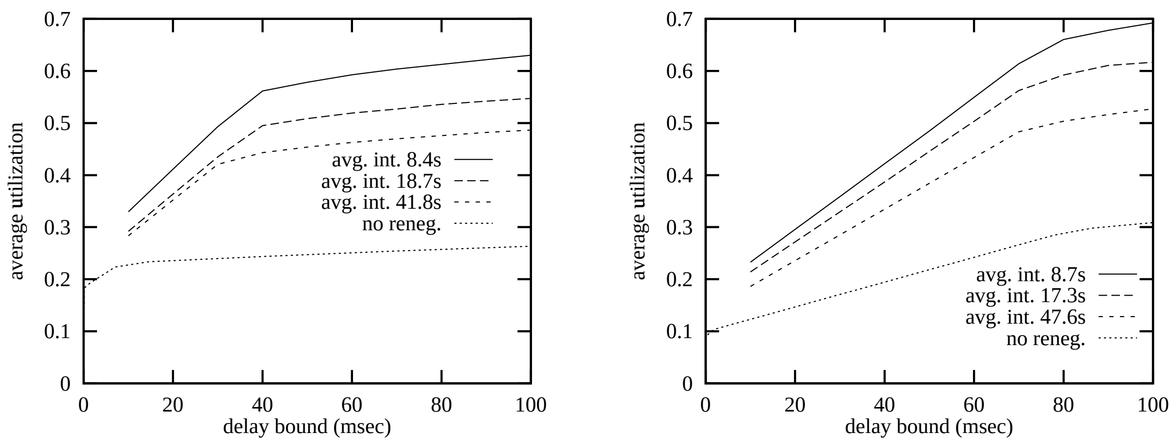
Finally, Figure 8 illustrates the range of multiplexing gains possible for the RED-VBR scheme. For the advertisement sequence of Figure 8(a), and AC with $Prob\{block\} = .1$, the utilization is improved from 24% to 61% at a delay bound of 50 msec. At 100 msec delay, the improvement is from 26% to 66%. Likewise, for the action movie of Figure 8(b), the utilization is improved from 22% to 47% at a delay bound of 50 msec, and from 31% to 65% at a delay bound of 100 msec. These represent respective multiplexing gains of up to 154% and 110% over a static deterministic VBR service.

As indicated by the admission control conditions of Section 4 and the trace driven simulations above, the segment level blocking probability $Prob\{block\}$ is affected by the delay bound and the number of multiplexed connections or utilization. For the off-line algorithm with $\psi = 0.7$, Figure 9 shows the blocking probability versus delay bound for average utilizations of .5 and .6. As indicated by the previous experiments, utilization increases with higher delay bounds and blocking probabilities. As well, the admission control can be seen to be slightly more pessimistic than the trace driven simulation, predicting higher $Prob\{block\}$'s than those actually



(a) Advertisements (b) Action Movie

Figure 9: Effect of Delay Bound on $Prob\{block\}$



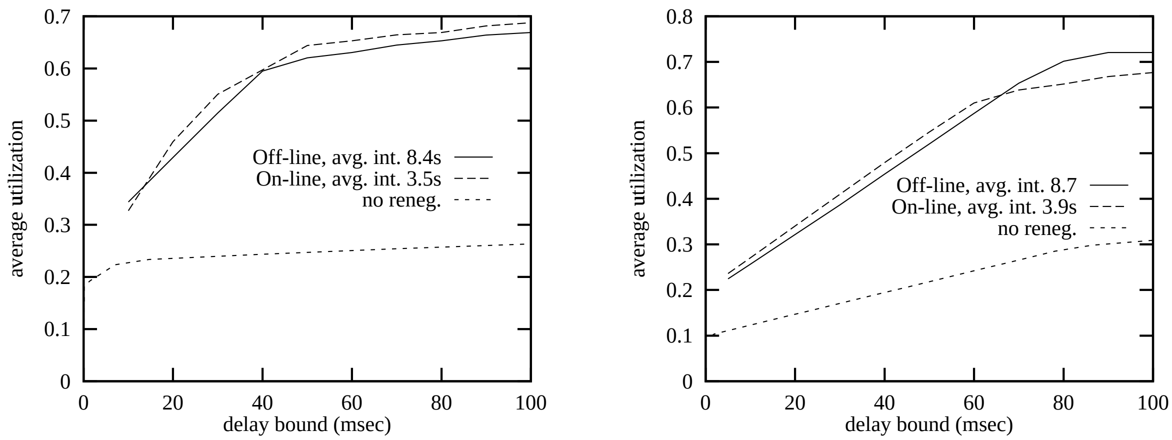
(a) Advertisements (b) Action Movie

Figure 10: Effect of Renegotiation Interval

attained. Figure 9 also indicates that for a fixed utilization, $Prob\{block\}$ decreases roughly exponentially with increased delay bound or buffering in several cases. In addition, the exponential decay rate is much higher for the lower utilizations of .5 than it is for the higher utilization of .6. Note that Figure 9 is not simply the tail of a Gaussian random variable since Equation (9) is a maximization over correlated random variables.

6.3 Tradeoffs Between Utilization and Renegotiation Frequency

In RED-VBR, there is a fundamental tradeoff between the achievable utilization in the network and the frequency of renegotiations. Figure 10 explores this tradeoff for the off-line segmentation algorithm. As described in Section 5, increasing ψ for the off-line algorithm increases the number of segments, thus increasing the renegotiation frequency. For various average renegotiation intervals, the figure shows utilization versus delay bound as in Figure 8. For Figure 10, only admission control is considered (i.e., TDS curves are not shown) and $Prob\{block\}$ is fixed at 0.01. In both Figures 10(a) and 10(b), the three upper curves correspond to $\psi =$ 0.5, 0.6, and 0.7. For the advertisements, these values of ψ correspond to respective average renegotiation intervals of 8.4, 18.7, and 41.8 seconds. For the action movie, the average renegotiation intervals are 8.7, 17.3, and 47.6 seconds. As shown, faster renegotiations (or smaller average renegotiation intervals) result in higher average utilization across the entire range of delay bounds. This alludes to the tradeoff mentioned in Section 3: a more aggressive segmentation policy allows higher network utilization. However, renegotiation intervals that are too small would overload the network's signaling components. Thus, the RED-VBR service can be "priced" to most efficiently utilize all of the network's resources, including buffers, bandwidth, and signaling components.



(a) Advertisements (b) Action Movie

Figure 11: On-line Renegotiation Performance

Figure 11 shows the performance of the on-line algorithm for the advertisement sequence and action movie. The three curves show the case of no renegotiations as a bench-mark, and one off-line case and one on-line case with $Prob\{block\} = 0.01$. Utilization is plotted versus delay bound as obtained by the trace-driven simulation. As shown, the on-line algorithm can achieve utilizations similar to that of the off-line algorithm. However, the on-line algorithm must renegotiate more frequently to achieve a utilization close to that achieved by the off-line algorithm. This is expected since the on-line algorithm does not have knowledge of "future" frame sizes and therefore cannot segment the video as efficiently as the off-line algorithm.

For example, for the advertisements, Figure 11(a) shows that for the on-line algorithm to achieve utilizations similar to those achieved by the off-line algorithm with an average renegotiation interval of 8.4 seconds, the on-line algorithm must have a smaller average renegotiation interval of 3.5 seconds. Likewise, with the action movie of Figure 11(b), the on-line algorithm requires an average renegotiation interval of 3.9 seconds to achieve utilizations similar to those achieved by the off-line algorithm using an average renegotiation interval of 8.7 seconds. For both of the traces, these experiments used parameters of $\beta = 1.5$ and $\alpha = 1.2$ for the on-line segmenting algorithm and $\psi = .7$ for the off-line algorithm.

While the focus here is not to propose the best on-line segmentation algorithm, the experiment of Figure 11 indicates that even simple on-line segmentation algorithms can achieve the same utilization as an off-line algorithm, with smaller, but same order-of-magnitude, average renegotiation intervals.

7 Conclusions

We have proposed a new service called REnegotiated Deterministic Variable Bit Rate or RED-VBR service for supporting transmission of delay-sensitive VBR video in packet-switched networks. The service is based on flexible renegotiation of traffic parameters with graceful degradation of QOS in the case where renegotiations fail. Each client determines its own renegotiation policies – when to renegotiate and what parameters are used. Between two adjacent renegotiation points, a deterministic network service is provided. We have shown that such an approach can achieve significant multiplexing gains without requiring an excessive signaling overhead. For example, with the D-VBR service, the network can only achieve 26% utilization when the video traffic is highly bursty over long intervals. However, using RED-VBR with a simple off-line segmenting algorithm and relatively low renegotiation frequencies of 8 to 42 seconds per renegotiation, high network utilization in the range of 50% to 80% can be achieved for connections with delay bounds of 100 msec. This represents improvements of up to 150% in network utilization compared to the D-VBR service. As well, the on-line algorithm achieves similar improvements but requires a smaller average renegotiation interval. For example, in one case the on-line algorithm required a 3.9 second average renegotiation interval to achieve utilizations close to those achieved with the off-line algorithm and a 8.7 second average renegotiation interval. The on-line algorithm also provides a practical solution to address the issue of specifying traffic parameters for live video. Compared to statistical and predicted service, RED-VBR allows more graceful and client-controlled QOS degradation during overload period.

While this paper demonstrates the effectiveness of the RED-VBR service, a number of issues remain to be explored in future works. Perhaps the most important issue is to design and implement efficient signaling support for renegotiations. Since most existing commercial signaling systems can support only 50 to 100 connection establishments per second, we need to increase the signaling throughput by two orders of magnitude's in order to make a renegotiation-based service feasible. Another area of research is to design more elaborate and general on-line and off-line renegotiation algorithms. These algorithms should be part of a software library that application programs can invoke without the knowledge of the actual algorithms.

References

[1] A. Banerjea, D. Ferrari, B. Mah, M. Moran, D. Verma, and H. Zhang. Tenet real-time prococol suite: Design, implementation, and experiences. *IEEE/ACM Transactions on Networking*, 1995. To appear.

[2] A. Banerjea, E. Knightly, F. Templin, and H. Zhang. Experiments with the Tenet real-time protocol suite on the Sequoia 2000 wide area network. In *Proceedings of the 2nd ACM International Conference on Multimedia*, San Francisco, CA, October 1994.

[3] T. Blackwell. Fast decoding of tagged message formats. To appear in IEEE INFOCOM'96.

[4] C. Chang. Stability, queue length, and delay of deterministic and stochastic queueing networks. *IEEE Transactions on Automatic Control*, 39(5):913–931, May 1994.

[5] 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.

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

[7] R. Cruz. A calculus for network delay, part I : Network elements in isolation. *IEEE Transactions on Information Theory*, 37(1):114–121, January 1991.

[8] A. Elwalid and D. Mitra. Analysis, appoximations and admission control of a multi-service multiplexing system with priorities. In *Proceedings of IEEE INFOCOM'95*, pages 463–472, Boston, MA, April 1995.

[9] A. I. Elwalid and D. Mitra. Effective bandwidth of general markovian traffic sources and admission control of high speed networks. *IEEE/ACM Transactions on Networking*, 1(3):329–43, June 1993.

[10] D. Ferrari and D. Verma. A scheme for real-time channel establishment in wide-area networks. *IEEE Journal on Selected Areas in Communications*, 8(3):368–379, April 1990.

[11] D. Le Gall. MPEG: A video compression standard for multimedia applications. *Communications of the ACM*, 34(4):46–58, April 1991.

[12] M. W. Garrett and W. Willinger. Analysis, modeling and generation of self-similar VBR video traffic. In *Proceedings of ACM SIGCOMM'94*, London, UK, August 1994.

[13] M. Gilge and R. Gusella. Motion video coding for packet switching networks – an integrated approach. In *Proceedings of SPIE Visual Communications and Image Processing '91*, pages 592–603, Boston, MA, November 1991.

[14] M. Grossglauser, S. Keshav, and D. Tse. The case against variable bit rate service. In *Proceedings of IEEE Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'95)*, pages 307–310, Durham, NH, April 1995.

[15] M. Grossglauser, S. Keshav, and D. Tse. RCBR: A simple and efficient service for multiple time-scale traffic. In *Proceedings of SIGCOMM'95*, pages 219–230, Boston, MA, September 1995.

[16] R. Guerin, H. Ahmadi, and M. Naghshineh. Equivalent capacity and its application to bandwidth allocation in high-speed networks. *IEEE Journal on Selected Areas in Communications*, 9(7):968–981, September 1991.

[17] S. Jamin, P. Danzig, S. Shenker, and L. Zhang. A measurement-based admission control algorithm for integrated services packet networks. In *Proceedings of SIGCOMM'95*, pages 2–13, Boston, MA, September 1995.

[18] H. Kanakia, P. Mishra, and A. Reibman. An adaptive congestion control scheme for real-time packet video transport. In *Proceedings of ACM SIGCOMM'94*, pages 20–31, San Francisco, CA, September 1993.

[19] 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.

[20] E. Knightly and P. Rossaro. Effects of smoothing on end-to-end performance guarantees for VBR video. In *Proceedings of 1995 International Symposium on Multimedia Communications and Video Coding*, New York, NY, October 1995.

[21] E. Knightly, D. Wrege, J. Liebeherr, and H. Zhang. Fundamental limits and tradeoffs for providing deterministic guarantees to VBR video traffic. In *Proceedings of ACM SIGMETRICS'95*, Ottowa, Ontario, May 1995.

[22] E. Knightly and H. Zhang. Traffic characterization and switch utilization using deterministic bounding interval dependent traffic models. In *Proceedings of IEEE INFOCOM'95*, pages 1137–1145, Boston, MA, April 1995.

[23] J. Kurose. On computing per-session performance bounds in high-speed multi-hop computer networks. In *Proceedings of ACM SIGMETRICS'92*, pages 128–139, Newport, Rhode Island, June 1992.

[24] A. Lazar, G. Pacifici, and D. Pendarakis. Modeling video sources for real-time scheduling. In *Proceedings of IEEE GLOBECOM'93*, pages 835–839, Houston, TX, November 1993.

[25] J. Liebeherr, D. Wrege, and D.Ferrari. Exact admission control for networks with bounded delay services. Technical Report CS-94-29, University of Virginia, Department of Computer Science, July 1994.

[26] M. Loève. *Probability Theory I.* Springer-Verlag, 4th edition, 1977.

[27] D. Lucantoni, M. Neuts, and A. Reibman. Methods for performance evaluation of VBR video traffic models. *IEEE/ACM Transactions on Networking*, 2(2):176–180, April 1994.

[28] C. Parris, H. Zhang, and D. Ferrari. Dynamic management of guaranteed performance multimedia connections. *Multimedia Systems Journal*, 1:267–283, 1994.

[29] G. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 34(4):46–58, April 1991.

[30] H. Zhang. Service disciplines for guaranteed performance service in packet-switching networks. *Proceedings of the IEEE*, 83(10):1374–1399, October 1995.

[31] H. Zhang and D. Ferrari. Rate-controlled static priority queueing. In *Proceedings of IEEE INFOCOM'93*, pages 227–236, San Francisco, CA, March 1993.

[32] H. Zhang and D. Ferrari. Improving utilization for deterministic service in multimedia communication. In *Proceedings of 1994 International Conference on Multimedia Computing and Systems*, pages 295–304, Boston, MA, May 1994.

[33] H. Zhang and E. Knightly. Providing end-to-end statistical performance guarantees with bounding interval dependent stochastic models. In *Proceedings of ACM SIGMETRICS'94*, pages 211–220, Nashville, TN, May 1994.

[34] H. Zhang and E. Knightly. RED-VBR: A new approach to support VBR video in packet-switching networks. In *Proceedings of IEEE Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'95)*, pages 275–286, Durham, NH, April 1995.