Smoothing Variable-Bit-Rate Video in an Internetwork

Jennifer Rexford and Don Towsley

***Abstract*— The burstiness of compressed video complicates the provisioning of network resources for emerging multimedia services. For stored video applications, the server can smooth the variable-bit-rate stream by transmitting frames into the client playback buffer in advance of each burst. Drawing on prior knowledge of the frame lengths and client buffer size, such bandwidth smoothing techniques can minimize the peak and variability of the rate requirements while avoiding underflow and overflow of the playback buffer. However, in an internetworking environment, a single service provider typically does not control the entire path from the stored-video server to the client buffer. This paper presents efficient techniques for transmitting variable-bit-rate video across a portion of the route, from an ingress node to an egress node. We develop efficient techniques for minimizing the network bandwidth requirements by characterizing how the peak transmission rate varies as a function of the playback delay and the buffer allocation at the two nodes. Drawing on these results, we present an efficient algorithm for minimizing both the playback delay and the buffer allocation, subject to a constraint on the peak transmission rate. We then describe how to compute an optimal transmission schedule for a sequence of nodes by solving a collection of independent single-link problems, and show that the optimal resource allocation places all buffers at the ingress and egress nodes. Experiments with motion-JPEG and MPEG traces show the interplay between buffer space, playback delay, and bandwidth requirements for a collection of full-length video traces.**

Keywords: **variable-bit-rate video, bandwidth smoothing, majorization, prefetching, internetwork**

1 Introduction

The emergence of high-speed internetworks facilitates a wide range of new multimedia applications, such as distance learning and entertainment services, that rely on the efficient transfer of compressed video. However, constant-quality compressed video typically exhibits significant burstiness on multiple time scales, due to the frame structure of the compression algorithm as well as natural variations within and between scenes [1–5]. The presence of this burstiness complicates the effort to allocate network resources to ensure continuous playback of the video at the client site. To avoid excessive loss and delay in the network, service providers can provision link and buffer resources to accommodate the variable bandwidth requirements. However, the high peak and variability of the transmission rates substantially increases the network resources required to transfer the video.

J. Rexford is with the Networking and Distributed Systems Research center at AT&T Labs – Research in Florham Park, New Jersey. D. Towsley is with the Department of Computer Science at the University of Massachusetts in Amherst, Massachusetts; part of his work was performed while on sabatical at AT&T Labs – Research.

To reduce the end-to-end resource requirements, a stored-video server can smooth the outgoing stream through workahead transmission into the client playback buffer. By initiating transmission early, the server can send large frames at a slower rate without disrupting the client application; the client system can then retrieve, decode, and display frames at the stream frame rate. Drawing on *a priori* knowledge of the frame lengths and client buffer size, *bandwidth smoothing* algorithms [6–14] generate a server transmission schedule that consists of a sequence of constant-bit-rate runs that avoid underflow and overflow of the playback buffer. With a modest client buffer size, effective smoothing algorithms can produce schedules with fairly small numbers of runs and a significant reduction in both the peak and variability of the transmission rates, as discussed in the overview in Section 2. Smoothing substantially reduces network resource requirements under several different network service models, including peak-rate reservation, bandwidth renegotiation, and effective bandwidth [9].

Previous work on bandwidth smoothing has focused on video-on-demand servers that store the entire sequence of frames and can directly coordinate access to the client playback buffer. However, in a realistic internetworking environment, a single service provider may not control the entire path from the stored video at the server, through the communication network, to the playback buffer at the client site. Instead, each network service provider has dominion over a portion of the route that may or may not include the server and client locations. Still, bandwidth smoothing can be extremely effective at reducing the overhead of transporting variable-bit-rate video within a subset of the route through the network. To minimize resource requirements in an internetworking environment, this paper generalizes the bandwidth smoothing model to a sequence of nodes along part of the path from the stored-video server to the playback buffer [15]. These nodes can represent individual switches or routers in the domain of a single network service provider, or dedicated proxy servers that perform bandwidth smoothing. For example, bandwidth smoothing may be performed at network peering points or at the gateway to the client access network.

Starting with a simple model of an ingress node that performs workahead transmission into an egress node, Section 3 describes how to compute the buffer allocation and start-up delay that allow bandwidth smoothing to achieve the maximum reduction in the peak transmission rate. By characterizing how the peak rate varies as a function of the buffer allocation and start-up delay, we show that a simple binary-search algorithm is sufficient for finding the optimal configuration. The proofs, presented in Appendix A, draw on the majorization smoothing algorithm introduced in [9] and additional properties established in [16]. Section 4 then describes how to minimize the buffer space and start-up latency subject to a constraint on the peak rate in the transmission schedule, drawing on earlier work [12,17] on stored video. We show that it is possible to jointly minimize the buffer space and start-up delay using an algorithm with complexity that is linear in the number of frames in the video stream. For the common case where the ingress node receives the original unsmoothed stream, we show that the optimal buffer allocation divides the buffer space evenly between the ingress and egress nodes.

In Section 5, we turn our attention to the tandem smoothing model with a sequence of n nodes, and describe how to efficiently compute a smooth transmission schedule for each link in the sequence. In concurrent work, the author of [18] describes a similar tandem smoothing problem and present heuristics for iteratively computing the n transmission schedules. We show that optimal transmission schedules can be computed by solving n independent instances of the single-link problem. Then, we show that an optimal allocation of resources places all buffers at the ingress and egress nodes, allowing us to apply the results from Section 3 and Section 4 to the more general tandem model. These results draw on the underlying properties of the majorization smoothing algorithm. In Section 6, we describe the results of experiments with full-length motion-JPEG and MPEG video traces. These experiments highlight the performance of the smoothing algorithm and provide guidelines for allocating buffer space and start-up latency. Section 7 summarizes the paper and discusses future research directions.

2 Smoothing Stored Video

A multimedia server can substantially reduce the rate requirements for prerecorded video by transmitting frames into the client playback buffer in advance of each burst. Previous work on bandwidth smoothing typically assumes an *infinite-source single-link* model, where the server stores the entire video stream and generates a single transmission schedule based on the overflow and underflow constraints on the client buffer.

2.1 Smoothing Constraints

As an example of constraints on bandwidth smoothing, consider a server that stores an entire sequence of N video frames with sizes $\ell_1, \ell_2, \ldots, \ell_N$ for transmission to a client with an x-byte playback buffer. For continuous playback of the video stream, the server must always transmit enough data to avoid buffer underflow, where $L_k = \sum_{i=1}^{k} \ell_i$ indicates the amount of data consumed at the client by time k, where $k = 0, 1, \ldots, N$. To prevent overflow of the playback buffer, the client should not receive more than $U_k = L_k + x$ bytes by time k, as shown in Figure 1. Any valid

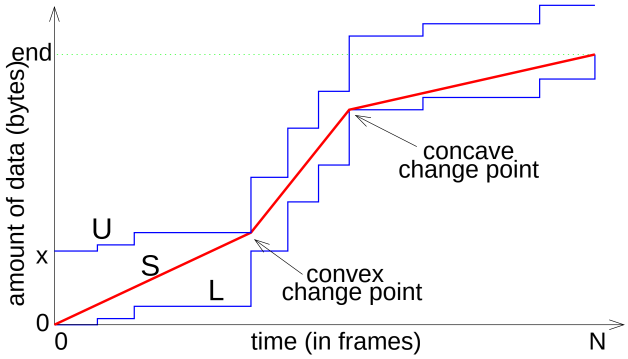


Figure 1: **Constraint Curves:** The sample transmission plan $\mathbf{S}$ is a non-decreasing curve consisting of three linear runs that stay between the upper ($\mathbf{U}$) and lower ($\mathbf{L}$) smoothing constraints. The second run increases the transmission rate to avoid an eventual buffer underflow, while the third run decreases the transmission rate.

server transmission plan should stay between these vertically equidistant functions. Workahead transmission occurs whenever the schedule lies above the underflow curve. To permit smoothing at the beginning of the video, the model can introduce a playback delay by shifting the underflow and overflow curves to the right by w time units, resulting in $L_k = \sum_{i=1}^{k} \ell_{i-w}$ for $k = 0, 1, \ldots, N+w$, where $\ell_j = 0$ for $j \leq 0$.

More generally, the vectors $\mathbf{L} = (L_0, \ldots, L_N)$ and $\mathbf{U} = (U_0, \ldots, U_N)$ can represent a wide variety of constraints on the transmission of data from the server site. To formalize the bandwidth smoothing problem, suppose that we are given three vectors, $\mathbf{L}, \mathbf{U}, \mathbf{S} \in I\!R^{N+1}$, such that $\mathbf{L} \leq \mathbf{U}$. We say that $\mathbf{S}$ is *feasible* with respect to $\mathbf{L}$ and $\mathbf{U}$ if and only if $S_0 = L_0$, $S_N = L_N$, and $\mathbf{L} \leq \mathbf{S} \leq \mathbf{U}$. The vector $\mathbf{R}(\mathbf{S}) = (S_1 - S_0, \ldots, S_N - S_{N-1})$ represents the schedule as a sequence of transmission rates, where R_k denotes the amount of data transferred during the time interval $[k-1, k)$, $k = 1, 2, \ldots, N$. When there is no possibility of confusion, we write the schedule $\mathbf{S}(\mathbf{L}, \mathbf{U})$ as $\mathbf{S}$ and its rate vector $\mathbf{R}(\mathbf{S})$ as $\mathbf{R}$. If $R_{k+1} \neq R_k$, then k is said to be a *change point* in the vector $\mathbf{S}$. Moreover, it is said to be a *convex* change point if $R_{k+1} > R_k$ and a *concave* change point if $R_{k+1} < R_k$. The example in Figure 1 has a convex change point, followed by a concave change point. Note that the run responsible for the peak transmission rate involves a bandwidth increase (convex change point) followed by a bandwidth decrease (concave change point).

2.2 Majorization Schedule

The constraints $\mathbf{L}$ and $\mathbf{U}$ typically result in multiple feasible schedules, each with different performance properties. Several smoothing algorithms compute schedules that minimize the peak rate $\max_k\{R_k\}$ to reduce the bandwidth requirements for transmitting the video stream. Minimizing other moments of the transmission rates can also reduce the video's empirical effective bandwidth [9,19], allowing the server and the network to multiplex a larger number of streams. The theory of majorization [20] offers a useful way to compare the smoothness properties of different schedules. For $X, Y \in I\!R^K$, let $X_{[i]}$ ($Y_{[i]}$) be the i-th largest component of X (Y). Vector Y *majorizes* X ($X \prec Y$) if $\sum_{i=1}^{k} X_{[i]} \leq \sum_{i=1}^{k} Y_{[i]}$ for $k = 1, \ldots, K-1$ and $\sum_{i=1}^{K} X_i \leq \sum_{i=1}^{K} Y_i$; intuitively, the elements of vector X are more "evenly distributed" than the elements of Y. The majorization relation $X \prec Y$ implies that $\phi(X) \leq \phi(Y)$ for any *Schur-convex* function $\phi : I\!R^K \rightarrow I\!R$. In addition to common smoothness metrics such as the peak and variance, this class of functions includes $\phi(X) = \sum_{i=1}^{K} f(X_i)$ for all convex $f : I\!R^K \rightarrow I\!R$.

Applying the theory of majorization to the bandwidth smoothing problem, there exists an $O(N)$ algorithm [9] which finds a vector $\mathbf{S}^* \in \mathcal{S}(\mathbf{L}, \mathbf{U})$ such that $\mathbf{R}(\mathbf{S}^*) \prec \mathbf{R}(\mathbf{S})$ for all $\mathbf{S} \in \mathcal{S}(\mathbf{L}, \mathbf{U})$. In constructing this majorization schedule $\mathbf{S}^*$, the algorithm computes linear trajectories that extend as far as possible between the lower and upper constraints, ultimately encountering both the $\mathbf{L}$ and $\mathbf{U}$ curves. Whenever a bandwidth increase (decrease) is necessary to avoid crossing the $\mathbf{L}$ ($\mathbf{U}$) curve, the algorithm starts a new trajectory as early as possible to limit the size of the change. Hence, change points always reside on the $\mathbf{L}$ or $\mathbf{U}$ curves, with $S_k^* = U_k$ at convex points (rate increases) and $S_k^* = L_k$ at concave points (rate decreases) [9], as in the example in Figure 1. The majorization algorithm produces a sequence of constant-bit-rate runs that substantially reduces the peak bandwidth requirement for transmitting stored video, as shown by the graphs in Figure 2 taken from [21].

2.3 Domination and Refinement Results

The majorization algorithm has several properties that facilitate comparisons between different instances of the bandwidth smoothing problem; in particular, the following *domination* and *refinement* lemmas [16] play an important role in analyzing the models introduced in Section 5 and Section 3.2. The domination result compares the $\mathbf{L}$ and $\mathbf{U}$ constraints of two smoothing configurations:

Lemma 2.1 *Let $\mathbf{L}_1 \leq \mathbf{U}_1$ and $\mathbf{L}_2 \leq \mathbf{U}_2$ be such that $\mathbf{L}_1 \leq \mathbf{L}_2$ and $\mathbf{U}_1 \leq \mathbf{U}_2$. Then $\mathbf{S}_1^* \leq \mathbf{S}_2^*$. Moreover, if $\mathbf{L}_1 = \mathbf{L}_2$, and k is a concave change point in $\mathbf{S}_2^*$, it is a concave change point in $\mathbf{S}_1^*$. Similarly, if $\mathbf{U}_1 = \mathbf{U}_2$, and k is a convex change point in $\mathbf{S}_1^*$, then it is a convex change point in $\mathbf{S}_2^*$.*

Proof. This was established in [16]. The proof of $\mathbf{S}_1^* \leq \mathbf{S}_2^*$ is by contradiction. If the schedule $\mathbf{S}_2^*$ dips below $\mathbf{S}_1^*$ at some time k, then the schedule must eventually have a rate increase. This rate increase would occur at a convex change point, where $\mathbf{S}_2^*$ hits $\mathbf{U}_2$. But, $\mathbf{U}_2 \geq \mathbf{U}_1 \geq \mathbf{S}_1^*$, which violates the assumption that $\mathbf{S}_2^*$ lies below $\mathbf{S}_1^*$. Hence, $\mathbf{S}_1^* \leq \mathbf{S}_2^*$. If $\mathbf{L}_1 = \mathbf{L}_2$ and $\mathbf{S}_2^*$ has a convex change point at time k (hitting $\mathbf{L}_2$), then $\mathbf{S}_1^* \leq \mathbf{S}_2^*$ implies that $\mathbf{S}_1^*$ also has a convex change point at time k (hitting $\mathbf{L}_1$). Similarly, if $\mathbf{U}_1 = \mathbf{U}_2$ and $\mathbf{S}_1^*$ has a concave change point at time k (hitting $\mathbf{U}_1$), then $\mathbf{S}_2^* \geq \mathbf{S}_1^*$ implies that $\mathbf{S}_2^*$ also has a concave change point at time k (hitting $\mathbf{U}_2$). ∎

Focusing on one set of constraints $(\mathbf{L}, \mathbf{U})$, the domination lemma implies that raising the $\mathbf{U}$ curve causes *concave* change points to gradually disappear in the corresponding majorization schedules; similarly, lowering the $\mathbf{L}$ curve causes *convex* change points to gradually disappear. These change points disappear at a series of critical buffer sizes.

Drawing on this result, the refinement property characterizes how the majorization schedule changes with the size of the playback buffer. With a slight rephrasing of the result in [16], we have:

Lemma 2.2 *Let $\mathbf{U}_1 = \mathbf{L}_1 + vec(b_1)$ and $\mathbf{U}_2 = \mathbf{L}_2 + vec(b_2)$ be such that $b_2 \geq b_1 > 0$, where $vec(b_i)$ is a vector whose components are equal to b_i. If $\mathbf{L}_1 = \mathbf{L}_2$, then the change points in S_1^* are a superset of the change points in S_2^*.*

Proof. This was established in [16] by invoking Lemma 2.1 twice. Applying the lemma to the original pair of smoothing constraints shows the result for concave change points; applying the lemma a second time, after raising the $\mathbf{L}_1$ and $\mathbf{U}_1$ curves by $b_2 - b_1$, implies the result for convex change points. ∎

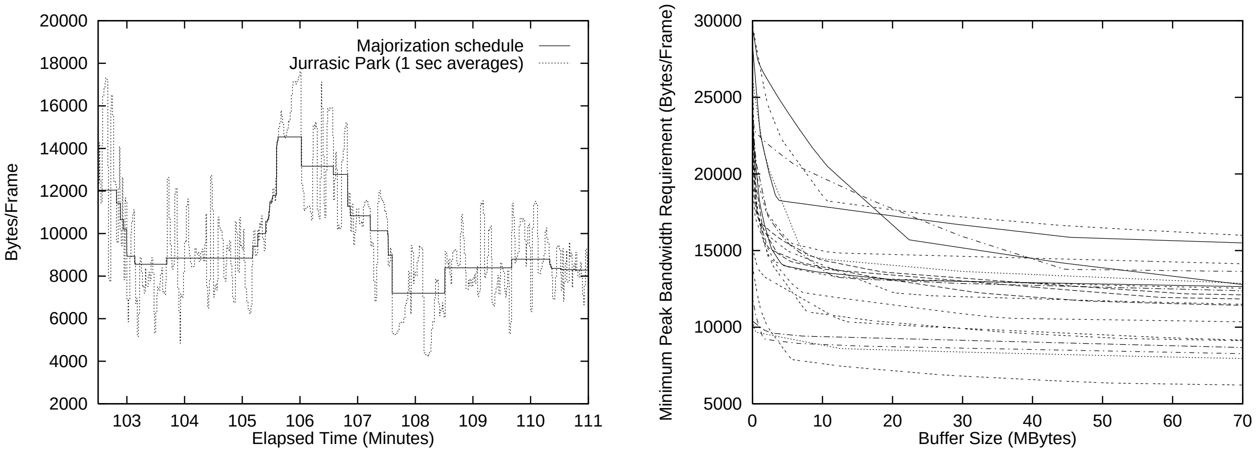
Capitalizing on the majorization algorithm and its properties, the next section analyzes a more general model where the ingress node receives the video stream and performs workahead transmission into the buffer at the egress node.

3 Minimizing the Peak Rate

We begin by considering bandwidth smoothing over the route between an ingress node and an egress node, and determine how to compute the buffer allocation and start-up delay that allow the the maximum reduction in the peak transmission rate. By characterizing the shape of the peak-rate curve, we show that a simple binary-search procedure is sufficient to determine the optimal allocation of buffer space between the ingress and egress nodes. A similar result holds for the selection of the start-up latency. The section ends with a brief discussion of practical issues, including propagation delay, jitter, and limited knowledge of frame sizes.

3.1 Single-Link Model

The single-link model consists of an ingress node that performs workahead transmission of the arriving video stream into the egress node, as shown in Figure 3. The incoming stream is characterized by an *arrival vector* $\mathbf{A} = (A_0, \ldots, A_N)$, where A_k corresponds to the amount of data which has arrived at the ingress node by time $k = 0, 1, \ldots, N$, where $A_k \geq A_{k-1}$, $k = 1, \ldots, N$. The stream has a *playout vector* $\mathbf{D} = (D_0, \ldots, D_N)$, where D_k corresponds to the amount of data that must be removed from the x-bit egress buffer by time $k = 0, 1, 2, \ldots, N$, where $D_0 = 0$ and $D_k \geq D_{k-1}$, $k = 1, \ldots, N$. For a valid playout vector for the arriving stream, we assume that $A_k \geq D_k$ for $k = 0, 1, \ldots, N$, with $A_N = D_N$ at the end of the transfer. At time $k =$



(a) Bandwidth plan with 1-megabyte buffer (b) Peak bandwidth vs. buffer size x

Figure 2: **Bandwidth Smoothing Example:** The first graph plots a portion of the majorization schedule for a motion-JPEG encoding of the movie *Jurassic Park*, while the second graph plots the peak bandwidth as a function of the buffer size x for twenty full-length motion-JPEG traces.

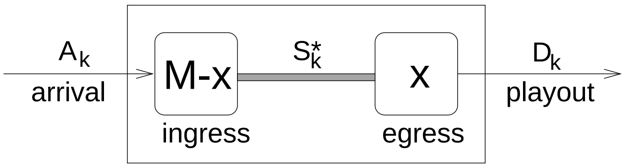


Figure 3: **Single-Link Model:** This figure shows a single-link system with an x-bit egress buffer and an $(M-x)$-bit ingress buffer, with an arrival vector $\mathbf{A}$, a playout vector $\mathbf{D}$, and the optimal transmission vector $\mathbf{S}^*$.

$0, 1, \ldots, N$, a total of $A_k - D_k$ bits are distributed between the x-bit egress buffer and the $(M-x)$-bit ingress buffer, requiring $A_0 \leq M - x$ and $A_k - D_k \leq M$ for $k = 0, 1, \ldots, N$.

Any valid transmission schedule $\mathbf{S}$ must avoid underflow and overflow of the egress buffer:

$$\mathbf{D} \leq \mathbf{S} \leq \mathbf{D} + vec(x),$$

and underflow and overflow of the ingress buffer:

$$\mathbf{A} - vec(M-x) \leq \mathbf{S} \leq \mathbf{A}.$$

Let $\mathbf{S}^*(\mathbf{A}, \mathbf{D}, (M-x, x))$ denote the majorization schedule, subject to these constraints. When there is no ambiguity, we refer to the majorization schedule as $S^*(x)$. The schedule has a peak rate of $p(x) = \max\{\mathbf{R}^*(x)\}$. In the next subsection, we determine how to compute the value of x that minimizes $p(x)$ by characterizing the shape of the $p(x)$ curve. Then, we follow a similar approach to compute the start-up delay that minimizes the peak rate, where a playout delay of w shifts $\mathbf{D}$ to the right by w time units.

3.2 Buffer Allocation

We first consider how to determine the optimal division of the M-bit buffer budget between the ingress and egress nodes, with the goal of minimizing the peak bandwidth of the smooth transmission schedule. To characterize how $p(x)$ varies with x, we parameterize the smoothing constraints:

$$\begin{aligned}
\mathbf{L}_1(x) &= \mathbf{A} + vec(x - M) \\
\mathbf{U}_1(x) &= \mathbf{D} + vec(x) \\
\mathbf{L}_2(x) &= \mathbf{D} \\
\mathbf{U}_2(x) &= \mathbf{A}.
\end{aligned}$$

Recall that $\mathbf{S}^*(x)$ is the majorization schedule associated with $\mathcal{S}(\max\{\mathbf{L}_1(x), \mathbf{L}_2(x)\}, \min\{\mathbf{U}_1(x), \mathbf{U}_2(x)\}, (M-x, x))$. Note that as more buffer space is allocated to the egress node, the pair $(\mathbf{L}_1, \mathbf{U}_1)$ rises relative to the pair $(\mathbf{L}_2, \mathbf{U}_2)$, which remains stationary.

By characterizing how the schedule $\mathbf{S}^*(x)$ changes with x, we can show that

Lemma 3.1 *The peak rate, $p(x, M)$, of the single-link system is a piecewise-linear, convex function of x.*

The proof, which extends the results in Section 2.3, is presented in Appendix A. The main consequence of the lemma is that the peak-rate curve has one of the shapes shown in Figure 4, where the slope of the curve increases in x. Consequently, a binary search is sufficient to determine a value of x that minimizes $p(x, M)$ for a given value of M. Optimizing x requires at most $\log M$ invocations of the $O(N)$ majorization smoothing algorithm, resulting in a worst-case complexity of $O(N \log M)$ to determine an optimal value x^* and an optimal transmission schedule $S^*(x^*)$.

To motivate the theoretical result, consider the simpler case of stored video, where $A_i = A_N$ for $k = 0, 1, \ldots N$ and the ingress buffer is large enough to store the entire video (i.e., A_N bits). The peak bandwidth decreases as the egress buffer size x increases, with diminishing returns for larger buffer sizes, as shown for several sample video traces in Figure 2(b). Typically, the peak bandwidth in a trans-

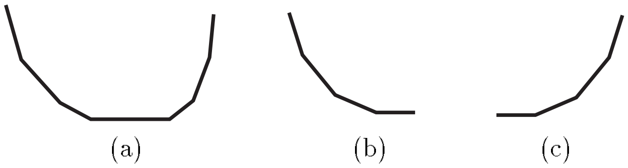


Figure 4: **Shape of Peak-Rate Curve:** As a function of the buffer allocation x, the peak-rate curve $p(x)$ has (a) shrinking and growing phases; (b) shrinking phase only; (c) growing phase only.

mission plan is determined by a region of large frames in a small portion of the stream (e.g., the second run in Figure 1). The upper constraint curve $\mathbf{U}$ rises as the egress buffer size x increases, causing a gradual decrease in the slope of the peak-rate run. In fact, the peak rate decreases in proportion to the increase in x, resulting in a linear segment in the peak-rate curve. These linear segments are clearly visible in Figure 2(b). Appealing to the domination result in Lemma 2.1, the buffer size eventually grows large enough for the convex change point to disappear, causing the peak rate run to occur in a different part of the video. This begins a new linear segment in the peak-rate curve. The diminishing returns account for the convexity of the peak-rate curve. The proof of Lemma 3.1 in Appendix A formalizes and generalizes this argument. With buffer constraints at both the ingress and egress nodes, the peak-rate curves typically have a cup shape, as shown in Figure 4(a).

3.3 Start-Up Delay

In addition to allocating buffer space to smoothing, a video service can minimize the bandwidth requirements of the variable-bit-rate stream by introducing a start-up latency of w at the egress node. Delaying playout by $w > 0$ time units effectively shifts the playout curve at the egress node, as discussed in Section 2.1. Hence, we define the vectors $\mathbf{D}(w)$ and $\mathbf{A}(w)$ to be

$$D_k(w) = \begin{cases} 0, & k = 0, \ldots, w-1, \\ D_{k-w}, & k = w, \ldots N+w \end{cases}$$

$$A_k(w) = \begin{cases} A_k, & k = 0, \ldots, N, \\ A_N, & k = N+1, \ldots N+w. \end{cases}$$

Given a buffer allocation $(M-x, x)$, the startup latency w is constrained to be less than

$$w_{\max} \equiv \min\left\{ w \mid \max_{0 \le k \le N+w} \{A_k(w) - D_k(w)\} \ge M \right\};$$

otherwise some windows of w frames would overflow the combined buffer space. Let $p(w)$ denote the peak rate as a function of startup latency for given arrival and playout vectors and buffer allocation. Define w^* to be startup latency that minimizes the peak rate,

$$w^* = \arg \min_{0 \le w \le w_{\max}} p(w).$$

To characterize how $p(w)$ varies with w, we parameterize the smoothing constraints:

$$\begin{aligned} \mathbf{L}_1(w) &= \mathbf{D}(w) \\ \mathbf{U}_1(w) &= \mathbf{D}(w) + vec(x) \\ \mathbf{L}_2(w) &= \mathbf{A} + vec(x - M) \\ \mathbf{U}_2(w) &= \mathbf{A}. \end{aligned}$$

As the start-up delay w increases, $(\mathbf{L}_1, \mathbf{U}_1)$ rise relative to the pair $(\mathbf{L}_2, \mathbf{U}_2)$, which remains stationary. Recall that $S^*(w)$ is the majorization schedule associated with $\mathcal{S}(\max\{\mathbf{L}_1(w), \mathbf{L}_2(w)\}, \min\{\mathbf{U}_1(w), \mathbf{U}_2(w)\}, (b_0, b_1))$. As in Section 3.2, we can show that

Lemma 3.2 *The peak rate, $p(w)$, of the single link system is a piecewise-linear convex function of w.*

The arguments in the proof of Lemma 3.1 in Appendix A are easily modified to apply here. Whereas the pair of vectors $\mathbf{L}_1(x)$ and $\mathbf{U}_1(x)$ in the buffer-allocation problem rise with respect to the other pair of vectors as x increases, $\mathbf{L}_1(w)$ and $\mathbf{U}_1(w)$ move to the right with respect to the pair of vectors $\mathbf{L}_2(w)$ and $\mathbf{U}_2(w)$. The points on the majorization schedule exhibit similar behavior under a latency change as under a buffer change.

Hence, the peak-rate curve $p(w)$ has one of the shapes shown in Figure 4. In the stored video case, increasing w always decreases the peak rate, since a longer start-up delay provides more opportunity for smoothing, though there are diminishing returns for longer delays. The diminishing returns account for the convexity of the peak-rate curve. When there is finite buffer space at both the ingress and egress nodes, the peak-rate curve typically has a cup shape, as in Figure 4(a). If the start-up delay is too short, the ingress node does not have sufficient time to perform workahead transmission into the egress buffer. On the other hand, if the start-up delay is too long, the ingress node must transmit aggressively to avoid overflow of the ingress buffer. The primary consequence of Lemma 3.2 is that a binary search can be used to determine the optimal startup latency w^* for a given buffer allocation. Given a maximum feasible start-up latency of $w_{\max}$, the optimization procedure has a worst-case complexity of $O(N \log w_{\max})$.

3.4 Practical Constraints

The smoothing model presumes prior knowledge of the arrival and playout vectors, and does not account for propagation delay and jitter. The basic model can be extended to incorporate these practical constraints. To account for a constant propagation delay of d time units, the playout curve $\mathbf{D}$ should be shifted to the left by d units, to ensure that each frame arrives at the egress node by its deadline. Delay jitter can be handled by making the smoothing constraints more conservative, based on the minimum and maximum number of bits that may arrive at any time [9]. For example, the overflow constraints can assume the maximum number of arriving bits at each time unit, while the underflow constraints can assume the minimum number of arriving bits at each unit of time.

In addition, the ingress node may not have full knowledge of the arrival vector, particularly in the case of live video. Experiments with motion-JPEG and MPEG traces show that having information about a small window of future frames is sufficient to approximate the performance of a scheme that has complete knowledge of frame sizes [22]. Our experiments show that relatively simple frame-size prediction schemes are effective [22]. To focus the discussion on the basic properties of the smoothing model, the remainder of the paper assumes that there is no propagation delay or jitter, and that the arrival vector $\mathbf{A}$ is known in advance.

4 Rate Constraints

The previous section described how to allocate buffer space and start-up delay to minimize the peak bandwidth in the transmission schedule. However, in many cases the network may impose a specific rate constraint, such as modem bandwidth, that drives the selection of the other smoothing parameters. This section presents efficient techniques for minimizing both the start-up delay and the buffer allocation, subject to an upper bound on the transmission rate between the ingress and egress nodes. Our approach is to consider greedy transmission schedules that send as late or as early as possible, subject to the rate constraint and the arrival and playout vectors. These greedy schedules can be used to determine the minimum sizes for the egress buffer (x), ingress buffer (y), and start-up delay (w).

4.1 Joint Minimization of Start-Up Delay and Egress/Ingress Buffer Size

Before investigating the general problem of rate-constrained smoothing on a single link, we discuss the important special case of stored video, where the ingress buffer stores the entire video in advance. For stored video with an ingress buffer of size $y \geq D_N$, the smoothing constraints simplify to $\mathbf{L}(x,w) = \mathbf{D}(w)$ and $\mathbf{U}(x,w) = \mathbf{D}(w) + vec(x)$, as in the example in Section 2.1 and Figure 1. In this context, it is possible to minimize both the start-up delay w and the egress buffer size x by generating a schedule $\mathbf{S}^{late}$ that transmits frames *as late as possible*, subject to the rate constraint r [12, 17]. Drawing on the playout curve $\mathbf{D}$, this algorithm, of complexity $O(N)$, starts from the last frame and sequences backwards to the beginning of the video; when necessary, the algorithm transmits frames at rate r, as shown in Figure 5. That is,

$$S_k^{late} = \begin{cases} D_N & k = N \\ \max\{S_{k+1}^{late} - r, D_k\} & k < N. \end{cases}$$

As a result, $\mathbf{S}^{late}$ closely follows the playout curve $\mathbf{D}$ except in regions with large frames, which require workahead transmission to avoid exceeding the rate constraint. The minimum buffer size x^* corresponds to the maximum distance that this schedule $\mathbf{S}^{late}$ lies above the lower constraint curve $\mathbf{L}$, while the minimum start-up latency w^* depends on the amount of workahead transmission required for the first frame in the video [12].

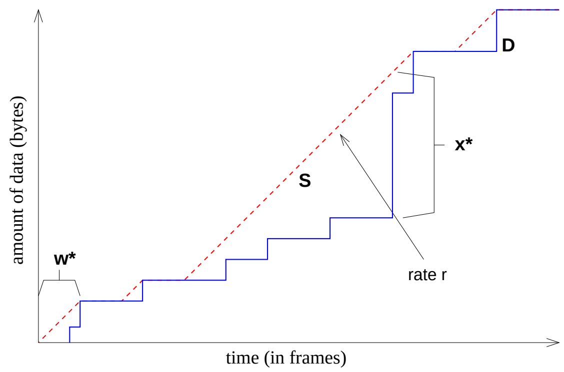


Figure 5: **Rate-Constrained Smoothing for Stored Video:** This figure shows a smoothing schedule that transmits data as late as possible, subject to a rate constraint r. No schedule that obeys the rate constraint r and the underflow constraint $\mathbf{D}$ can enter the region between $\mathbf{S}^{late}$ and $\mathbf{D}$.

The schedule that transmits as late as possible serves as a building block for minimizing start-up latency and buffer sizes in the general case, where $y \leq D_N$ and $A_k \leq D_N$ for all k. We begin by minimizing w and x without constraining the ingress buffer size y. This results in the constraint vectors $\mathbf{L}(x,w) = \mathbf{D}(w)$ and $\mathbf{U}(x,w) = \min\{\mathbf{D}(w) + vec(x), \mathbf{A}\}$. As in the case of stored video, the minimum buffer size x^* can be computed from the schedule $\mathbf{S}^{late}$ that transmits frames as late as possible, based on the playout curve $\mathbf{D}$; that is,

$$x^* = \max_k\{S_k^{late} - D_k\}.$$

However, the start-up latency w now depends on the maximum amount of workahead transmission necessary at *any* point in the schedule, relative to the arrival vector $\mathbf{A}$. Based on the schedule $\mathbf{S}^{late}$ that transmits data as late as possible, the minimum start-up latency is

$$w^* = \min\{w \mid A_k - S_{k-w}^{late} \geq 0\}$$

which can be determined in $O(N)$ time by sequencing through the vectors $\mathbf{A}$ and $\mathbf{S}^{late}$, independent of x. By transmitting frames as late as possible, this process jointly minimizes both x and w, since it is not possible to reduce either the egress buffer utilization or start-up latency by sending data any earlier than necessary.

A similar approach can be used to jointly minimize the *ingress* buffer size y and the start-up latency w without any constraint on the egress buffer size x by initially generating a schedule $\mathbf{S}^{early}$ that transmits frames as early as possible, subject to the rate constraint r and the arrival curve $\mathbf{A}$, with $A_0 = 0$. In particular,

$$S_k^{early} = \begin{cases} 0 & k = 0 \\ \min\{S_{k-1}^{early} + r, A_k\} & k > 0. \end{cases}$$

Based on the schedule $\mathbf{S}^{early}$, the optimum values of y and w are

$$\begin{aligned} y^* &= \max_k\{A_k - S_k^{early}\} \\ w^* &= \min\{w \,|\, S_k^{early} - D_{k-w} \geq 0\}. \end{aligned}$$

We observe that the optimizations based on $\mathbf{S}^{late}$ and the optimization based on $\mathbf{S}^{early}$ must achieve the same minimum start-up delay w^*. In the next subsection, we show how to minimize the ingress (egress) buffer, subject to jointly minimizing the egress (ingress) buffer and the start-up latency. Both approaches generate a feasible schedule in $O(N)$ time, subject to the rate constraint r, with the minimum start-up latency w^* and a minimum size for at least one of the buffers (x^* or y^*). Both approaches also minimize the *total* buffer allocation $M = x + y$. Also, in the special case of $\mathbf{A} \equiv \mathbf{D}$, it is possible to jointly minimize both x and y, with $x^* = y^*$. After determining the values for x, y, and w, the majorization algorithm can compute the optimally smooth transmission schedule.

4.2 Joint Minimization of Start-Up Delay and Total Buffer Allocation

After jointly minimizing the start-up latency and the egress (resp. ingress) buffer size, it is possible to minimize the ingress (resp. egress) buffer size based on the schedule $\mathbf{S}^{late}$ (resp. $\mathbf{S}^{early}$). More formally, let $y'(w)$ be the smallest ingress buffer size, given a start-up latency $w \geq w^*$ and the minimum egress buffer size x^*; similarly, let $x'(w)$ be the smallest egress buffer size, given a start-up latency $w \geq w^*$ and the minimum ingress buffer size y^*. We focus on the derivation of $y'(w)$ after determining x^*. While the egress buffer size x^* was determined from the $\mathbf{S}^{late}$ schedule subject to $\mathbf{L}(x,w) = \mathbf{D}(w)$, the ingress buffer size y' can be found by generating a schedule $\mathbf{S}^{early'}$ that sends as *early* as possible subject to $\mathbf{U} = \min\{\mathbf{D}(w) + vec(x^*), \mathbf{A}\}$. Consequently,

$$y'(w) = \max_k\{A_k - S_k^{early'}\}, \;\; w \geq w^*,$$

as shown in Figure 6(a). Similarly, x' can be determined from the schedule $\mathbf{S}^{late'}$ that transmits as late as possible subject to $\mathbf{L} = \max\{\mathbf{D}(w), \mathbf{A} - vec(y^*)\}$, resulting in

$$x'(w) = \max_k\{S_k^{late'} - D_k\}, \;\; w \geq w^*.$$

Based on these derivations, we can show that $x^* + y'(w) \equiv x'(w) + y^*$ for all $w \geq w^*$. The proof of this result draws on properties of the greedy schedules that transmit as late, or as early, as possible:

Lemma 4.1 *Given a rate constraint r, let S^{late_1} and S^{late_2} be the schedules that send as late as possible, subject to constraints $\mathbf{L}_1$ and $\mathbf{L}_2$, respectively. The schedule $\mathbf{S}^{late_{\max}}$ that sends as late as possible, subject to $\mathbf{L}_{\max} = \max\{\mathbf{L}_1, \mathbf{L}_2\}$, satisfies $\mathbf{S}^{late_{\max}} = \max\{\mathbf{S}^{late_1}, \mathbf{S}^{late_2}\}$. Similarly, the schedule $S^{early_{\min}}$ that sends as late as possible subject to $\mathbf{U}_{\min} = \min\{\mathbf{U}_1, \mathbf{U}_2\}$ satisfies $\mathbf{S}^{early_{\min}} = \min\{\mathbf{S}^{early_1}, \mathbf{S}^{early_2}\}$, where $\mathbf{S}^{early_1}$ and $\mathbf{S}^{early_2}$ are the schedules that send as early as possible, subject to constraints $\mathbf{U}_1$ and $\mathbf{U}_2$, respectively.*

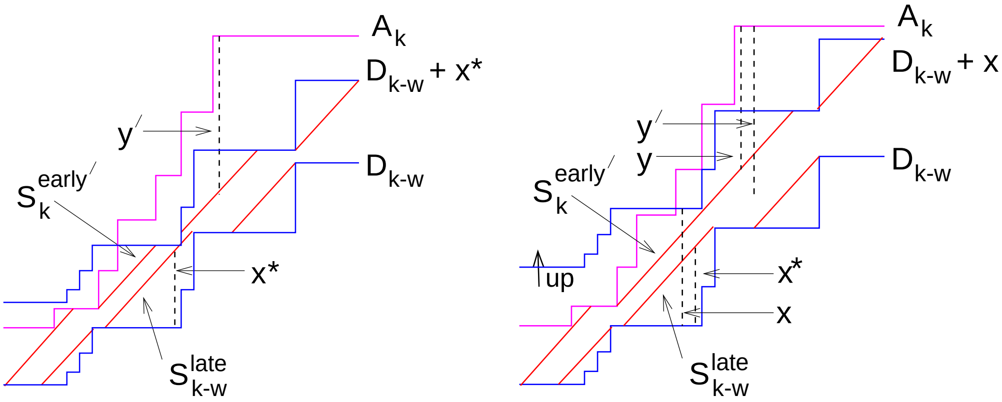
Proof: First, we observe that $\max\{\mathbf{S}^{late_1}, \mathbf{S}^{late_2}\}$ is a *feasible* rate-constrained schedule subject to the lower constraint $\mathbf{L}_{\max}$, since $\mathbf{S}^{late_1} \geq \mathbf{L}_1$ and $\mathbf{S}^{late_2} \geq \mathbf{L}_2$ imply that $\max\{\mathbf{S}^{late_1}, \mathbf{S}^{late_2}\} \geq \mathbf{L}_{\max}$. Next, we show by contradiction that no feasible schedule can send frames later than $\max\{\mathbf{S}^{late_1}, \mathbf{S}^{late_2}\}$. Assume that the schedule $\mathbf{S}^{late_{\max}}$ lies below $\max\{\mathbf{S}^{late_1}, \mathbf{S}^{late_2}\}$ at some time k. Without loss of generality, assume that $\mathbf{S}^{late_1}$ dominates $\mathbf{S}^{late_2}$ at time k (i.e., $S_k^{late_1} \geq S_k^{late_2}$). This implies that $S_k^{late_{\max}} < S_k^{late_1}$. But, this violates the definition of $\mathbf{S}^{late_1}$ as the schedule that transmits as late as possible subject to $\mathbf{L}_1$. Hence, $\mathbf{S}^{late_{\max}} = \max\{\mathbf{S}^{late_1}, \mathbf{S}^{late_2}\}$. A similar result holds for $\mathbf{S}^{early_{\min}}$. ∎

In effect, $\mathbf{S}^{late_1}$ and $\mathbf{S}^{late_2}$ each define an inadmissible region above $\mathbf{L}_1$ and $\mathbf{L}_2$, respectively, that cannot support any schedule with rate constraint r; a similar result holds for $\mathbf{S}^{early_1}$ and $\mathbf{S}^{early_2}$. Lemma 4.1 shows that, when there are multiple smoothing constraints, the inadmissible region is the *union* of the regions defined by each individual constraint.

Applying these results to $\mathbf{S}^{late'}$, we see that $S_k^{late'} = S_k^{late}$ in regions where $\mathbf{S}^{late'}$ is determined by $\mathbf{D}(w)$. Similarly, $S_k^{early'} = S_k^{early}$ in regions where $\mathbf{S}^{early'}$ is determined by $\mathbf{A}$. Drawing on this observation, we can show that:

Theorem 4.1 *Given a rate constraint r, an arrival vector $\mathbf{A}$, a playout vector $\mathbf{D}$, and a startup latency $w \geq w^*$, the total buffer size $M(w)$ is minimized by allocating an egress buffer of size x^* and an ingress buffer of size $y'(w)$. Similarly, $M(w)$ is minimized by allocating an egress buffer of size $x'(w)$ and an ingress buffer of size y^*. Selecting $w = w^*$ results in the minimum total buffer $M^*(w^*) = x^* + y'(w^*) = x'(w^*) + y^*$.*

Proof: Consider the buffer allocation with x^* and y' at the egress and ingress, respectively, where x^* is derived from $\mathbf{S}^{late}$ and y' is derived from $\mathbf{S}^{early'}$, as shown in Figure 6(a). To see that no other allocation can decrease the total size of the two buffers, consider the effect of increasing the egress buffer size x in Figure 6(b). Suppose that y' is determined by the value of $\mathbf{S}^{early'}$ at time k. Drawing on Lemma 4.1, we know that $\mathbf{S}_k^{early'}$ is determined by the schedule that sends as early as possible subject to either $\mathbf{D} + vec(x)$ or $\mathbf{A}$. If $S_k^{early'}$ is determined by $\mathbf{D} + vec(x)$, then an increase in x raises $\mathbf{S}^{early'}$ by the same amount; as a result, the increase in x is exactly offset by the corresponding decrease in y. If S_k^{early} is determined by $\mathbf{A}$, then increasing x does not decrease y, and the total buffer allocation increases. Finally, if increasing x moves the selection of y to a different part of the video, then the decrease in y does not offset the increase in x, and again the total buffer allocation increases. Hence, it is not possible to reduce the total buffer allocation below $M^*(w) = x^* + y'(w)$. A similar result holds for $x'(w) + y^*$, ensuring that both allocations minimize the total buffer requirement for a given value of



(a) Allocating ingress buffer size y' (b) Increasing egress buffer size $x \geq x^*$

Figure 6: **Egress and Ingress Buffer Allocation:** This left figure shows how to compute the minimum ingress buffer size y' subject to the minimum egress buffer size x^*. The right figure shows that increasing the egress buffer size does not decrease the minimum total buffer requirement $x + y$.

$w \geq w^*$. Since $y'(w)$ and $x'(w)$ are nondecreasing in w, the choice of $w = w^*$ results in the minimum total allocation of $M^*(w^*)$. ∎

One consequence of Theorem 4.1 is that the total buffer allocation $M^*(w) = x^* + y'(w) = x'(w) + y^*$ can be minimized through the calculation of either $\mathbf{S}^{late}$ or $\mathbf{S}^{early}$. More generally, any buffer allocation $M \geq M^*(w)$ can satisfy the rate constraint r, so long as $x \geq x^*$ and $y \geq y^*$:

Corollary 4.1 *Under start-up latency $w \geq w^*$, any buffer allocation with $x \geq x^*$ and $y \geq y^*$ such that $x + y = M^*(w)$ results in a transmission schedule with a minimum peak rate of r.*

Proof: As shown in Lemma 3.1, the peak rate $p(x)$ has a decreasing region, followed by a constant region, and ending with an increasing region. Since allocations of $x = x^*$ ($y = y'(w)$) and $x = x'(w)$ ($y = y^*$) both produce the same peak rate r, the function $p(x)$ cannot have a *larger* value in the interval $x \in (x^*, x'(w))$. Also, note that an egress buffer with $x^* < x < x'(w)$ cannot result in a *smaller* peak rate under the same value of M, since it is not possible to reduce the rate r without increasing the total buffer allocation. Hence, any allocation of $x^* \leq x \leq x'(w)$ can satisfy the rate constraint r with the minimum total buffer allocation. ∎

A stronger version of Theorem 4.1 holds for the important special case of $\mathbf{A} \equiv \mathbf{D}$. As before, the joint minimization of x and w stems from the schedule $\mathbf{S}^{late}(w)$ that transmits frames as late as possible subject to the playout curve $\mathbf{D}(w)$ and the rate r. The schedule also minimizes the egress buffer size y:

Theorem 4.2 *Given a rate constraint r and $\mathbf{A} \equiv \mathbf{D}$, $x^* = y^* = x'(w^*) = y'(w^*)$ and $M^*(w^*) = 2x^* = 2y^*$.*

Proof: When $\mathbf{A} \equiv \mathbf{D}$, a single value of k determines both $x^* = \max_k\{S_k^{late} - D_k\}$ and $w^* = \min\{w \mid D_k - S_{k-w}^{late} \geq 0\}$, as shown in Figure 7. In fact, the two values are directly related by the rate constraint r, with $x^* = rw^*$. To determine y', the schedule $\mathbf{S}^{early'}$ is computed from the two upper constraints $\mathbf{A} \equiv \mathbf{D}$ and $\mathbf{D}(w^*) + vec(x^*)$. These two constraints have exactly the same shape ($\mathbf{D}$), except for the constant time and space offsets (w^* and x^*, respectively). Hence, the schedules that send as early as possible are $\mathbf{S}^{early}$ and $\mathbf{S}^{early}(w^*) + vec(x^*)$, respectively. Since $x^* = rw^*$ and the two schedules obey the rate constraint r, we have $S_k^{early} \leq S_{k-w^*}^{early} + x^*$ for all k. That is, the upper constraint $\mathbf{A}$ always dominates, and $\mathbf{S}^{early'} \equiv \mathbf{S}^{early}$ by Lemma 4.1. Increasing x above x^* does not influence the constraint $\mathbf{A}$ and, hence, does not allow us to decrease y. To show that $y' = x^*$, note that the segments in $\mathbf{S}^{early}$ are parallel to $\mathbf{S}^{late}$. In fact, the ingress buffer size y' is determined by the same run that determines the egress buffer size x^*. Since it is not possible to decrease y by increasing x, we also have $y^* = y'$, resulting in $x^* = y^* = x' = y'$. ∎

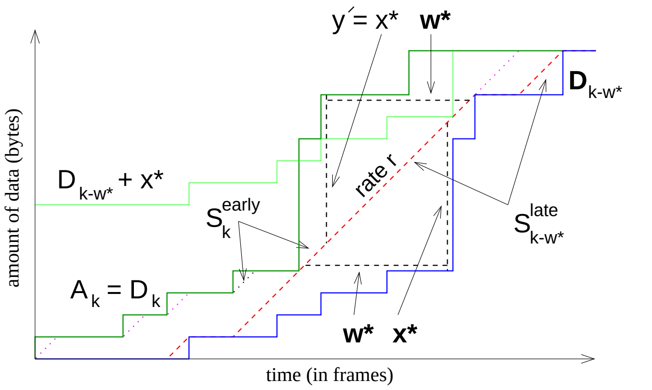


Figure 7: **Buffer Size and Playback Delay for A ≡ D:** This figure shows the playback curve $\mathbf{D}(w^*)$ and the upper smoothing constraints for the special case of $\mathbf{A} \equiv \mathbf{D}$. A single segment of the video stream determines both the egress buffer size x^* and the start-up latency w^* under the rate constraint r, where $x^* = rw^*$.

The theorem shows that when the ingress node receives an unsmoothed version of the video stream, the optimal buffer allocation divides the resources evenly between the ingress and egress nodes.

5 Optimal Tandem Smoothing

The previous sections have focused on the single-link system, where the buffer space resides solely at the ingress and egress nodes. In this section, we generalize the results to a tandem of nodes, and show how to efficiently compute a smooth transmission schedule for each link in the sequence. After introducing the tandem model, we derive an efficient technique for computing an optimal schedule for a given allocation of buffers in the tandem system. In particular, we show that it is possible to compute an optimal tandem schedule by applying the majorization algorithm on a collection of independent single-link problems. Then, we show that an optimal allocation of resources places all buffers at the ingress and egress nodes, allowing us to apply the results from Section 3 and Section 4 to the tandem model.

5.1 Tandem Model

The tandem smoothing model consists of $n+1$ nodes with node i feeding node $i+1$, $i = 0, 1, \ldots, n-1$, as shown in Figure 8. Each node i includes a buffer of size $0 \leq b_i < \infty$. Consider a stream of data which arrives at node 0 destined for node n. The stream has an *arrival vector* $\mathbf{A} = (A_0, \ldots, A_N)$, where A_k corresponds to the amount of data which has arrived at node 0 by time $k = 0, 1, \ldots, N$. It is assumed that $A_k \geq A_{k-1}$, $k = 1, \ldots, N$. Similarly, the stream has a *playout vector* $\mathbf{D} = (D_0, \ldots, D_N)$, where D_k corresponds to the amount of data that must be removed from node n by time $k = 0, 1, 2, \ldots, N$. It is assumed that $D_0 = 0$ and $D_k \geq D_{k-1}$, $k = 1, \ldots, N$. For a valid playout vector for the arriving stream, we assume that $A_k \geq D_k$ for $k = 0, 1, \ldots, N$, with $A_N = D_N$ at the end of the transfer. To ensure that the smoothing buffers can store any outstanding data in the tandem system, we also require that $A_0 \leq b_0$ and

$$A_k - D_k \leq \sum_{i=0}^{n} b_i$$

for $k = 0, 1, \ldots, N$.

Each node $i = 0, 1, \ldots, n$ has a smoothing schedule $\mathbf{S}_i = (S_{i,0}, \ldots, S_{i,N})$, where $S_{i,k}$ denotes the cumulative amount of data that node $i-1$ has sent to node i by time $k = 0, 1, \ldots, N$. As in Section 2, $S_{i,k} \geq S_{i,k-1}$, $k = 1, \ldots N$. Each schedule $\mathbf{S}_i$ has a corresponding rate vector $\mathbf{R}_i = (R_{i,1}, \ldots, R_{i,N})$ where $R_{i,k} = S_{i,k} - S_{i,k-1}$, $k = 1, 2, \ldots, N$. In deriving the smoothing constraints on the tandem system, we focus on the schedule $\mathbf{S}_i$, as shown in Figure 8. To avoid buffer underflow and overflow at node i, a feasible schedule for node i must satisfy the inequality

$$\mathbf{S}_{i+1} \leq \mathbf{S}_i \leq \mathbf{S}_{i+1} + vec(b_i). \qquad (1)$$

A similar inequality holds for the schedule $\mathbf{S}_{i-1}$. Consequently, the set of link schedules in the tandem system must satisfy the constraints

$$\max\{\mathbf{S}_{i-1} - vec(b_{i-1}), \mathbf{S}_{i+1}\} \leq \mathbf{S}_i \leq \min\{\mathbf{S}_{i+1} + vec(b_i), \mathbf{S}_{i-1}\}$$

for $i = 1, \ldots n$, where $\mathbf{S}_0 \equiv \mathbf{A}$ and $\mathbf{S}_{n+1} \equiv \mathbf{D}$. Let $S(\mathbf{A}, \mathbf{D}, \mathbf{b})$ denote the set of feasible schedules for the buffer allocation $\mathbf{b} = (b_0, b_1, \ldots, b_n)$. For the case of $n = 1$, this system reduces to the single-link smoothing problem with $\mathbf{L} = \max(\mathbf{D}, \mathbf{A} - vec(b_0))$ and $\mathbf{U} = \min(\mathbf{D} + vec(b_1), \mathbf{A})$, as discussed in Section 3.

5.2 Optimal Transmission Schedules

To relate the tandem model to the single-link problem, observe that the link entering node i has upstream buffer space from nodes $0, 1, \ldots, i-1$ and downstream buffer space from nodes $i, i+1, \ldots, n$, as shown in Figure 8. Let $\mathbf{S}_i^*$ be the majorization schedule associated with a single-link system with a source buffer of capacity $\sum_{j=0}^{i-1} b_j$ and a receiver buffer of capacity $\sum_{j=i}^{n} b_j$. That is, $\mathbf{S}_i^*$ is the majorization schedule associated with the constraints

$$\begin{aligned} \mathbf{L}_i &= \max\{\mathbf{A} - vec\left(\textstyle\sum_{j=0}^{i-1} b_j\right), \mathbf{D}\} \\ \mathbf{U}_i &= \min\{\mathbf{A}, \mathbf{D} + vec\left(\textstyle\sum_{j=i}^{n} b_j\right)\} \end{aligned}$$

for $i = 1, \ldots, n$. Note that these constraints are typically looser than those imposed on the link schedules in the tandem system, since each link schedule in the tandem system also depends on the schedule chosen for the upstream and downstream links. The main result in this subsection is to show that, despite having tighter constraints, the tandem system has the same optimal transmission schedule for each link. To establish this result, we first observe that the optimal single-link schedules are also a feasible solution to the tandem system:

Lemma 5.1 *The schedule $\mathbf{S}_i^*$ satisfies the following constraints for $i = 1, 2 \ldots n$:*

$$\max\{\mathbf{S}_{i-1}^* - vec(b_{i-1}), \mathbf{S}_{i+1}^*\} \leq \mathbf{S}_i^* \leq \min\{\mathbf{S}_{i+1}^* + vec(b_i), \mathbf{S}_{i-1}^*\}.$$

Proof. It suffices to establish the following,

$$\mathbf{S}_{i+1}^* \leq \mathbf{S}_i^* \leq \mathbf{S}_{i+1}^* + vec(b_i), \qquad (2)$$

for $i = 1, 2, \ldots n$. Observe that $\mathbf{L}_i$ and $\mathbf{U}_i$ are nonincreasing in i. Moreover, $\mathbf{L}_{i+1} \leq \mathbf{L}_i \leq \mathbf{L}_{i+1} + vec(b_i)$ and $\mathbf{U}_{i+1} \leq \mathbf{U}_i \leq \mathbf{U}_{i+1} + vec(b_i)$. Hence, the inequalities in (2) follow from an application of Lemma 2.1. ∎

Then, we show that these optimal single-link solutions $\{\mathbf{S}_i^*\}_{i=1}^n$ are also the optimal link transmission schedules for the tandem system:

Theorem 5.1 *The majorization schedules $\{\mathbf{S}_i^*\}_{i=1}^n$ associated with the finite-source, single-link problems with playout function $\mathbf{D}$ and client buffers $\sum_{j=i}^n b_j$ satisfy the following relations,*

$$\mathbf{R}_i^* \prec \mathbf{R}_i, \qquad \forall \textit{ feasible } \{S_i\}.$$

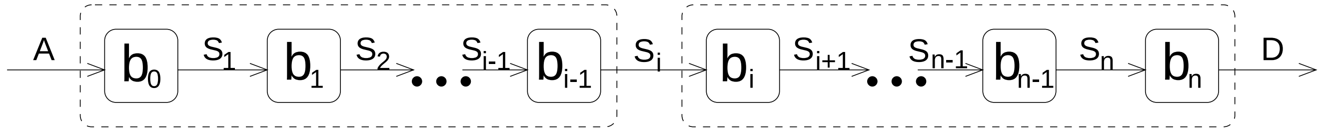


Figure 8: **Tandem Smoothing Model:** The tandem model consists of nodes $i = 0, 1, \ldots, n$ with buffer space b_i. The video stream has an arrival vector $\mathbf{A}$ at node 0 and a playout vector $\mathbf{D}$ at node n. Given these constraints, a solution to the tandem smoothing problem produces link transmission schedules $\mathbf{S}_i$ for $i = 1, 2, \ldots, n$.

Proof. We begin by showing that any feasible solution for the tandem model is a feasible solution for the set of single-link models; that is,

$$\max\{\mathbf{A} - vec\left(\textstyle\sum_{j=0}^{i-1} b_j\right), \mathbf{D}\} \leq \max\{\mathbf{S}_{i-1} - vec(b_{i-1}), \mathbf{S}_{i+1}\}$$

$$\min\{\mathbf{S}_{i+1} + vec(b_i), \mathbf{S}_{i-1}\} \leq \min\{\mathbf{A}, \mathbf{D} + vec\left(\textstyle\sum_{j=i}^{n} b_j\right)\}.$$

This is accomplished by establishing the following four inequalities for $i = 0, \ldots, n+1$

$$\mathbf{A} - vec\left(\textstyle\sum_{j=0}^{i-1} b_j\right) \leq \mathbf{S}_i \quad (3)$$

$$\mathbf{D} \leq \mathbf{S}_i \quad (4)$$

$$\mathbf{S}_i \leq \mathbf{A} \quad (5)$$

$$\mathbf{S}_i \leq \mathbf{D} + vec\left(\textstyle\sum_{j=i}^{n} b_j\right) \quad (6)$$

Inequality (3) is satisfied by $\mathbf{S}_0$ because it is defined as $\mathbf{S}_0 = \mathbf{A}$; for $\mathbf{S}_i$, inequality (3) follows through repeated application of inequality (1). Similarly, as $\mathbf{S}_{n+1} = \mathbf{D}$, repeated application of (1) yields (4). Inequalities (5) and (6) are established in the same manner. Lemma 5.1 can now be applied to establish the theorem. ∎

Henceforth, we shall let $\mathbf{S}_i^*(\mathbf{A}, \mathbf{D}, \mathbf{b})$ denote the majorization schedules associated with $\mathcal{S}(\mathbf{A}, \mathbf{D}, \mathbf{b})$. The schedules can be computed in $O(nN)$ time by solving the n independent single-link problems. The resulting rate vectors $\mathbf{R}_i^*$ minimize any function of the form $g(f_1(\mathbf{R}_1), \ldots, f_n(\mathbf{R}_n))$ where g is a non-decreasing function and the f_i are Schur-convex functions.

5.3 Optimal Buffer Allocation

Next, we consider how to optimally allocate buffer resources across the nodes in the tandem system. We initially focus on an important special case of the tandem smoothing model, where the ingress node has sufficient buffer space to store the entire video; for example, this scenario arises when the ingress node is a server that stores a prerecorded video stream. When $b_0 \geq A_N$, the buffer allocation problem reduces to allocating $M > 0$ bytes among nodes $1, 2, \ldots, n$. Let $\mathcal{B}(M) = \{(b_1, \ldots, b_n) \,|\, b_i \geq 0, i = 1, \ldots, n; \sum_{i=1}^{n} b_i = M\}$ be the set of feasible buffer allocations. Then:

Lemma 5.2 *If the ingress node has sufficient buffer to store the entire video ($b_0 \geq A_N$), then the optimal allocation of M units of buffer to the remaining nodes is $\mathbf{b}^* = (0, \ldots, 0, M)$ in the sense that*

$$\mathbf{R}_i^*(\mathbf{b}^*) \prec \mathbf{R}_i(\mathbf{b}), \quad \forall \mathbf{b} \in \mathcal{B}(M).$$

Proof: The proof follows from Theorem 5.1. ∎

As a consequence of this majorization relationship, the optimal buffer allocation $\mathbf{b}^* = (0, \ldots, 0, M)$ minimizes cost functions of the form $g(f_1(\mathbf{R}_1), \ldots, f_n(\mathbf{R}_n))$ where g is a non-decreasing function and the f_i are (possibly different) Schur-convex functions. A similar result holds when the *egress* node has an arbitrarily large buffer (i.e., $b_n \geq A_N$) and $M > 0$ bytes are allocated among the remaining nodes.

A similar, albeit slightly weaker, result applies to the more general problem of allocating M bytes among all $n+1$ nodes. A feasible smoothing schedule requires $M \geq \max_k\{A_k - D_k\}$ to ensure that the tandem of nodes can accommodate the outstanding data at any time k. For the set of $\mathcal{B}(M) = \{(b_0, \ldots, b_n) \,|\, b_0 \geq A_0; b_i \geq 0, i = 1, \ldots, n; \sum_{i=0}^{n} b_i = M\}$ of feasible buffer allocations, we have:

Lemma 5.3 *For any buffer allocation $\mathbf{b} \in \mathcal{B}(M)$ there exists a buffer allocation $\mathbf{b}' = (b_0', 0, \ldots, 0, b_n')$, with $\mathbf{b}' \in \mathcal{B}(M)$, such that*

$$g(f(\mathbf{R}_1^*(\mathbf{b}')), \ldots, f(\mathbf{R}_n^*(\mathbf{b}'))) \leq g(f(\mathbf{R}_1^*(\mathbf{b})), \ldots, f(\mathbf{R}_n^*(\mathbf{b})))$$

whenever g is non-decreasing and f is Schur-convex.

Proof. Let $j_0 = \arg\min f(\mathbf{R}_i^*(\mathbf{b}))$. Clearly

$$g(f(\mathbf{R}_{j_0}^*(\mathbf{b})), \ldots, f(\mathbf{R}_{j_0}^*(\mathbf{b}))) \leq g(f(\mathbf{R}_1^*(\mathbf{b}), \ldots, f(\mathbf{R}_n^*(\mathbf{b}))).$$

However, this can be achieved by simply allocating $\sum_{i=0}^{j_0-1} b_j$ units of buffer to the ingress node (node 0) and $\sum_{i=j_0}^{n} b_j$ units of buffer to the egress node (node n). ∎

Note that Lemma 5.3 shows that the buffer allocation $\mathbf{b} = (b_0, 0, \ldots, 0, b_n)$ with $b_0, b_n > 0$ minimizes non-decreasing cost functions that apply the *same* Schur-convex function f to each link, in contrast to the stronger result in Lemma 5.2. Still, for a wide set of performance metrics, such as the peak and the variability of the transmission rates, Lemma 5.3 shows that we need only consider allocations with $\mathbf{b} = (b_0, 0, \ldots, 0, b_n)$. Also, we know from Theorem 5.1 that we need only consider majorization schedules associated with particular buffer allocations.

Once the tandem model is reduced to a single-link system, the results for the single-link model in Section 3 and Section 4 can be applied. For example, suppose that each link i in the tandem model imposes a peak rate constraint

$\mathbf{R}_i \leq r_i$, $i = 1, 2, \ldots, n$. Given these rate constraints, we attempt to minimize the total buffer allocation $M = \sum_{i=0}^{n} b_i$ and the start-up latency w. Drawing on Lemma 5.3, we can reduce this problem to a single-link system with buffers at the ingress and egress nodes. In particular, applying Lemma 5.3 with $f, g = \max$ shows that it suffices to consider buffer allocations $\mathbf{b}' = (M - x, 0, \ldots, 0, x)$. Hence, the optimal allocation for the tandem system derives from analyzing a single-link system with a constraint $r = \min_i\{r_i\}$ on the transmission rates.

6 Performance Evaluation

Service providers can minimize the bandwidth requirements for transferring variable-bit-rate video through careful provisioning of their buffer resources and the start-up delay. Using full-length motion-JPEG and MPEG video traces, simulation experiments evaluate a network with a x-byte egress buffer and a y-byte ingress buffer. The experiments assume that the ingress node receives an unsmoothed sequence of N video frames ($\mathbf{A} \equiv \mathbf{D}$), which are removed from the egress buffer w time units later. The simulation experiments investigate how x, y, and w affect the peak and the variability of the transmission rates in the majorization schedule $\mathbf{S}^*$. The empirical results motivate simple heuristics for selecting these parameters to minimize the bandwidth requirements of the video stream.

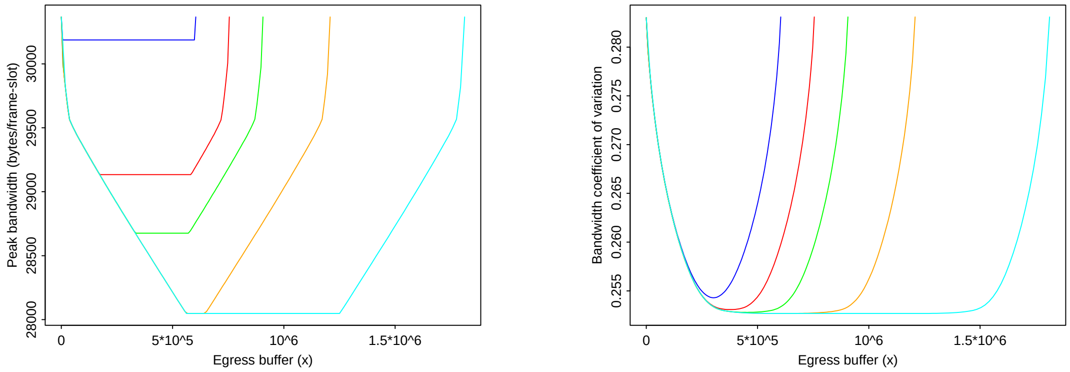
6.1 Buffer Allocation

To investigate the impact of the buffer allocation policy, Figure 9 plots the peak and the coefficient of variation (standard deviation divided by the mean) of the transmission rates for a motion-JPEG encoding of the movie *Beauty and the Beast* with a 20-frame start-up delay. Each curve corresponds to a different total buffer size M, where the x-axis varies the fraction allocated to the egress buffer. To avoid overflow, the combined ingress and egress buffers must hold at least $W \equiv \max_k\{D_k - D_{k-w}\}$ bytes; for this video trace, a sliding window of $w = 20$ frames has a maximum of $W = 604{,}036$ bytes. From top to bottom, the curves in Figure 9 correspond to increasing values of M; as expected, larger buffer configurations reduce both the peak and variability of the bandwidth requirements. Plots for other video traces and other values of w show the same basic trends. Placing 0.5–2.0 megabytes of buffer space at gateway routers or network proxies for each video stream is quite reasonable given the relatively low cost of memory. In exchange, smoothing offers a substantial reduction in network bandwidth requirements.

Consistent with Lemma 3.1, the peak-rate curves $p(x, M)$ are piecewise-linear and convex in the egress buffer size. Unbalanced buffer allocations limit the amount of workahead transmission (when x is small) or require excessive workahead transmission (when $y = M - x$ is small) during regions of large frame sizes. As a result, each curve reaches its maximum at the end points $x = 0$ and $x = M$, since no smoothing is possible when the network has only ingress or egress buffering; in these cases, the peak rate stems from the maximum frame size (30367 bytes), while the coefficient of variation arises from the variability in the frame sizes (a standard deviation of 3580 bytes and a mean of 12,661 bytes, resulting in a ratio of 0.28). Allocating some buffer space at both the ingress and egress of the network has an immediate effect on both metrics. This effect is especially dramatic for MPEG videos (not shown), since a small buffer can remove the significant amount of burstiness that occurs within a group-of-pictures.

Following the left-hand side of the curves in Figure 9(a) shows how the minimum egress buffer size x^* changes as a function of the rate $p(x^*, M)$ for each value of M. Observe that, in the decreasing region, all of the curves share the same minimum buffer size x^* for each peak rate r. Consistent with the discussion in Section 4.1, the minimum buffer size x^* does not depend on the total buffer size M, as long as the combined ingress and egress buffers are large enough to satisfy the rate constraint r. Similarly, each curve reaches the same peak rate r at $x = M - x^*$, when the ingress buffer has size $y = y^* = x^*$. As a result, the peak-rate curves are each symmetric with $p(x, M) = p(M - x, M)$ for $x \in [0, M]$. As a consequence of Theorem 4.2, each peak-rate curve has a minimum at $x = y = M/2$. Smaller values of M offer limited opportunities for smoothing, resulting in a nearly flat peak-bandwidth function when $M = W$, as seen in the top curve in Figure 9(a). Once M is large enough to permit a moderate amount of workahead transmission, the benefits of smoothing become more sensitive to the distribution of buffer space between the ingress and egress points.

For a given value of M, a more balanced distribution of buffer space gives greater latitude for smoothing throughout the entire video, even in regions with relatively small frame sizes compared to the peak rate. As M grows from W to $2W$, the constant region of the peak-rate curve gradually shrinks, while remaining symmetric around $x = M/2$. The curves for $M = 2W$ and $M = 3W$ have the same minimum peak rate of 28048 bytes/frame-slot, with larger values of M resulting in wider regions of optimal buffer allocations in Figure 9(a). In these flat portions of the peak-rate curve, the ingress and egress buffers are both larger than W bytes, ensuring that each buffer can hold any sliding window of w frames; this results in smoothing constraints $L_k = D_{k-w}$ and $U_k = D_k$ that do not depend on the exact distribution of the buffer space. Even when the peak-rate curve is flat, moving closer to a symmetric buffer allocation of $x = M/2$ offers further reductions in the bandwidth variability, as shown in Figure 9(b). Although the bandwidth variability curves appear to have the symmetry properties of the peak-rate curve, the other smoothness metrics are not always perfectly symmetric. Having the same peak rate does not necessarily imply that two schedules match in other parts of the smoothed video. Still, the bandwidth variability metrics are effectively symmetric across a range of traces and buffer sizes.



(a) Peak bandwidth (b) Bandwidth variability

Figure 9: **Buffer Allocation (x):** The graphs show bandwidth requirements for a motion-JPEG encoding of *Beauty and the Beast* after smoothing with a $(M-x)$-byte ingress buffer and x-byte egress buffer using a window $w=20$ frames (for a minimum of $W = 590$ kilobytes). From upper left to lower right, the curves correspond to $M = W, 1.25W, 1.5W, 2W, 3W$.

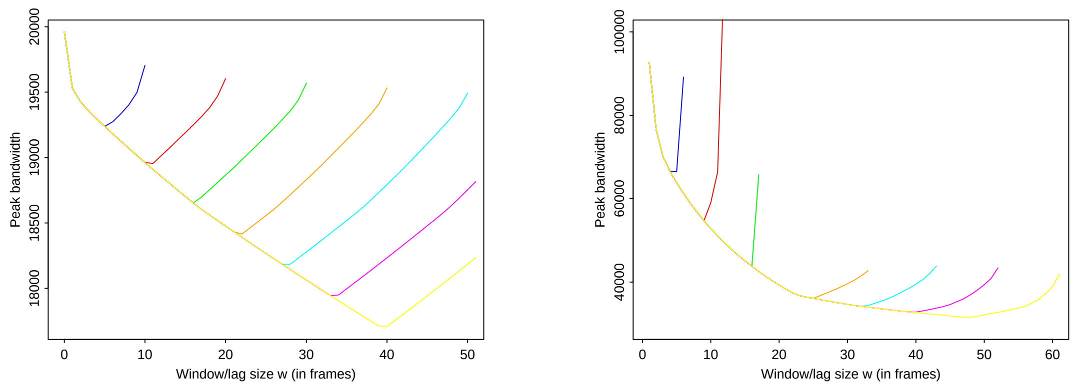
6.2 Start-up Latency

To study the sensitivity of smoothing to the start-up delay, Figure 10 plots the peak bandwidth as a function of w for a motion-JPEG encoding of *E.T. – The Extra Terrestrial* and an MPEG encoding of *Star Wars* [2]. Each peak-rate curve corresponds to a different value of M with an optimal buffer distribution of $x=y=M/2$. From upper left to lower right, the curves in Figure 10(a) correspond to $M = 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4$ megabytes, whereas the curves in Figure 10(b) correspond to $M=0.6, 1.0, 1.4, 1.8, 2.2, 2.6, 3.0$ megabytes. Consistent with the Lemma 3.2, each peak-rate curve is piecewise-linear and convex. Large values of w limit smoothing during regions with large frames, since the combined ingress and egress buffers are nearly full. At the other extreme, small values of w limit the number of frames available for workahead transmission, resulting in larger peak-rate requirements; when $w = 0$, the peak rate equals the maximum frame size, since no smoothing can occur.

Following the left-hand side of the curves in Figure 10 shows how the minimum window size w^* changes as a function of the rate $p(w^*, M)$. Consistent with the discussion in Section 4.1, the minimum start-up latency w^* does not depend on the total buffer size M, similar to the trends in Figure 9. Interestingly, the *E.T.* plots in Figure 10(a) are nearly symmetric in w, while the *Star Wars* curves in Figure 10(b) exhibit significant variations as the start-up latency grows. These irregularities are a byproduct of the video encoding scheme. MPEG compression results in a mixture of large I frames and (typically) smaller P and B frames within a group of pictures (GOP); the *Star Wars* video in Figure 10(b) has a GOP size of 12 frames. For MPEG video, a small change in w can substantially increase the minimum buffer requirement (W), limiting the ability to smooth regions with large frames. In contrast, motion-JPEG compression does not introduce much short-term variability in bandwidth requirements, since each frame is encoded independently.

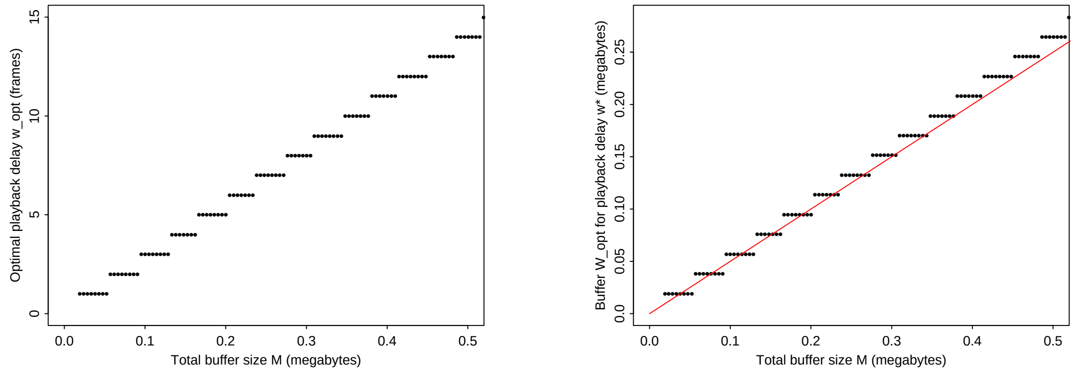
Investigating the relationship between start-up delay and buffer size more closely, Figure 11 plots the window size w^{opt} that minimizes the peak rate across a range of buffer sizes M for the *E.T.* trace; experiments with other video traces show similar trends. As expected, $w^{opt}(M)$ size grows as a function of the total buffer size M, where the optimal configuration has $x = y = M/2$. Plots of w^{opt} consistently have a staircase shape since the start-up delay is an integer number of frames. Except for this discretization effect, these plots are nearly linear, corresponding to the even spacing of the peak-rate curves in Figure 10(a). To minimize the peak rate, the start-up delay w^{opt} must be large enough to permit aggressive smoothing without allowing a window of large frames to consume too much of the aggregate buffer space.

Figure 11(b) investigates the relationship between the total memory size M and the minimum buffer requirement for a window of w^{opt} frames. The buffer requirement W^{opt} is consistently just over half of the available memory M. Conceptually, the optimal start-up delay w^{opt} is typically the smallest window size w that removes the buffer size constraints, resulting in $L_k = D_{k-w}$ and $U_k = D_k$. Smaller values of w limit smoothing since the ingress node cannot transmit sufficient data to capitalize on the egress buffer space; similarly, larger values of w result in a large buffer requirement W^{opt} that strains the storage space in the ingress buffer. Hence, the service could effectively utilize an M-byte buffer budget by assigning $x = M/2$ and by selecting the minimum start-up delay that satisfies $W^{opt} \geq M/2$. This serves as a useful guideline for minimizing the peak and variability of the bandwidth requirements for the smoothed video stream.



(a) *E.T.* (motion-JPEG) (b) *Star Wars* (MPEG)

Figure 10: **Start-up Latency (w):** These graphs plot the peak bandwidth requirements for two videos after smoothing with a start-up delay of w frames. Each curve corresponds to a different buffer allocation of $x = M/2$, with seven evenly spaced values of M increasing from the upper leftmost curve to the lower rightmost curve.



(a) Optimal start-up delay w^{opt} (b) Playback buffer requirement W^{opt}

Figure 11: **Optimal Buffer Allocation and Start-up Delay:** These graphs plot the optimal start-up delay w^{opt} and the corresponding minimum buffer requirement W^{opt} for different buffer sizes M for a motion-JPEG encoding of *E.T.* In (b), the solid line corresponds to $M = 2W$.

7 Conclusions and Future Work

In this paper, we have investigated bandwidth smoothing across a sequence of nodes, where a service provider has control over a subset of the path between the video server and the client site. Starting with a single-link system, with buffer space at ingress and egress nodes, we show that a simple binary search can determine the buffer allocation and start-up latency that minimize the peak rate in the transmission schedule. Drawing on these results, we develop efficient techniques for minimizing the buffer sizes and start-up latency, subject to a constraint on the transmission rate. Then, we generalize these results to the tandem model, consisting of a sequence of multiple nodes and links. We first show how to compute the transmission schedule on each link for a given buffer allocation. Then, we show that the optimal resource allocation places buffers at the ingress and egress nodes. This allows us to apply the results of the single-link system to the tandem model. Finally, experiments with full-length MPEG and motion-JPEG traces verify the analytic results and provide guidelines for allocating buffer space and start-up latency in a real system.

These initial simulation results focus on the important special case where the ingress node receives the original unsmoothed version of the video stream (i.e., $\mathbf{A} \equiv \mathbf{D}$). In a realistic internetworking environment, the arrival vector $\mathbf{A}$ may depend on the network conditions at the upstream nodes, or on the policies and peering arrangements of the neighboring service provider. Experiments with other arrival vectors $\mathbf{A} \neq \mathbf{D}$ can allow us to investigate how the traffic characteristics of the incoming stream affect resource requirements across the tandem of nodes. Along similar lines, we have applied our smoothing model to the online smoothing problem, where the ingress node does not have a priori knowledge of the frame sizes [22]. This study shows that an ingress node does not need to have prior knowledge of frame sizes to achieve the performance gains of bandwidth smoothing. With good estimates of a small window of future frame sizes, an online smoothing algorithm can approximate the results in Section 6. Based on these promising results, we are in the process of building a prototype of a proxy smoothing service for transmitting variable-bit-rate video in an internetworking environment.

A Shape of the Peak-Rate Curve

In this appendix, we prove the key lemma from Section 3:

Lemma 3.1 *The peak rate, $p(x, M)$, of the single-link system is a piecewise-linear, convex function of x.*

To characterize the shape of the peak-rate curve, we first generalize the concept of *change points* in the majorization schedule to apply to the smoothing constraints $\mathcal{V}(x) = \{\mathbf{L}_1(x), \mathbf{L}_2(x), \mathbf{U}_1(x), \mathbf{U}_2(x)\}$. Point k $(0 \leq k \leq N)$ is said to be an *anchor point* (or anchored) on $\mathbf{V}(x) \in \mathcal{V}(x)$ if and only if $V_k(x) = S^*_k(x)$. Observe that some points may not be anchored on any vector in $\mathcal{V}(x)$. Such points will be referred to as *floating points*. Note that change points are anchor points but that the reverse is not necessarily the case. A segment (k_1, k_2) of the schedule $S^*(x)$ is anchored on $\mathbf{V}_1(x), \mathbf{V}_2(x) \in \mathcal{V}(x)$ if k_i is anchored on $\mathbf{V}_i(x)$, $i = 1, 2$ and the remaining points on the segment are floating, i.e., k is floating, $k_1 < k < k_2$. Note that $\mathbf{V}_1(x) = \mathbf{V}_2(x)$ is permitted as is $k_1 = k$, $k_2 = k+1$. Similar to the results in Section 2.3, we draw on the fact that anchor points appear or disappear at certain critical buffer sizes.

A.1 Piecewise-Linear Function

We say that b is a *buffer switching value* if either there exist a pair of vectors $\mathbf{V}_1(b), \mathbf{V}_2(b) \in \mathcal{V}(b)$ and a point k, $0 \leq k \leq N$ such that k is anchored on $\mathbf{V}_1(x)$ but not on $\mathbf{V}_2(x)$ for $x = b^-$ and k is anchored on $\mathbf{V}_2(x)$ but not on $\mathbf{V}_1(x)$ for $x = b^+$ or there exists a vector $\mathbf{V}(b) \in \mathcal{V}(b)$ and a point k such that either k is anchored on $\mathbf{V}(x)$ for $x = b^-$ and is floating for $x = b^+$ or vice versa, i.e., it is a floating point when $x = b^-$ and an anchor point on $\mathbf{V}(x)$ when $x = b^+$. The single-link system exhibits the following behavior as x is varied. If we focus on a point k, we observe that for $0 < x < M$, it follows one of the progressions given in Figure 12. Consequently, as x varies, the system has a finite number, r, of buffer switching values $0 < b_1 < \cdots < b_r < M$, analogous to the results in Section 2.3. The following properties are sufficient to establish the progressions given in Figure 12:

1. $\mathbf{S}^*(x)$ is non-decreasing in $x \in (0, M + D_0 - A_0)$,
2. $dS^*_i(x)/dx \leq 1$, $x \in (0, M + D_0 - A_0)$, $i = 0, \ldots, N$

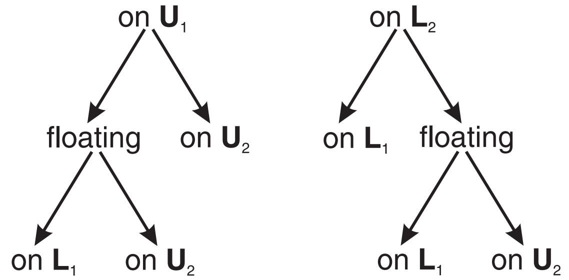


Figure 12: **Increasing Client Buffer Size:** This figure illustrates how the majorization schedule at time k can vary as the client buffer size increases. At a series of critical buffer allocations, the schedule at time k changes from an anchor point on one of the curves, to a floating point or an anchor point on a different constraint curve. The schedule eventually settles at an achor point on either $\mathbf{L}_1$ or $\mathbf{U}_2$.

3. $dU_{1,i}(x)/dx = dL_{1,i}(x)/dx = 1$, $x \in (0, M + D_0 - A_0)$, $i = 0, \ldots, N$
4. $\mathbf{U}_i \geq \mathbf{L}_j$, $i, j = 1, 2$.

The non-decreasingness property follows from the fact that $\mathbf{U}_1(x)$ and $\mathbf{L}_2(x)$ increase in x coupled with Lemma 2.1. Consider the second item. If i is an anchor point, then either $dS^*_i(x)/dx = 0$ because i is anchored on $\mathbf{L}_2$ or $\mathbf{U}_2$, both of which are stationary or $dS^*_i(x)/dx = 1$ because i is anchored on $\mathbf{L}_1$ or $\mathbf{U}_1$, both of which move up at rate 1. If i is floating on a segment attached to $i_1 \neq i$ and $i_2 \neq i$, $i_1 < i_2$, then either $dS^*_{i_1}(x)/dx \leq dS^*_i(x)/dx \leq dS^*_{i_2}(x)/dx$ or $dS^*_{i_2}(x)/dx < dS^*_i(x)/dx < dS^*_{i_1}(x)/dx$. The second item follows in both cases because $dS^*_{i_1}(x)/dx, dS^*_{i_2}(x)/dx \in \{0, 1\}$. The last two items follow from the definitions of $\mathbf{L}_1(x)$, $\mathbf{U}_1(x)$, $\mathbf{L}_2(x)$, and $\mathbf{U}_2(x)$.

The frame rate $R^*_k(x)$ is piecewise-linear with the segments potentially changing at $\{b_i\}$, i.e., $R^*_k(x) = a_{i,k} + x c_{i,k}$, $b_i < x \leq b_{i+1}$, $k = 1, \ldots, N$. Let (k_1, k_2) be the segment of $\mathbf{S}^*(x)$ containing k, $k_1 < k \leq k_2$. Now,

$$a_{i,k} = \frac{V_{2,k_2}(b_i) - V_{1,k_1}(b_i)}{k_2 - k_1}.$$

If k_1 and k_2 are anchored on $\mathbf{V}_1(x)$ and $\mathbf{V}_2(x)$ respectively where the pair $(\mathbf{V}_1(x), \mathbf{V}_2(x))$ comes from the following, $(\mathbf{L}_1(x), \mathbf{L}_2(x))$, $(\mathbf{L}_1(x), \mathbf{U}_2(x))$, $(\mathbf{U}_1(x), \mathbf{U}_2(x))$, $(\mathbf{L}_2(x), \mathbf{U}_1(x))$ for $b_i < x \leq b_{i+1}$, then

$$c_{i,k} = \begin{cases} 1/(k_2 - k_1), & \mathbf{V}_1(x) \in \{\mathbf{L}_2(x), \mathbf{U}_2(x)\}; \\ & \mathbf{V}_2(x) \in \{\mathbf{L}_1(x), \mathbf{U}_1(x)\} \\ 1/(k_1 - k_2), & \mathbf{V}_1(x) \in \{\mathbf{L}_1(x), \mathbf{U}_1(x)\}; \\ & \mathbf{V}_2(x) \in \{\mathbf{L}_2(x), \mathbf{U}_2(x)\} \\ 0, & \text{otherwise.} \end{cases}$$

The piecewise linearity of $p(x)$ follows from the fact that the maximum of a finite set of piecewise-linear functions is piecewise-linear.

A.2 Convex Function

Next, we establish that the convexity of $p(x)$. We know that $p(x)$ is a piecewise linear function of x. Let x_0, x_1, and x_2 be such that $p(x) = a_1 - d_1 x$ for $x_0 \leq x < x_1$, $p(x) = a_2 - d_2 x$, for $x_1 \leq x < x_2$. It suffices to show that $d_1 > d_2$. Note that $a_1 - d_1 x_1 = a_2 - d_2 x_1$. We consider two cases:

(i) The peak rate is determined by the same frame, say k, over the entire interval $[x_0, x_2)$. In other words,

$$p(x) = R_k(x) = \begin{cases} a_1 - d_1 x, & x_0 \leq x < x_1, \\ a_2 - d_2 x, & x_1 \leq x < x_2. \end{cases}$$

In this case, x_1 is a buffer switching value. Let (k_1, k_2) and (j_1, j_2) be the segments containing k when $b_0 \leq x < b_1$ and $x_1 \leq x < x_2$ respectively. We begin with the case that the segment length either increases or remains unchanged, $k_2 - k_1 \leq j_1 - j_2$. There are several possibilities according to which constraint curves k_1 and k_2 are anchored:

k_1 **on** $\mathbf{U}_1$**,** k_2 **on** $\mathbf{L}_2$**:** Note that $d_1 > 0$. Suppose that either k_1 or k_2 become floating points at $x = x_1$, $(j_2 - j_1 > k_2 - k_1)$. In this case, k now lies on a longer segment and $d_2 < d_1$. Suppose that k_1 shifts to $\mathbf{U}_2$; then $d_2 = 0 < d_1$. Similarly, if k_2 attaches to $\mathbf{L}_1$, then $d_2 = 0 < d_1$.

k_1 **on** $\mathbf{U}_1$**,** k_2 **on** $\mathbf{L}_1$**:** Note that $d_1 = 0$. The only possibility here is that k_1 attaches itself to $\mathbf{U}_2$ in which case $d_2 < 0 = d_1$.

k_1 **on** $\mathbf{U}_1$**,** k_2 **on** $\mathbf{L}_1$**:** Note that $d_1 = 0$. The only possibility here is that k_2 attaches itself to $\mathbf{L}_1$ in which case $d_2 < 0 = d_1$.

The situation where the segment size shrinks $(k_2 - k_1 > j_2 - j_1)$ is handled in a similar manner.

(ii) The peak rate is determined by different frames, j and k, in the intervals $[x_0, x_1)$ and $[x_1, x_2)$. In other words,

$$p(x) = \begin{cases} R_j(x) &=& a_1 - d_1 x, & x_0 \leq x < x_1, \\ R_k(x) &=& a_2 - d_2 x, & x_1 \leq x < x_2. \end{cases}$$

Once x increases to x_1, the peak rate is no longer determined at frame j, resulting in $R_j(x) < R_k(x)$ for $x_1 \leq x < x_2$. It follows necessarily that $d_1 > d_2$.

Acknowledgments

The authors would like to acknowledge Subhabrata Sen for his constructive comments, and for discovering a mistake in our initial proof of Theorem 4.1. We also thank Jayanta Dey for discussions and suggestions on an earlier draft of the paper.

Jennifer Rexford received a B.S.E. degree in electrical engineering from Princeton University in 1991, and M.S. and Ph.D. degrees in computer science and engineering from the University of Michigan in Ann Arbor, in 1993 and 1996, respectively. She is currently a member of the Networking and Distributed Systems Center at AT&T Labs – Research in Florham Park, New Jersey. Her research focuses on network measurement, routing protocols, and video streaming, with an emphasis on efficient network support for performance guarantees. Her e-mail address is jrex@research.att.com and her home page is at http://www.research.att.com/~jrex.

Don Towsley received the B.A. degree in physics and the Ph.D. degree in Computer Science from University of Texas in 1971 and 1975 respectively. From 1976 to 1985 he was a member of the faculty of the Department of Electrical and Computer Engineering at the University of Massachusetts, Amherst. He is currently a Distinguished University Professor at the University of Massachusetts in the Department of Computer Science. During 1982-1983, he was a Visiting Scientist at the IBM T.J. Watson Research Center, Yorktown Heights, NY. and during the year 1989-1990, he was a Visiting Professor at the Laboratoire MASI, Paris, France. His research interests include high speed networks and multimedia systems. He currently serves on the Editorial board of *Performance Evaluation* and has previously served on several editorial boards including those of the *IEEE Transactions on Communications* and *IEEE/ACM Transactions on Networking*. He was a Program Co-chair of the joint ACM SIGMETRICS and PERFORMANCE '92 conference. He is a member of ACM, ORSA and the IFIP Working Groups 6.3 and 7.3. He has received two best conference paper awards from ACM SIGMETRICS and has been elected Fellow of both the ACM and IEEE.

References

[1] E. P. Rathgeb, "Policing of realistic VBR video traffic in an ATM network," *International Journal of Digital and Analog Communication Systems*, vol. 6, pp. 213–226, October–December 1993.

[2] M. Garrett and W. Willinger, "Analysis, modeling and generation of self-similar VBR video traffic," in *Proceedings of ACM SIGCOMM*, pp. 269–280, September 1994.

[3] M. Grossglauser, S. Keshav, and D. Tse, "RCBR: A simple and efficient service for multiple time-scale traffic," *IEEE/ACM Transactions on Networking*, vol. 5, pp. 741–755, December 1997.

[4] O. Rose, "Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems," in *Proceedings of the Conference on Local Computer Networks*, pp. 397–406, October 1995.

[5] T. V. Lakshman, A. Ortega, and A. R. Reibman, "Variable bit-rate (VBR) video: Tradeoffs and potentials," *Proceedings of the IEEE*, vol. 86, pp. 952–973, May 1998.

[6] T. Ott, T. Lakshman, and A. Tabatabai, "A scheme for smoothing delay-sensitive traffic offered to ATM networks," in *Proceedings of IEEE INFOCOM*, pp. 776–785, May 1992.

[7] W. Feng and S. Sechrest, "Critical bandwidth allocation for delivery of compressed video," *Computer Communications*, vol. 18, pp. 709–717, October 1995.

[8] W. Feng, F. Jahanian, and S. Sechrest, "Optimal buffering for the delivery of compressed prerecorded video," *ACM/Springer-Verlag Multimedia Systems Journal*, pp. 297–309, September 1997.

[9] 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," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 397–410, August 1998.

[10] J. M. McManus and K. W. Ross, "Video on demand over ATM: Constant-rate transmission and transport," *IEEE Journal on Selected Areas in Communications*, pp. 1087–1098, August 1996.

[11] J. M. del Rosario and G. C. Fox, "Constant bit rate network transmission of variable bit rate continuous media in video-on-demand servers," *Multimedia Tools and Applications*, vol. 2, pp. 215–232, May 1996.

[12] W. Feng, "Rate-constrained bandwidth smoothing for the delivery of stored video," in *Proceedings of IS&T/SPIE Multimedia Networking and Computing*, pp. 316–327, February 1997.

[13] J. Zhang and J. Hui, "Applying traffic smoothing techniques for quality of service control in VBR video transmissions," *Computer Communications*, vol. 21, pp. 375–389, April 1998.

[14] Z. Jiang and L. Kleinrock, "General optimal video smoothing algorithm," in *Proceedings of IEEE INFOCOM*, pp. 676–684, April 1998.

[15] J. Rexford and D. Towsley, "Smoothing variable-bit-rate video in an internetwork," in *Proc. SPIE Conference on Performance and Control of Network Systems*, pp. 345–357, November 1997.

[16] G. Sanjay and S. V. Raghavan, "Fast techniques for the optimal smoothing of stored video." To appear in *ACM/Springer-Verlag Multimedia Systems Journal*.

[17] J. K. Dey, S. Sen, J. Kurose, D. Towsley, and J. Salehi, "Playback restart in interactive streaming video applications," in *Proc. IEEE Conference on Multimedia Computing and Systems*, pp. 458–465, June 1997.

[18] J. Zhang, "Using multiple buffers for smooth VBR video transmission over the network," in *Proc. International Conference on Communications Technologies*, October 1998.

[19] J. Y. Hui, *Switching and Traffic Theory for Integrated Broadband Networks*. Kluwer, 1990.

[20] A. W. Marshall and I. Olkin, *Inequalities: Theory of Majorization and Its Applications*. Academic Press, 1979.

[21] W. Feng and J. Rexford, "A comparison of bandwidth smoothing techniques for the transmission of prerecorded compressed video," in *Proceedings of IEEE INFOCOM*, pp. 58–66, April 1997.

[22] S. Sen, J. Rexford, J. Dey, J. Kurose, and D. Towsley, "Online smoothing of variable-bit-rate streaming video," Tech. Rep. 98-75, University of Massachusetts Computer Science Department, 1998.