Smoothing Variable-Bit-Rate Video in an Internetwork

Jennifer Rexford[a] and Don Towsley[b]

[a]AT&T Labs – Research, Florham Park, NJ 07932-0971 USA

[b]University of Massachusetts, Amherst, MA 01003-4610 USA

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 prefetching frames into the client playback buffer in advance of each burst. Drawing on *a priori* 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. To develop efficient techniques for transmitting variable-bit-rate video across a *portion* of the route, we investigate bandwidth smoothing across a tandem of nodes, which may or may not include the server and client sites. We show that it is possible to compute an optimal transmission schedule for the tandem system by solving a collection of independent single-link problems. To develop efficient techniques for minimizing the network bandwidth requirements, we characterize how the peak rate varies as a function of the buffer allocation and the playback delay. Simulation experiments illustrate the subtle interplay between buffer space, playback delay, and bandwidth requirements for a collection of full-length video traces. These analytic and empirical results suggest effective guidelines for provisioning network services for the transmission of compressed video.

Keywords: compressed video, bandwidth smoothing, majorization, prefetching, internetwork, tandem model

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, video traffic 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.[1–4] 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 increase the network resources required to transfer the video.

To reduce the end-to-end resource requirements, a stored-video server can smooth the outgoing stream by prefetching frames into the client playback buffer in advance of each burst. 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[5–9] can 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 a fairly small number of runs and a significant reduction in both the peak and variability of the transmission rates, as discussed in the overview in Section 2.

Previous work on smoothing prerecorded video has been based on a video-on-demand model of a server that stores the entire sequence of frames and can directly coordinate access to the client prefetch buffer. However, in a realistic networking environment, a single service provider may not have control over the entire path from the stored video

---

Contact information:
J. Rexford: Email: jrex@research.att.com; Web: http://www.research.att.com/~jrex; Telephone: 973-360-8728; Fax: 973-360-8871.
D. Towsley: Email: towsley@cs.umass.edu; Telephone: 413-545-0207; Fax: 413-545-1249. The work of this author was performed while visiting AT&T Labs – Research.

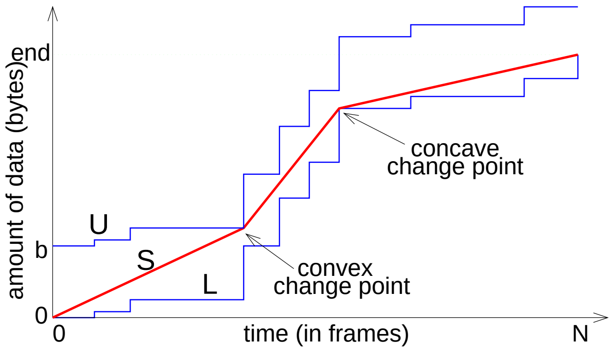


Figure 1. Constraint Curves: The sample transmission plan **S** is a non-decreasing curve consisting of three linear runs that stay between the upper (**U**) and lower (**L**) smoothing constraints. The second run increases the transmission rate to avoid an eventual buffer underflow, while the third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

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 overheads for 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 the path from the stored-video server to the playback buffer. Concurrent work[10] formulates a similar tandem smoothing problem and presents heuristic schemes for iteratively computing transmission schedules for each node. Instead, we derive efficient techniques that compute *globally* optimal schedules under a variety of network constraints and cost metrics.

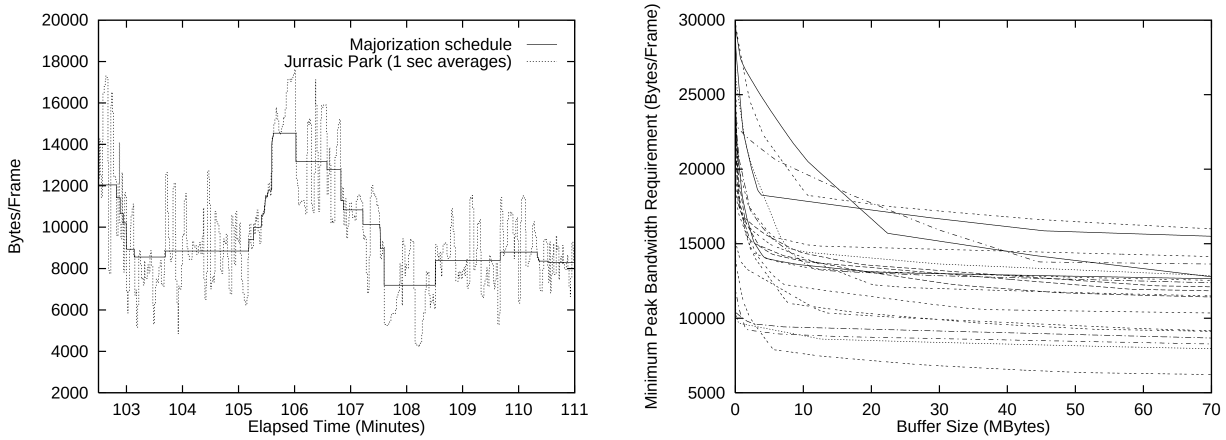
Section 3 shows that, given the distribution of buffer space at the nodes in the route, it is possible to determine a globally optimal transmission schedule by solving a collection of single-link problems. For an N-frame video that traverses n links, this optimal schedule can be computed in $O(nN)$ time by applying an existing $O(N)$ algorithm[8] for smoothing stored video; the decomposition result for the tandem system relies on the unique properties[8, 11, 12] of this algorithm, as reviewed in Section 2. Then, Section 4 and Section 5 characterize how a video's peak-bandwidth requirement varies as a function of the buffer distribution and start-up delay in the tandem system. These results show that a simple binary-search algorithm is sufficient to compute buffer and delay allocations that minimize the peak rate. In Section 6, simulation experiments with full-length MPEG and motion-JPEG video traces highlight the performance of the smoothing algorithm and suggest effective guidelines for provisioning network services for the transmission of compressed video. Section 7 concludes the paper with a discussion of future research directions.

2. BANDWIDTH SMOOTHING

A multimedia server can substantially reduce the rate requirements for transmitting prerecorded video by prefetching frames into the client playback buffer. 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 that prefetches frames 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 a b-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 $k = 0, 1, \ldots, N$. To prevent overflow of the playback buffer, the client should not receive more than $U_k = L_k + b$ bytes by time k, as shown in Figure 1. Any valid server transmission plan should stay between these vertically equidistant functions. Prefetching occurs whenever the schedule lies above the underflow curve. To permit prefetching 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 $j \leq 0$.



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

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 b for twenty full-length motion-JPEG traces.

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 maximum transmission rate occurs when a bandwidth increase (convex change point) is 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 effective bandwidth, allowing the server and the network to multiplex a larger number of streams. The theory of majorization[11] offers an effective 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 $\phi : I\!R^K \to I\!R$. 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 $f : I\!R^K \to I\!R$.

Applying the theory of majorization to the bandwidth smoothing problem, there exists an $O(N)$ algorithm[8] 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),[8] 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[13].

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[12] play an important role in analyzing the models introduced in Section 3 and Section 4. The domination result compares the $\mathbf{L}$ and $\mathbf{U}$ constraints of two smoothing configurations:

LEMMA 2.1. *Let* $\mathbf{L}_1 \le \mathbf{U}_1$ *and* $\mathbf{L}_2 \le \mathbf{U}_2$ *be such that* $\mathbf{L}_1 \le \mathbf{L}_2$ *and* $\mathbf{U}_1 \le \mathbf{U}_2$. *Then* $\mathbf{S}_1^* \le \mathbf{S}_2^*$. *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[12]. □

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 prefetch buffer. With a slight rephrasing of the result in,[12] 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* $b_2 \ge b_1 > 0$, *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 by invoking Lemma 2.1 twice.[12] 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 that consists of a sequence of smoothing nodes with finite buffer space, where the source node may consist of a finite buffer that cannot store the entire incoming video stream.

3. OPTIMAL SMOOTHING IN A TANDEM SYSTEM

This section formally introduces the tandem smoothing model and develops an efficient technique for computing link transmission schedules. By applying the majorization algorithm to a collection of independent single-link problems, we determine the transmission schedules that minimize any Schur-convex function of the link rate vectors across the tandem system.

3.1. 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 3. Each node i includes a buffer of size $0 \le b_i < \infty$. 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 $A_k \ge A_{k-1}$, $k = 1, \ldots, N$. 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 $D_k \ge D_{k-1}$, $k = 1, \ldots, N$. For a valid playout vector for the arriving stream, we assume that $A_k \ge D_k$ 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 $A_0 \le b_0$ and

$$\max_{0 \le k \le N} \{A_k - D_k\} \le \sum_{i=0}^{n} b_i.$$

Hence, we implicitly assume that the links do not introduce propagation delay or jitter. Propagation delay would add a constant term to the end-to-end latency, while jitter tolerance would require nominal additional buffer space at each node.

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, $S_{i,k} \ge S_{i,k-1}$, $k = 1, \ldots N$. Each schedule $\mathbf{S}_i$ has a corresponding rate vector $\mathbf{R}_i = (R_{i,1}, \ldots, R_{i,N})$ where $R_{i,k} = S_{i,k} - S_{i,k-1}$,

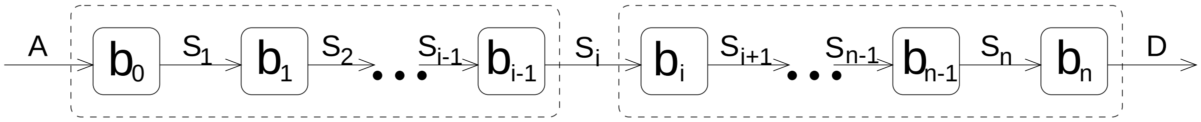


Figure 3. Tandem Smoothing Model: The tandem model consists of nodes $i = 0, 1, \ldots, n$ with buffer space b_i. The video stream has an arrival vector $\mathbf{A}$ at node 0 and a playout vector $\mathbf{D}$ at node n. Given these constraints, a solution to the tandem smoothing problem produces link transmission schedules $\mathbf{S}_i$ for $i = 1, 2, \ldots, n$.

$k = 1, 2, \ldots, N$. In deriving the smoothing constraints on the tandem system, we focus on the schedule $\mathbf{S}_i$, as shown in Figure 3. To avoid buffer underflow and overflow at node i, a feasible schedule for node i must satisfy the inequality

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

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

$$\max\{\mathbf{S}_{i-1} - vec(b_{i-1}), \mathbf{S}_{i+1}\} \leq \mathbf{S}_i \leq \min\{\mathbf{S}_{i+1} + vec(b_i), \mathbf{S}_{i-1}\}, \quad i = 1, \ldots n \tag{2}$$

where $\mathbf{S}_0 \equiv \mathbf{A}$ and $\mathbf{S}_{n+1} \equiv \mathbf{D}$. Here the maximum and minimum of two vectors are taken componentwise.

Let $\mathcal{S}(\mathbf{A}, \mathbf{D}, \mathbf{b})$ denote the set of feasible schedules for the buffer allocation $\mathbf{b} = (b_0, \ldots, b_n)$. For the case $n = 1$, this system reduces to a single-link smoothing problem with $\mathbf{L} = \max(\mathbf{D}, \mathbf{A} - vec(b_0))$ and $\mathbf{U} = \min(\mathbf{D} + vec(b_1), \mathbf{A})$. Hence, the $O(N)$ majorization algorithm can determine a schedule $\mathbf{S}^*$ that minimizes any Schur-convex function of the rate vector $\mathbf{R}^*$, as discussed in Section 2. The next subsection shows how to decompose the tandem smoothing problem in Figure 3 into a collection of n independent single-link problems. In particular, Theorem 3.2 implies that the rate vectors $\mathbf{R}_i^*$ associated with the link majorization schedules 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 (possibly different) Schur-convex functions. Furthermore, these majorization schedules can be obtained in $O(nN)$ time.

3.2. Optimal schedules

To relate the tandem smoothing 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 3. 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 $i = 1, \ldots, n$. Drawing on these definitions, we have the following result:

LEMMA 3.1. *The schedule $\mathbf{S}_i^*$ satisfies the following constraints.*

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

Proof. It suffices to establish that $\mathbf{S}_{i+1}^* \leq \mathbf{S}_i^* \leq \mathbf{S}_{i+1}^* + vec(b_i)$ for $i = 1, \ldots, n$. 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 follow from an application of Lemma 2.1. □

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

THEOREM 3.2. *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 relation $\mathbf{R}_i^* \prec \mathbf{R}_i$, for all feasible $\{S_i\}$.*

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,

$$\begin{aligned}\max\{\mathbf{A} - vec\left(\sum_{j=0}^{i-1} b_j\right), \mathbf{D}\} &\leq \max\{\mathbf{S}_{i-1} - vec(b_{i-1}), \mathbf{S}_{i+1}\}\\ \min\{\mathbf{S}_{i+1} + vec(b_i), \mathbf{S}_{i-1}\} &\leq \min\{\mathbf{A}, \mathbf{D} + vec\left(\sum_{j=i}^{n} b_j\right)\}.\end{aligned}$$

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

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

$$\mathbf{D} \leq \mathbf{S}_i \tag{4}$$

$$\mathbf{S}_i \leq \mathbf{A}, \tag{5}$$

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

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 3.1 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.

4. BUFFER ALLOCATION

The previous section describes how to compute optimal link schedules for a tandem system based on the size of the smoothing buffer at each node. Based on these results, we now investigate how to allocate an M-byte buffer budget among the nodes to minimize the cost of transmitting the video stream. For many important cost metrics, such as the peak transmission rate, selecting an optimal buffer allocation in the tandem system reduces to solving a single-link optimization problem. By characterizing the shape of the peak-rate curve, we show that a simple binary search algorithm is sufficient to determine a buffer allocation that minimizes the maximum bandwidth of the smoothed video stream.

4.1. Reducing the tandem model to a single-link problem

Initially, we consider 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 the actual server that stores the 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 4.1. *If the ingress node has sufficient buffer to store the entire video ($b_0 > A_N$), then the optimal allocation of M units of buffer to the remaining nodes is $\mathbf{b}^* = (0, \ldots, 0, M)$ in the sense that $\mathbf{R}_i^*(\mathbf{b}^*) \prec \mathbf{R}_i(\mathbf{b})$, for all $\mathbf{b} \in \mathcal{B}(M)$.*

Proof. The proof follows from Theorem 3.2. □

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 4.2. *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 4.2 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 4.1. Still, for a wide set of performance metrics, such as the peak and the variability of the transmission rates, Lemma 4.2 shows that we need only consider allocations with $\mathbf{b} = (b_0, 0, \ldots, 0, b_n)$. Also, we know from Theorem 3.2 that we need only consider majorization schedules associated with particular buffer allocations. The remainder of the section focuses on minimizing the peak bandwidth ($g, f = \max$) in a single-link system by showing that a locally optimal buffer allocation is a global optimum.

4.2. Minimizing the peak rate in a single-link system

Consider the problem of allocating an M-byte buffer budget in a single-link system, with an x-byte egress buffer and an $(M-x)$-byte ingress buffer, where $0 \leq x \leq M + D_0 - A_0$ (to account for any data in the ingress buffer at time 0). Let $p(x, M)$ denote the peak rate associated with the majorization schedule $S^*(\mathbf{A}, \mathbf{D}, (M - x, x))$. When there is no ambiguity, we shall refer to the peak rate as $p(x)$ and the associated majorization schedule as $S^*(x)$. To characterize how $p(x)$ varies with x, we parameterize the smoothing constraints:

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

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. Recall that $\mathbf{S}^*(x)$ is the majorization schedule associated with $\mathcal{S}(\max\{\mathbf{L}_1(x), \mathbf{L}_2(x)\}, \min\{\mathbf{U}_1(x), \mathbf{U}_2(x)\}, (M - x, x))$.

By characterizing how the schedule $\mathbf{S}^*(x)$ changes with x, we can show that $p(x)$ is a piecewise-linear function in x with a decreasing phase, followed by a constant phase, and ending with an increasing phase. First, we 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. First, we show that:

LEMMA 4.3. *The peak rate,* $p(x, M)$, *of the single-link system is a piecewise-linear function of* x.

Proof. 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 4. 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 4:

1. $\mathbf{S}^*(x)$ is non-decreasing in $x \in (0, M + D_0 - A_0)$,

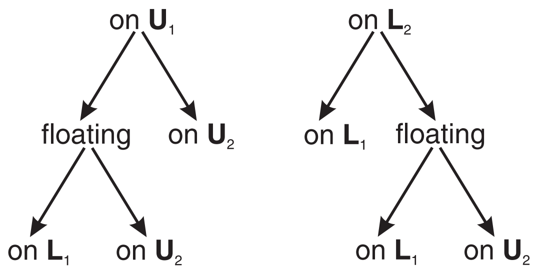


Figure 4. Increasing Client Buffer Size: This figure illustrates how the majorization schedule at time k can vary as the client buffer size increases. At a series of critical buffer allocations, the schedule at time k changes from an anchor point on one of the curves, to a floating point or an anchor point on a different constraint curve. The schedule eventually settles at an achor point on either L_1 or U_2.

2. $dS_i^*(x)/dx \leq 1$, $x \in (0, M + D_0 - A_0)$, $i = 0, \ldots, N$
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 $\mathbf{L}_1(x), \mathbf{U}_1(x), \mathbf{L}_2(x), \mathbf{U}_2(x)$.

The frame rate $R_k^*(x)$ is piecewise-linear with the segments potentially changing at $\{b_i\}$, i.e., $R_k^*(x) = a_{i,k} + xc_{i,k}$, $b_i < x \leq b_{i+1}$, $k = 1, \ldots, N$. 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. □

Next, we establish that any local minimum of $p(x)$ is also a global minimum. In particular:

LEMMA 4.4. *There exist $b_1, b_2 \in [A_0, M]$ ($b_1 \leq b_2$) such that $p(x)$ is decreasing in $[A_0, b_1)$, constant in $[b_1, b_2)$ and increasing in $[b_2, M]$.*

Proof. We shall refer to these three phases respectively as shrinking, constant, and growing regions. In the region $[A_0, b_1)$, $p(x)$ is determined by points lying on segments that are anchored on $\mathbf{U}_1$ and $\mathbf{L}_2$. In the range $[b_1, b_2)$, $p(x)$ is determined by a single segment anchored either on $\mathbf{U}_1$ and $\mathbf{L}_1$, on $\mathbf{U}_2$ and $\mathbf{L}_2$, or both on $\mathbf{L}_2$. In the latter case, the point $k = 0$ is necessarily an anchor point. Last, in the range $[b_2, M]$, $p(x)$ is determined by points lying on segments anchored on $\mathbf{U}_2$ and $\mathbf{L}_1$ or, in the case of $A_0 = D_0$, $\mathbf{L}_2$ and $\mathbf{L}_1$. Note that either of the shrinking or the growing regions may be absent (Figures 5(b) or 5(c), but not both).

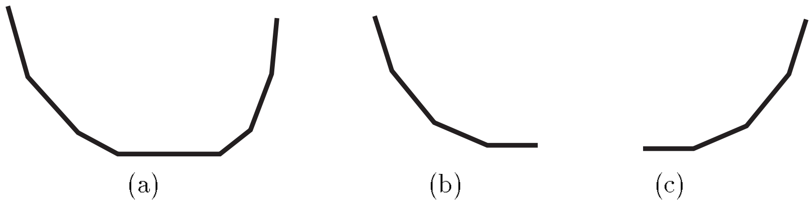


Figure 5. Shape of Peak-Rate Curve: As a function of the buffer allocation x, the peak-rate curve $p(x)$ has (a) shrinking and growing phases; (b) shrinking phase only; (c) growing phase only.

We establish this property through contradiction. Assume that there are buffer sizes $x_0 < x_1 < x_2$ such that $p(x) = a_0 + d_0 x$, $x \in [x_0, x_1)$ where $d_0 \geq 0$, and $p(x) = a_1 + d_1 x$, $x \in [x_1, x_2)$ where $d_1 < 0$. Note that $a_0 + d_0 x_1 = a_1 + d_1 x_1$. In the range $[x_1, x_2)$ $p(x)$ is determined by a point k on a line segment anchored between a convex change point on $\mathbf{U}_1$ and a concave change point on $\mathbf{L}_2$. If this is so, then $R_k^*(x) \geq a_1 + d_1 x > a_0 + d_0 x_1 \geq a_0 + d_0 x$ for $x < x_1$ which contradicts the assertion that $p(x) = a_0 + d_0 x$ is the peak rate in the interval $[x_0, x_1)$. Next, consider the case $p(x) = a_0 + d_0 x$, $x \in [x_0, x_1)$, where $d_0 > 0$ and $p(x) = a_0$, $x \in [x_1, x_2)$. The previous argument can be modified to show $a_0 < R_k^*(x)$ for $x_1 < x$ contradicting the assertion that $p(x) = a_0$ is the peak rate in the interval $(b_1, b_2]$. □

With additional work, it is also possible to establish that $p(x)$ is a *convex* function in x. As a consequence of Lemma 4.4, a simple binary search algorithm is sufficient to determine a value of x that minimizes the peak rate.

5. START-UP LATENCY

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 at the egress node. This section shows that the peak rate of the smoothed stream is a piecewise-linear function of the start-up latency with a decreasing phase, followed by a constant phase, and ending with an increasing phase, drawing on the arguments in Section 4.

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 playout vector $\mathbf{D}(w)$ where

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

Given a buffer allocation (b_0, b_1), the startup latency w is constrained to be less than

$$w_{\max} \equiv \min \left\{ w \mid \max_{0 \leq k \leq N+w} \{A_k(w) - D_k(w)\} \geq b_0 + b_1 \right\};$$

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

Although we are unable to derive a closed form expression for w^*, $p(w)$ exhibits a property that facilitates its computation. To characterize how $p(w)$ varies with w, we parameterize the smoothing constraints:

$$\begin{aligned} \mathbf{L}_1(w) &= \mathbf{D}(w) \\ \mathbf{U}_1(w) &= \mathbf{D}(w) + vec(b_1) \\ \mathbf{L}_2(w) &= \mathbf{A} - vec(b_0) \\ \mathbf{U}_2(w) &= \mathbf{A}. \end{aligned}$$

Note that $\mathbf{L}_1(w)$ and $\mathbf{U}_1(w)$ are stationary with respect to each other as are $\mathbf{L}_2(w)$ and $\mathbf{U}_2(w)$. 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 4, we show that

LEMMA 5.1. *The peak rate, $p(w)$, of the single-link system is a piecewise-linear function of w, the startup delay. Moreover, there exist $t_1, t_2 \in [0, w_{\max}]$ ($t_1 \le t_2$) such that $p(w)$ is decreasing in $[0, t_1)$, constant in $[t_1, t_2)$ and increasing in $[t_2, w_{\max}]$.*

Proof. Observe that, 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 $\mathbf{U}_2(w)$. However, the behavior of the points on the majorization schedules exhibit similar behavior under a latency change as under a buffer change. Hence, the arguments used in the proof of Lemma 4.3 and Lemma 4.4 are easily modified to apply here. □

The primary consequence of Lemma 5.1 is that we can now use a binary search algorithm to determine the optimal startup latency w^* for a given buffer allocation.

6. PERFORMANCE EVALUATION

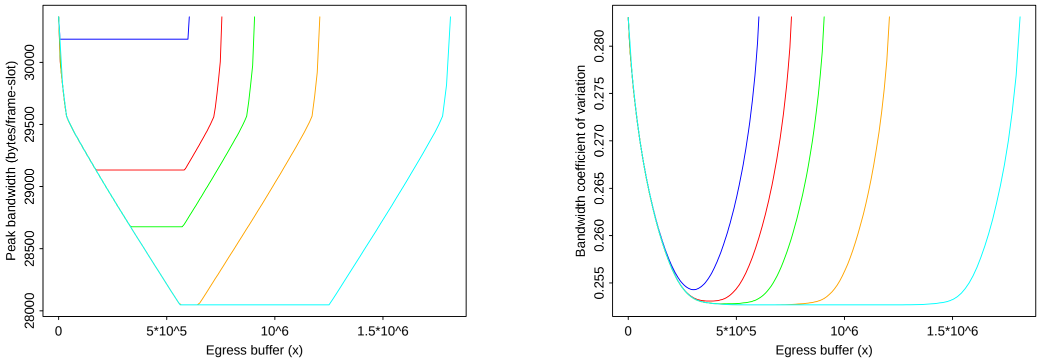
Service providers can minimize the bandwidth requirements for transferring stored video through careful provisioning of the buffer resources and the start-up delay. Using full-length video traces, simulation experiments evaluate a network with a x-byte egress buffer and an $(M-x)$-byte ingress buffer. The experiments assume that the ingress node receives an unsmoothed sequence of N video frames ($A_k = D_k$), which are removed from the egress buffer w time units later. Hence, the smoothing problem has constraints $L_k = \max\{D_{k-w}, D_k + x - M\}$ and $U_k = \min\{D_{k-w} + x, D_k\}$ for $k = 0, 1, \ldots, N + w$. Playback begins at time $k = w$, where $D_k = 0$ for $k \le 0$; similarly, playback ends at time $k = N + w$, where $D_k = D_N$ for $k > N$. The simulation experiments investigate how M, x, 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 6 plots the peak and the coefficient of variation 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 6 correspond to $M = W, 1.25W, 1.5W, 2W, 3W$; 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.

Consistent with the theorems in Section 4, the peak-rate curves are piecewise-linear in the buffer size and consist of decreasing, constant, and increasing phases. 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; for 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.

Smaller values of M offer limited opportunities for smoothing, resulting in a nearly flat peak-bandwidth curve when $M = W$. Once M is large enough to permit a moderate amount of prefetching, the benefits of smoothing become more sensitive to the distribution of buffer space between the ingress and egress points. As M grows larger than W, the flat region gradually shrinks, while remaining symmetric around $x = M/2$, as shown in Figure 6(a). The peak rate often stems from relatively small regions of large frames in the video stream. Unbalanced buffer allocations limit the amount of prefetching (when x is small) or require excessive prefetching (when $M-x$ is small) during these regions of large frame sizes. Even when the peak-rate curve is flat, moving closer to a symmetric buffer allocation of $x = M/2$ reduces the variability in the transmission rates, as shown in Figure 6(b).



(a) Peak bandwidth

(b) Bandwidth variability

Figure 6. Buffer Allocation (B and b): 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 $M = W, 1.25W, 1.5W, 2W, 3W$.

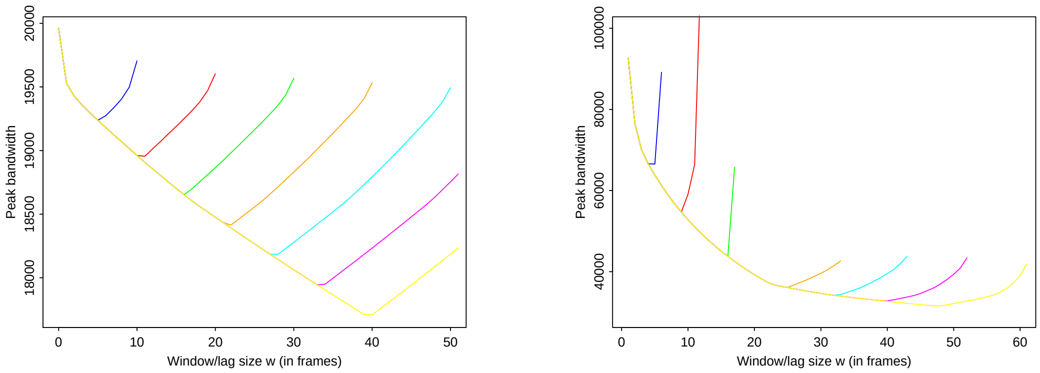
The curves for $M = 2W$ and $M = 3W$ have the same minimum peak rate of 28048 bytes/frame-slot, with larger values of M resulting in wider regions of optimal buffer allocations in Figure 6(a). In these flat portions of the peak-rate curve, the ingress and egress buffers are both larger than W bytes, 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. Consequently, these buffer configurations also produce optimal schedules with the same rate variability, as shown in Figure 6(b). Allocating more than $M = 2W$ does not offer any additional smoothing benefits. These simulation results in Figure 6, and similar results for other motion-JPEG and MPEG traces, suggest that service providers should allocate up to $2W$ bytes of data, with the buffer space split evenly between the ingress and egress nodes.

6.2. Start-up delay

To study the sensitivity of smoothing to the start-up delay, Figure 7 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 curve corresponds to a different value of M with a buffer allocation of $x = M/2$. From upper left to lower right, the curves in Figure 7(a) correspond to $M = 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4$ megabytes, whereas the curves in Figure 7(b) correspond to $M = 0.6, 1.0, 1.4, 1.8, 2.2, 2.6, 3.0$ megabytes. The curves start at $w = 0$ and end when the start-up delay exceeds the available buffer space. Consistent with the Lemma 5.1 in Section 5, each peak-rate curve is piecewise-linear and has decreasing, constant, and increasing phases. Small values of w limit the number of frames available for prefetching, resulting in larger peak-rate requirements; in fact, for small w the peak rate is largely independent of the buffer size M, resulting in $L_k = D_{k-w}$ and $U_k = D_k$; as a result, each curve has the same peak bandwidth in this region.

On the other extreme, large values of w limit smoothing during regions with large frames, since the combined ingress and egress buffers are nearly full. Hence, for large and small values of w, the peak rate stems from the maximum frame size. Comparing the two graphs, the *E.T.* plots exhibit a significant amount of symmetry, since motion-JPEG compression does not introduce as much short-term variability, since each frame is encoded independently. In contrast, 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 7(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.

Investigating the relationship between start-up delay and buffer size more closely, Figure 8(a) graphs the optimal start-up delay w^* as a function of the total buffer size M for the *E.T.* trace; experiments with other video traces show similar trends. As expected, the optimal window size grows as a function of the total buffer size M. Plots



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

Figure 7. Start-up Delay (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.

of w^* 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 7(a). To minimize the peak rate, the start-up delay w^* must be large enough to permit aggressive prefetching without allowing a window of large frames to consume too much of the aggregate buffer space.

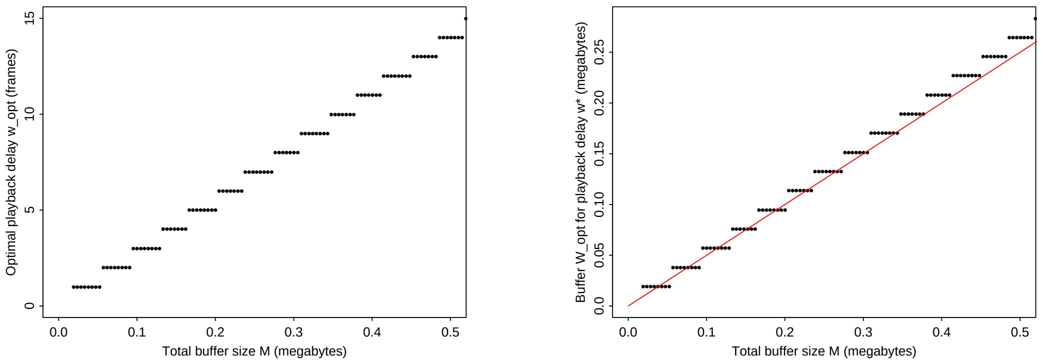
Figure 8(b) investigates the relationship between the total memory size M and the minimum buffer requirement for a window of w^* frames. The buffer requirement W^* is consistently just over half of the available memory M. Conceptually, the optimal start-up delay w^* is typically the smallest window size w that removes the buffer size constraints, resulting in $L_k = D_{k-w}$ and $U_k = D_k$. Smaller values of w limit prefetching since the ingress node cannot prefetch sufficient data to capitalize on the egress buffer space; similarly, larger values of w result in a large buffer requirement W^* 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 $W^* \geq M/2$. This serves as a useful heuristic for minimizing the peak and variability of the bandwidth requirements.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we have investigated bandwidth smoothing across a tandem of nodes, where a service provider has control over a subset of the path between the video server and the client site. By reducing the tandem system to a collection of independent single-link problems, we show that the majorization algorithm[8] can be used to compute an optimal schedule for each of the nodes. Then, by characterizing the shape of the peak-bandwidth curve, we show that simple binary-search algorithms can be used to optimize the allocation of buffer space and start-up delay. Finally, simulation experiments with full-length MPEG and motion-JPEG traces verify the analytic results and suggest effective heuristics for allocating buffer space and start-up latency in an internetwork.

As part of ongoing work[14], we have proven that a symmetric division of buffer space (i.e., $b = B = M/2$) always minimizes the peak rate of the smoothed video stream when $\mathbf{A} \equiv \mathbf{D}$. More generally, we have also determined how to compute the optimal buffer allocation and start-up delay for arbitrary arrival vectors, subject to a constraint on the peak transmission rate in the transmission schedule. Finally, we are also applying the tandem smoothing model to the transmission of *live* video, where the ingress node does not know the sizes of future frames; this work extends the initial approach in[15] to incorporate constraints on the size of the ingress smoothing buffer. Further generalizations of the tandem smoothing model can include other practical constraints, such as delay jitter and efficient video distribution in a multicast tree.

Acknowledgments: The authors would like to acknowledge discussions with and constructive feedback on earlier versions of this paper from Jayanta Dey and Subhabrata Sen.



(a) Optimal start-up delay w^* (b) Playback buffer requirement W^*

Figure 8. Optimal Buffer Allocation and Start-up Delay: These graphs plot the optimal start-up delay w^* and the corresponding minimum buffer requirement W^* for different buffer sizes M for a motion-JPEG encoding of *E.T. – The Extra Terrestrial.* In (b), the solid line corresponds to $M = 2W$.

REFERENCES

1. E. P. Rathgeb, "Policing of realistic VBR video traffic in an ATM network," *International Journal of Digital and Analog Communication Systems* **6**, pp. 213–226, October–December 1993.
2. M. Garrett and W. Willinger, "Analysis, modeling and generation of self-similar VBR video traffic," in *Proc. ACM SIGCOMM*, September 1994.
3. A. R. Reibman and A. W. Berger, "Traffic descriptors for VBR video teleconferencing over ATM networks," *IEEE/ACM Transactions on Networking* **3**, pp. 329–339, June 1995.
4. O. Rose, "Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems," in *Proc. Conference on Local Computer Networks*, pp. 397–406, October 1995.
5. T. Ott, T. Lakshman, and A. Tabatabai, "A scheme for smoothing delay-sensitive traffic offered to ATM networks," in *Proc. IEEE INFOCOM*, pp. 776–785, May 1992.
6. W. Feng and S. Sechrest, "Critical bandwidth allocation for delivery of compressed video," *Computer Communications* **18**, pp. 234–242, October 1995.
7. W. Feng, F. Jahanian, and S. Sechrest, "Optimal buffering for the delivery of compressed prerecorded video," in *Proc. IASTED/ISMM International Conference on Networks*, January 1995. Extended version to appear in *ACM/Springer-Verlag Multimedia Systems Journal.*
8. 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. ACM SIGMETRICS*, pp. 222–231, May 1996.
9. 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**, pp. 1087–1098, August 1996.
10. J. Zhang and J. Hui, "Multi-node buffering and traffic smoothing in VBR video transmission," Tech. Rep. CAIP-TR-217, Rutgers University, 1997.
11. A. W. Marshall and I. Olkin, *Inequalities: Theory of Majorization and Its Applications*, Academic Press, 1979.
12. G. Sanjay, "Work-ahead smoothing of video traffic for interactive multimedia applications." Bachelor of Technology Project Report, Computer Science and Engineering, Indian Institute of Technology, Madras, May 1997.
13. W. Feng and J. Rexford, "A comparison of bandwidth smoothing techniques for the transmission of prerecorded compressed video," in *Proc. IEEE INFOCOM*, pp. 58–66, April 1997.
14. J. Rexford and D. Towsley, "Smoothing variable-bit-rate video in an internetwork." Extended version, available via `http://www.research.att.com/~jrex`, July 1997.
15. J. Rexford, S. Sen, J. Dey, W. Feng, J. Kurose, J. Stankovic, and D. Towsley, "Online smoothing of live, variable-bit-rate video," in *Proc. Network and OS Support for Digital Audio and Video*, pp. 249–257, May 1997.