Impact of Video Scheduling on Bandwidth Allocation for Multiplexed MPEG Streams*

Marwan Krunz and Satish K. Tripathi
Institute for Advanced Computer Studies
Department of Computer Science
University of Maryland
College Park, MD 20742
Email: [krunz,tripathi]@cs.umd.edu

Abstract

We present efficient schemes for scheduling the delivery of VBR MPEG-compressed video with stringent QoS requirements. Video scheduling is being used to improve bandwidth allocation at a video server that uses statistical multiplexing to aggregate video streams prior to transporting them over a network. A video stream is modeled using a traffic envelop that provides a deterministic time-varying bound on the bit rate. Because of the periodicity in which frame types in an MPEG stream are typically generated, a simple traffic envelop can be constructed using only five parameters. Using the traffic-envelop model, we show that video sources can be statistically multiplexed with an *effective bandwidth* that is often less than the source peak rate. Bandwidth gain is achieved without sacrificing the stringency of the requested QoS. The effective bandwidth depends on the *arrangement* of the multiplexed streams, which is a measure of the lag between the GOP periods of various streams. For homogeneous streams, we give an optimal scheduling scheme for video sources at a VOD server that results in the minimum effective bandwidth. We give the form of the "best" arrangement for which the scheduling is optimal. For heterogeneous sources, a sub-optimal scheduling scheme is given which achieves acceptable bandwidth gain. Numerical examples based on traces of MPEG-coded movies are used to demonstrate the effectiveness of our schemes.

Keywords: bandwidth allocation, MPEG, statistical multiplexing, video-on-demand

* This research was supported by NSF grant # CCR 9318933, and by a Shared University Research IBM grant.

1 Introduction

The advent of broadband networks such as BISDN/ATM spurred a strong interest in a variety of multimedia applications, including video-on-demand (VOD) and multimedia-conferencing. The bulk of the traffic generated by these applications consists of digital video that is stored and transported over the network in a compressed format. Providing network support for video traffic without underutilizing bandwidth resources is a major challenge in traffic management, particularly when a video stream is transported at a constant or near-constant frame rate, in which case it exhibits a variable bit rate (VBR) that depends on the scene dynamics and the compression technique. Although ATM networks are expected to provide a VBR network service [2], it is unlikely that such a service will support deterministic quality-of-service (QoS) guarantees.

In this paper, we present efficient bandwidth allocation schemes for VBR video with stringent deterministic QoS (e.g., no cell losses and small, bounded delay). We consider MPEG-coded streams that are compressed and stored on disk prior to their transmission, which is the case in VOD systems where clients can remotely browse through a library of pre-recorded movies and request a particular movie to be delivered over the network. We focus on the issue of efficient bandwidth allocation for the delivery of pre-recorded video. Other related issues such as storage, media synchronization, and operating system support have been extensively researched in other papers (for example, [15, 14, 12]), and will not be addressed here. Allocation efficiency is measured by the *effective bandwidth* per stream, or equivalently, the maximum number of connections that can be simultaneously transported using some fixed total bandwidth. To provide deterministic guarantees, only constant-bit-rate (CBR) or piecewise-CBR (*renegotiated*) [4] network services can be used to transport the VBR traffic from a video server.

Efficient bandwidth allocation for VBR traffic transported using a CBR or a piecewise-CBR network service requires reducing the variability of the bit rate. In principle, two fundamental approaches can be used for variability reduction: *temporal averaging (smoothing)* on a stream-by-stream basis and *spatial averaging (or aggregation)* by means of statistical multiplexing (**SM**). For real-time applications (e.g., teleconferencing), temporal averaging is done using a FIFO buffer, which video frames are sent to just after their compression [10, 6, 13]. This approach introduces variable delay that is caused by buffering. For stored video, temporal averaging can be done by a *work-ahead* approach [1, 11, 17] in which frames are sent ahead of their playback time. A smoothing algorithm that attempts to minimize some function of the bit-rate variation is used to determine an appropriate transmission schedule for a video stream. Although smoothing by *work-ahead* does not introduce queueing delays, it often requires exact knowledge of the end-to-end network delay to avoid buffer overflow and/or underflow at the client side. In practice, network delay cannot be

exactly determined since it varies depending on network conditions.

As an alternative to temporal smoothing, we propose the use of SM to reduce the variability of VBR video *while providing stringent deterministic guarantees*. Typically, SM has been used in ATM networks to improve the utilization by allowing various streams to share buffer and bandwidth resources on demand. Bandwidth gain is attained by allocating a total amount of bandwidth that is less than the sum of the peak rates of the individual streams. Thus, the aggregate input rate at the multiplexer can temporarily exceed the output rate. Such conventional use of SM results in possible cell queueing and buffer overflow, the amounts of which depend on the model being used to characterize the traffic. Because typical traffic models are stochastic in nature, it is often believed that SM can be used advantageously only when providing statistical guarantees. Contrary to this general belief, we show that in the case of VBR MPEG-coded video traffic, SM can be used to improve bandwidth utilization while supporting stringent and deterministic QoS guarantees. For this purpose, we model an MPEG source using a traffic envelop which provides a deterministic upper bound on the bit rate. The use of traffic envelops to characterize MPEG video sources was first mentioned in [7]. By exploiting the periodic manner in which various types of frames are generated by an MPEG encoder, we provide a simple time-varying traffic envelop that is less conservative than a constant-peak-rate characterization. Based on such characterization, we present simple recursive procedures for computing the effective bandwidth of multiplexed video streams that guarantees no losses and small delay. In most scenarios, the effective bandwidth is less than the source peak rate, and some bandwidth gain can be realized. The amount of this gain depends on the *arrangement* of the multiplexed streams, which is a measure of the synchronization among the Group-of-Pictures (GOP) patterns of the various streams. By appropriately scheduling the starting times of MPEG streams at the server, it is possible to achieve significant bandwidth gain from SM without sacrificing the stringency of the QoS requirements. For homogeneous sources, which are characterized by the same envelop, we provide the optimal scheduling scheme for video sources at a VOD server that results in the minimum effective bandwidth. For heterogeneous traffic envelops, we provide a sub-optimal scheme for the scheduling of video connections at a server. Numerical examples based on several real video traces from actual movies (compressed with MPEG-I encoders) are used to illustrate the advantages of the proposed scheduling schemes.

The rest of the paper is structured as follows. Section 2 describes the traffic model used in this paper. The effective bandwidth for video streams is introduced in Section 3. Online procedures for computing the effective bandwidth are described in Section 4. Efficient scheduling schemes of video streams at a VOD server are presented in Section 5 for homogeneous and heterogeneous streams. Our main results are summarized in Section 6.

2 Video Source Model

A standard MPEG encoder generates three types of compressed frames: Intra-coded (I), Predictive (P), and Bidirectional (B) frames. Different combinations of compression modes are used to encode the different frame types. In general, I frames are larger in size than P frames which, in turn, are larger than B frames (the frame size refers to the number of bits in an encoded frame). For VBR constant-quality video, frames are compressed at a constant frame rate (e.g., 30 f/s). When compressing a video sequence, typical MPEG encoders use a pre-defined GOP pattern to determine frame types. The specification of a fixed GOP pattern prior to compression is not required by the MPEG standards, but is often used to simplify the implementation of the encoder. Also, fixing the GOP pattern results in a more predictive traffic behavior that can be exploited in resource allocation. We assume throughout this paper that each MPEG stream is compressed using one GOP pattern. Different streams can have different GOP patterns. An example of a stream that uses a single GOP pattern is shown in Figure 1. In practice, the encoding and transmission orders

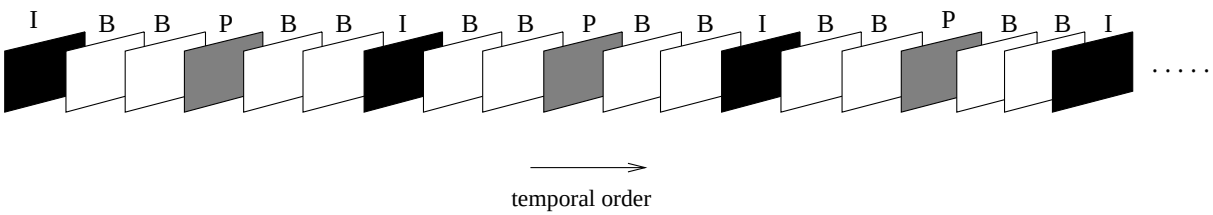

Figure 1: Example of an MPEG stream with a fixed GOP pattern.

of MPEG frames are different due to the non-causal nature of B-frames compression. Nevertheless, after the first GOP, both the transmission and encoding orders look alike with respect to frame types. For simplicity, we will ignore the first GOP in the transmission order, and assume that a stream is periodic in its GOP pattern. Furthermore, we will primarily deal with *regular* GOP patterns in which the number of *successive* B frames in a given stream is constant. This restriction is not crucial for effective-bandwidth allocation, but is needed to obtain tractable analytical results. Regular GOP patterns can be specified by two parameters:

- L : number of frames between two consecutive I frames in an MPEG stream.
- Q : number of frames between an I frame and the subsequent I/P frame (whichever comes first) in an MPEG stream.

The regularity of the GOP pattern implies that L is a multiple of Q. Notice that it is possible to have $L = Q = 1$, in which case only I frames are generated (i.e., JPEG-like stream).

To provide deterministic guarantees for video traffic, we characterize a video stream by a traffic envelop that is similar, to some extent,[1] to the D-BIND model introduced in [7]. In our model, a stream consists of a sequence of compressed frames generated at a constant frame rate according to a *regular* GOP pattern. The traffic envelop for the ith stream, s_i, is given by the time-varying periodic function $\overline{b}_i(t)$ (with period $L^{(i)}$) which is parameterized by the 5-tuple $\boldsymbol{E}_i = \left(I_{max}^{(i)},\ P_{max}^{(i)},\ B_{max}^{(i)},\ L^{(i)},\ Q^{(i)}\right)$, where the first three parameters denote, respectively, the maximum sizes of I, P, and B frames in s_i (frame sizes are given in ATM cells which are evenly distributed over a frame period). $L^{(i)}$ and $Q^{(i)}$ describe the GOP pattern of s_i. An example of $\overline{b}_i(t)$ with constant parameters is shown in Figure 2. Throughout this paper, it is assumed that $\boldsymbol{E}_i$ satisfies $I_{max}^{(i)} > P_{max}^{(i)} > B_{max}^{(i)}$ for all i.

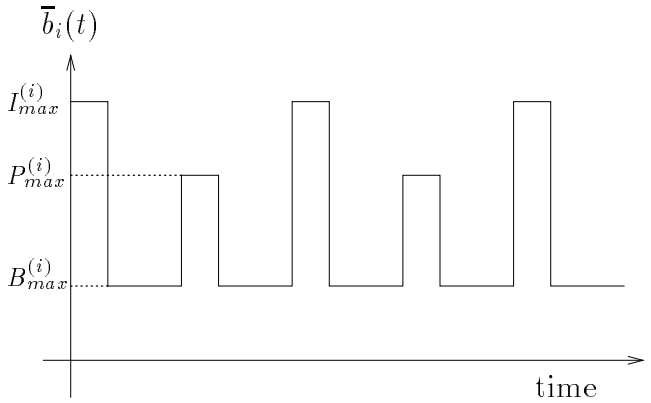

Figure 2: Traffic envelop with $L^{(i)} = 6$ and $Q^{(i)} = 3$.

3 Effective Bandwidth for Multiplexed MPEG Streams

In this section, we demonstrate the advantage of statistical multiplexing in reducing the bandwidth requirements of MPEG video streams, while providing stringent deterministic guarantees. We quantify the attainable bandwidth gain using the notion of effective bandwidth. This notion, also known as *equivalent capacity*, was investigated in several previous studies within a stochastic framework [5]. In this paper, the effective bandwidth is defined within a deterministic framework to guarantee zero cell loss rate and negligible queueing delays.

Consider N MPEG streams, $s_1, \ldots, s_N$, with stringent deterministic QoS requirements to be transported over an ATM network. Typically, such requirements are met by allocating bandwidth based on the peak bit rate of each source. Let $\overline{b}_i(t)$ be the traffic envelop for s_i, parameterized

[1] The D-BIND model provides a time-invariant bound on the cumulative arrivals. Our model is more restrictive since it provides a time-varying bound on the *arrival rate*.

by $\boldsymbol{E}_i$ (assume, for now, that $L^{(i)} = L$ for all i). First, let's consider the general case in which a video stream s_i starts sending frames into the multiplexer at any arbitrary time t_i. This situation normally occurs at intermediate switching nodes. When multiplexing takes place at a VOD server, the server can, to some extent, control the starting times of video sources, which we exploit in Section 5 to provide efficient scheduling of video sources. Let $t_1 \stackrel{\triangle}{=} 0$, so that the first stream is used as a time reference. The lag in frame durations between a GOP of s_1 and the next immediate GOP of s_i is given by $u_i = t_i \bmod L$. The vector $\boldsymbol{u} = (u_2, u_3, \ldots, u_N)$, referred to as the *arrangement*, completely specifies the synchronization structure for the N streams with regard to their GOPs (note that $u_1 \stackrel{\triangle}{=} 0$). Denote the traffic envelop for the *superposition* of the N streams by $\overline{b}_{tot}(t)$, where $\overline{b}_{tot}(t) = \sum_i \overline{b}_i(t - u_i)$. Note that $\overline{b}_{tot}(t)$ is periodic with period L. We define the effective bandwidth (per stream) for N multiplexed streams with arrangement $\boldsymbol{u}$ by:

$$C(\boldsymbol{u}, N) \stackrel{\triangle}{=} \frac{1}{N} \max_{t \geq 0} \overline{b}_{tot}(t) = \frac{1}{N} \max_{t \geq 0} \left(\sum_{i=1}^{N} \overline{b}_i(t - u_i) \right) \tag{1}$$

Because of the periodicity of $\overline{b}_{tot}(t)$, it is sufficient to take the maximum over an interval of length L. Equation (1) can also be written as:

$$C(\boldsymbol{u}, N) = \frac{\sum_{j \in \Lambda_I} I_{max}^{(j)} + \sum_{j \in \Lambda_P} P_{max}^{(j)} + \sum_{j \in \Lambda_B} B_{max}^{(j)}}{N} \tag{2}$$

where $\Lambda_I, \Lambda_P, \Lambda_B$ are pairwise mutually disjoint sets with $\Lambda_I \bigcup \Lambda_P \bigcup \Lambda_B = \{s_1, \ldots, s_N\}$.

When $L^{(i)}$ varies with i, (1) and (2) are still valid with $\overline{b}_{tot}(t)$ having a period $\widetilde{L}$, where

$$\widetilde{L} = \textit{least common multiple of} \{L^{(1)}, L^{(2)}, \ldots, L^{(N)}\} \tag{3}$$

and the maximization in (1) is taken over a time interval of length $\widetilde{L}$ (also in the definition of u_i, L should be replaced by $\widetilde{L}$).

Given that $I_{max}^{(i)} > P_{max}^{(i)} > B_{max}^{(i)}$ for all i, it is easy to see that $NC(\boldsymbol{u}, N) < \sum_i I_{max}^{(i)}$ for most values of $\boldsymbol{u}$. One obvious case for which $NC(\boldsymbol{u}, N) = \sum_i I_{max}^{(i)}$ is when $\boldsymbol{u}$ is the zero vector (i.e., all streams start simultaneously). As an illustrative example, consider the situation in Figure 3, where two streams are statistically multiplexed. Suppose that both s_1 and s_2 are characterized by the same traffic envelop $\overline{b}(t)$ with $L = 6$, $Q = 3$, and $\boldsymbol{u} = (0, 1)$ (i.e., s_2 starts $1 + jL$ frame periods after the start of s_1, for any nonnegative integer j). Then,

$$C(\boldsymbol{u}, 2) \stackrel{\triangle}{=} \frac{1}{N} \max_{t \geq 0} \overline{b}_{tot}(t) = \frac{1}{N} \max_{t \geq 0} \left(\overline{b}(t) + \overline{b}(t - 1) \right) = \frac{I_{max} + B_{max}}{2} < I_{max}$$

By superposing the two streams and allocating bandwidth for the aggregate traffic, the required bandwidth per source decreases from I_{max} (under source-peak-rate allocation) to $(I_{max} + B_{max})/2$. A small buffer of N cells is needed at the input to the multiplexer in case cells from several sources arrive simultaneously. Note that bandwidth gain from SM is obtained from spatial averaging, and not temporal averaging.

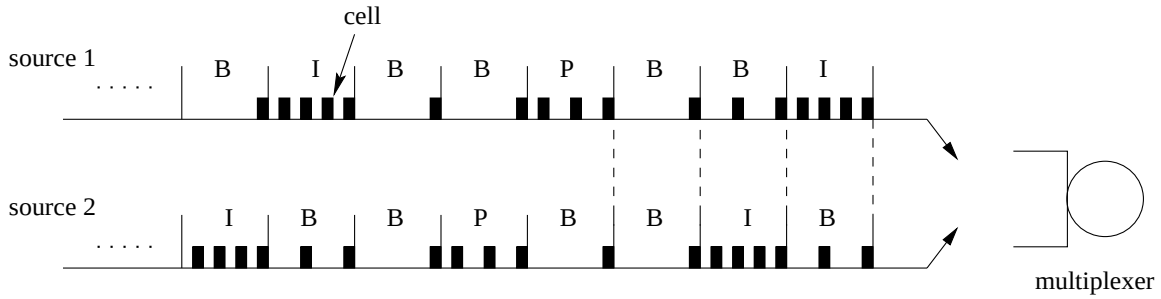

Figure 3: An example of the statistical multiplexing of two MPEG streams with aligned boundaries.

4 Online Computation of Effective Bandwidth

We now give efficient procedures for recursive computation of $C(\boldsymbol{u}, N)$ or, in some cases, an upper bound on it. In practice, the bandwidth allocated to video traffic at a multiplexing node must be updated dynamically upon the admittance of a new video connection or the termination of an ongoing one. Allocation based on $C(\boldsymbol{u}, N)$ can be used when the multiplexing node does not control the starting times of video connections, as in the case of intermediate switches. We consider video streams with heterogeneous traffic envelops, each with constant parameters. The following two cases are considered in the computation of $C(\boldsymbol{u}, N)$ and its upper bound, depending on the values of the elements of $\boldsymbol{u}$.

4.1 Aligned Boundaries Case

Suppose that $\boldsymbol{u}$ takes only integer values in $\{0, 1, \ldots, \widetilde{L} - 1\}$, implying that frame boundaries of different sources are aligned in time (as in Figure 3). Alignment of frame boundaries can be enforced by introducing a fixed amount of delay (less than one frame period) in the path of a video connection before entering the multiplexer. Such delay has negligible impact on the perceived QoS. Recall that $\overline{b}_{tot}(t)$ is piecewise-constant and periodic with period $\widetilde{L}$. Given that frame boundaries are aligned, for any integer k, $\overline{b}_{tot}(t)$ is constant for all $t \in (k, k+1)$ (time is measured in frame periods, a frame period = 1/30 sec). Accordingly, $\overline{b}_{tot}(t)$ can be written as a discrete-time periodic sequence of $\widetilde{L}$ values, for which $C(\boldsymbol{u}, N)$ can be recursively computed online. Because of the periodicity of $\overline{b}_{tot}(t)$,

computation of $C(\boldsymbol{u}, N)$ requires only maintaining the values of the traffic envelops for the first $\widetilde{L}$ slots (from 0 to $\widetilde{L}-1$). Such slots are referred to as *phases*. Let $\overline{b}_{i,j}$ be the value of $\overline{b}_i(t)$ during phase j. Thus,

$$\overline{b}_{i,j} = \overline{b}_i(\tau - u_i) \;\; \forall \tau \in (j, j+1) \tag{4}$$

(for $t \leq 0$, we extend $\overline{b}_i(t)$ along the negative time axis). To compute $C(\boldsymbol{u}, N)$, the multiplexing node maintains a matrix $M = [m_{ij}]$ of size $N \times \widetilde{L}$. Each video stream is associated with one row in the table. For $i = 1, \ldots, N$, and $j = 1, \ldots, \widetilde{L}$, $m_{ij} = \overline{b}_{i,j-1}$. In addition, the node maintains a row vector $V = \left[v_1, \ldots, v_{\widetilde{L}}\right]$, where

$$v_j = \sum_{i=1}^{N} m_{ij} \;\; \forall \;\; j \tag{5}$$

which gives the value of $\overline{b}_{tot}(t)$ during phase $j-1$. Now, $C(\boldsymbol{u}, N)$ is simply given by:

$$C(\boldsymbol{u}, N) = \frac{1}{N} \max_{0 \leq j \leq \widetilde{L}-1} v_j \tag{6}$$

Upon the arrival of the $(N+1)$th stream, a row is added to M based on $\overline{b}_{N+1}(t)$ and u_{N+1}. For heterogeneous streams with different $L^{(i)}$ values, the updating of M can be simplified by choosing $\widetilde{L}$ based on all anticipated values of $L^{(i)}$ (which are few in practice). Thus, the number of columns of M is kept constant and only the rows are added or deleted during the updating process. The effective bandwidth is re-computed by updating V (using $v_j := v_j + m_{N+1,j}$), and then applying (6) with $N+1$ replacing N. A similar procedure is used to update $C(\boldsymbol{u}, N)$ when an ongoing connection is terminated. Clearly, very few operations are needed to re-compute the effective bandwidth upon adding/dropping a video stream.

4.2 Non-Aligned Boundaries Case

As a generalization of the previous case, suppose that $u_2, u_3, \ldots$, take real values in $[0, \widetilde{L})$. Because $\boldsymbol{u}$ is not necessarily integer-valued, using a table of $\widetilde{L}$ columns as in the previous case is not sufficient for computing $C(\boldsymbol{u}, N)$, since $\overline{b}_{tot}(t)$ could take up to $N\widetilde{L}$ different values within a period of $\widetilde{L}$ (compared to $\widetilde{L}$ values in the aligned boundaries case). With N continuously varying, the size of the table and the cost of updating it become impractical for online computations. Instead, we provide an upper bound on the effective bandwidth, which can be efficiently updated. As before, time is slotted in units of frame periods with slots being locally synchronized. A matrix $\widehat{M} = [\widehat{m}_{ij}]$ of dimensions $2N \times \widetilde{L}$ is maintained at the node. Each ongoing stream, s_i, is associated with two adjacent rows of $\widehat{M}$; the $(2i-1)$th and the $(2i)$th rows. In the first row, the $\widetilde{L}$ possible values of $\overline{b}_i(t)$ are recorded assuming that the I frames of s_i are *exactly* aligned with phase $\lfloor u_i \rfloor$. The second

row contains the $\widetilde{L}$ values of $\overline{b}_i(t)$ as if the I frames of s_i are *exactly* aligned with phase $\lceil u_i \rceil$ mod $\widetilde{L}$. Hence,

$$\widehat{m}_{ij} = \begin{cases} \overline{b}_{(i+1)/2,j-1} & \text{if } i \text{ is odd} \\ \overline{b}_{i/2,j-2} & \text{if } i \text{ is even} \end{cases} \tag{7}$$

where $\overline{b}_{i,j}$ is now defined by $\overline{b}_{i,j} \stackrel{\triangle}{=} \overline{b}_i(\tau - \lfloor u_i \rfloor)$ for all$\tau \in (j, j+1)$. In addition to $\widehat{M}$, the node maintains a row vector $\widetilde{V} = \left[\widetilde{v}_1, \ldots, \widetilde{v}_{\widetilde{L}}\right]$, where

$$\widetilde{v}_j = \sum_{i=1}^{N} \max\left\{\widehat{m}_{2i-1,j},\ \widehat{m}_{2i,j}\right\} \quad \forall\ j \tag{8}$$

$\widetilde{v}_j$ gives the peak bit rate of the aggregate traffic during phase $j-1$.

Proposition 1 *An upper bound on $C(\boldsymbol{u}, N)$ is given by:*

$$\overline{C}(\boldsymbol{u}, N) = \frac{1}{N} \max_{1 \le j \le \widetilde{L}} \widetilde{v}_j \tag{9}$$

The proof of Proposition 1 is given in the appendix. Upon the arrival of a new video stream to a node with N ongoing streams, two rows are added to $\widehat{M}$ based on $\overline{b}_{N+1}(t)$ and u_{N+1}, and $\widetilde{v}_j$ is updated using

$$\widetilde{v}_j := \widetilde{v}_j + \max\left\{\widehat{m}_{2(N+1)-1,j},\ \widehat{m}_{2(N+1),j}\right\} \quad \forall\, j \tag{10}$$

The bound on the effective bandwidth is updated using (9) (with $N+1$ replacing N). When an ongoing connection, s_i, is terminated, $\widetilde{v}_j$ is updated using

$$\widetilde{v}_j := \widetilde{v}_j - \min\left\{\widehat{m}_{2i-1,j},\ \widehat{m}_{2i,j}\right\} \quad \forall\, j \tag{11}$$

The following example shows the benefits of $C(\boldsymbol{u}, N)$ and $\overline{C}(\boldsymbol{u}, N)$ allocation. Consider three homogeneous sources with a common traffic envelop that is parameterized by $\boldsymbol{E} = (894, 742, 157, 15, 3)$ (frame sizes in cells). These values are computed from *Wizard of Oz* trace (see Section 6). Figure 4 gives the percentage of $C(\boldsymbol{u}, N)/I_{max}$ for different values of $\boldsymbol{u}$ in the aligned boundaries case. For simplicity, $\boldsymbol{u} = (u_1, u_2, u_3)$ is varied by varying u_3 over the range $\{0, \ldots, L-1\}$ and taking $u_2 = 0, 1, 2$ ($u_1 \stackrel{\triangle}{=} 0$). Except when $\boldsymbol{u} = (0,0,0)$, the effective bandwidth is less than the source peak rate. In fact, for some values of $\boldsymbol{u}$ the allocated bandwidth is less than 50% of the source peak rate. In the case of non-aligned boundaries, Figure 5 depicts $\overline{C}(\boldsymbol{u}, N)/I_{max}$ (in percentage) for different

values of $\boldsymbol{u}$ (u_3 is varied continuously in $[0, L)$ and $u_2 = 0, 1$) As expected, $\overline{C}(\boldsymbol{u}, N)$-based allocation results in less gain than $C(\boldsymbol{u}, N)$-based allocation. For homogeneous streams, if we assume that $\boldsymbol{u}$ is random with a discrete uniform distribution over $\{0, 1, \ldots, L-1\}$ for the aligned boundaries case, and a continuous uniform distribution over $[0, L)$ in the non-aligned boundaries case, then $\Pr\{C(\boldsymbol{u}, N) = I_{max}\} = 1/L^{N-1}$ and $\Pr\{\overline{C}(\boldsymbol{u}, N) = I_{max}\} = 2/L^{N-1}$.

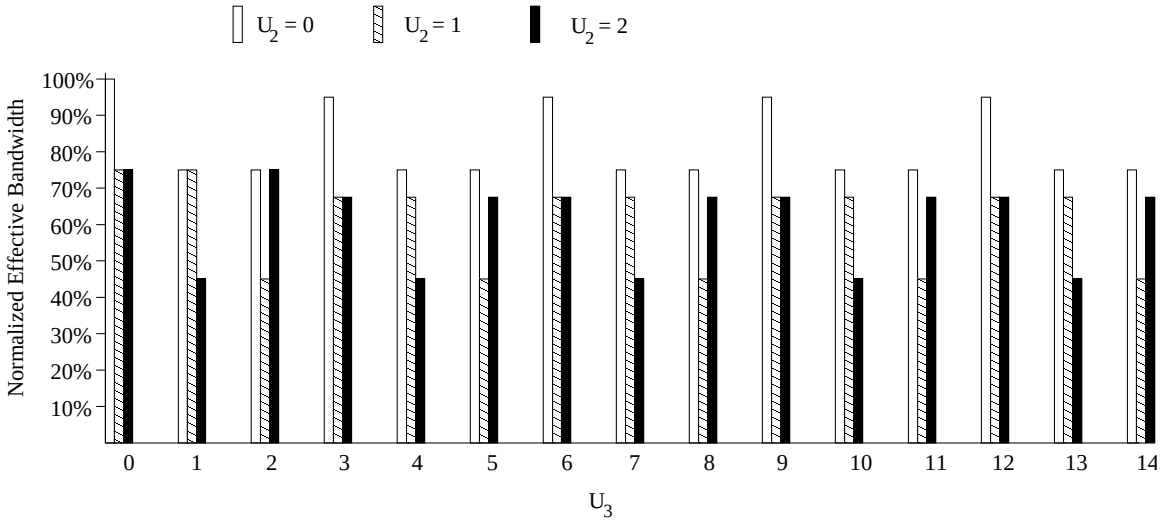

Figure 4: Percentage of $C(\boldsymbol{u}, N)/I_{max}$ for different values of $\boldsymbol{u}$ in the aligned boundaries case.

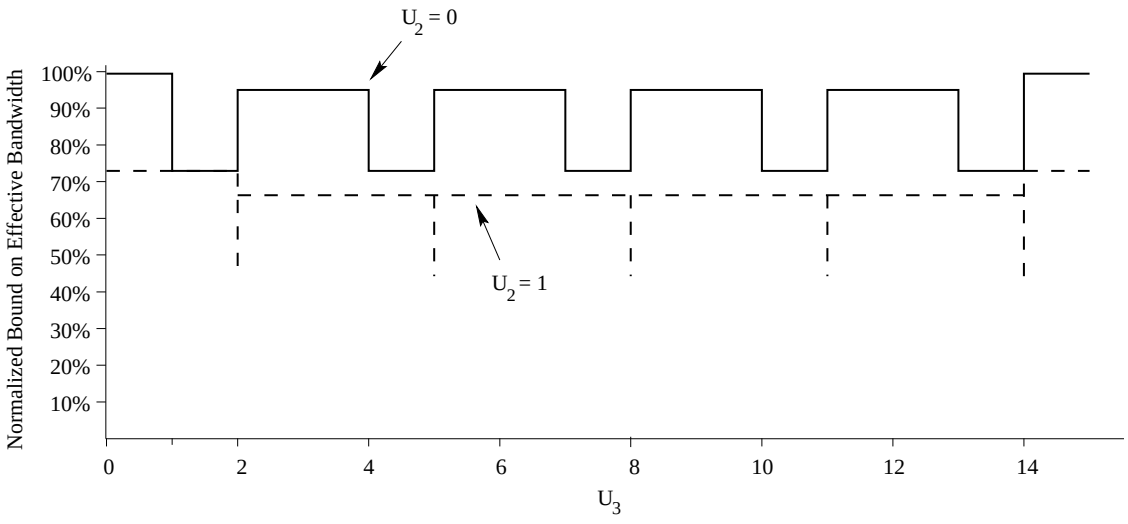

Figure 5: Percentage of $\overline{C}(\boldsymbol{u}, N)/I_{max}$ for different values of $\boldsymbol{u}$ in the non-aligned boundaries case.

5 Efficient Scheduling of Video Streams

Since the effective bandwidth depends on the arrangement of the multiplexed video streams, it is natural to look for the *best* arrangement that results in the *minimal* effective bandwidth. A best arrangement can be used in a VOD system to provide optimal scheduling of video streams for transmission over the network. The server in a VOD system has some flexibility in controlling the starting times of new connections. This flexibility allows the server to efficiently schedule the transmission of requested movies at the expense of delaying the start of a new stream by no more than a GOP period (1/2 second), which is not noticeable by the client. Efficient scheduling schemes for video are given in this section. In the homogeneous case (identical envelops), our scheduling scheme is proven to be optimal. A sub-optimal scheme is provided for heterogeneous envelops.

5.1 Optimal Scheduling in the Homogeneous Case

Suppose that the same traffic envelop $\overline{b}(t)$ with parameters $\boldsymbol{E} = (I_{max},\ P_{max},\ B_{max},\ L,\ Q)$ is used to characterize every stream at the multiplexer. This case, which we refer to as the *homogeneous* case, applies directly to a situation in which several copies of the same movie are requested at different instants of time. In addition, a homogeneous case results from using a slightly conservative common traffic envelop to characterize heterogeneous streams with relatively close but different maximum frame sizes and similar L and Q values. Such an envelop is constructed by taking I_{max} to be the largest $I_{max}^{(i)}$ over all i, and similarly for P_{max} and B_{max}.
We define the minimal effective bandwidth by:

$$C_{min}(N) = C(\boldsymbol{u}^*, N) \stackrel{\triangle}{=} \min_{u \in \mathcal{U}} C(\boldsymbol{u}, N) \tag{12}$$

where $\mathcal{U}$ is the set of all possible *distinct* arrangements of N streams, and $\boldsymbol{u}^*$ is a *best* arrangement that results in the minimal effective bandwidth. Using combinatorial techniques, it can be shown that the size of $\mathcal{U}$ is

$$|\mathcal{U}| = \sum_{i=1}^{m} \left(\begin{array}{c} L \\ i \end{array} \right) \left(\begin{array}{c} N-2 \\ i-1 \end{array} \right) \quad \text{where } m = \min\{N-1, L\} \tag{13}$$

which increases rapidly with N. Therefore, obtaining $C_{min}(N)$ from (12) by exhaustive search is computationally prohibitive for moderate and large N. Instead, we give a closed-form expression for $\boldsymbol{u}^*$. We assume, without loss of generality, that frame boundaries are aligned. This assumption is justified by the fact that the effective bandwidth for an arrangement with non-aligned boundaries can be shown to be greater than or equal the effective bandwidth for some arrangement with aligned

boundaries. Thus, $\boldsymbol{u}^*$ is necessarily an arrangement with aligned boundaries. Table 1 provides the form of $\boldsymbol{u}^*$ and the expression for $C_{min}(N)$. Although the structure of $\boldsymbol{u}^*$ is quite intuitive, proving its optimality is not trivial. The optimality of $\boldsymbol{u}^*$ and the associated expression for $C_{min}(N)$ in (12) are proved in the next section. Note that $\boldsymbol{u}^*$ is not necessarily unique.

A best arrangement of N streams, $N \geq 1$, that have similar traffic envelops is given by:

$$\boldsymbol{u}^* = (\underbrace{0, 1, 2, \ldots, L-1, 0, 1, 2, \ldots, L-1, \ldots,}_{w \text{ times}} 0, 1, 2, \ldots, N - wL - 1) \tag{14}$$

and the minimal effective bandwidth is given by:

$$C_{min}(N) = C(\boldsymbol{u}^*, N) = \frac{(w+1)I_{max} + (m-w)P_{max} + (N-1-m)B_{max}}{N} \tag{15}$$

where

$$w \stackrel{\triangle}{=} \text{ largest nonnegative integer } k \text{ that satisfies } N > kL$$
$$m \stackrel{\triangle}{=} \text{ largest nonnegative integer } k \text{ that satisfies } N > kQ$$

Table 1: A best arrangement of N streams and the associated minimal effective bandwidth.

Given that N ongoing streams are scheduled according to $\boldsymbol{u}^*$, a new stream can be added to the existing ones resulting in a *best* arrangement of $(N+1)$ streams without disrupting the original structure of the N streams. In other words, $\boldsymbol{u}^*$ of $(N+1)$ streams can be obtained by simply concatenating a single number to $\boldsymbol{u}^*$ of N streams. When N streams are arranged according to $\boldsymbol{u}^*$ and $N \leq L$, the removal of *any* stream will still result in a best arrangement of $N-1$ streams. When $N > L$, only the removal of certain streams preserves the optimality of the arrangement.

As N increases, $C_{min}(N)$ decreases slowly in a non-monotonic manner. The *asymptotic* value of $C_{min}(N)$ can be obtained by taking the limit of $C_{min}(N)$ in (15) with respect to N. For large N, $w \approx N/L$ and $m \approx N/Q$. Thus,

$$C^*_{min} \stackrel{\triangle}{=} \lim_{N \to \infty} C_{min}(N) = (1/L)I_{max} + (1/Q - 1/L)P_{max} + (1 - 1/Q)B_{max} \tag{16}$$

In fact, this limiting value is reachable when $N = kL$, for $k = 1, 2, 3, \ldots$, implying that the highest gain from multiplexing in the homogeneous case is achieved whenever the number of multiplexed streams is a multiple of L.

5.2 Proof of Optimality

In this section, we prove the optimality of $\boldsymbol{u}^*$ as given in (14). For homogeneous streams with aligned boundaries, $C(\boldsymbol{u}, N)$ can be written as

$$C(\boldsymbol{u}, N) = \frac{1}{N} \max_{0 \le j \le L-1} \left(\sum_{i=1}^{N} \overline{b}_{i,j} \right) \tag{17}$$

where $\overline{b}_{i,j}$ was defined in (4). We can also write $C(\boldsymbol{u}, N)$ in the following form:

$$C(\boldsymbol{u}, N) = \frac{n_I I_{max} + n_P P_{max} + (N - n_I - n_P) B_{max}}{N} \tag{18}$$

for some nonnegative integers n_I, n_P, and n_B with $n_I + n_P + n_B = N$. Only L values of $\overline{b}_{tot}(t)$ (in the first L slots) are needed to compute $C(\boldsymbol{u}, N)$ in the homogeneous case with aligned boundaries. We use the notation $\overline{b}_{tot,i}$ to refer to $\overline{b}_{tot}(\tau)$ for $\tau \in (i, i+1)$. A stream s_i is said to belong to phase k, where $k = 0, \ldots, L-1$, if $u_i = k$, i.e., s_i sends its I frames during phase k. Define

$r_k \stackrel{\triangle}{=}$ number of streams that belong to phase k

$z_k \stackrel{\triangle}{=}$ number of streams that belong to phases that differ from k by a nonzero multiple of Q

r_k and z_k give the numbers of streams sending I and P frames, respectively, during phase k. The following proposition follows directly from the periodicity of the GOP patterns.

Proposition 2 *Consider any two streams s_i and s_j with phases u_i and u_j, $u_i \neq u_j$. If during phase u_i s_j sends a B frame, then during phase u_j s_i sends a B frame. Also, if during phase u_i s_j sends a P frame, then during phase u_j s_i sends a P frame.* □

From Proposition 2, it is easy to see that for any two phases, m and n, with $\mid m - n \mid =$ a multiple of Q, we have $r_m + z_m = r_n + z_n$.

Proposition 3 *In (18), $n_I \ge 1$ for any arrangement $\boldsymbol{u} = (u_1, \ldots, u_N)$.*

Proof (by contradiction): Suppose that $n_I = 0$.
First, consider the case when $n_P = 0$. Then $C(\boldsymbol{u}, N) = NB_{max}/N$. Since $u_1 \stackrel{\triangle}{=} 0$, $r_0 \ge 1$. Thus, during phase 0 the aggregate peak rate $\overline{b}_{tot,0} \ge I_{max} + (N-1)B_{max} > NC(\boldsymbol{u}, N)$, which contradicts the definition of $C(\boldsymbol{u}, N)$.

Next, consider the case when $n_P \geq 1$. Let phase k be the phase for which $\overline{b}_{tot,k}/N = C(\boldsymbol{u}, N)$. By our assumption, $r_k = 0$. Since $n_P \geq 1$, there exists at least one stream, say s_j, with phase j such that $|j-k|$ = a multiple of Q. During phase j, s_j sends I frames. Also, any other stream that sends P frames during phase k will be sending either I frames or P frames during phase j (from Proposition 2). Thus, $\overline{b}_{tot,j} > \overline{b}_{tot,k}$, which contradicts the definition of $C(\boldsymbol{u}, N)$. Hence, $n_I \geq 1$. □

To prove the optimality of $\boldsymbol{u}^*$, we first show that $C(\boldsymbol{u}^*, N)$ is given by the RHS of (15). Then, we show that $C_{min}(N)$ is also given by the RHS of (15). An inspection of (14) reveals that when N streams are arranged according to $\boldsymbol{u}^*$, there are exactly $m+1$ streams whose phases differ, pairwise, by a nonnegative multiple of Q. Among those, $w+1$ streams belong to the same phase (m and w were defined in Table 1). It is obvious that $C(\boldsymbol{u}^*, N)$ is obtained from a phase i in which $r_i = w+1$ and $z_i = m+1-(w+1) = m-w$. Thus,

$$C(\boldsymbol{u}^*, N) = \frac{(w+1)I_{max} + (m-w)P_{max} + (N-1-m)B_{max}}{N} \tag{19}$$

Now consider an arbitrary arrangement $\boldsymbol{u} = (u_1, \ldots, u_N)$. If we can show that $C(\boldsymbol{u}, N)$ satisfies

$$C(\boldsymbol{u}, N) \geq \frac{sI_{max} + lP_{max} + (N-s-l)B_{max}}{N} \tag{20}$$

with $s \geq w+1$ and $s+l \geq m+1$, then $C(\boldsymbol{u}, N)$ must be greater than or equal the RHS of (19), which proves the optimality of $\boldsymbol{u}^*$. To prove (20) for an arbitrary $\boldsymbol{u}$, we consider two cases.

5.2.1 Arbitrary Arrangement with Distinct Elements

Suppose that the elements of $\boldsymbol{u}$ are distinct (i.e., $u_i \neq u_j$ for all $i \neq j$), which is only possible when $N \leq L$ (thus $w = 0$). At least $m+1$ of these streams belong to phases that differ pairwise by a multiple of Q. (In general, for a set of distinct $kX+1$ integers where k and X are nonnegative integers and $X \neq 0$, there are at least $k+1$ integers that differ pairwise by a multiple of X). Thus, $\overline{b}_{tot,j} \geq I_{max} + mP_{max} + (N-1-m)B_{max}$ for some phase j. By definition, $C(\boldsymbol{u}, N) \geq \overline{b}_{tot,i}/N$ for all i, and thus, $C(\boldsymbol{u}, N) \geq \overline{b}_{tot,j}/N \geq (I_{max} + mP_{max} + (N-1-m)B_{max})/N$. Therefore, $C(\boldsymbol{u}, N)$ satisfies (20) with $s = w+1$ and $l = m-w$ ($w = 0$ in this case).

5.2.2 Arbitrary Arrangement with Repeated Elements

Suppose that the elements of $\boldsymbol{u}$ are not distinct. Let

$$\alpha \stackrel{\triangle}{=} \max_{0 \leq j \leq L-1} r_j \tag{21}$$

Clearly, $\alpha \geq \max\{2, w+1\}$. We use the term *chain* to refer to a subset of the N streams whose phases differ pairwise by a multiple of Q (including the ones that belong to the same phase). For example, if $N = 9$, $L = 15$, $Q = 3$, and $\boldsymbol{u} = (0,0,0,1,2,3,4,5,6)$, then the first chain consists of the sources $\{s_1, s_2, s_3, s_6, s_9\}$, the second chain consists of $\{s_4, s_7\}$, and the last chain consists of $\{s_5, s_8\}$ as shown in Figure 6. Here, $\alpha = 3$. Observe that no more than Q chains can exist in any

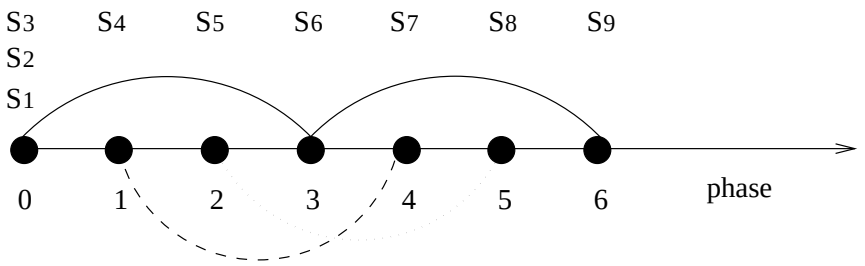

Figure 6: Resulting chains when $\boldsymbol{u} = (0,0,0,1,2,3,4,5,6)$, $L = 15$, and $Q = 3$.

arrangement. Let q be the number of chains in $\boldsymbol{u}$ ($q \leq Q$). Denote these chains by $W_1, W_2, \ldots, W_q$, with corresponding sizes $\eta_1, \eta_2, \ldots, \eta_q$ ($\sum_j \eta_j = N$). From Proposition 2, if two streams that belong to phases m and n are in the same chain, say W_i, then $r_m + z_m = r_n + z_n = \eta_i$. For each chain W_j, let $C_j(\boldsymbol{u}, N)$ be the maximum aggregate peak rate divided by N, with the maximization taken only over the phases of the streams in W_j. For $j = 1, \ldots, q$, $C_j(\boldsymbol{u}, N)$ can be written as

$$C_j(\boldsymbol{u}, N) = \frac{n_I^{(j)} I_{max} + n_P^{(j)} P_{max} + n_B^{(j)} B_{max}}{N} \tag{22}$$

where for all j, $n_I^{(j)}$, $n_P^{(j)}$, and $n_B^{(j)}$ are nonnegative integers; $n_I^{(j)} \geq 1$ (from Proposition 3); and $n_I^{(j)} + n_P^{(j)} + n_B^{(j)} = N$. The total number of streams sending I or P frames during the phase of any stream in W_j is given by η_j. At least one of the chains, say W_1, contains α streams that belong to the same phase, say phase i. Consequently, $C_1(u, N) = \overline{b}_{tot,i}/N$ and $n_I^{(1)} = \alpha$. Based on the definition of $C(\boldsymbol{u}, N)$,

$$C(\boldsymbol{u}, N) = \max_{1 \leq j \leq q} C_j(u, N) \tag{23}$$

We consider two cases, depending on the value of η_1. First, suppose that $\eta_1 \geq m + 1$. Then,

$$C(\boldsymbol{u}, N) \geq C_1(u, N) = \frac{\alpha I_{max} + (\eta_1 - \alpha) P_{max} + (N - \eta_1) B_{max}}{N} \tag{24}$$

Since $\alpha \geq w + 1$ and $\eta_1 \geq m + 1$, $C(\boldsymbol{u}, N)$ satisfies (20), and $\boldsymbol{u}^*$ is optimal.

Next, suppose $\eta_1 < m + 1$. Thus, $\sum_{j=2}^{q} \eta_j \geq N - m$. There must be at least one chain, say W_j,

for which

$$\eta_j \geq \frac{N-m}{q-1} \tag{25}$$

(otherwise, $\sum_{j=2}^{q} \eta_j < N-m$). Accordingly,

$$\eta_j \geq \frac{N-m}{q-1} > \frac{N-N/Q}{q-1} = \frac{N(Q-1)/Q}{q-1} > \frac{N}{Q} > m \tag{26}$$

where we use the fact that $m < N/Q \leq m+1$ and $q \leq Q$. Since η_j is an integer, $\eta_j > m$ implies that $\eta_j \geq m+1$. The streams in W_j belong to no more than L/Q phases. For at least one of these phases, say phase i, we have $r_i \geq \eta_j/(L/Q)$. But $n_I^{(j)} \geq r_k$ for all values of k that represent the phases of streams in W_j. Consequently,

$$n_I^{(j)} \geq \frac{\eta_j}{L/Q} \geq \frac{1+m}{L/Q} \geq \frac{N/Q}{L/Q} = \frac{N}{L} > w \tag{27}$$

The last inequality follows from the definition of w which implies that $w < N/L \leq w+1$. Accordingly, $n_I^{(j)} \geq w+1$. Based on the above, we have

$$C(\boldsymbol{u}, N) \geq C_j(u, N) = \frac{n_I^{(j)} I_{max} + (\eta_j - n_I^{(j)}) P_{max} + (N-\eta_j) B_{max}}{N} \tag{28}$$

Since $n_I^{(j)} \geq w+1$ and $\eta_j \geq m+1$, $C(\boldsymbol{u}, N)$ satisfies (20), and $\boldsymbol{u}^*$ is optimal.

5.3 Sub-optimal Scheduling in the Heterogeneous Case

The scheduling scheme presented in the previous section is proved to be optimal for streams with identical traffic envelops. Using the same envelop to characterize different video movies can be conservative if, for example, these movies differ in their resolution or quantization levels. In such situations, the use of different traffic envelops is quite advantageous. When video sources at the multiplexer are characterized by different envelops, $\boldsymbol{u}^*$ in (14) is no longer optimal. In fact, it can be easily shown that the optimal schedule in this case depends on the exact values of the traffic envelops parameters. Therefore, it is not possible to obtain a general expression for a *best* arrangement of heterogeneous sources. And even if an optimal scheduling for N heterogeneous streams was found and used to transport these streams, the addition of a new stream would require disrupting the original structure of the N streams if the $N+1$ streams are to be optimally scheduled. Such disruption can happen at any time during video delivery, and as frequently as the rate at which connections are added and terminated, resulting in noticeable discontinuities in the motion picture at the client side. Additionally, a *best* arrangement of heterogeneous streams

cannot be computed recursively; therefore, an exhaustive search is needed every time a connection is admitted or terminated, which is computationally impractical.

A more practical approach is to provide an efficient scheme for scheduling heterogeneous sources, which is practical to implement but not necessarily optimal. As before, the parameters of the traffic envelop that model stream s_i are given by the 5-tuple vector $\boldsymbol{E}_i$. The server maintains a table M which similar to the one used to compute the effective bandwidth for arbitrary arrangements with aligned boundaries (see Section 4.1). Each row in this table gives the $\widetilde{L}$ values associated with $\overline{b}_i(\tau - u_i)$. In contrast to the situation in Section 4.1, u_i for each source s_i can be determined by the server just before s_i is started. Once s_i starts, it keeps on sending frames without any delay. The server also maintains a vector $\boldsymbol{V}$ similar to the one in Section 4.1. Given that N streams have already been scheduled and are being multiplexed, once a new stream is being requested, the server schedules the $(N+1)$th stream to start in a phase i, $i \in \{0, \ldots, \widetilde{L} - 1\}$ for which the total bit rate for the ongoing N streams is minimum. Using the same notation of Section 4.1,

$$u_{N+1} = i \text{ where } v_i = \min_{0 \le j \le \widetilde{L}-1} v_j \tag{29}$$

Once u_{N+1} is obtained, the server updates V and computes $C(\boldsymbol{u}, N+1)$ using the recursions in Section 4.1. A similar updating procedure is also used when a connection is terminated.

6 Numerical Results

We tested the efficiency of our scheduling schemes on real MPEG traces that were captured by several researchers [3, 7, 8, 16] for various types of video, including action movies, advertisements, and a lecture. These traces are listed in Table 2 along with their traffic envelops parameters (frames are packetized into 48-byte cells). They were all generated using MPEG-I encoders (see the references for the compression details). The last column in the table gives the maximum attainable multiplexing gain (given as a percentage of the source peak rate).

Trace	Length (in frames)	I_{max}	P_{max}	B_{max}	L	Q	$(C^*_{min}/I_{max}) \times 100\%$
Star Wars [3]	174136	483	454	169	12	3	55%
Wizard of Oz [8]	41760	894	742	157	15	3	41%
Advertisements [7]	16316	215	214	162	6	3	84%
Lecture [7]	16316	131	92	32	6	3	45%
Silence of the Lambs [16]	40000	350	231	144	12	3	53%

Table 2: Empirical MPEG traces for different video movies with various compression patterns (frame sizes in cells). The last column shows C^*_{min} as a percentage of source peak rate.

Figure 7 depicts the normalized minimum effective bandwidth, $(C_{min}(N)/I_{max}) \times 100\%$, versus N for homogeneous streams. As N increases, $C_{min}(N)$ decreases non-monotonically to C^*_{min}. For moderate and large N, $C_{min}(N)$ is very insensitive to the variation in N. Clearly, the reduction in bandwidth strongly depends on the values of the traffic envelop parameters. For example, when several *Wizard of Oz* streams are multiplexed, the bandwidth per source is about 41% of the source peak rate, whereas it is about 85% for *Lecture* streams.

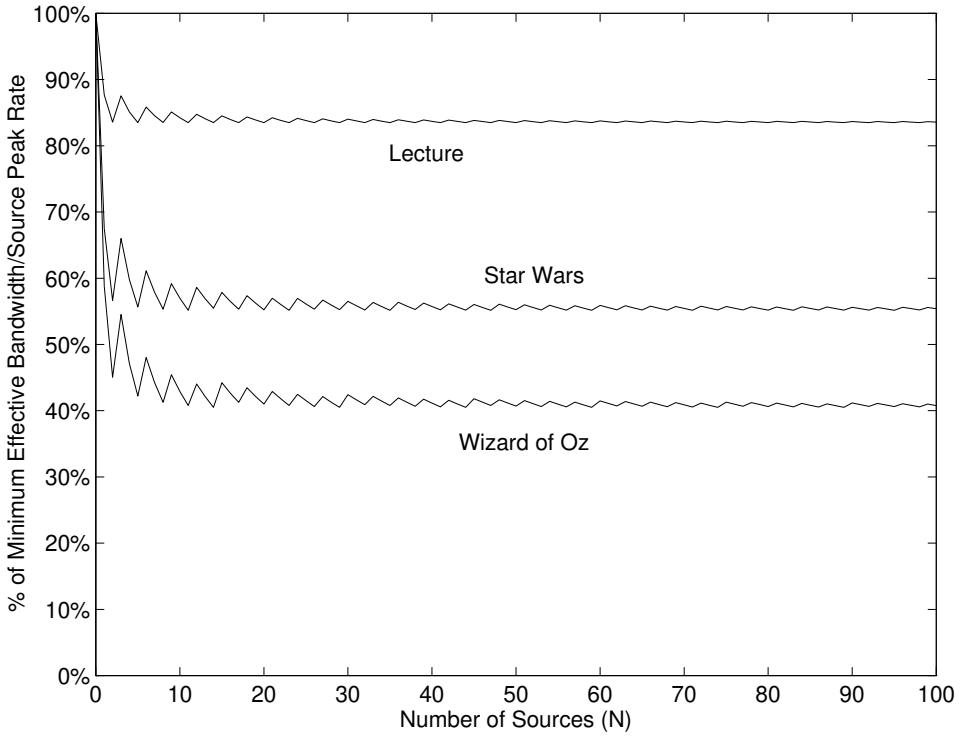

Figure 7: Percentage of $C_{min}(N)/I_{max}$ versus N for different MPEG traces (homogeneous case).

Another way to measure the multiplexing gain under optimal scheduling is by computing the number of homogeneous video connections that can be simultaneously transported using a fixed total bandwidth. This is shown in Figure 8 for two movies. The abscissa in this figure depicts the total bandwidth normalized to I_{max}.

To study the impact of L and Q on the minimum effective bandwidth, we examined a segment of 12600 frames from *Wizard of Oz* movie (from frame # 29191 to frame # 41790 in the movie). This segment was compressed several times using different L and Q values. Table 3 depicts the GOP patterns that were used and the resulting I_{max}, P_{max}, and B_{max}. Unexpectedly, the GOP pattern seems to have little impact on the maximum frames sizes (note, however, that the GOP pattern has significant impact on the average frame size of each frame type). This can be justified by the fact that a movie consists of several scenes, where a scene can be loosely defined as a segment

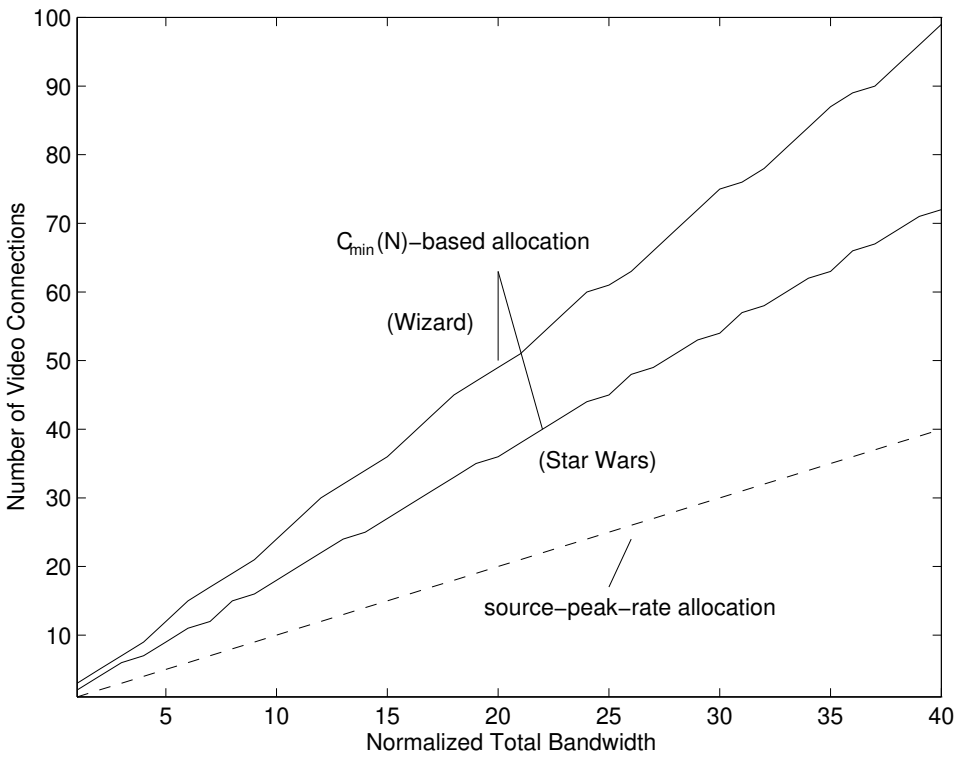

Figure 8: Number of homogeneous video connections that can be simultaneously transported based on minimum effective bandwidth versus total capacity (normalized with respect to I_{max}).

Compression Pattern	L	Q	I_{max}	P_{max}	B_{max}	$(C^*_{min}/I_{max}) \times 100\%$
I	1	1	908	—	—	100%
IP	2	1	898	756	—	92.1%
IPP	3	1	898	756	—	89.5%
$IPPP$	4	1	896	756	—	88.3%
$IPPPP$	5	1	896	740	—	86.1%
$IBPB$	4	2	896	733	161	54.4%
$IBPBPB$	6	2	898	742	161	53.2%
$IBPBPBPB$	8	2	889	742	161	52.9%
$IBPBPBPBPB$	10	2	894	742	161	52.2%
$IBBPBB$	6	3	898	719	157	41.7%
$IBBPBBPBB$	9	3	896	742	157	41.2%
$IBBPBBPBBPBB$	12	3	896	742	157	40.7%
$IBBPBBPBBPBBPBB$	15	3	893	742	157	40.5%

Table 3: Encoding of a video segment using different compression patterns.

of the movie that exhibits uniformity in the video dynamics. Sizes of I frames (similarly, P and B frames) within one scene are close in value. Since on the average a scene lasts for several seconds [9], changing the compression pattern (whose time scale is smaller than one second) has little effect on the maximum sizes of I, P, and B frames within a scene. The last column in Table 3 gives the limiting value of $C_{min}(N)$ computed from (16). It is obvious that L has a very negligible effect on C^*_{min}, whereas increasing Q results in a significant reduction in C^*_{min}. This is expected since for the examined traces, P_{max} is closer to I_{max} than to B_{max}. When $P_{max} \approx I_{max}$, C^*_{min} in (16) reduces to $(1/Q)P_{max} + (1 - 1/Q)B_{max}$ which does not depend on L. However, using a large Q (i.e., more B frames in a GOP) is not desirable from the decoder's perspective. Hence, Q should be chosen such that it provides a good compromise between the decoder complexity (and the associated decoding delay) and the multiplexing gain.

Using the sub-optimal scheduling scheme for heterogeneous sources, the normalized effective bandwidth is plotted in Figure 9 as a function of the multiplexed streams. Here, we consider a simple scenario in which the heterogeneous mix consists of two different envelops (e.g., two movies). Starting with $N = 1$, we increment N by adding streams one at a time to the multiplexer, and recursively computing the effective bandwidth according to the sub-optimal scheme. When adding streams, we alternate between the two movies (for example, we start with an *Advertisement* stream, then add a *Lecture* stream, then add another *Advertisement* stream, and so on). The effective bandwidth is normalized with respect to the average source peak rate $\sum_{i=1}^{N} I^{(i)}_{max}/N$. Similar to the homogeneous case, it is observed that effective-bandwidth allocation, though not optimal, results in significant bandwidth gain.

7 Summary

Bandwidth allocation for VBR video with stringent deterministic QoS requirements is typically done based on the source peak rate. Such an allocation strategy underutilizes the network capacity. By exploiting the periodic manner in which frame types are generated in MPEG compressors, we showed that significant bandwidth gain can be achieved by means of statistical multiplexing of video, while supporting stringent deterministic QoS guarantees. When streams are multiplexed, bandwidth allocation can be done based on the effective bandwidth per source which is often less than the source peak rate. We provided efficient, recursive procedures for computing the effective bandwidth, as well as an upper bound on it. These procedures can be easily implemented at a multiplexing node to provide dynamic resource allocation for video. The amount of gain obtained from effective-bandwidth allocation depends largely on the *arrangement* of the multiplexed streams,

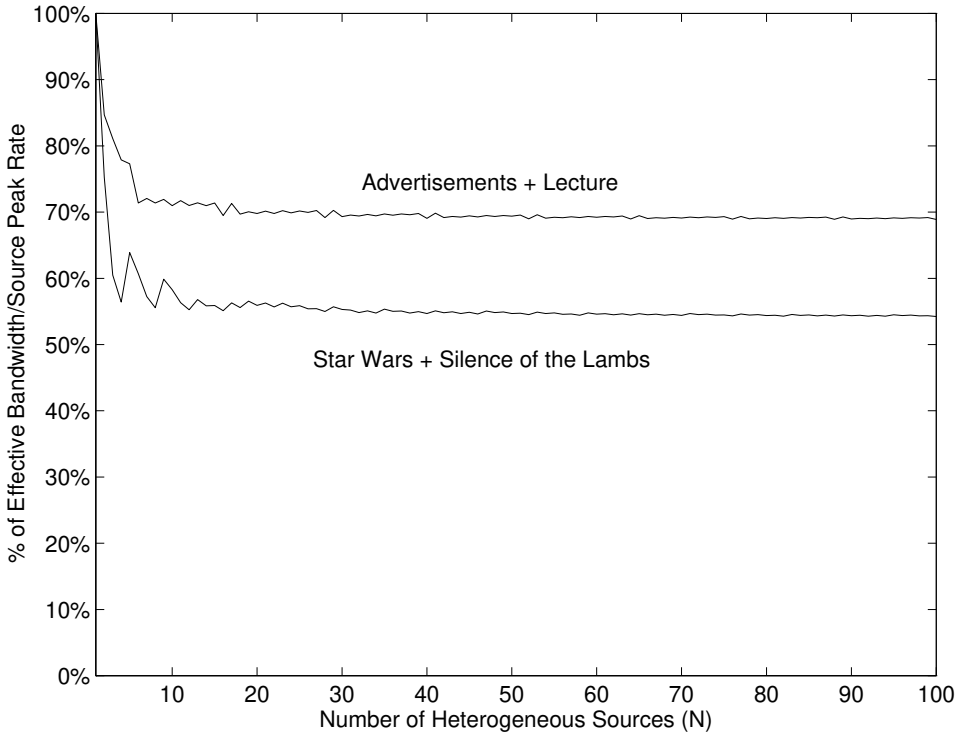

Figure 9: Percentage of allocated bandwidth normalized by average source peak rate as a function of N for heterogeneous streams.

which, in the case of VOD systems, is determined by the way these streams are scheduled for transmission over the network. Given that a video server has some flexibility in determining the starting times of new connections (and thus, their transmission schedule), we provided an optimal scheduling scheme for homogeneous video sources, which achieves the minimum effective bandwidth. We proved mathematically the optimality of this scheme, and gave the form of the associated best arrangement. In several cases, optimal scheduling of video resulted in more than 50% decrease in the allocated bandwidth. We also gave an efficient scheduling scheme for heterogeneous sources, which achieves sub-optimal performance. Examples of real MPEG traces from various compressed movies were used to demonstrate the advantages of our scheduling schemes. Traffic envelops with constant parameters were used in this paper to characterize video traffic. A more accurate characterization can be provided using traffic envelops with time-varying parameters. In a future work, we will extend the results of this paper to video streams that are characterized by traffic envelops with time-varying parameters.

Acknowledgement

The authors would like to thank Ron Sass for providing the traces of the *Wizard of Oz* movie.

Appendix

A Proof of Proposition 1

We show that $\overline{C}(\boldsymbol{u}, N)$ given in (9) is an upper bound on $C(\boldsymbol{u}, N)$. Substitute (7) in (8) to obtain

$$\widetilde{v}_j = \sum_{i=1}^{N} \max\left\{\overline{b}_{i,j-1},\ \overline{b}_{i,j-2}\right\} \tag{30}$$

$$= \sum_{i=1}^{N} \max_{j-2\le\tau\le j}\left\{\overline{b}_i\left(\tau - \lfloor u_i \rfloor\right)\right\} = \sum_{i=1}^{N} \max_{j-2-\lfloor u_i\rfloor\le\tau\le j-\lfloor u_i\rfloor}\left\{\overline{b}_i\left(\tau\right)\right\} \tag{31}$$

But $j-2-\lfloor u_i\rfloor \le j-1-u_i < j-u_i \le j-2-\lfloor u_i\rfloor$. Thus,

$$\widetilde{v}_j \ge \sum_{i=1}^{N} \max_{j-1-u_i\le\tau\le j-u_i}\left\{\overline{b}_i\left(\tau\right)\right\} = \sum_{i=1}^{N} \max_{j-1\le\tau\le j}\left\{\overline{b}_i\left(\tau - u_i\right)\right\} \tag{32}$$

From (9) and (32), we have

$$\begin{aligned}
\overline{C}(\boldsymbol{u}, N) &\ge \frac{1}{N}\max_{1\le j\le\widetilde{L}}\left\{\sum_{i=1}^{N}\max_{j-1\le\tau\le j}\left\{\overline{b}_i\left(\tau-u_i\right)\right\}\right\} \\
&\ge \frac{1}{N}\max_{1\le j\le\widetilde{L}}\left\{\max_{j-1\le\tau\le j}\left\{\sum_{i=1}^{N}\overline{b}_i\left(\tau-u_i\right)\right\}\right\} \\
&= \frac{1}{N}\max_{0\le\tau\le\widetilde{L}}\left\{\sum_{i=1}^{N}\overline{b}_i\left(\tau-u_i\right)\right\} = C(\boldsymbol{u}, N)
\end{aligned} \tag{33}$$

(in the above equations, the maximization over j is taken on integer values while the maximization over τ is taken on real values). □

References

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

[2] A. Forum. Traffic management specification, version 4.0, Aug. 1995.

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

[4] M. Grossglauser, S. Keshav, and D. Tse. RCBR: A simple and efficient service for multiple time-scale traffic. In *Proceedings of the ACM SIGCOMM '95 Conference*, pages 219–230, Aug. 1995.

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

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

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

[8] M. Krunz and H. Hughes. A traffic model for MPEG-coded VBR streams. In *Proc. of the ACM SIGMETRICS/PERFORMANCE '95 Conference*, pages 47–55, May 1995.

[9] M. Krunz, S. Tripathi, and H. Hughes. A source model for MPEG-coded video movies. In *First IEEE Workshop on ATM Networks*, Washington, DC, Oct. 1995.

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

[11] J. M. McManus and K. W. Ross. Video on demand over ATM: Constant-rate transmission and transport. In *Proc. of IEEE INFOCOM '96*, pages 1357–1362, 1996.

[12] D.-Y. Oh, S. SampathKumar, and P. V. Rangan. Content-based inter-media synchronization. In *Proceedings of SPIE Conference*, pages 202–214, Feb. 1995.

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

[14] C. Papadimitriou, S. Ramanathan, P. V. Rangan, and S. SampathKumar. Multimedia information caching for personalized video-on-demand. *Computer Communications*, 18(3):204–216, Mar. 1995.

[15] P. V. Rangan and H. M. Vin. Efficient storage techniques for digital continuous multimedia. *IEEE Transactions on Knowledge and Data Engineering*, 5(4):564–573, Aug. 1993.

[16] O. Rose. Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems. In *Proceedings of the 20th Annual Conference on Local Computer Networks*, Minneapolis, MN, 1995.

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