

Optimally Scheduling Video-on-Demand to Minimize Delay When Server and Receiver Bandwidth May Differ*

William Evans[†]

David Kirkpatrick[†]

Abstract

We establish tight bounds on the intrinsic cost (either minimizing delay d for fixed server and receiver bandwidths, or minimizing server bandwidth for fixed delay and receiver bandwidth) of broadcasting a movie of length m over a channel of bandwidth S in such a way that a receiver (with bandwidth R), starting at an arbitrary time t , can download the movie so that it can begin playback after a delay of at most d time units.

Our bounds are realized by a simple abstract protocol that partitions the movie into a fixed number of segments, partitions the server bandwidth into an equivalent number of equal bandwidth subchannels, and broadcasts each segment repeatedly on its own subchannel. This protocol can be implemented as a concrete discrete protocol in which movie information is packaged into discrete fixed length packets using only a modest overhead (measured in terms of increased delay or server bandwidth).

Our primary contribution is a lower bound on the required delay that applies in a very general model of communication. This lower bound matches the behaviour of our abstract protocol in the limit as the number of segments approaches infinity. We are also able to relate its behaviour to arbitrary protocols that have a fixed number of segments.

1 Introduction

Suppose we broadcast a movie using bandwidth S (measured in units of movie bandwidth). A person may tune their television, which can receive any R/S fraction of the bandwidth, to this channel at any time in order to watch the movie. We would like to minimize the delay between when they tune to the channel and when they can start watching the movie. Let m be the length of the movie (in minutes). We can certainly guarantee a delay of at most m minutes simply by repeatedly broadcasting the movie back-to-back (as long as S and R are at least

1); no matter when a person tunes in, they know the movie will start within m minutes.

If the television has memory and can record from the channel into the memory while simultaneously displaying from the memory, we can do better. Of course, if we knew what movie a person would request, we could store the entire movie in the television memory and there would be no delay in showing the movie. We assume that this is not possible and, in fact, that the television memory is empty when the person selects a movie (i.e. tunes to a channel).

How small of a delay could we hope to achieve?

In this paper, we obtain asymptotically tight upper and lower bounds on the delay achievable in this model. In particular, both bounds approach

$$\frac{1}{-1 + \sum_{j=0}^{\lfloor S/R \rfloor} \frac{(jR-S)^j}{j!} e^{S-jR}}$$

of the total movie, as we increase the number of segments into which we divide the movie. In Section 3, we define our model of broadcast protocols. In Section 4, we outline our results. In Section 5, we describe a broadcast protocol that achieves the optimal delay in the limit as the number of segments into which it partitions the movie increases. In Sections 6 and 7, we show that this protocol is optimal by demonstrating a lower bound on the delay of any protocol within our model. In Section 8, we discuss the impact on our protocol of insisting on a fixed, smallest unit of transmission. We start by outlining some related work.

2 Related work

Many video-on-demand protocols have been designed for repeatedly broadcasting a movie so that a receiver may receive and watch the movie by listening to the broadcast. *Staggered broadcast* is perhaps the simplest example of such a protocol. It repeatedly transmits a copy of the movie on each of c channels (each with bandwidth equal to the movie bandwidth). By starting the movie at equally spaced time intervals across the channels, a receiver need wait at most $1/c$ of the movie to start watching it.

More complicated protocols require the receiver to

*This research was supported in part by the Natural Sciences and Engineering Research Council of Canada.

[†]Department of Computer Science, University of British Columbia, Vancouver B.C. Canada, V6T 1Z4. Email: {will, kirk}@cs.ubc.ca

store what it reads from the broadcast into memory and then to play what it stores immediately or at a later time, perhaps while simultaneously reading additional broadcast data into memory. The family of *pyramid protocols* [8] partition a movie into some number of unequal size, contiguous segments and then transmit these segments on channels or sub-channels of equal bandwidth. On the other hand, the family of *harmonic protocols* [6] partition a movie into equal size segments and transmit them on unequal bandwidth channels: the i th segment receiving bandwidth $1/i$.

Early protocols in both families were designed like the staggered protocol so that a receiver could begin watching the movie immediately on receiving the first segment. In other words, a receiver could wait until the start of the first segment to begin watching and buffering the broadcast. Pâris, Carter, and Long [7] recognized that the receiver's wait could be decreased if the receiver buffered data from the moment they started receiving the broadcast. Their *polyharmonic protocol* (part of the harmonic family) forced every receiver to wait the same amount of time, regardless of arrival time. If the number of segments increases, the receiver's wait approaches a $1/(e^S - 1)$ fraction of the movie, where S is the bandwidth of both the receiver and server. Hu, Nikolaidis, and van Beek's *greedy equal bandwidth protocol* [4] is a pyramid family protocol that also requires a forced delay. It achieves the same asymptotic performance as the polyharmonic protocol.

All of these protocols assume that the receiver and server have the same bandwidth. However, in 1998 Gao, Kurose, and Towsley [3] proposed a formal framework for studying broadcast protocols that permit limited receiver bandwidth. They described the *greedy disk-conserving protocol* for limited receiver bandwidth and illustrated its performance. They also proved that any broadcasting scheme with server bandwidth S will cause a client to wait at least $1/(e^S - 1)$ fraction of the movie, for some arrival time. Hu [5] also claims this lower bound on client delay, but only for what he calls *optimally-structured broadcasting protocols* (which are similar to our early delivery stripped schedules). Engebretsen and Sudan [2] obtain the same $1/(e^S - 1)$ lower bound by relating broadcast protocols to priority encoded transmission schemes of a particular type, for which they prove a lower bound. Bar-Noy, Ladner, and Tamir [1] also prove this lower bound, generalized to M movies as $1/(e^{S/M} - 1)$. None of the lower bounds depend on receiver bandwidth.

3 Model

We start by formalizing our notion of a broadcast protocol.

A *transmission function* $\Theta(t, x)$ is the instantaneous bandwidth allocated to movie instant $x \in [0, m]$ at absolute time $t \in [0, \infty)$, where m is the length of the movie. A *reception function* $\Psi(t, \delta, x)$ is the instantaneous bandwidth received of movie instant x at absolute time $t + \delta$ by a receiver that arrives at time t . A transmission/reception function pair (Θ, Ψ) defines a *continuous t/r protocol*. Such a protocol is *consistent* if

$$\Psi(t, \delta, x) \leq \Theta(t + \delta, x)$$

for all $t \in [0, \infty)$, $\delta \in [0, \infty)$, and $x \in [0, m]$;

has sender bandwidth at most S if

$$\int_0^m \Theta(t, x) dx \leq S$$

for all $t \in [0, \infty)$;

has receiver bandwidth at most R if

$$\int_0^m \Psi(t, \delta, x) dx \leq R$$

for all $t \in [0, \infty)$ and $\delta \in [0, \infty)$;

and achieves a length m video transfer with delay at most d if

$$\int_0^{a'} \Psi(t, d + a', x) dx - \int_0^a \Psi(t, d + a, x) dx \geq a' - a$$

for all $t \in [0, \infty)$, $a \in [0, m]$, and $a' \in [0, m]$.

It should be clear that every discrete movie t/r protocol can be expressed as a transmission/reception function pair. An advantage of considering protocols in this generality is that it allows us to prove the equivalence of any protocol with a natural regularized form of it. (Note this regular form is implicitly assumed in Hu's lower bound proof [5] even though it is not proved.)

A transmission function $\Theta(t, x)$ is *regular* if $\Theta(t, x) = \Theta(t', x)$ for all $t, t' \in [0, \infty)$. Regularity states that the amount of bandwidth used to transmit a movie instant x is constant over time. Similarly, a reception function $\Psi(t, \delta, x)$ is *regular* if $\Psi(t, \delta, x) = \Psi(t', \delta, x)$ for all $t, t' \in [0, \infty)$.

THEOREM 3.1. *If (Θ, Ψ) achieves a length m video transfer with delay d using server bandwidth S and receiver bandwidth R , then there exists a regular transmission/reception function pair (Θ', Ψ') with the same property.*

The proof is given in Appendix A and consists of regularizing (averaging) (Θ, Ψ) to obtain (Θ', Ψ') .

3.1 Stripped protocols *Stripped* protocols subdivide unit bandwidth channels into a fixed number of equal bandwidth subchannels (strips). The movie is partitioned into as many segments as there are strips, and each strip is dedicated to the transmission of information from one and only one segment (in an unconstrained fashion). Thus stripped protocols have a kind of discrete regularity – the bandwidth devoted to a single segment is constant over all time instants.

A continuous t/r protocol (Θ, Ψ) with sender bandwidth S and movie length m is *k-stripped* if there exist x_i for $0 \leq i \leq kS$, where $0 = x_0 \leq x_1 \leq \dots \leq x_{kS} = m$, (called *movie partition points*) such that

$$\int_{x_{i-1}}^{x_i} \Theta(t, x) dx \leq 1/k$$

for all $t \in [0, \infty)$ and $1 \leq i \leq kS$.

Stripped protocols are interesting because (i) many real discrete protocols are stripped protocols, and (ii) in the limit as the number of strips approaches infinity, stripped protocols become arbitrary regular continuous t/r protocols.

3.2 Early and late delivery We consider two models for stripped protocols. In each case we denote by k the fragmentation factor and assume that the total broadcast bandwidth can be subdivided into subchannels of bandwidth $1/k$, each of which is dedicated to the broadcast of one movie segment. The models differ in their requirement for segment delivery. In the *early delivery stripped model* we insist that the information sufficient to play the entire segment is in hand before playback of the segment begins. In the *late delivery stripped model* we insist only that information sufficient to play the entire segment is in hand before playback of the segment completes. (In fact, our models are even weaker and hence more inclusive, requiring only that the quantity of bits received be sufficient for playback.)

(Θ, Ψ) achieves *early delivery* with delay d if

$$\int_{x_{i-1}}^{x_i} \Psi(t, d + x_{i-1}, x) dx \geq x_i - x_{i-1}$$

for all $1 \leq i \leq kS$.

Note that a protocol may meet the specifications of early delivery without actually achieving full video transfer – early delivery ensures only that we have received enough bits of each segment (with duplicates counted multiply), not necessarily the correct bits.

(Θ, Ψ) achieves *late delivery* with delay d if

$$\int_{x_{i-1}}^{x_i} \Psi(t, d + x_i, x) dx \geq x_i - x_{i-1}$$

for all $1 \leq i \leq kS$.

Again, late delivery ensures only that the right quantity of bits are in hand by the time the segment finishes playing.

4 Results

Our results are the following:

1. There is a simple stripped protocol (Section 5) that fits the early delivery model and achieves optimal movie length among all such protocols with the same fragmentation factor (Section 6).
2. This same protocol approaches the optimal movie length among all protocols in the late delivery model as k goes to infinity (Section 7). The rate of convergence is $O(1/k)$.
3. It follows from 2. that our simple stripped protocol approximates the optimal movie length for any regular continuous t/r protocol (and hence, by Theorem 3.1, any even non-regular continuous t/r protocol.)
4. Our simple stripped protocol can be realized by a fully discrete protocol (using packets of a fixed size), but this causes a modest increase in the delay (Section 8).

5 An optimal early delivery stripped schedule

We divide the server’s bandwidth into *strips* of bandwidth $1/k$ for some integer k . Each strip is devoted to transmitting, repeatedly, a single, distinct, contiguous piece of the movie called a *segment*. The server transmits a segment on each strip in such a way that the receiver will obtain the entire segment if it reads from the strip starting at any point for time $k\delta$ where δ is the duration of the segment (measured in bandwidth units). A simple way to achieve this goal is for the server to transmit the segment, repeatedly, on the strip.

At time 0, the receiver starts reading the kR strips that carry the first kR segments of the movie. Let d be the time when the receiver has the entire first segment of the movie. At this point, the receiver stops reading the first strip (carrying the first segment) and uses that bandwidth to read from strip $kR + 1$ (carrying segment $kR + 1$). The receiver follows this strategy for the duration of the movie: after receiving segment i , the receiver stops reading strip i and uses the now-available bandwidth to read from strip $kR + i$. The

total number of strips available to carry the movie is kS . Thus receiving stops at the end of segment kS .

In order to be able to show the movie without interruption starting at time d , the first segment of the movie must be in hand at time $t_0 = d$. Since the protocol is stripped, playback of the first segment completes at time $t_1 = d + d/k$, at which point we must have the second segment in hand. In general, for $i \geq 0$, the completion time t_{i+1} of segment $i + 1$ is the completion time of segment i plus the duration of segment $i + 1$. Thus,

$$t_{i+1} = t_i + \delta_{i+1}$$

where $t_0 = d$.

The duration of the first segment, δ_1 , is $\frac{t_0}{k} = \frac{d}{k}$. The duration of segment $i + 1$ is the amount of time the receiver spends reading from strip $i + 1$ divided by k . For the first kR strips, the receiver spends time t_i reading strip $i + 1$. However, the receiver only reads strip $kR + i + 1$ (for $i \geq 0$) after it has stopped reading strip $i + 1$. So the time spent reading strip $kR + i + 1$ is $t_{kR+i} - t_i$. Defining $t_i = 0$ for $i < 0$, we obtain the following relation for $i \geq 0$:

$$\delta_{i+1} = \frac{t_i - t_{i-kR}}{k}.$$

Combining this with the previous recurrence yields:

$$(5.1) \quad t_i = \begin{cases} 0 & \text{if } i < 0 \\ d & \text{if } i = 0 \\ t_{i-1} + \frac{t_{i-1} - t_{i-kR-1}}{k} & \text{if } i > 0. \end{cases}$$

Note that the sequence $t_i - d$, for $i = 0 \dots kS$, are the movie partition points of the schedule. Figure 1 shows at what times the receiver reads from each strip in a movie transmission schedule when $S = 6$, $R = 2$, and $k = 7$.

Viewing this figure, the reader may mistakenly conclude that segment duration is monotonically increasing. Figure 2, which eases the comparison of segment lengths, should confound this belief. This is an odd property for a supposedly optimal schedule: some later parts of the movie are transmitted more frequently than earlier parts. It follows from our results in Section 6 that an optimal schedule, for certain R and S , must have this surprising property.

Before proving optimality, we calculate the delay that is guaranteed by the above scheme. As a first step, we restate the recurrence equation (5.1) in a slightly more general form and then present some of its basic properties and a general solution.

$$(5.2) \quad t_i = \begin{cases} 0 & \text{if } i < 0 \\ d & \text{if } i = 0 \\ t_{i-1} + \frac{t_{i-1} - t_{i-a-1}}{k} & \text{if } i > 0. \end{cases}$$

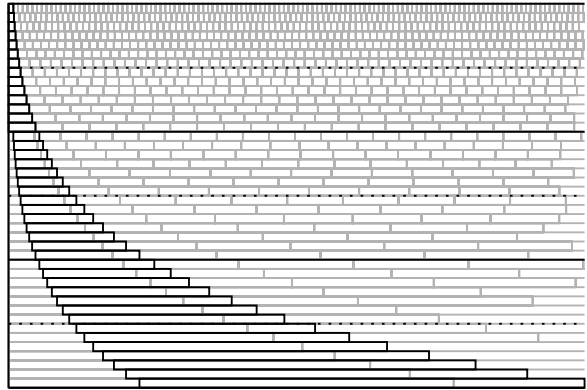


Figure 1: The x -axis measures time while the y -axis represents server bandwidth that has been divided into many equal bandwidth strips. The black rectangles (one per strip) illustrate the times at which a receiver with bandwidth $R = 2$ will read each of the strips. The gray rectangles represent the transmitted movie segments. Server bandwidth is $S = 6$ and $k = 7$.

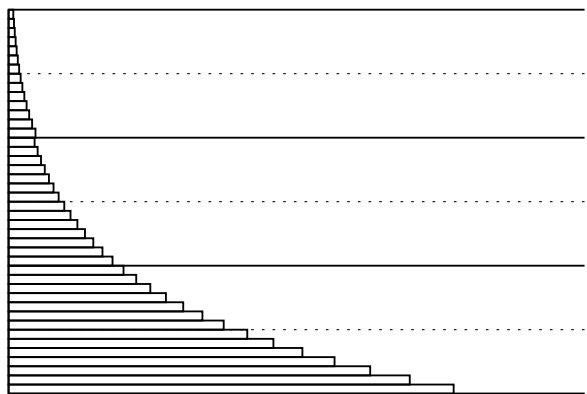


Figure 2: The same segments as in Figure 1 but left-justified to show their non-monotonicity.

LEMMA 5.1. *If the sequence $\{t_i\}$ satisfies the recurrence 5.2 then*

- (i) $t_i = d \left(1 + \frac{1}{k}\right)^i$, for $0 \leq i \leq a$
- (ii) $t_i = d + \frac{1}{k} \sum_{j=i-a}^{i-1} t_j$, for $i \geq 0$, and
- (iii) $t_i = d \left(1 + \frac{1}{k}\right)^i \sum_{j=0}^{\lfloor \frac{i}{a+1} \rfloor} (-\lambda)^j \binom{i-j}{j} a^j$, for $i \geq 0$, where $\lambda = \frac{1}{k(1+\frac{1}{k})^{a+1}}$.

Proof. The equation (i) follows immediately from the fact that $t_0 = d$ and $t_i = t_{i-1} \left(1 + \frac{1}{k}\right)$, for $1 \leq i \leq a$. Equation (ii) follows from 5.2 by induction on i .

Equation (i) provides the basis for an inductive proof of the equation (iii). For the inductive step, assume that it is true for all integer values i from 0 to $n \geq a$.

$$\begin{aligned} t_{n+1} &= t_n + \frac{t_n - t_{n-a}}{k} = \left(1 + \frac{1}{k}\right) t_n - \frac{t_{n-a}}{k} \\ &= \left(1 + \frac{1}{k}\right) d \left(1 + \frac{1}{k}\right)^n \sum_{j=0}^{\lfloor \frac{n}{a+1} \rfloor} (-\lambda)^j \binom{n-j}{j} a^j \\ &\quad - \frac{d \left(1 + \frac{1}{k}\right)^{n-a} \sum_{j=0}^{\lfloor \frac{n-a}{a+1} \rfloor} (-\lambda)^j \binom{n-(j+1)a}{j} a^j}{k} \\ &\quad \text{(by ind. hypothesis)} \\ &= d \left(1 + \frac{1}{k}\right)^{n+1} \left[\sum_{j=0}^{\lfloor \frac{n}{a+1} \rfloor} (-\lambda)^j \binom{n-j}{j} a^j \right. \\ &\quad \left. - \lambda \sum_{j=1}^{\lfloor \frac{n-a}{a+1} \rfloor} (-\lambda)^j \binom{n-(j+1)a}{j} a^j \right] \end{aligned}$$

When $n+1$ is not an integer multiple of $a+1$, say $n+1 = m(a+1) + r$, where $1 \leq r < a+1$, then $m = \lfloor \frac{n+1}{a+1} \rfloor = \lfloor \frac{n}{a+1} \rfloor + 1$. In this case, collecting powers of λ ,

$$\begin{aligned} t_{n+1} &= d \left(1 + \frac{1}{k}\right)^{n+1} \\ &\quad \cdot \left[1 + \sum_{j=1}^m (-\lambda)^j \left(\binom{n-j}{j} a^j + \binom{n-j}{j-1} a^{j-1} \right) \right] \\ &= d \left(1 + \frac{1}{k}\right)^{n+1} \sum_{j=0}^{\lfloor \frac{n+1}{a+1} \rfloor} (-\lambda)^j \binom{n+1-j}{j} a^j. \end{aligned}$$

Here we use the fact that $\binom{n-a}{0} = 1$ and $\binom{n-j}{j} a^j + \binom{n-j}{j-1} a^{j-1} = \binom{n+1-j}{j} a^j$.

When $n+1 = m(a+1)$ for some integer m , then $m = \lfloor \frac{n+1}{a+1} \rfloor = \lfloor \frac{n}{a+1} \rfloor + 1 = \lfloor \frac{n-a}{a+1} \rfloor + 1$. In this case, again

collecting powers of λ ,

$$\begin{aligned} t_{n+1} &= d \left(1 + \frac{1}{k}\right)^{n+1} \left[1 + (-\lambda)^m \binom{n-ma}{m-1} \right. \\ &\quad \left. + \sum_{j=1}^{m-1} (-\lambda)^j \left(\binom{n-ja}{j} + \binom{n-ja}{j-1} \right) \right] \\ &= d \left(1 + \frac{1}{k}\right)^{n+1} \sum_{j=0}^{\lfloor \frac{n+1}{a+1} \rfloor} (-\lambda)^j \binom{n+1-ja}{j} a^j. \end{aligned}$$

Here we use the additional fact that when $n+1 = m(a+1)$, $\binom{n-ma}{m-1} = \binom{n+1-ma}{m}$ (they both equal 1).

THEOREM 5.1. *The schedule described in this section achieves a delay of*

$$\frac{1}{-1 + \sum_{j=0}^{\lfloor S/R \rfloor} \frac{(jR-S)^j}{j!} e^{S-jR}}$$

of the total movie, in the limit as k goes to infinity.

Proof. The final segment finishes playing at time t_{kS} , where t_i is defined by recurrence equation (5.1). Thus the total movie time is $t_{kS} - d$. If we can split a movie into an arbitrary number of segments, then we are interested in this value as k goes to infinity. Applying lemma 5.1(iii) with $a = kR$ and $i = kS$, we get

$$(5.3) \quad \lim_{k \rightarrow \infty} t_{kS} = d \sum_{j=0}^{\lfloor S/R \rfloor} \frac{(jR-S)^j}{j!} e^{S-jR}.$$

6 Bounds on movie length under early delivery

Suppose that (Θ, Ψ) is a k -stripped protocol with server bandwidth S and receiver bandwidth R that achieves early delivery with delay d . Let $0 = r_0, r_1, \dots, r_{kS} = m$ be the movie partition points.

Since (Θ, Ψ) is k -stripped, it follows that at time $d + r_i$ the receiver has received at most $(d + r_i)/k$ units of segment $i+1$ for $i \geq 0$. Since all of the movie interval $[r_i, r_{i+1}]$ must be received by time $d + r_i$, it follows that

$$(6.4) \quad r_{i+1} - r_i \leq (d + r_i)/k$$

for $i \geq 0$.

LEMMA 6.1.

$$r_j + d \leq d(1 + 1/k)^j$$

for $j \geq 0$.

Proof. When $j = 0$ the result follows from the fact that $r_0 = 0$. From (6.4) and induction, we have $r_{j+1} \leq (1 + 1/k)r_j + d/k \leq d(1 + 1/k)^{j+1} - d$.

Since (Θ, Ψ) has receiver bandwidth R , at time $d+r_i$ the receiver has received at most $R(d+r_i)$ movie units. By the definition of early delivery, the entire movie interval $[0, r_{i+1}]$ must be in hand at time $d+r_i$. Thus the information received by time $d+r_i$ that could be used for segments following the $(i+1)$ st segment is at most $R(d+r_i) - r_{i+1}$ movie units. During time interval $[d+r_i, d+r_{j-1}]$ only $(r_{j-1} - r_i)/k$ units of segment j are broadcast (since (Θ, Ψ) is k -stripped). Since the total size of segment j is $r_j - r_{j-1}$ and the segment must be received by time r_{j-1} , the part of segment j that must be received before $d+r_i$ is at least

$$(r_j - r_{j-1}) \dot{\div} \frac{r_{j-1} - r_i}{k}$$

where $a \dot{\div} b \equiv \max\{a - b, 0\}$. Since this holds for all $j > i+1$, we have,

$$\sum_{j>i+1} \left((r_j - r_{j-1}) \dot{\div} \frac{r_{j-1} - r_i}{k} \right) \leq R(d+r_i) - r_{i+1}.$$

For $i \geq 1$, summing over j such that $i+1 < j \leq i+kR+1$ and replacing $\dot{\div}$ by $-$, this implies

$$\sum_{j=i+2}^{i+kR+1} \left((r_j - r_{j-1}) - \frac{r_{j-1} - r_i}{k} \right) \leq R(d+r_i) - r_{i+1}.$$

Simplifying the telescoping sum yields,

$$r_{i+kR+1} - r_{i+1} - \frac{1}{k} \sum_{j=i+1}^{i+kR} r_j + Rr_i \leq R(d+r_i) - r_{i+1}$$

or equivalently

$$(6.5) \quad r_{i+kR+1} \leq Rd + \frac{1}{k} \sum_{j=i+1}^{i+kR} r_j$$

for $i \geq 1$.

THEOREM 6.1.

$$r_j + d \leq t_j$$

for $j \geq 0$.

Proof. Lemmas 5.1(i) and 6.1 establish the base cases ($0 \leq j \leq kR$). Since

$$t_{i+kR} = d + \frac{1}{k} \sum_{j=i}^{i+kR-1} t_j$$

holds (by Lemma 5.1(ii), using the recurrence (5.1)) the result follows by induction on j (using (6.5)).

This establishes that, for fixed delay d , the movie length, $t_{kS} - d$, realized by our optimal schedule is at least as long as that achieved by *any* protocol in the early delivery, k -stripped model.

7 Bounds on movie length under late delivery

In the preceding section, we showed that the delay achieved by our k -stripped protocol is exactly the delay achieved by the best possible early delivery, k -stripped protocol. The early delivery model requires the entire i th segment to have been received by the time the $(i-1)$ th segment finishes playing. More general protocols, that still could play the movie without interruption, may receive part of the i th segment during the process of playing the i th segment. To include these more general protocols, we consider a model that only requires the i th segment to have been received by the time the i th segment finishes playing. This is the late delivery model and in this section we compare it to our k -stripped protocol.

It is convenient to compare our k -stripped protocol to any late model $(k+1)$ -stripped protocol. Let (Θ, Ψ) be a $(k+1)$ -stripped protocol with server bandwidth S and receiver bandwidth R that achieves late delivery with delay d . Let $0 = s_0, s_1, \dots, s_{(k+1)S} = m$ be its movie partition points.

At time $d + s_{i+1}$, the receiver has at most $(d + s_{i+1})/(k+1)$ units of segment $i+1$. The late model insists that all of segment $i+1$ (i.e. movie interval $[s_i, s_{i+1}]$) must be received by time $d + s_{i+1}$. Thus, for $i \geq 0$

$$(7.6) \quad s_{i+1} - s_i \leq (d + s_{i+1})/(k+1),$$

or equivalently,

$$(7.7) \quad s_{i+1} - s_i \leq (d + s_i)/k.$$

It follows immediately from Lemma 6.1 and the correspondence between equations (6.4) and (7.7) that

$$(7.8) \quad s_i + d \leq d(1 + 1/k)^i,$$

for $i \geq 0$.

Since (Θ, Ψ) has receiver bandwidth R , at time $d+s_i$ the receiver has at most $R(d+s_i)$ movie units. By the definition of late delivery, the entire movie interval $[0, s_i]$ must be in hand at time $d+s_i$. Thus, the information received by time $d+s_i$ that could be used for segments following the i th segment is at most $R(d+s_i) - s_i$ movie units. During time interval $[d+s_i, d+s_j]$ only $(s_j - s_i)/(k+1)$ units of segment j are broadcast (since (Θ, Ψ) is $(k+1)$ -stripped). Since the total size of segment j is $s_j - s_{j-1}$ and the segment must be received by time s_j , the part of segment j that must be received before $d+s_i$ is at least

$$(s_j - s_{j-1}) \dot{\div} \frac{s_j - s_i}{k+1}.$$

Since this holds for all $j > i$, we have,

$$\sum_{j>i} \left((s_j - s_{j-1}) \dot{-} \frac{s_j - s_i}{k+1} \right) \leq R(d + s_i) - s_i.$$

For $i \geq 0$, summing over j such that $i < j \leq i + (k+1)R$ and replacing $\dot{-}$ by $-$, this implies

$$\sum_{j=i+1}^{i+(k+1)R} \left((s_j - s_{j-1}) - \frac{s_j - s_i}{k+1} \right) \leq R(d + s_i) - s_i.$$

Simplifying the telescoping sum yields,

$$s_{i+(k+1)R} - s_i - \frac{1}{k+1} \sum_{j=i+1}^{i+(k+1)R} s_j + R s_i \leq R d + R s_i - s_i$$

or (by collecting terms and scaling by $(k+1)/k$)

$$(7.9) \quad s_{i+(k+1)R} \leq R d(k+1)/k + \frac{1}{k} \sum_{j=i+1}^{i+(k+1)R-1} s_j$$

for $i \geq 1$.

Let $\{u_i\}$ be the sequence defined by:

$$(7.10) \quad u_i = \begin{cases} 0 & \text{if } i < 0 \\ d(k+1)/k & \text{if } i = 0 \\ u_{i-1} + \frac{u_{i-1} - u_{i-(k+1)R}}{k} & \text{if } i > 0. \end{cases}$$

Since the recurrence (7.10) is a special case of our generic recurrence (5.2), it follows from Lemma 5.1(iii) that

$$(7.11) \quad u_i = d \left(1 + \frac{1}{k} \right)^{i+1} \cdot \sum_{j=0}^{\lfloor \frac{i}{(k+1)R} \rfloor} (-\lambda)^j \binom{i - j((k+1)R - 1)}{j}$$

for $i \geq 0$, where $\lambda = \frac{1}{k(1+\frac{1}{k})^{(k+1)R}}$.

By direct analogy with Theorem 6.1 we have:

THEOREM 7.1.

$$s_j + d \leq u_{j-1}$$

for $j \geq 1$.

This means that the movie length, $s_{(k+1)S}$, of any protocol in the late, $(k+1)$ -stripped model with fixed delay d is upper bounded by $u_{(k+1)S-1} - d$. It remains to see how $u_{(k+1)S-1} - d$ compares to the movie length, $t_{kS} - d$, of our optimal schedule.

It is straight-forward to verify that both $u_{(k+1)S-1}$ and t_{kS} converge to the same limit (equation (5.3)) as k goes to infinity; $u_{(k+1)S-1}$ from above and t_{kS} from below. The rate at which they converge is upper bounded by their difference, which (by inspection of the absolute values of the differences of their corresponding terms) is seen to decrease asymptotically as $O(1/k)$.

8 Discretizing transmission

One objection to the preceding continuous models is that they require the movie to be split into many small, arbitrary sized pieces as k increases. This may not be possible or desirable since we can't split a digital movie into pieces smaller than a single bit and the communication protocol may require packets to contain some integral number of bits.

Another objection is our "parallel strips" view of the channel. One might argue that a channel transmits a *sequence* of bits or packets, and that we must convert from our parallel strips view to a serialized packet view. There is an obvious approach to this conversion: simply read and transmit a constant amount (a packet) from each strip in round-robin order (also known as *time-division multiplexing*). However, we may have movie segments that are not divisible into an integral number of constant size packets. If we follow the obvious approach, we may create a packet that spans the start/end boundary of a strip's segment. The number of packets we need to obtain the segment from this strip is then one more than it should be and our serialized stripped schedule may fail to deliver the segment in time. To remedy this problem, we use our existing algorithm to obtain a stripped schedule for a channel with less than the actual available bandwidth. We then *fatten* each strip by a constant factor, enough to permit the transmission of each segment in one fewer packet.

Let S be the server's bandwidth and R the receiver's bandwidth. We pretend that the server can send and we can receive only an α fraction of the regular bandwidth for some $\alpha < 1$. We divide the server channel into kS strips, each with bandwidth α/k . Using our optimal algorithm (Theorem 5.1), we create a schedule with delay $d = m\Delta^{(k/\alpha)}(\alpha R, \alpha S)$ for a movie of length m where

$$\Delta^{(k)}(R, S) = \frac{1}{-1 + \left(1 + \frac{1}{k}\right)^{kS} \sum_{j=0}^{\lfloor \frac{kS}{kR+1} \rfloor} (-\lambda)^j \binom{kS-jkR}{j}}$$

Let b be the length of the movie that can be played using one packet's worth of information. To read the first segment (of length $d\alpha/k$) we need $\lceil \frac{d\alpha/k}{b} \rceil$ packets and they must be received by time d when transmitted on the fattened strip (of bandwidth $1/k$). Thus, α must be small enough so that

$$\left\lceil \frac{d\alpha/k}{b} \right\rceil b \leq d \frac{1}{k}.$$

This is satisfied for $\alpha = 1 - \frac{b}{d/k}$. Notice that for α to be positive, b must be less than d/k , reflecting the need for the packet size to be smaller than the number

of bits needed to transmit the first segment. Since the first segment is smaller than any other segment, this choice of α insures that the required number of packets to carry segment i arrive on the fattened strip before segment i arrived on the original strip, for all i .

The cost of this packetization is an increase in delay. With server bandwidth S and receiver bandwidth R , we would expect our delay to be $m\Delta^{(k)}(R, S)$. Our delay is $m\Delta^{(k/\alpha)}(\alpha R, \alpha S)$. For example, if $R = S$, this is larger by a factor of

$$\frac{(1 + 1/k)^{kS} - 1}{(1 + \alpha/k)^{kS} - 1} \approx e^{\frac{b}{d/k}S}.$$

One should bear in mind that the length b of the movie that can be shown using one packet's worth of information is typically much smaller than the delay length d , so even for a rather large fragmentation factor k , $\frac{b}{d/k}$ is small.

9 Conclusion

We consider broadcast protocols for video on demand when server bandwidth S and receiver bandwidth R may differ. We describe a simple stripped protocol in the early model (where each segment is received entirely before being played) that achieves optimal movie length for a fixed delay among all such protocols with the same fragmentation factor.

10 Acknowledgements

We would like to thank Richard Ladner for introducing us to some of the problems in this area.

References

- [1] A. Bar-Noy, R. E. Ladner, and Tami Tamir. Scheduling techniques for media-on-demand. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 791–800, January 2003.
- [2] L. Engebretsen and M. Sudan. Harmonic broadcasting is optimal. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 431–432, January 2002.
- [3] L. Gao, J. Kurose, and D. Towsley. Efficient schemes for broadcasting popular videos. In *Proc. IEEE Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, July 1998.
- [4] A. Hu, I. Nikolaidis, and P. van Beek. On the design of efficient video-on-demand broadcast schemes. In *Proc. Symp. on Modeling, Analysis and Simulation of Computer and Telecommunications Systems (MASCOTS)*, pages 262–269, 1999.
- [5] Ailan Hu. Video-on-demand broadcasting protocols: a comprehensive study. In *Proc. IEEE INFOCOM*, pages 508–517, April 2001.

- [6] L. Juhn and L. Tseng. Harmonic broadcasting for video-on-demand service. *IEEE Trans. on Broadcasting*, 43(3):268–271, September 1997.
- [7] J.-F. P aris, S. W. Carter, and D. D. E. Long. A low bandwidth broadcasting protocol for video on demand. In *Proceedings of the 7th International Conference on Computer Communications and Networks*, pages 690–697, October 1998.
- [8] S. Viswanathan and T. Imielinski. Metropolitan area video-on-demand service using pyramid broadcasting. *Multimedia Systems*, 4(4):197–208, 1996.

A Appendix

Proof. [of Theorem 3.1] Let

$$\Theta'(t, x) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \Theta(t, x) dt$$

and

$$\Psi'(t, \delta, x) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \Psi(t, \delta, x) dt.$$

We note that (Θ', Ψ') is consistent

$$\begin{aligned} \Psi'(t, \delta, x) &= \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \Psi(t, \delta, x) dt \\ &\leq \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \Theta(t + \delta, x) dt \\ &= \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \Theta(t, x) dt \\ &= \Theta'(t, x); \end{aligned}$$

has receiver bandwidth at most R

$$\begin{aligned} \int_0^m \Psi'(t, \delta, x) dx &= \int_0^m \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \Psi(t, \delta, x) dt dx \\ &= \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \int_0^m \Psi(t, \delta, x) dx dt \\ &\leq \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T R dt \\ &= R; \end{aligned}$$

and has sender bandwidth at most S

$$\begin{aligned} \int_0^m \Theta'(t, x) dx &= \int_0^m \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \Theta(t, x) dt dx \\ &= \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \int_0^m \Theta(t, x) dx dt \\ &\leq \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T S dt \\ &= S. \end{aligned}$$

Now suppose (Θ', Ψ') does not achieve a length m video transfer with delay d . Then there exists a time t and a movie interval $[a, a']$ such that

$$\int_0^{a'} \Psi'(t, d + a', x) dx - \int_0^a \Psi'(t, d + a, x) dx < a' - a.$$

But

$$\begin{aligned} & \int_0^{a'} \Psi'(t, d + a', x) dx \\ &= \int_0^{a'} \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \Psi(t, d + a', x) dt dx \\ &= \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \int_0^{a'} \Psi(t, d + a', x) dx dt \\ &\geq \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \left(\int_0^a \Psi(t, d + a, x) dx + a' - a \right) dt \\ &= a' - a + \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \int_0^a \Psi(t, d + a, x) dx dt \\ &= a' - a + \int_0^a \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \Psi(t, d + a, x) dt dx \\ &= a' - a + \int_0^a \Psi'(t, d + a, x) dx \end{aligned}$$

This contradiction proves that (Θ', Ψ') does in fact achieve a length m video transfer with delay d .