Scheduling and Bandwidth Allocation for the Distribution of Archived Video in VOD Systems*

Marwan Krunz[1] †, Wei Zhao[2], and Ibrahim Matta[3]
1 Department of Electrical and Computer Engineering
University of Arizona
2 Department of Computer Science
University of Maryland
3 College of Computer Science
Northeastern University

Abstract

Providing cost-effective video-on-demand (VOD) services necessitates reducing the required bandwidth for transporting video over high-speed networks. In this paper, we investigate efficient schemes for transporting archived MPEG-coded video over a VOD distribution network. A video stream is characterized by a time-varying traffic envelope, which provides an upper bound on the bit rate. Using such envelopes, we show that video streams can be scheduled for transmission over the network such that the per-stream allocated bandwidth is significantly less than the source peak rate. In a previous work [13], we investigated stream scheduling and bandwidth allocation using *global* traffic envelopes and homogeneous streams. In this paper, we generalize the scheduling scheme in [13] to include the heterogeneous case. We then investigate the allocation problem under *window-based* traffic envelopes, which provide tight bounds on the bit rate. Using such envelopes, we introduce three stream-scheduling schemes for multiplexing video connections at a server. The performance of these schemes is evaluated under static and dynamic scenarios. Our results indicate a significant reduction in the per-stream allocated bandwidth when stream scheduling is used. While this reduction is obtained through statistical multiplexing, the transported streams are guaranteed *stringent, deterministic* quality of service (i.e., zero loss rate and small, bounded delay). In contrast to video smoothing, our approach require virtually no buffer at the set-top box since frames are delivered at their playback rate.

Keywords: bandwidth allocation, MPEG, video scheduling, video-on-demand, traffic envelope

1 Introduction

The advent of broadband asynchronous transfer mode (ATM) networks spurred a large interest in multimedia applications that use the underlying network to exchange textual and audiovisual information. A large volume of the traffic that is generated by these applications consists of video frames transported in a compressed format. To maintain constant-quality motion picture, a video

*Part of this paper was presented at the IEEE ICC '97 Conference.
†Author of correspondence. Address: Department of ECE, ECE Building, University of Arizona, Tucson AZ 85721. Email: krunz@ece.arizona.edu. Tel: (520) 621-8731. Fax: (520) 621-3862.

encoder generates frames that vary in size depending on the scene dynamics in the video. If the video stream is transmitted at a constant frame rate, it exhibits a variable-bit-rate (VBR) behavior. Although ATM networks include a VBR transport service that is geared towards VBR sources, such a service is expected to only support the quality-of-service (QoS) on a *statistical* basis. Statistical guarantees may not be adequate for all types of video, specially when their uniformity cannot be ensured over finite connection hold times.

In this paper, we investigate the transport of MPEG-compressed video streams that require stringent QoS guarantees (no losses and and small bounded delay). Such guarantees are commonly supported by transporting a stream at its peak rate, at the expense of poor channel utilization. Improving the utilization necessitates reducing the variability of the bit rate by means of temporal averaging (i.e., *smoothing*) on a stream-by-stream basis or spatial averaging (i.e., *aggregation*) via statistical multiplexing (stream batching and multicast-based approaches have also been proposed [1, 17, 15]). Video smoothing has been the focus of many studies (for example, [3, 10, 14, 16, 18, 20, 21, 23, 25]). This approach is particularly appropriate for archived-video applications in which the traffic profile (i.e., frame sizes) is known long before transporting the video, thus allowing the server to implement an appropriate transmission schedule that results in high resource utilization. Smoothing of archived video takes a work-ahead approach, whereby video frames are transported prior to their playback times. To reduce the required bandwidth, a video request must be issued before the actual playback time to allow for some buffer build-up in the client's set-top box before the actual commencement of the movie. Despite its general appeal, the work-ahead approach has a number of drawbacks. First, to provide an "optimal" schedule for transporting video frames (which avoids buffer overflow and underflow in the set-top buffer), the video server must have exact knowledge of the end-to-end delay. One can use suboptimal transmission schedules that do not require knowledge of network delay, but this comes at the expense of reduced utilization. Second, to achieve a significant reduction in bandwidth, a large buffer is needed at the client set-top box. As the buffer size increases, so does the build-up delay, with the implication that the client must initiate the request long before the actual commencement of the movie. The amount of build-up delay depends on the utilization and the frame-size characteristics of the movie, and can be as large as few hours for a bursty stream at 100% bandwidth utilization.

As an alternative to video smoothing, we propose a different approach for transporting archived video. Our approach is tailored to MPEG-coded video streams that are generated by VBR encoders (i.e., constant-quality video). Bandwidth gain is achieved by means of statistical (more properly, asynchronous) multiplexing of video connections at the server, but with *stringent, deterministic* QoS guarantees. In principle, supporting deterministic QoS necessitates the use of deterministic traffic models. For this purpose, we use a a time-varying traffic envelope that provides an upper bound on the bit rate. The main idea behind this envelope is to exploit the periodic nature of the groups-of-pictures (GOP) pattern in MPEG compression. The parameters of this envelope are constant over a time-window of a predefined size, and they vary from one time-window to another. With smaller window sizes, the bounds are tighter and the bandwidth allocation strategy is more effective. As a limiting case, the window size can be taken as the whole duration of the movie, resulting in a *global* traffic envelope, which was used in [13]. By using window-based envelopes and by appropriate stream scheduling at the server, the *aggregate* traffic has a peak rate that is much smaller than the sum of the peak rates of individual sources. Thus, allocation can be made based on the peak rate of the aggregate traffic, resulting in significant bandwidth gain. We present several scheduling schemes for heterogeneous sources. In some cases, our schemes achieve about 85% reduction in peak rate; a figure comparable to the best gain from temporal smoothing [10, 23]. Since video streams in our schemes are transmitted at a constant frame rate instead of a constant bit rate, almost no buffer is needed at client side (a 1-frame buffer may be needed by the decoder for synchronization purposes). The startup delay is determined only by the scheduling delay, which is quite acceptable for small window sizes.

The rest of the paper is organized as follows. Section 2 introduces the traffic-envelope model. The basic bandwidth allocation strategy is presented in Section 3. Scheduling schemes are proposed in Section 4. The performance of these schemes is investigated in Section 5 using real video traces. Section 6 summarizes the main findings of the paper.

2 Traffic-Envelope Model

A typical MPEG encoder generates three types of frames: Intra-coded (I), Predictive (P), and Bidirectional (B) frames [8, 9]. On average, I frames are the largest in size, followed by P frames, and finally B frames. The three types of frames are typically generated according to a predefined GOP pattern. Although not mandated by the standards, many available MPEG encoders, particularly those that target constant-quality compression, use the same GOP pattern repeatedly in compressing all frames of a given video sequence. Typically, "regular" GOP patterns are used, where the number of successive B frames between two reference frames (I or P) is fixed (see Figure 1). Throughout this paper we assume regular GOP patterns. A regular GOP pattern is characterized by two parameters:

N : Frame distance between an I frame and the subsequent I frame

M : Frame distance between an I frame and the subsequent P frame

If no P frames are used, then $M \triangleq N$. Since the coding of B frames involves non-causal prediction, before transmitting a video sequence over the network the compressed frames are rearranged according to their decompression order. However, and except for the first few frames, the encoding and transmission orders of an MPEG sequence exhibit similar periodic behavior in the GOP pattern, although the dependency structure is now different. This is illustrated in Figure 1. It is thus reasonable to assume that a stream consists of replications of a given GOP pattern.

A deterministic time-varying bound on the bit rate of a source s_i can be constructed from the 5-tuple $(I^{(i)}_{max}, P^{(i)}_{max}, B^{(i)}_{max}, N^{(i)}, M^{(i)})$, where $I^{(i)}_{max}$ is the largest frame of s_i (typically, an I frame), $P^{(i)}_{max}$ is the largest P or B frame (typically, a P frame), and $B^{(i)}_{max}$ is the largest B frame. It is assumed here that the bit rate over a frame period is uniform, so the frame size is an equivalent representation of the bit rate during that frame period. Note that $I^{(i)}_{max} \geq P^{(i)}_{max} \geq B^{(i)}_{max}$. The parameters $N^{(i)}$ and $M^{(i)}$ describe the GOP pattern of s_i. The five parameters were used in [13] to construct a "global" traffic envelope, which is depicted in Figure 2 (solid line). The simplicity of the global traffic envelope makes it attractive even for real-time video, where the maximum frame sizes may be estimated or policed according to predetermined values. However, the bound represented by

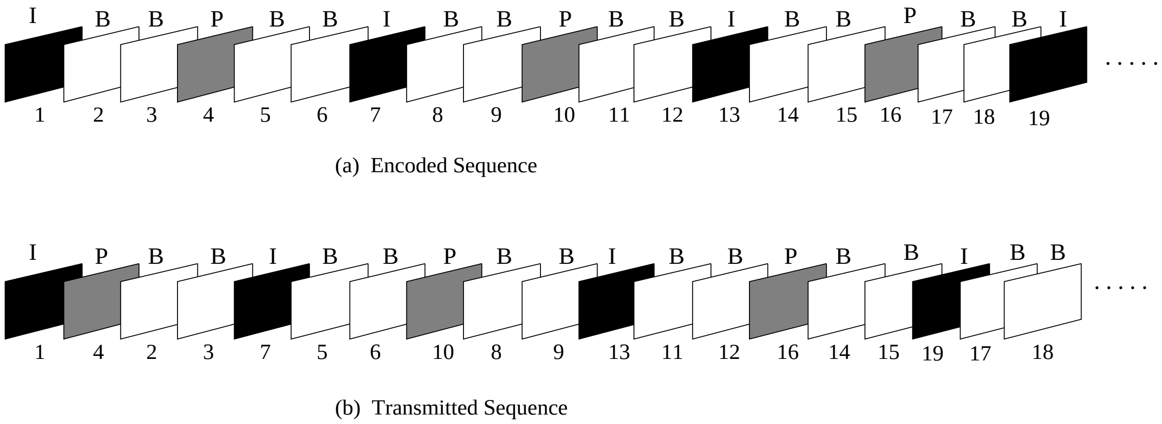


Figure 1: Encoding and transmission orders of an MPEG sequence ($N = 6, M = 3$).

such an envelope can be quite loose since the maximum frame sizes for each frame type (taken over the entire stream) are not representative of most parts of the stream. To obtain a tighter bound, we use a *window-based* traffic envelope. Here, an MPEG stream is divided into segments called *windows* (since a VBR stream is transported at a constant frame rate, a window can be regarded as a time interval and a sequence of frames. Given a pre-coded video stream, a traffic envelope is constructed for each window with the maximum frame sizes taken over frames in that window. For simplicity, we use windows of fixed length W which is taken to be a multiple of N.

The window-based traffic envelope for stream s_i is a piecewise-constant function $\bar{b}^{(i)}(t)$ that is parameterized by

$$E^{(i)} = \left(\boldsymbol{I}^{(i)}_{max}, \boldsymbol{P}^{(i)}_{max}, \boldsymbol{B}^{(i)}_{max}, N^{(i)}, M^{(i)}\right)$$

where $\boldsymbol{I}^{(i)}_{max}$, $\boldsymbol{P}^{(i)}_{max}$, and $\boldsymbol{B}^{(i)}_{max}$ are *vectors* of $W^{(i)}_{max}$ elements; $W^{(i)}_{max}$ being the number of windows in stream s_i. The jth element of each of these vectors represents the maximum size of the corresponding frame type in the jth window. This element is indicated by $I^{(i)}_{max}(j)$, $P^{(i)}_{max}(j)$, and $B^{(i)}_{max}(j)$, for I, P, and B frame types, respectively. Note that $\bar{b}^{(i)}(t)$ is periodic within each window. An example of global and window-based traffic envelopes is depicted in Figure 2.

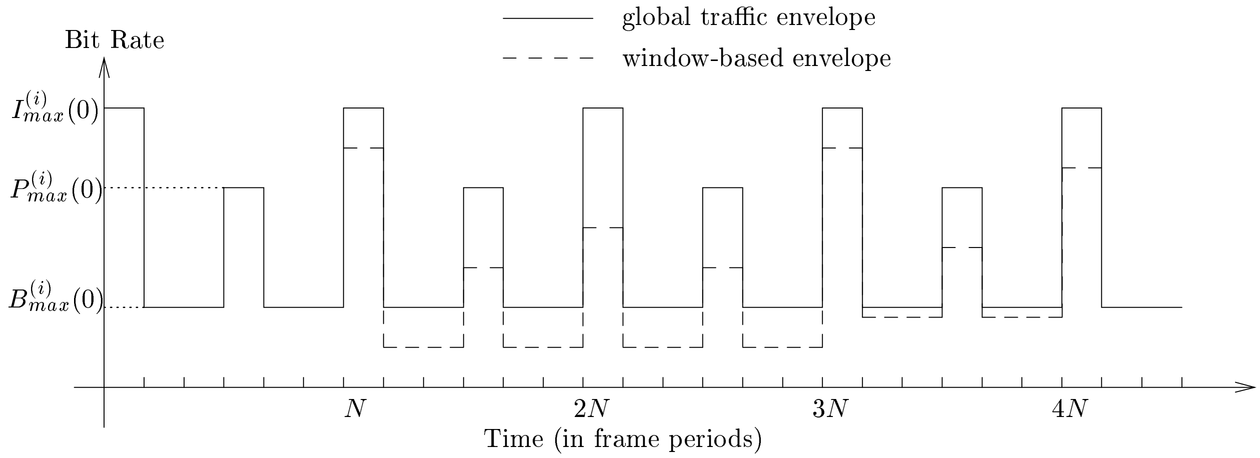


Figure 2: Example of global and window-based traffic envelopes ($W = N = 6$, $M = 3$).

3 Bandwidth Allocation for Multiplexed Streams

By characterizing video streams using the above model, one can intuitively expect that in most cases the peak rate of the aggregate traffic (i.e., after multiplexing) is less than the sum of the individual peak rates. The peak rate of the aggregate traffic depends on the degree of overlap among the GOPs of different streams, with the extreme case when all streams are sending I frames simultaneously. A video server can take advantage of this fact and minimize the total required bandwidth (equivalently, maximize the number of simultaneously transported streams) by appropriate scheduling of video streams at the multiplexer.

Consider a plausible scenario for a video distribution system shown in Figure 3. In this scenario, video streams are transported from the server to several head-end (HE) switches over a public network. Each HE switch provides access to several VOD clients. The distribution network consists of several fixed-capacity bandwidth "pipes" (e.g., CBR ATM virtual paths) over which video connections are transported. The server tries to maximize the utilization of these pipes by implementing stream-scheduling and multiplexing strategies for connections destined to the same HE switch. Since each HE switch is expected to provide access to as many as 1000 connections [17], multiplexing of such connections should intuitively result in significant bandwidth gain.

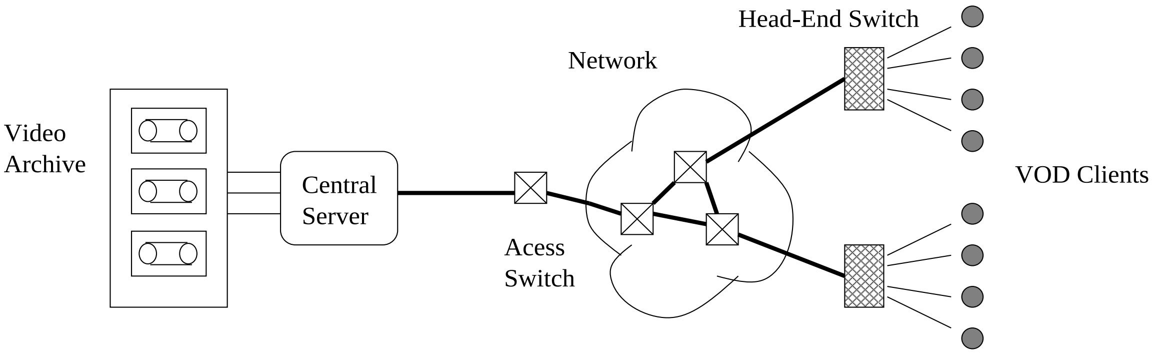


Figure 3: Scenario for video distribution over a wide-area network.

Suppose that n streams, $s_0, s_1, \ldots, s_{n-1}$, are being multiplexed. Let t_i be the starting time of s_i, as determined by the scheduling scheme at the server. We set $t_0 \stackrel{\triangle}{=} 0$ to be used as a time reference. Without loss of generality, we assume that for all i, t_i is integer-valued (in units of frame periods). Consequently, frame boundaries of different streams are aligned in time. Frame sizes are given in ATM cells, with cells being evenly distributed over a frame period. The traffic envelope for the superposition of n streams is given by

$$\bar{b}_{tot}(t) \stackrel{\triangle}{=} \sum_{i=0}^{n-1} \bar{b}^{(i)}(t - t_i) \tag{1}$$

We divide the time axis into windows of size W. New streams are initiated only at the start of a time-window. This will result in a negligible startup delay of no more than W frame periods. From the periodicity of the individual traffic envelopes, $\bar{b}_{tot}(t)$ is also periodic over each window with period $\widetilde{N}$, where $\widetilde{N}$ is the least common multiple of $N^{(0)}, N^{(1)}, \ldots$. Thus, during the jth time-window, $\bar{b}_{tot}(t)$ is completely specified by $\widetilde{N}$ values, which we simply indicate by $\bar{b}_{tot}(0), \bar{b}_{tot}(1), \ldots, \bar{b}_{tot}(\widetilde{N}-1)$, with the understanding that these values are specific to a given window (henceforth, $\bar{b}_{tot}(.)$ with an argument other than t refers to one of these values, while $\bar{b}_{tot}(t)$ will still be used to indicate the general form of this envelope). Likewise, we use $\bar{b}^{(i)}(k)$ to indicate one of the $N^{(i)}$ values of $\bar{b}^{(i)}(t)$ over a given window. Note that $\bar{b}_{tot}(t)$ and $\bar{b}^{(i)}(t)$ are piecewise-constant functions of time. For simplicity, we let the window size be a multiple of $\widetilde{N}$.

Our bandwidth allocation strategy is as follows: During the jth time-window, the aggregate traffic is allocated an amount of bandwidth $B(j)$ that is equal to the maximum value of $\bar{b}_{tot}(t)$ during that window, i.e.,

$$B(j) = \max_{j\text{th window}} \bar{b}_{tot}(t) = \max_{i=0,\ldots,\widetilde{N}-1} \bar{b}_{tot}(i) \tag{2}$$

where $\bar{b}_{tot}(i)$, $i = 0, 1, \ldots, \widetilde{N}-1$, are for the jth window. Under this allocation strategy, a buffer of n cells at the multiplexer (n being the number of ongoing streams) ensures no losses and a maximum cell delay of $n/B(j)$ for all cells transported during the jth time-window. The buffer is needed to handle simultaneous cell arrivals from different streams.

4 Video Scheduling

In this section, we present several stream-scheduling schemes that can be used for efficient multiplexing of video connections at a server. The main objective of these schemes is to achieve significant reduction in the per-stream allocated bandwidth while simultaneously support stringent, deterministic QoS. We consider two cases. In the first case, a video stream is characterized by a *global* traffic envelope, which provides a simple yet relatively conservative model. In the second case, streams are characterized by window-based envelopes. As the window size increases, the window-based envelope approaches the global envelope. We use B (without an argument) to indicate the total allocated bandwidth under global traffic envelopes.

4.1 Scheduling Using Global Envelopes

Given n ongoing streams we define their *arrangement* by the vector $\boldsymbol{u} = (u_0, u_1, \ldots, u_{n-1})$, where $u_i \stackrel{\triangle}{=} t_i \bmod \widetilde{N}$ is referred to as the *phase* of the ith stream. The vector $\boldsymbol{u}$ completely specifies the degree of overlap among the GOPs of different streams. In the homogeneous case (i.e., identical traffic envelopes), it was shown [13] that an "optimal" arrangement that minimizes the allocated bandwidth for n multiplexed streams is given by:

$$\boldsymbol{u}^* = (\underbrace{0, 1, \ldots, N-1, 0, 1, \ldots, N-1, \ldots,}_{w \text{ times}} 0, 1, \ldots, n - wN - 1) \tag{3}$$

where

$$w \stackrel{\triangle}{=} \text{ largest nonnegative integer } k \text{ that satisfies } n > kN$$

The form of $\boldsymbol{u}$ is intuitive; given i ongoing streams, the $(i+1)$th stream is scheduled such that its GOPs are one frame period lag from the GOPs of the ith stream. For heterogeneous streams (streams with different envelopes), $\boldsymbol{u}^*$ is not necessarily optimal. In fact, it can be shown that the optimal arrangement in this case depends on the parameters of all traffic envelopes. And even if such an optimal arrangement is obtained for a given number of streams, the start or termination of streams makes it necessary to rearrange the remaining streams to maintain the optimality, thus disrupting the continuity of the displayed video. Alternatively, we provide efficient suboptimal scheduling schemes for streams that are characterized by heterogeneous envelopes.

Minimal-Rate Phase (MRP) Scheduling

Given n ongoing streams at the multiplexer, a new stream in MRP scheduling is initiated at the phase with the lowest aggregate bit rate. MRP scheduling under global envelopes is illustrated in Figure 4.

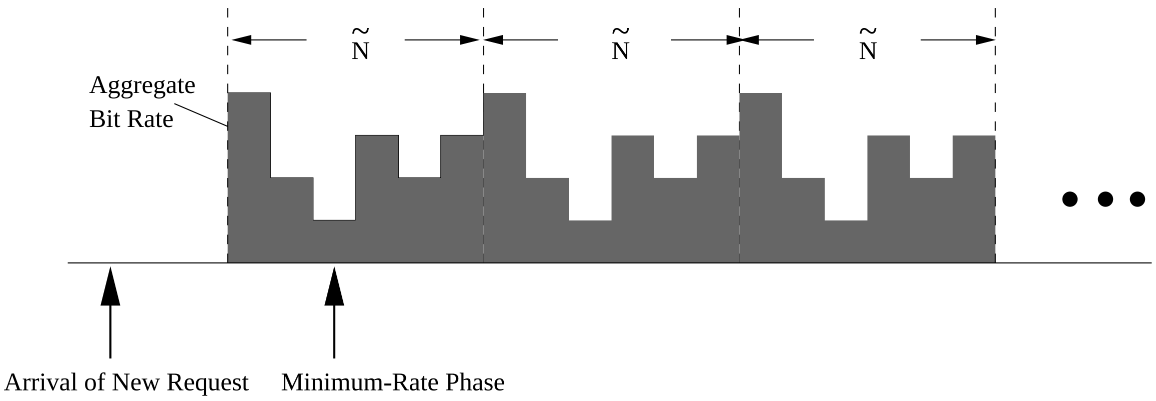


Figure 4: Minimal-Rate Phase scheduling under global traffic envelopes.

In the homogeneous case, a MRP schedule reduces to $\boldsymbol{u}^*$ of (3). In the heterogeneous case, we will show that MRP is asymptotically optimal (i.e., as $n \to \infty$). For simplicity, we assume that the envelopes may differ in their maximum frame sizes but they all have the same GOP pattern. Our results can be extended to the case of different GOP patterns, though the proofs are more involved. Hence, we limit our treatment to the case of common GOP, which is not too unrealistic given that movies in a particular video library will probably be compressed using the same GOP pattern. First, we introduce some preliminary results. We assume that the scheduling of n streams is done incrementally, starting with one stream and adding streams one at a time. Each addition of a stream is called a *step*. Hence, n steps are needed to schedule n streams. While in practice streams may terminate before the addition of new streams, this will have no effect on our asymptotic results for large n. Let $\bar{b}_{tot}^{(i)}(j)$ be the value of $\bar{b}_{tot}(j)$ after the ith step (i.e., after scheduling the ith stream). We refer to $\bar{b}_{tot}^{(i)}(j)$ as the ith partial sum during phase j. Note that $\bar{b}_{tot}(j) = \bar{b}_{tot}^{(n)}(j)$ and $\bar{b}_{tot}^{(i+1)}(j) = \bar{b}_{tot}^{(i)}(j) + \bar{b}^{(i+1)}(j - u_{i+1})$. A *chain* is defined as the set of phases which differ pairwise by a multiple of M. For example, Phases 0, M, and $2M$ belong to the same chain. If the phases of two streams belong to the same chain, then either both streams send I frames simultaneously, or one stream sends I frames while the other is sending P frames (and vice versa).

Lemma 4.1 *The difference between any two partial sums on the same chain is at most A. Formally,*

$$\forall j, k \text{ such that } (j-k) \bmod M = 0, \; \left|\bar{b}_{tot}^{(i)}(j) - \bar{b}_{tot}^{(i)}(k)\right| \leq A \tag{4}$$

where $A = \max_{i \in \{0,1,\ldots,n-1\}} I_{max}^{(i)}$.

Lemma 4.2 *The difference between any two partial sums after any step i is at most $2A$. Formally,*

$$\forall j, k, \; \left|\bar{b}_{tot}^{(i)}(j) - \bar{b}_{tot}^{(i)}(k)\right| \leq 2A \tag{5}$$

The two lemmas are proved in the appendix. The following theorem follows directly from Lemma 4.2.

Theorem 4.1 *Given n streams which are characterized by global traffic envelopes, the aggregate bit-rate function $\bar{b}_{tot}(t)$ resulting from MRP scheduling satisfies:*

$$\left|\bar{b}_{tot}(i) - \bar{b}_{tot}(j)\right| \leq 2A, \; \forall \; 0 \leq i, j \leq N-1 \tag{6}$$

The mean rate for the traffic envelope of stream s_i is given by

$$\overline{r}_i \triangleq \frac{1}{N^{(i)}} \sum_{j=0}^{N^{(i)}-1} \bar{b}^{(i)}(j) = (1/N^{(i)}) I_{max}^{(i)} + (1/M^{(i)} - 1/N^{(i)}) P_{max}^{(i)} + (1 - 1/M^{(i)}) B_{max}^{(i)} \tag{7}$$

$\overline{r}_i$ is a lower bound on the minimum bandwidth required to transmit the stream without delay. For n streams, a lower bound on the total required bandwidth is given by $\sum_{i=0}^{n-1} \overline{r}_i$.

Corollary 4.1 *For n heterogeneous streams with similar GOP patterns, the allocated bandwidth B under MRP scheduling is within $2A$ of a lower bound on the minimum required bandwidth, i.e.,*

$$\sum_{i=0}^{n-1} \overline{r}_i \leq B \leq 2A + \sum_{i=0}^{n-1} \overline{r}_i \tag{8}$$

Proof:

From (2),

$$B \;=\; \max_{j=0,\ldots,\tilde{N}} \bar{b}_{tot}(j) \geq \frac{1}{N} \sum_{j=0}^{N-1} \bar{b}_{tot}(j) = \frac{1}{N} \sum_{j=0}^{N-1} \sum_{i=0}^{n-1} \bar{b}^{(i)}(j) = \frac{1}{N} N \sum_{i=0}^{n-1} \overline{r}_i = \sum_{i=0}^{n-1} \overline{r}_i \tag{9}$$

which proves the left inequality. Since $B = \max_{j \in \{0,1,\ldots,N-1\}} \bar{b}_{tot}(j)$, from Theorem 4.1 we have $B - \bar{b}_{tot}(j) \leq 2A \; \forall j$. Thus,

$$B - \sum_{i=0}^{n-1} \overline{r}_i = B - \frac{1}{N} \sum_{j=0}^{N-1} \bar{b}_{tot}(j) = \sum_{j=0}^{N-1} \frac{B - \bar{b}_{tot}(j)}{N} \leq \sum_{j=0}^{N-1} \frac{2A}{N} = 2A \tag{10}$$

which proves the right inequality. $\square$

The above corollary says that under MRP scheduling the bandwidth allocated for n streams is upper-bounded by $2A + \sum_{i=0}^{n-1} \overline{r}_i$. As n increases, this bound approaches $\sum_{i=0}^{n-1} \overline{r}_i$. However, $\sum_{i=0}^{n-1} \overline{r}_i$ is also a lower bound on the peak rate of the superposition of n envelopes, and the allocated bandwidth under any scheduling scheme is greater than or equal this lower bound. Therefore, MRP is asymptotically optimal.

4.2 Scheduling Using Window-Based Envelopes

Bandwidth allocation based on global traffic envelopes can be relatively conservative if frame sizes vary significantly from one part of the movie to another. To achieve further reduction in bandwidth allocation, we model MPEG streams by window-based traffic envelopes. Throughout this section, we consider heterogeneous envelopes with possibly different GOP patterns. We assume, as before, that n streams $s_0, s_1, \ldots, s_{n-1}$ have been scheduled and are already being multiplexed. The goal of a scheduling scheme is to determine an appropriate phase for initiating a new stream s_n.

Scheme A: Minimal-Stream Phase Scheduling

Here, s_n is scheduled to start at the beginning of the phase with the least number of ongoing streams. Let $P(i)$ be the set of streams with phase i, $P(i) = \{s_j \mid u_j = i\}$. Then,

$$u_n = k \in \{0, 1, \ldots, \widetilde{N} - 1\} \text{ such that } \forall i \in \{0, 1, \ldots, \widetilde{N} - 1\}, |P(k)| \leq |P(i)| \qquad (11)$$

In a sense, Scheme A tries to approximate $\boldsymbol{u}^*$ by distributing streams over different phases as evenly as possible. This, however, may not always give good results, especially when the multiplexed streams vary significantly in their traffic envelopes.

Scheme B: Minimal-Rate Phase Scheduling

Instead of the number of streams in a phase, scheduling can be based on the aggregate bit rate during a phase. This is essentially a MRP scheduling adapted for window-based envelopes. Here, s_n is scheduled to start at the phase with the least aggregate bit rate *in the next window*. The idea is illustrated in Figure 5. Note that s_n arrives during the current window, but is initiated in the subsequent window.

While Scheme A works well for homogeneous streams, Scheme B is expected to perform better when the multiplexed streams are heterogeneous. The reason is that Scheme B tries to evenly distribute the aggregate bit rate among all phases, irrespective of the number of streams in each phase, giving rise to a smoother envelope than the one obtained from Scheme A. We showed earlier that MRP scheduling with global traffic envelopes is asymptotically optimal. This need not be true under window-based traffic envelopes, since u_n is selected based on the minimum aggregate bit rate over one time-window only.

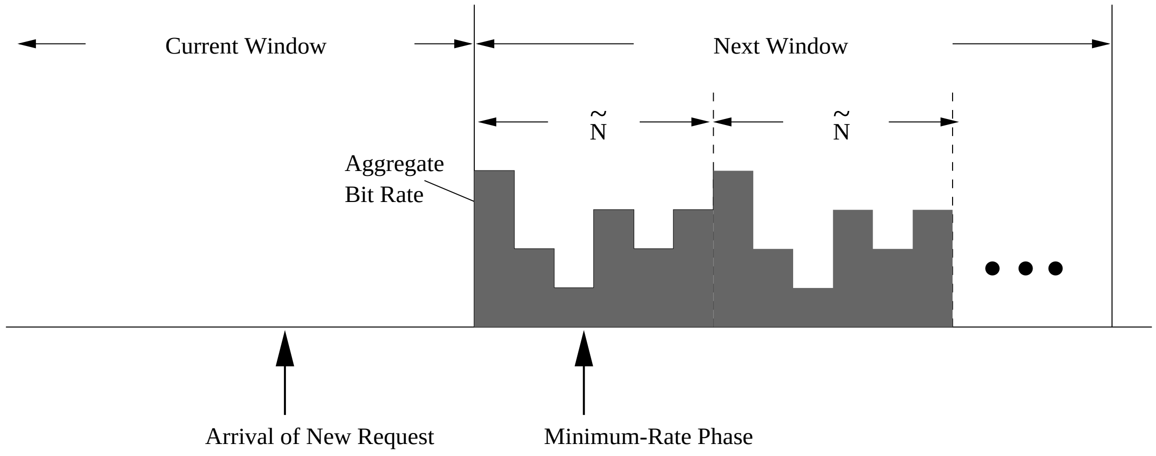


Figure 5: Minimal-Rate Phase scheduling under window-based envelopes.

Scheme C: Most-Window Minimal-Rate Phase Scheduling

The effectiveness of Scheme B can be improved by considering *all* relevant time-windows in the determination of the minimal-rate phase. In Scheme C, s_n is scheduled to start in the phase with the minimum aggregate bit rate in most *active* time-windows. Let $C(p)$ be the set of windows for which phase p is the minimal-rate phase. Then, Scheme C schedules the new stream as follows:

$$u_n = k \in \{0, 1, \ldots, \widetilde{N} - 1\}, \text{ such that } \forall i, |C(k)| \geq |C(i)| \tag{12}$$

To implement Scheme C, $\widetilde{N}$ counters are needed to record the sizes of the sets $C(k)$ for $k = 0, \ldots, \widetilde{N} - 1$. An MRP schedule is first obtained for all time-windows during which s_n will be active (active windows are determined from the length of the requested movie and the window size). Every time MRP selects a phase the counter associated with that phase is incremented by one. $\widetilde{N}$ comparisons are needed to determine the MRP schedule in each time-window. If s_n contains L_n frames, then the computational complexity of Scheme C is $\mathcal{O}(\widetilde{N}\, L_n/W)$, which is quite feasible. For a two-hour movie with a frame rate of 30 frames/s, even with the minimum allowable $W = \widetilde{N}$, the computations are in the order of milliseconds, which is quite acceptable for VOD services. Online bandwidth calculations are of similar order of complexity.

Since bandwidth computation incur minor delay, the startup delay is mainly determined by the delay due to scheduling, which is composed of a phase delay and a window delay. Phase delay is at most $\widetilde{N}$ frame periods ($\sim 1/2$ second) for the three scheduling schemes. Based on our setting, a new stream must be initiated no later than $\widetilde{N}$ frame periods after the start of the time-window that follows the arrival of the video request. Thus, a maximum window delay of W is possible, which is tolerable provided W is not too large.

If the system has a fixed bandwidth B_{avail}, we can use a static admission control that checks the maximum bandwidth $B_{max} = \max_j B(j)$ against B_{avail}. On the other hand, if the underlying network has a renegotiated CBR service, then $B(j)$ can be renegotiated for each window j. However, appropriate actions should be taken if renegotiation fails, such as dropping connections or reducing quality. In our experiments, we use static admission control to compute the admissible connections.

5 Performance Results

In this section, we evaluate the performance of the previous scheduling schemes using four real MPEG-1 video traces. These traces were captured by different researchers [4, 7, 10, 12] for various types of video (see references for compression details). Table 1 summarizes the main characteristics of the examined traces.

Trace	Length (in frames)	% of Mean/Peak	N	M
Wizard of Oz [12]	41760	12.2%	15	3
Star Wars [4]	174136	8.5%	12	3
Lecture [11]	16316	22.3%	6	3
Last Action Hero [7]	238680	6.1%	12	3

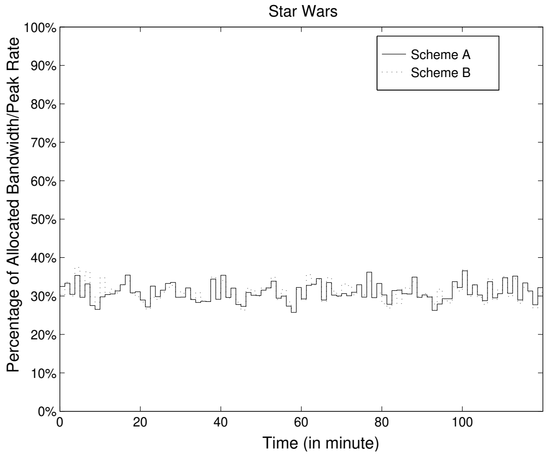
Table 1: Traces used to evaluate the performance of video-scheduling schemes.

Two types of results are reported. One for the "static" scenario in which the number of multiplexed streams is fixed with the assumption that video requests are evenly spread out over the time duration of a particular movie. Such a scenario provides insight about the behavior of the system at equilibrium. The second type of results is for a dynamic scenario in which video requests arrive at the server according to a Poisson process, and they terminate after fixed time intervals that correspond to movie durations. In this case, both the number of ongoing connections and the per-stream bandwidth fluctuate continuously with time. Our performance measures for both types of results are the average allocated bandwidth per stream (in percentage of the source peak rate) and the number of admissible streams. For the latter measure, admissibility is determined based on the worst-case bandwidth over all subsequent windows during which the examined stream will be active if it was admitted.

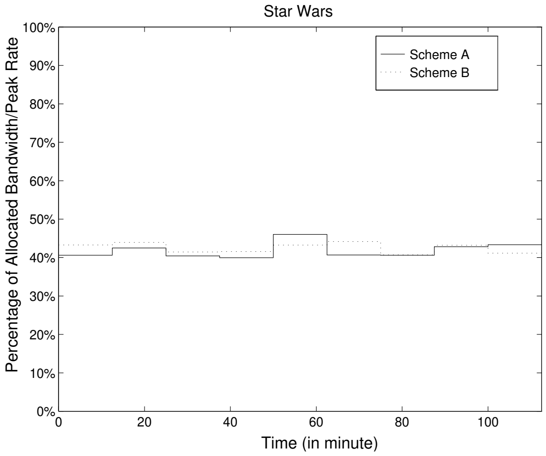
5.1 Static Scenario

Figure 6 shows the allocated bandwidth as a function of time for ten *Star Wars* [5] sources using Schemes A and B. As the window size increases from $W = 1800$ frames (Figure 6 (a)) to $W = 18000$ frames (Figure 6 (b)), the average allocated bandwidth tends to increase. Schemes A and B have comparable performance, with Scheme A being slightly better on the average. Although Scheme B chooses a minimal rate phase, which tends to smooth out the difference between phases, the choice is only based on the first window. Because the envelope parameters differ from one window to another, the selected phase may not be a minimal-rate phase in these windows. This explains why Scheme B fails to perform better in this case. Note that even with a small number of multiplexed streams, 60-70% reduction from the peak rate is achieved.

To examine the impact of the window size on the allocated bandwidth, we tested schemes A, B, and C with different window sizes using *Star Wars* and *Wizard of Oz* traces (taken separately). The average bandwidth per stream is shown in Figure 7 as a function of the number of multiplexed streams. Intuitively, the smaller the window size, the tighter the traffic-envelope bound and the more efficient the allocation strategy. As W goes to infinity, the window-based traffic envelope reduces to the global traffic envelope. The three scheduling schemes have comparable performance, with



(*a*) $W = 1800$ frames (1 min)



(*b*) $W = 18000$ frames (10 min)

Figure 6: Allocated bandwidth as a function of time for ten multiplexed streams.

Scheme B being slightly worse than the other two. Schemes A and C perform similarly, with one outperforming the other in certain cases and vice versa. In fact, the plots of schemes A and C overlap significantly, especially in the case of *Star Wars* streams. Note that the bandwidth per stream for all window sizes converges very quickly, with the multiplexing gain of 20 sources almost as good as that of 200 sources. For a small window size ($W = 60$ frames = 2 seconds), the three schemes achieve an average bandwidth of about 15% of the peak rate. Even for large windows ($W = 1800$), the allocated bandwidth is less than 30% of the peak rate.

It would be interesting to compare bandwidth gain from our schemes to that achieved using video smoothing. One problem in doing such a comparison is that the gain in video smoothing depends on the buffer size in the set-top box and on the startup delay. According to one "optimal" video smoothing approach [23], a reduction of 75-87% of the peak rate was reported for a client buffer of about 1 MB and a startup delay of less than a second. In our schemes, a window size of 60 frames results in 80-85% reduction of the peak rate (see Figure 7), which is quite comparable to the maximum gain from video smoothing. However, in contrast to video smoothing, our approach require no buffer at the set-top box since frames are transported at their playback rate. With smaller window sizes (e.g., $W = 30$), more gain can be achieved at the expense of more bandwidth computations at the server.

The performance of the stream-scheduling schemes in terms of the overall utilization can be obtained by contrasting the third column in Table 1 to the plots in Figure 7. For example, for *Wizard of Oz* the mean rate is about 12.2% of the peak rate. When multiplexing more than twenty *Wizard of Oz* streams, the three scheduling schemes result in per-stream bandwidth of about 17% of the source peak rate (with $W = 60$). Thus, the utilization in this case is about 74%. For $W = 30$ (not shown) the utilization reaches up to 90%. For *Star Wars* the utilization is about 57% when $W = 60$ and 79% when $W = 30$.

Figure 8 (a) depicts the number of admissible connections based on our scheduling schemes versus its counterpart based on source-peak-rate allocation for *Star Wars* with $W = 300$. The number of admissible connections based on our schemes is more than four times that of source-peak-rate allocation. The figure also depicts the performance using the global traffic envelope ($W = \infty$). Note that the number of admissible connections based on our scheduling schemes is almost a linear function of available bandwidth with a slope several times the slope of the corresponding curve under peak-rate allocation. This suggests that VBR connections are admitted almost like CBR connections, but with an effective bandwidth that is a fraction of the peak rate. Figure 8(b) shows the impact of the window size on the number of admissible connections. As expected, more connections can be admitted with smaller window sizes.

We also tested our schemes when video streams of different movies are multiplexed. Figure 9 shows the performance for two mixes; each consisting of three different movies. The average allocated bandwidth (over all windows and over all ongoing streams) is depicted as a function of the number of streams. To multiplex a given number of streams, we incrementally add streams while alternating among the three movies until reaching the desired number of streams. Comparing both figures to the previous figures for the multiplexing of a particular movie, it can be noted that for mixed sources, schemes B and C outperform Scheme A, with Scheme C yielding the best result. The reason is that the phase with the least number of streams which is used in Scheme A is no longer necessarily a good phase to schedule the start of a new stream. Due to the repetitive fashion of our stream selection, Scheme A always schedules streams belonging to a certain movie at a certain set of phases. Because of the difference in the traffic envelopes from one source to another, the aggregate bit-rate in some phases is significantly greater than that of other phases, causing an increase in the per-stream allocated bandwidth.

5.2 Dynamic Scenario

The results in the previous section were computed numerically assuming a fixed number of streams in a given experiment. It would be useful to evaluate the performance in a dynamic setting that simulates the situation at a video server. In the following simulations, we assume that video requests arrive at a server according to a Poisson process with rate λ. The server maintains the set of movies given in Table 1. Each movie has a fixed length, which differ from the lengths of the other movies. These lengths are determined based on the lengths of the corresponding traces assuming that movies are encoded at a rate of 30 frames/second. Movies are requested with equal probabilities (one can also assign unequal probabilities using, for example, Zipf's law). When a request arrives during a time-window, an admission test is performed. If the connection is admitted, the corresponding video stream is scheduling to start in an appropriate phase in the *subsequent* time-window. Let $\rho \triangleq \lambda\overline{\sigma}$, where $\overline{\sigma}$ is the average duration of a movie (in case of multiple movies, $\overline{\sigma}$ is the arithmetic average of the durations of these movies). ρ represents the average number of transported connections at equilibrium, excluding the ones that arrive in the middle of a time window and are waiting to be initiated in the next window.

Figure 10 depicts a sample path for the number of connections in the system assuming a particular movie (the *Wizard of Oz*) is always being requested. The figure shows both the number of admitted connections and the number of actually transported connections. The two quantities differ due to the fact that accepted requests are serviced only at the beginning of a time-window. Consequently,

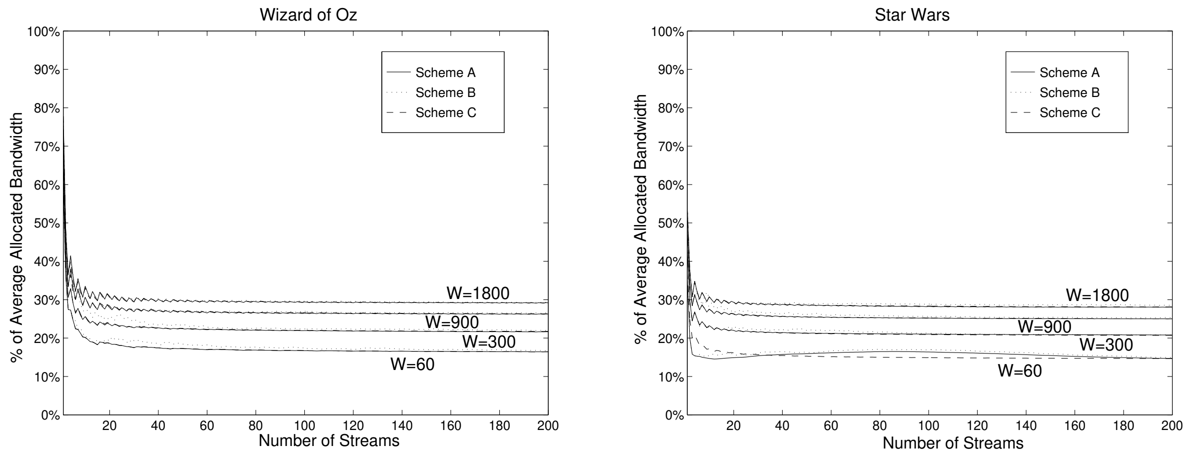
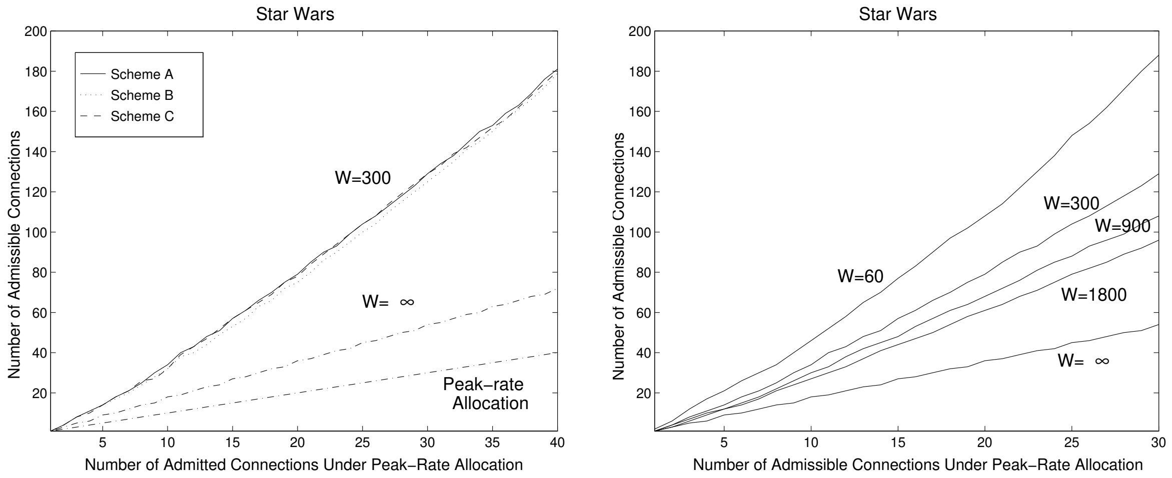


Figure 7: Ratio of average allocated bandwidth/peak rate versus number of multiplexed streams.



(a) $W = 300$ (b) different window sizes

Figure 8: Number of admissible connections.

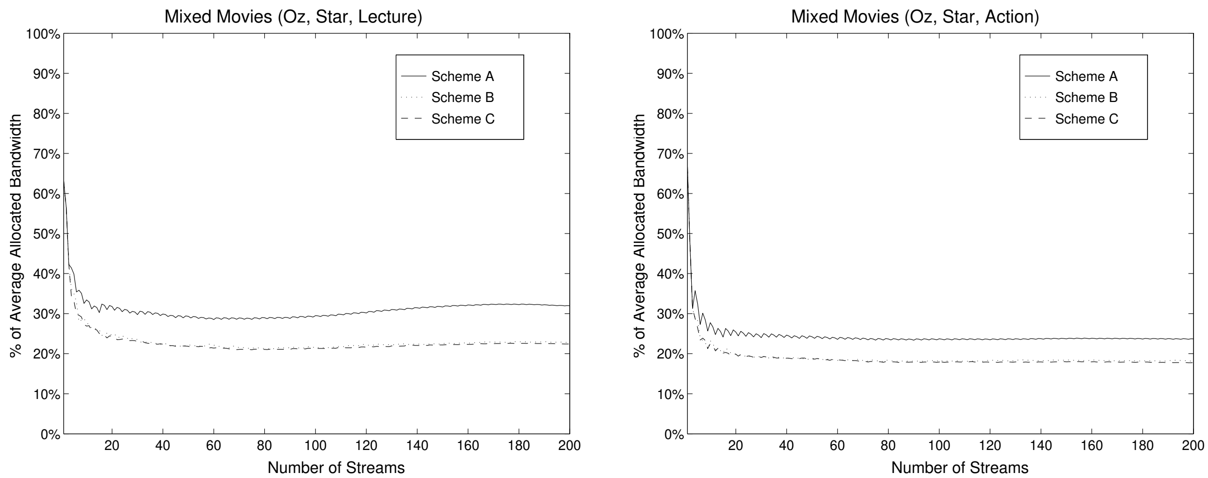


Figure 9: Ratio of average allocated bandwidth/peak rate versus number of streams ($W = 300$).

the number of transported connections increases at time instants $W,\ 2W, \ldots$, and decreases at time instants $W+\overline{\sigma},\ 2W+\overline{\sigma},\ 3W+\overline{\sigma}, \ldots$.

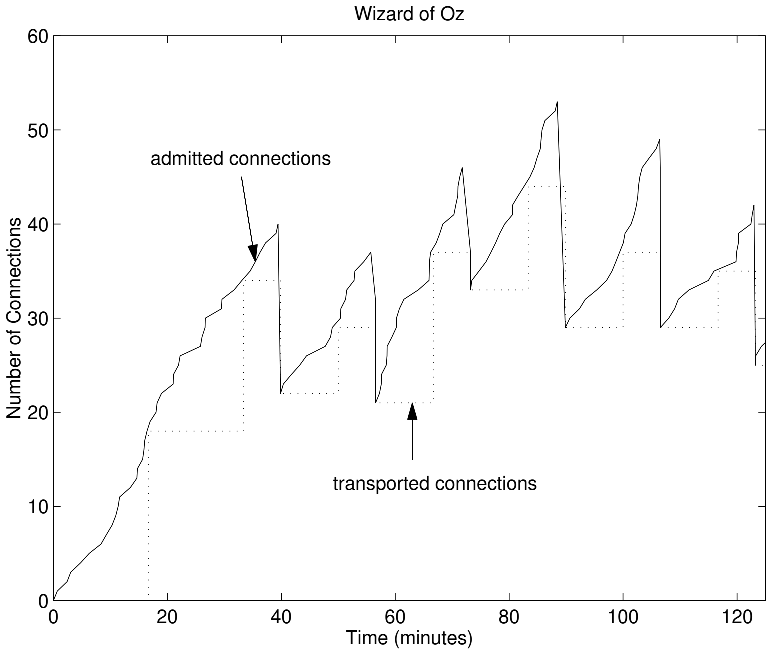


Figure 10: Numbers of admitted and transported connections as a function of time ($\overline{\sigma} = 23.2$ min, $W = 16.667$ min).

It would be useful to evaluate the *average* delay incurred due to windowing in our schemes (the worst-case delay is given by W). This can also be used to determine the average number of admitted connections. Let d be the windowing delay that is experienced by a newly admitted connection whose request arrive in the middle of a time-window, and let $\tilde{d} = W - d$, which represents the time between the start of the current window and the arrival of the request during that window. First, we will determine $E[\tilde{d}]$. Let A be the number of requests that arrive during the current window. Clearly, A has a Poisson distribution with parameter λW. Let x_i be the difference between the arrival time of the ith request of the current window and the beginning of the current window; $i = 1, \ldots, A$. x_i is the sum of i independent and exponentially distributed interarrival times. Thus, conditioned on $A = k$, x_i has a gamma distribution with scale parameter λ and shape parameter i, for $i = 1, \ldots, k$. Accordingly,

$$E[\tilde{d}/N = k] = E\left[\frac{\sum_{j=1}^{k} x_j}{k}\right] = \frac{1}{k}\sum_{j=1}^{k} E[x_j] = \frac{1}{k}\sum_{j=1}^{k}\frac{j}{\lambda} = \frac{k+1}{2\lambda} \tag{13}$$

From (13) we obtain

$$\begin{aligned} E[\tilde{d}] &= \sum_{k=1}^{\infty} \frac{k+1}{2\lambda} \Pr[N=k] = \frac{1}{2\lambda}\left\{\sum_{k=1}^{\infty} k \Pr[N=k] + \sum_{k=1}^{\infty} \Pr[N=k]\right\} \\ &= \frac{1}{2\lambda}\left\{\sum_{k=0}^{\infty} k \Pr[N=k] + 1 - \Pr[N=0]\right\} \end{aligned}$$

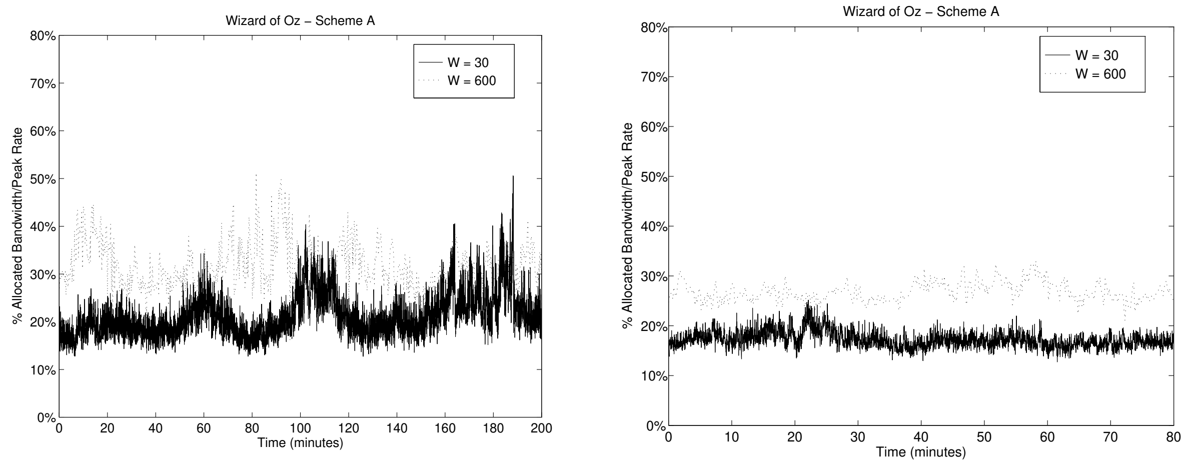
$$= \frac{1}{2\lambda}\left\{\lambda W + 1 - e^{-\lambda W}\right\} = \frac{W}{2} + \frac{1 - e^{-\lambda W}}{2\lambda} \tag{14}$$

Therefore,

$$E[d] = \frac{W}{2} - \frac{1 - e^{-\lambda W}}{2\lambda} \tag{15}$$

From (15) one can determine the average number of admitted connections, which is given by $\lambda(\overline{\sigma}E[d])$.

Figure 11 depicts the fluctuations of the allocated bandwidth with time in the dynamic scenario using Scheme A. In contrast to the static scenario (see Figure 6), bandwidth updates are performed when a stream terminates or when a new window starts (bandwidth updates due to new streams are also performed at the start of a window). As ρ increases, the level of bandwidth fluctuations decreases, since now the addition or termination of a stream has less impact on the per-stream bandwidth.



$(a)\ \rho = 10$ $(b)\ \rho = 30$

Figure 11: Allocated bandwidth/peak rate as a function of time in the dynamic scenario.

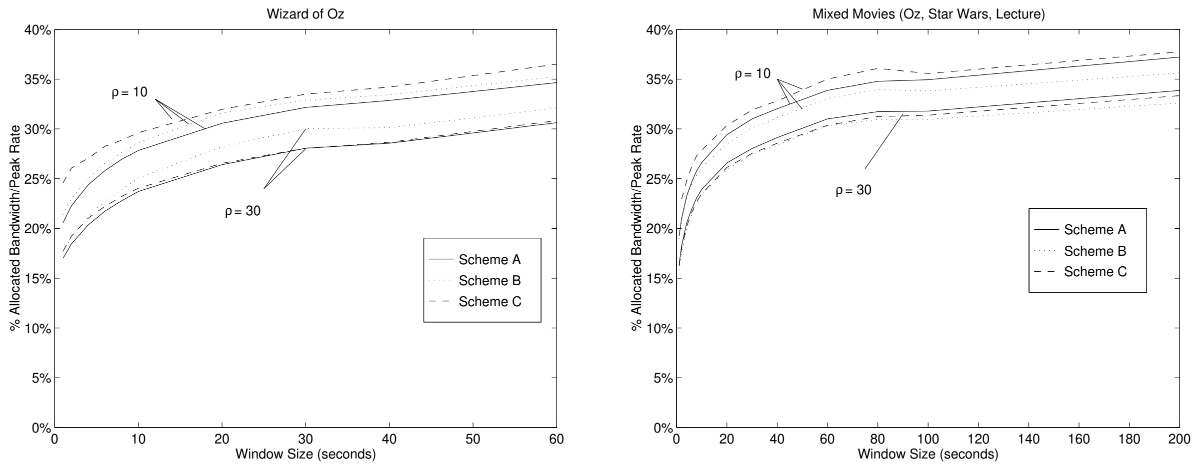


Figure 12: Average allocated bandwidth/peak rate for different window sizes in the dynamic scenario.

The average allocated bandwidth (as a percentage of the source peak rate) versus the window size is shown in Figure 12 for each of the window-based scheduling schemes. Each point in the figure represents a time average of the per-stream bandwidth computed over the duration of the simulation experiment. Each simulation experiment was run for 1440 minutes of simulated time. As expected, more bandwidth gain is achieved with smaller window sizes and with larger ρ. However, for $W \geq$ few hundreds of frames, the window size starts to have less impact on the gain. Furthermore, as ρ increases its impact on the bandwidth starts to fade away (which is predicted from the asymptotic behavior in Figure 7). What is not intuitive in Figure 12 is the relative performance of the three scheduling schemes. One would have expected Scheme C to achieve the best performance. However, in the single-movie case (i.e., no mixing) and at $\rho = 10$, the performance of Scheme A is slightly better than that of Schemes B and C. For larger values of ρ, Scheme A and C have almost the same performance, which is slightly better than that of Scheme B. When streams of different movies are multiplexed, Scheme B shows a slightly better performance than the other two. Overall, the three schemes achieve comparable performance in the dynamic scenario. This is justified by the fact that the scheduling criteria for Schemes B and C do not take into account the dynamic nature of active connections at the server. For example, the scheduling criterion for Scheme C is based on the minimal-rate phase with the largest number of occurrences in future active windows *as determined by currently active streams.* The continuous addition and termination of streams reduces the scheduling efficiency of Scheme C. Note that as ρ increases and for a particular movie, the impact of a single stream is reduced, and Scheme C starts to be more efficient than Scheme B. In the dynamic scenario, we expect that further gain can be obtained using a variant of Scheme C which is based on the minimal-rate phase over a number of time-windows for which no more that one arrival or termination is anticipated. This is currently being investigated and results will be reported in a future paper.

6 Summary

In this paper, we proposed an efficient strategy for transporting VBR MPEG-coded video streams from a video server to clients over a distribution network. Our approach achieves a significant reduction in the allocated bandwidth (compared to peak-rate allocation), while simultaneously providing stringent, deterministic QoS guarantees. To achieve such gain, we exploit the periodic structure of MPEG video by characterizing video streams using time-varying traffic envelopes. Based on such characterization, stream-scheduling and multiplexing are used to reduce the per-stream bandwidth at the server. Unlike video smoothing approaches, video frames in our approach are transmitted at a constant frame-rate and no extra buffering is needed at the receiving end. For heterogeneous sources, we introduced a minimal-rate phase scheduling and established its asymptotic optimality under global traffic envelopes. Window-based traffic envelopes are introduced to provide a tighter bound on the actual bit-rate of MPEG streams. Based on such envelopes, three stream-scheduling schemes were presented for heterogeneous sources, which were shown to achieve significant bandwidth gain. The effectiveness of these schemes was demonstrated via numerical results and simulations using real video traces.

Appendix

A Proof of Lemma 4.1

We prove the lemma by induction on the number of "steps". For the first stream, the partial sums are $I^{(1)}_{max}$, $P^{(1)}_{max}$, or $B^{(1)}_{max}$. So the difference of partial sums between any two phases is at most A. Suppose that (4) holds after the mth step. At the $(m+1)$th step, MRP chooses u_{m+1} for stream s_{m+1} such that

$$\forall p \in \{0, 1, \ldots, \widetilde{N} - 1\},\ \bar{b}^{(m)}_{tot}(u_{m+1}) \leq \bar{b}^{(m)}_{tot}(p) \tag{16}$$

The partial sum at phase j is incremented by $\bar{b}^{(m+1)}(j - u_{m+1})$; that is $\bar{b}^{(m+1)}_{tot}(j) = \bar{b}^{(m)}_{tot}(j) + \bar{b}^{(m+1)}(j - u_{m+1})$. We now show that (4) still holds. For any two phases, j and k, in the same chain, compare the values of $\bar{b}^{(m+1)}_{tot}(j)$ and $\bar{b}^{(m+1)}_{tot}(k)$. The only possibilities for the partial sum increments are:

1. $\bar{b}^{(m+1)}(j - u_{m+1}) = \bar{b}^{(m+1)}(k - u_{m+1}) = B^{(m+1)}_{max}$ (or $P^{(m+1)}_{max}$). Thus, $\bar{b}^{(m+1)}_{tot}(j)$ and $\bar{b}^{(m+1)}_{tot}(k)$ are both increased by $B^{(m+1)}_{max}$ (or $P^{(m+1)}_{max}$) from the previous step, and their difference remains the same.

2. $\bar{b}^{(m+1)}(j - u_{m+1}) = I^{(m+1)}_{max}$ and $\bar{b}^{(m+1)}(k - u_{m+1}) = P^{(m+1)}_{max}$ (or vice versa). Since $(j - u_{m+1})$th frame is an I frame, $j = u_{m+1}$. But u_{m+1} is the minimal-rate phase. By (16), $\bar{b}^{(m)}_{tot}(j) \leq \bar{b}^{(m)}_{tot}(k)$. Using (4) and the fact that $P^{(m+1)}_{max} \leq I^{(m+1)}_{max} \leq A$,

$$\begin{aligned} \left|\bar{b}^{(m+1)}_{tot}(j) - \bar{b}^{(m+1)}_{tot}(k)\right| &= \left|(\bar{b}^{(m)}_{tot}(j) - \bar{b}^{(m)}_{tot}(k)) + (I^{(m+1)}_{max} - P^{(m+1)}_{max})\right| \\ &\leq \max\{\bar{b}^{(m)}_{tot}(k) - \bar{b}^{(m)}_{tot}(j), I^{(m+1)}_{max} - P^{(m+1)}_{max}\} \leq A \end{aligned}$$

By induction, the lemma is proved. □

B Proof of Lemma 4.2

(By induction on the number of steps). When one stream is scheduled, the lemma clearly holds. Suppose (5) holds after the mth step. After the $(m+1)$th step, compare the values of $\bar{b}^{(m+1)}_{tot}(j)$ and $\bar{b}^{(m+1)}_{tot}(k)$. If j and k are in the same chain, by Lemma 4.1, their partial sum difference is at most A, and (5) holds. If j and k are not in the same chain, but are incremented by the same value, still (5) holds. The remaining possibility is that j is in the same chain as u_{m+1}, with an increment of either $I^{(m+1)}_{max}$ or $P^{(m+1)}_{max}$, and k is in a different chain, with an increment of $B^{(m+1)}_{max}$ (or vice versa). We consider the case of $I^{(m+1)}_{max}$ (the case of an increment of $P^{(m+1)}_{max}$ is similar).

1. $\bar{b}^{(m)}_{tot}(j) \leq \bar{b}^{(m)}_{tot}(k)$. From (5),

$$\begin{aligned} \left|\bar{b}^{(m)}_{tot}(j) - \bar{b}^{(m)}_{tot}(k)\right| &= \left|(\bar{b}^{(m)}_{tot}(j) - \bar{b}^{(m)}_{tot}(k)) + (I^{(m+1)}_{max} - B^{(m+1)}_{max})\right| \\ &\leq \max\{\bar{b}^{(m)}_{tot}(k) - \bar{b}^{(m)}_{tot}(j), I^{(m+1)}_{max} - B^{(m+1)}_{max}\} \leq 2A \end{aligned}$$

2. $\bar{b}_{tot}^{(m)}(j) > \bar{b}_{tot}^{(m)}(k)$. From (16), we have $\bar{b}_{tot}^{(m)}(j) > \bar{b}_{tot}^{(m)}(k) \geq \bar{b}_{tot}^{(m)}(u_{m+1})$. Since j and u_{m+1} are in the same chain, the difference between $\bar{b}_{tot}^{(m)}(j)$ and $\bar{b}_{tot}^{(m)}(u_{m+1})$ is at most A by (4). Thus, the difference between $\bar{b}_{tot}^{(m)}(j)$ and $\bar{b}_{tot}^{(m)}(k)$ is bounded by A.

$$\begin{aligned} \left|\bar{b}_{tot}^{(m+1)}(j) - \bar{b}_{tot}^{(m+1)}(k)\right| &= \left|(\bar{b}_{tot}^{(m)}(j) - \bar{b}_{tot}^{(m)}(k)) + (I_{max}^{(m+1)} - B_{max}^{(m+1)})\right| \\ &\leq (\bar{b}_{tot}^{(m)}(j) - \bar{b}_{tot}^{(m)}(k)) + (I_{max}^{(m+1)} - B_{max}^{(m+1)}) \leq 2A \end{aligned}$$

Therefore, (5) holds for the $(m+1)$th step. By induction, the assertion is proved. □

References

[1] K. C. Almeroth and M. H. Ammar. The use of multicast delivery to provide a scalable and interactive video-on-demand service. *IEEE Journal on Selected Areas in Communications*, 14(6):1110–1122, Aug. 1996.

[2] W. Feng, F. Jahanian, and S. Sechrest. An optimal bandwidth allocation strategy for the delivery of compressed prerecorded video. Technical Report CSE-TR260-95, Department of EECS, University of Michigan, Ann Arbor, 1995.

[3] W. Feng and S. Sechrest. Smoothing and buffering for the delivery of prerecorded compressed video. In *IS&T/SPIE Multimedia Computing and Networking*, pages 234–244, Feb. 1995.

[4] M. W. Garrett and M. Vetterli. Congestion control strategies for packet video. In *Proc. of Fourth Int. Workshop on Packet Video*, Aug. 1991. Kyoto, Japan.

[5] M. W. Garrett and W. Willinger. Analysis, modeling, and generation of self-similar VBR video traffic. In *Proceedings of the SIGCOMM'94 Conference (Computer Communications Review)*, pages 269–280, Sept. 1994.

[6] M. Graf. VBR video over ATM: Reducing network resource requirements through endsystem traffic shaping. In *Proc. of IEEE INFOCOM '97*, pages 48–57, 1997.

[7] C. Huang, M. Devetsikiotis, I. Lambadaris, and A. Kaye. Modeling and simulation of self-similar variable bit rate compressed video: A unified approach. In *Proc. of SIGCOMM '95*, 1995.

[8] ISO/MPEG. ISO CD11172-2: Coding of moving pictures and associated audio for digital storage media at up to about 1.5 mbits/s, Nov. 1991.

[9] ISO/MPEG II. ISO CD11172-2: Coding of moving pictures and associated audio, Dec. 1992.

[10] E. W. Knightly, D. Wrege, J. Liebeherr, and H. Zhang. Fundamental limits and tradeoffs of providing deterministic guarantees to VBR video traffic. In *Proc. of the ACM SIGMETRICS/PERFORMANCE '95 Conference*, pages 98–107, May 1995.

[11] E. W. Knightly and H. Zhang. Traffic characterization and switch utilization using a deterministic bounding interval dependent traffic model. In *Proc. of IEEE INFOCOM '95*, pages 1137–1145, 1995.

[12] M. Krunz, R. Sass, and H. Hughes. Statistical characteristics and multiplexing of MPEG streams. In *Proc. of the IEEE INFOCOM '95 Conference*, pages 455–462, Boston, Apr. 1995.

[13] M. Krunz and S. K. Tripathi. Impact of video scheduling on bandwidth allocation for multiplexed MPEG streams. *Multimedia Systems Journal*, Nov. 1997. (to appear).

[14] S. S. Lam, S. Chow, and D. K. Y. Yau. An algorithm for lossless smoothing for MPEG video. In *Proceedings of the ACM SIGCOMM '94 Conference*, Aug. 1994.

[15] V. O. K. Li, W. Liao, X. Qiu, and E. Wong. Performance model of interactive video-on-demand systems. *IEEE Journal on Selected Areas in Communications*, 14(6):1099–1109, Aug. 1996.

[16] J. M. McManus and K. W. Ross. Video-on-demand over ATM: Constant-rate transmission and transport. *IEEE Journal on Selected Areas in Communications*, 14(6):1087–1098, Aug. 1996.

[17] J.-P. Nussbaumer, B. V. Patel, F. Schaffa, and J. P. G. Sterbenz. Networking requirements for interactive video on demand. *IEEE Journal on Selected Areas in Communications*, 13(5):779–787, 1995.

[18] T. Ott, T. Lakshman, and A. Tabatabai. A scheme for smoothing delay-sensitive traffic offered to ATM networks. In *Proc. of IEEE INFOCOM '92*, pages 776–785, May 1992.

[19] P. V. Rangan, H. M. Vin, and S. Ramanathan. Designing an on-demand multimedia service. *IEEE Communications Magazine*, pages 56–64, July 1992.

[20] A. R. Reibman and A. W. Berger. Traffic descriptors for VBR video teleconferencing over ATM networks. *IEEE/ACM Transactions on Networking*, 3(3):329–339, June 1995.

[21] D. J. Reininger, D. Raychaudhuri, and J. Y. Hui. Bandwidth renegotiation for VBR video over ATM networks. *IEEE Journal on Selected Areas in Communications*, 14(6):1076–1086, Aug. 1996.

[22] M. Reisslein and K. W. Ross. Join-the-Shortest-Queue prefetching for VBR video on demand. (technical report), University of Pennsylvania, Department of Systems Engineering, 1997.

[23] J. D. Salehi, Z.-L. Zhang, J. F. Kurose, and D. Towsley. Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing. In *Proc. of the ACM SIGMETRICS/PERFORMANCE '96 Conference*, May 1996.

[24] S. SampathKumar, S. Ramanathan, and P. V. Rangan. Technologies for distribution of interactive multimedia to residential subscribers. In *Proceedings of the 1st Int. Workshop on Community Networking: Integrated Multimedia Services to the Home*, pages 151–160, July 1994.

[25] N. Shroff and M. Schwartz. Video modeling within networks using deterministic smoothing at the source. In *Proceedings of IEEE INFOCOM '94*, volume 1, pages 342–349, 1994.