Optimal Dynamic Video-On-Demand using Adaptive Broadcasting

Therese Biedl[1], Erik D. Demaine[2], Alexander Golynski[1], Joseph D. Horton[3], Alejandro López-Ortiz[1], Guillaume Poirier[1], and Claude-Guy Quimper[1]

[1] School of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada, {agolynski, alopez-o, cquimper, gpoirier, biedl}@uwaterloo.ca
[2] MIT Laboratory for Computer Science, 200 Technology Square, Cambridge, MA 02139, USA, edemaine@mit.edu
[3] Faculty of Computer Science, University of New Brunswick, P. O. Box 4400, Fredericton, N. B. E3B 5A3, Canada, jdh@unb.ca

Abstract. We consider the transmission of a movie over a broadcast network to support several viewers who start watching at arbitrary times, after a wait of at most t_{wait} minutes. A recent approach called harmonic broadcasting optimally solves the case of many viewers watching a movie using a constant amount of bandwidth. We consider the more general setting in which a movie is watched by an arbitrary number v of viewers, and v changes dynamically. A natural objective is to minimize the amount of resources required to achieve this task. We introduce two natural measures of resource consumption and performance—total bandwidth usage and maximum momentary bandwidth usage—and propose strategies which are optimal for each of them. In particular, we show that an adaptive form of pyramid broadcasting is optimal for both measures simultaneously, up to constant factors. We also show that the maximum throughput for a fixed network bandwidth cannot be obtained by any online strategy.

1 Introduction

Video-on-demand. A drawback of traditional TV broadcasting schemes is that the signal is sent only once and all viewers wishing to receive it must be listening at time of broadcast. To address this problem, viewers rely on recording devices (VCR, TiVo) that allow them to postpone viewing time by recording the program at time of broadcast for later use. A drawback of this solution is that the viewer must predict her viewing preferences in advance or else record every single program being broadcast, neither of which is practical. One proposed solution is to implement a video-on-demand (VoD) distribution service in which movies or TV programs are sent at the viewer's request. Considerable commercial interest has inspired extensive study in the networking literature [CP01,DSS94,EVZ00,EVZ01,JT97,JT98,ME+01,PCL98a,PCL98b,VI96,Won88] and most recently in SODA 2002 [ES02,B-NL02].

Previous approaches. The obvious approach—to establish a point-to-point connection between the provider and each viewer to send each desired show at

each desired time—is incredibly costly. *Pay-per-view* is a system in which the system broadcasts a selected set of titles and viewers make selections among the titles offered. Each movie that is, say, n minutes long gets broadcast over k channels at equally spaced intervals. The viewer then waits at most n/k minutes before she can start watching the movie. If k were sufficiently large, pay-per-view would become indistinguishable from video-on-demand from the viewer's perspective. This property is known as *near video-on-demand* (nVoD) or simply VoD. In practice, movies are roughly 120 minutes long, which would require an impractical number of channels per movie for even a 5-minute wait time.

Viswanathan and Imielinski [VI96] observed that if viewers have specialized hardware available (such as a TiVo, DVD-R, or digital decoder standard with current cable setups), then it is possible to achieve video-on-demand with substantially lower bandwidth requirements. In practice, the physical broadcast medium is typically divided into *physical channels*, each of which has precisely the required bandwidth for broadcasting a single movie at normal play speed (real time). The idea is to simultaneously transmit different segments of a movie across several channels. The set-top device records the signals and collates the segments into viewing order.

Harmonic broadcasting. Juhn and Tseng [JT97] introduced the beautiful concept of *harmonic broadcasting* which involves dividing a movie into n equal sized *segments* of length t_{wait}. Throughout the paper, it is assumed that movies are encoded at a constant bitrate. The n segments are broadcast simultaneously and repeatedly, but at different rates; refer to Figure 1. Specifically, if we label the segments $S_1, \ldots, S_n$ in order, then segment S_i is sent at a rate of $1/i$. In other words, we set up n virtual channels, where virtual channel C_i has the capacity of $1/i$ of a physical channel, and channel C_i simply repeats segment S_i over and over. Whenever a viewer arrives, the first i segments will have been broadcast after monitoring for $i + 1$ time *units* (one time unit is t_{wait} minutes), so the viewer can start playing as soon as the first segment has arrived. The maximum waiting time for a viewer in this scheme is t_{wait} minutes. The number of physical channels required by this scheme is the sum of the virtual channel capacity, $1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$, which is the nth Harmonic number H_n. Asymptotically, $H_n \approx \ln n + \gamma + O(1/n)$ where $\gamma \approx 0.5572$ is Euler's constant.

Pâris et al. [PCL98b] improved this scheme and gave more precise bounds on the required bandwidth and needed waiting time. Engebretsen and Sudan [ES02] proved that harmonic broadcasting is an optimal broadcasting scheme for one movie with a specified maximum waiting time. This analysis assumes that the movie is encoded at a constant bitrate, and that at every time interval $[it_{\text{wait}}, (i+1)t_{\text{wait}}]$ $i \in \{0, 1, \ldots, n-1\}$, at least one viewer starts watching the movie. Hence, harmonic broadcasting is effective provided there are at least as many viewers as segments in the movie (e.g., 24 for a 120-minute movie and a 5-minute wait), but overkill otherwise.

Adaptive broadcasting. We introduce a family of *adaptive* broadcasting schemes which adapt to a dynamic number v of viewers, and use considerably less bandwidth for lower values of v.
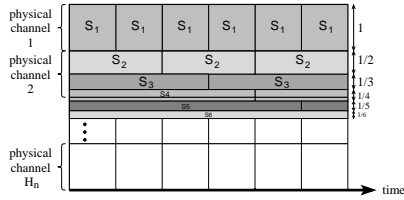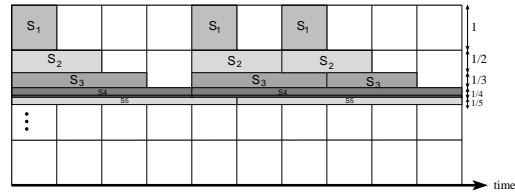
Fig. 1. Harmonic broadcasting.   **Fig. 2.** Adaptive harmonic broadcasting.

To simplify the analysis, we decompose the entire broadcasting duration (which might be infinite) into *timespans* of $T = mt_{\text{wait}}$ minutes in length, (i.e., m segments long), and consider requests from some number v of viewers arriving within each such timespan. Notice that some segments sent to a viewer who started watching in timespan $[0, T]$ are actually broadcast in the next timespan $[T, 2T]$. We ignore the cost of such segments counting only those segments *received* during the current timespan. Provided that T is big enough, the ignored cost is negligible compared to the cost induced by non-overlapping viewers. Let v denote the number of viewers in a timespan, ignoring any leftover viewers who started watching in the previous timespan. The number v can change from timespan to timespan, and thus bounds stated in terms of v for a single timespan adapt to changes in v. The bounds we obtain can also be combined to apply to longer time intervals: in a bound on total bandwidth, the v term becomes the average value of v, and in a bound on maximum bandwidth used at any time, the v term becomes the maximum value of v over the time interval.

The viewer arrival times are unknown beforehand, and hence the algorithm must react in an online fashion. In this scenario, our goal is to minimize bandwidth required to support VoD for v viewers, where the maximum waiting time is a fixed parameter t_{wait}. Such an algorithm must adapt to changing (and unknown) values of v, and adjust its bandwidth usage according. In particular, it is clear that in general harmonic broadcasting is suboptimal, particularly for small values of v. Carter et al. [CP01] introduced this setting of a variable number of viewers, and proposed a heuristic for minimizing bandwidth consumption; here we develop algorithms that are guaranteed to be optimal.

Objectives. For a given sequence of viewer arrival times, we propose three measures of the efficiency of an adaptive broadcasting strategy:

1. Minimizing the total amount of data transmitted, which models the requirement of a content provider who purchases capacity by the bit from a data carrier. The total capacity required on the average is also a relevant metric if we assume a large number of movies being watched with requests arriving randomly and independently.
2. Minimizing the maximum number of channels in use at any time, which models the realistic constraint that the available bandwidth has a hard upper limit imposed by hardware, and that bandwidth should be relatively balanced throughout the transmission.

3. Obtaining a feasible schedule subject to a fixed bandwidth bound, which models the case where the content provider pays for a fixed amount of bandwidth, whether used or not, and wishes to maximize the benefit it derives from it. (In contrast to the previous constraints, this measure favors early broadcasting so that the bandwidth is always fully used.)

Broadcasting schemes can be distinguished according to two main categories: *integral* which distribute entire segments of movies from beginning to end in physical channels, in real time, and *nonintegral* which distribute segments at various nonintegral rates and allow viewers to receive segments starting in the middle. For example, pay-per-view is a simple integral scheme, whereas harmonic broadcasting is nonintegral. Integral broadcasting is attractive in its simplicity.

Our results. We propose and analyze three adaptive broadcasting schemes, as summarized in Table 1. In Section 2, we show that a lazy integral broadcasting scheme is exactly optimal under Measure 1, yet highly inefficient under Measure 2, which makes this scheme infeasible in practice. Nonetheless this result establishes a theoretical baseline for Measure 1 against which to compare all other algorithms. Then in Section 3 we analyze an adaptive form of harmonic broadcasting (which is nonintegral) that uses at most $\ln n$ channels at any time, as in harmonic broadcasting, but whose average number of channels used over the course of a movie is $\ln \min\{v, n\} + 1$ plus lower-order terms (Section 3). The latter bound matches, up to lower-order terms, the optimal bandwidth usage by the lazy algorithm, while providing much better performance under Measure 2. However, $\ln n$ channels is suboptimal, and in Section 4, we show that an integral adaptive pyramid broadcasting scheme is optimal under Measure 2 up to lower-order terms while still being optimal up to a constant multiplicative factor of $\lg e \approx 1.4427$ under Measure 1. Lastly in Section 5 we show that a natural greedy strategy is suboptimal for Measure 3, and furthermore that no online strategy matches the offline optimal performance when multiple movies are involved.

Broadcasting alg.		Integral?	Total bandwidth usage	Max. bandwidth usage
Harmonic [JT97, ...]		No	$\text{OPT}(n) \sim n \ln n + \gamma n$	$\ln n + \gamma$
Lazy	[§2]	Yes	$\text{OPT}(n, v) \sim n \ln \min\{v, n\}$ $+ (2\gamma - 1) n$	$\sim n^{\ln 2 / \ln \ln n}$
Adapt. harmonic	[§3]	No	$m \ln \min\{(n+1)v/m, n+1\}$ $+ m + v$	$\ln n + \gamma$
Adapt. pyramid	[§4]	Yes	$m \lg \min\{(n+1)v/m, n+1\}$ $+ O(m)$	$\min\{v(t), \lg n\}$ $+ O(1)$

Table 1. Comparison of our results and harmonic broadcasting.

2 Lazy Broadcasting

First we consider Measure 1 in which the objective is to minimize the total amount of data transmitted (e.g., because we pay for each byte transferred).
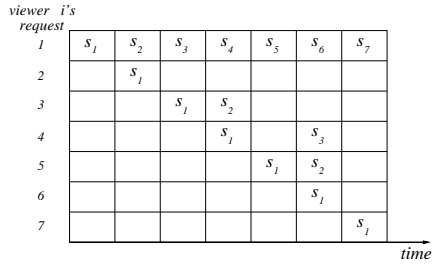
viewer i's request							
1	s_1	s_2	s_3	s_4	s_5	s_6	s_7
2		s_1					
3			s_1	s_2			
4				s_1		s_3	
5					s_1	s_2	
6						s_1	
7							s_1

time

Fig. 3. Lazy broadcasting schedule with $v = n$ viewers (one every segment).

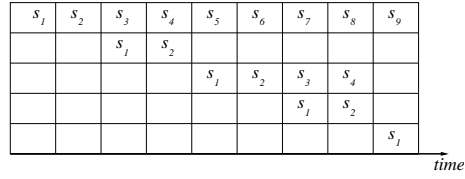|								
s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9
		s_1	s_2					
				s_1	s_2	s_3	s_4	
						s_1	s_2	
								s_1

time

Fig. 4. Lazy broadcasting schedule adapting to $v = n/2$ viewers (one every second segment).

Here the goal is to maximize re-use among multiple (offset) transmissions of the same movie.[4] For this case, we propose *lazy broadcasting* and show that it is exactly optimal under this measure. This algorithm has a high worst-case bandwidth requirement, and hence is impractical. However, it provides the optimal baseline against which to compare all other algorithms. In particular, we show that adaptive harmonic broadcasting is within lower-order terms of lazy, and adaptive pyramid broadcasting is within a constant factor of lazy, and therefore both are also roughly optimal in terms of total data transmitted. Because the worst-case bandwidth consumption of harmonic and pyramid broadcasting is much better than that of lazy broadcasting, these algorithms will serve as effective compromises between worst-case and total bandwidth usage. The lazy algorithm sends each segment of the movie as late as possible, the moment it is required by one or more viewers; see Figure 3. All transmissions proceed at a rate of 1 (real time / play speed).

Theorem 1. *The total amount of data transmitted by the lazy algorithm is the minimum possible.*

Proof. Consider any sequence of viewers' arrival times, and a schedule A which satisfies requests of these viewers. Perform the following two operations on movie segments sent by schedule A, thereby changing A. For each time that a movie segment is sent by A:

1. If the movie segment is not required by A because every viewer can otherwise record the movie segment beforehand, delete the movie segment.
2. If the movie segment is not required when A transmits it but is required at a later time, delay sending the segment until the earliest time at which it is required and then send at full rate.

---

[4] Amortization of data transfers has been observed empirically over internet service provider connections. In this case, it is not uncommon to sell up to twice as much capacity than physically possible over a given link, based on the observed tendency that it is extremely uncommon for all users to reach their peak transmission rate simultaneously.

After processing all movie segments, repeat until neither operation can be done to any movie segment. This process is finite because the number of segments and time intervals during which they can be shown is finite, and each operation makes each segment nonexistent or later. The claim is that the resulting schedule is the same as the lazy schedule. The proof can be carried out by induction on the number of segments requested.

$\square$

Now that we know that the lazy schedule optimizes the total amount of data transmitted, we give analytic bounds on this amount. First we study a full broadcast schedule (nonadaptive). This is equivalent to the setting in which a new viewer arrives at every time boundary between segments, see Figure 3. Notice that the ith segment is sent at time i to satisfy the request of the first viewer. The other first $i-1$ viewers also see this transmission and record the segment. On the other hand, the $(i+1)$st viewer did not witness this transmission, and requests the segment i time units after it started watching the movie, i.e., after time i. Hence S_i must be resent at time $2i$. In general, the ith segment must be sent at precisely those times that are multiples of i. Thus the ith segment is sent a $1/i$ fraction of the time, which shows that the total amount of bandwidth required is $n(H_n + O(1))$ for a timespan of n segments. In fact, we can obtain a more precise lower bound by observing that, at time i, we transmit those segments whose index divides i, and hence the total amount of bandwidth required is the sum of the divisors of i for $i = 1, 2, \ldots, n$.

Theorem 2. *The total amount of data transmitted by the lazy algorithm for n viewers arriving at equally spaced times during a timespan of n segments for a movie n segments long is $n \ln n + (2\gamma - 1)\, n + O(\sqrt{n})$ segments.*

The lazy algorithm is similar to the harmonic broadcasting algorithm, the only difference being that harmonic broadcasting transmits the ith segment of the movie evenly over each period of i minutes, whereas the lazy algorithm sends it in the last minute of the interval. Comparing the bound of Theorem 2 with the total bandwidth usage of harmonic broadcasting, $n \ln n + \gamma n + O(1)$ segments, we find a difference of $\approx 0.4228\, n + o(n)$. Thus, harmonic broadcasting is nearly optimal under the total bandwidth metric, for $v = n$ viewers.

In contrast to harmonic broadcasting, which uses $H_n \sim \ln n$ channels at once, the worst-case bandwidth requirements of the lazy algorithm can be substantially larger:

Theorem 3. *The worst-case momentary bandwidth consumption for lazy transmission of a movie with n segments and n viewers is, asymptotically, at least $n^{\ln 2 / \ln \ln n}$.*

In the case of $v < n$ viewers (see Figure 4), Theorem 1 still shows optimality, but the bounds in Theorem 2 become weak. The next theorem gives a lower bound on the bandwidth consumed by the lazy algorithm. A matching upper bound seems difficult to prove directly, so instead we rely on upper bounds for

other algorithms which match up to lower-order terms, implying that lazy is at least as good (being optimal).

Theorem 4. *The total amount of data transmitted by the lazy algorithm for v viewers arriving at equally spaced times during a timespan of n segments for a movie n segments long is at least $n \ln \min\{v, n\} + (2\gamma - 1)n + O(\sqrt{n})$ in the worst case.*

3 Adaptive Harmonic Broadcasting

In this section we propose a variation of harmonic broadcasting, called *adaptive harmonic broadcasting*, that simultaneously optimizes total bandwidth (within a lower-order term) and worst-case bandwidth usage at any moment in time. The key difference with our approach is that it adapts to a variable number of viewers over time, $v(t)$. In contrast, harmonic broadcasting is optimal only when viewers constantly arrive at every of the n movie segments.

Adaptive harmonic broadcasting defines virtual channels as in normal harmonic broadcasting, but not all virtual channels will be broadcasting at all times, saving on bandwidth. Whenever a viewing request arrives, we set the global variable t_{request} to the current time, and turn on all virtual channels. If a channel C_i was silent just before turning it on, it starts broadcasting S_i from the beginning; otherwise, the channel continues broadcasting S_i from its current position, and later returns to broadcasting the beginning of S_i. Finally, and most importantly, channel C_i stops broadcasting if the current time ever becomes larger than $t_{\text{request}} + i\, t_{\text{wait}}$.

Figure 2 illustrates this scheme with viewers arriving at times $t = 0, 4, 6$.

Theorem 5. *The adaptive harmonic broadcasting schedule broadcasts a movie n segments long to v active viewers with a maximum waiting time t_{wait} using at most H_n channels at any time and with a total data transfer of $m \min\{\ln(n + 1) - \ln(m/v), \ln(n + 1)\} + m + v$ segments during a timespan of $T = m\, t_{\text{wait}}$ minutes.*

Proof. Let t_i denote the time at which viewer i arrives, where $t_1 \leq t_2 \leq \cdots \leq t_v$ and $t_v - t_1 \leq T$. Let $g_i = (t_{i+1} - t_i)/t_{\text{wait}}$ denote the normalized gaps of time between consecutive viewer arrivals.

To simplify the analysis, we do the following discretization trick. Process viewers that arrived on $[0, T]$ interval from left to right. Delete all viewers that arrived within time t_{wait} from the last viewer we considered (t_{last}) and replace them by one viewer with arrival time $t_{\text{last}} + t_{\text{wait}}$. Clearly, this only can increase bandwidth requirements in both intervals $[t_{\text{last}}, t_{\text{last}} + t_{\text{wait}}]$ and $[t_{\text{last}} + t_{\text{wait}}, T]$. Choose the next viewer that have not been considered so far and repeat the procedure. This discretizing procedure gives a method of counting "distinct" viewers, namely, on the interval $[0, T]$ there are no more than m of them and $g_i \geq 1$ for all i. In particular, if the number of viewers is more than m then adaptive

harmonic broadcasting scheme degrades to simple harmonic broadcasting and theorem holds. Consider the case $v \leq m$.

The total amount of bandwidth used by all viewers is the sum over all i of the bandwidth $B(i)$ recorded by viewer i in the time interval between t_i and t_{i+1}. Each $B(i)$ can be computed locally because we reset the clock t_{request} at every time t_i that a new viewer arrives. Specifically, $B(i)$ can be divided into (1) the set of virtual channel transmissions that completed by time t_{i+1}, that is, transmissions of length at most g_i; and (2) the set of virtual channel transmissions that were not yet finished by time t_{i+1}, whose cost is trimmed. Each channel C_j of the first type ($j \leq g_i$) was able to transmit the entire segment in the time interval of length g_i, for a cost of one segment. Each channel C_j of the second type ($j \geq g_i$) was able to transmit for time g_i at a rate of $1/j$, for a cost of a g_i/j fraction of a segment. Thus, $B(i)$ is given by the formula

$$B(i) = \sum_{j=1}^{\lfloor g_i \rfloor} 1 + \sum_{j=\lfloor g_i \rfloor + 1}^{n} g_i/j \;=\; \lfloor g_i \rfloor + g_i \cdot \left(H_n - H_{\lfloor g_i \rfloor} \right).$$

We have the bound $B(i) \leq g_i(1 + \ln((n+1)/g_i)) + 1$, proof omitted. Now the total amount of data transmitted can be computed by summing over all i, which gives

$$B = \sum_{i=1}^{v} B(i) \;\leq\; \sum_{i=1}^{v} \left[g_i \left(1 + \ln \frac{n+1}{g_i} \right) + 1 \right] \;\leq\; m + v + \sum_{i=1}^{v} g_i \ln \frac{n+1}{g_i}.$$

The last summation can be rewritten as $n + 1$ times $\sum_{i=1}^{v} (g_i/(n+1)) \ln((n+1)/g_i)$. This expression is the entropy H, which is maximized when $g_1 = g_2 = \cdots = g_v = m/v$. Hence the total amount of bandwidth B is at most $m + v + m \ln(v(n+1)/m)$, as desired when $v \leq m$. □

This proof in fact establishes a tighter bound on the number of channels required, namely, the base-e entropy of the request sequence. Sequences with low entropy require less bandwidth.

4   Adaptive Pyramid Broadcasting

In this section we propose an integral adaptive broadcasting scheme which is optimal up to constant factors for both Measure 1 and Measure 2, that is, total amount of data transmitted and minimizing the maximum number of channels in use at any time.

Viswanathan and Imielinski [VI96] proposed the family of *pyramid broadcasting* schemes in which the movie is split into *chunks* of geometrically increasing size, that is, $|S_i| = \alpha|S_{i-1}|$ for some $\alpha \geq 1$, with each segment being broadcast at rate 1 using an entire physical channel.

In our case, we select $\alpha = 2$ and $|S_0| = t_{\text{wait}}$. Thus, there are $N = \lceil \lg(n+1) \rceil$ chunks $S_0, \ldots, S_{N-1}$, each consisting of an integral number of segments. Chunk

S_i has length $2^i t_{\text{wait}}$ (except for the last one), and hence covers the interval $[(2^i - 1)t_{\text{wait}}, (2^{i+1} - 1)t_{\text{wait}}]$ of the movie. We first analyze the bandwidth used by any protocol satisfying two natural conditions.

Lemma 1. *Consider any broadcast protocol for sending segments $S_0, \ldots, S_{N-1}$ satisfying the following two conditions: (A) for every viewer, every segment is sent at most once completely, (B) no two parts of the same segment are sent in parallel; then the total bandwidth usage for v viewers within a timespan of $T = m\, t_{\text{wait}}$ minutes is at most $m \min\{N, N - \lg(m/v) + 1\}$ segments.*

Proof. Because there are $N = \lceil \lg(n + 1) \rceil$ chunks, and no chunk is sent more than once at a time (Property B), we surely use at most $m\,N$ bandwidth. This bound proves the claim if $v \geq m$, so assume from now on that $v < m$.

We classify chunks into two categories. The *short* chunks are chunks S_j for $j = 0, \ldots, t - 1$ for some parameter t to be defined later. By Property A, chunk S_j is sent at most v times, and hence contributes at most $v\,2^j$ to the bandwidth. Thus, the total bandwidth used by short chunks is at most $\sum_{j=0}^{t-1} v\,2^j = v\,(2^t - 1)$.

The *long* chunks are chunks S_j for $j = t, \ldots, N - 1$. By Property B, at most one copy of S_j is sent at any given moment, so chunk S_j contributes at most m segments to the total bandwidth. Thus, the total bandwidth used by long chunks is at most $m\,(N - t)$. Total bandwidth $v\,(2^t - 1) + m\,(N - t)$ is minimized for the value of t roughly $\lg(m/v)$. Thus the total bandwidth is at most $m(N - \lg(m/v) + 1) - v$ segments as desired. $\square$

There are several protocols that satisfy the conditions of the previous lemma. In particular, we propose the following adaptive variant of pyramid broadcasting; see Figure 5. Suppose there are v viewers arriving at times $t_1 \leq t_2 \leq \cdots \leq t_v$. We discretize the viewer arrival schedule as follows: all viewers arrived in the interval $(i\,t_{\text{wait}}, (i + 1)\,t_{\text{wait}}]$ are considered as one viewer arriving at time $(i + 1)\,t_{\text{wait}}$ for $i = 0, \ldots, T/t_{\text{wait}}$ and this made-up viewer will start watching the movie immediately (i.e. at time $(i + 1)\,t_{\text{wait}}$). Thus the waiting time for any user is at most t_{wait}, however the average waiting time is twice as less if viewers are arriving in the uniform fashion.

If at time $t_j + 2^i - 1$ viewer j has not seen the beginning of segment S_i, then this segment S_i is broadcast in full on channel i, even if viewer j has already seen parts of S_i. By this algorithm, viewer j is guaranteed to see the beginning of segment S_i by time $t_j + 2^i - 1$. Because we send segments from beginning to end, and in real time, viewer j will therefore have seen every part of S_i by the time it is needed. Furthermore, this protocol sends every segment at most once per viewer, satisfying Property A.

It is less obvious that we never send two parts of S_i in parallel. Suppose viewer j requests segment S_i. This means that we never started broadcasting segment S_i between when j arrived (at time t_j) and when S_i is requested (at time $t_j + 2^i$). Because S_i has length 2^i, and because we send segments in their entirety, this means that at time $t_j + 2^i$ we are done with any previous transmission of S_i. Lemma 1 therefore applies, proving the first half of the following theorem:
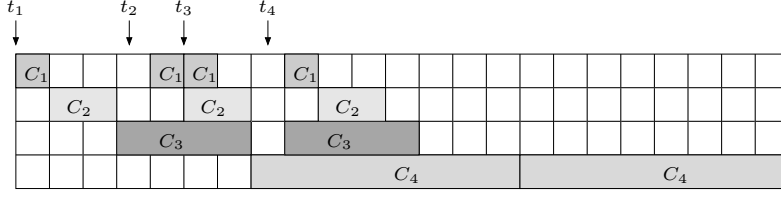
Fig. 5. Adaptive pyramid broadcasting.

Theorem 6. *The total bandwidth usage of adaptive pyramid broadcasting is at most $m \min\{\lceil \lg(n+1) \rceil, \lg((n+1)v/m) + 1\}$ segments, which is within a factor of $\lg e \approx 1.4427$ (plus lower order terms) of optimal. Furthermore, the maximum number of channels in use at any moment t is at most $\min\{v(t), \lceil \lg(n+1) \rceil\}$.*

Finally, we prove that the maximum number of channels used by adaptive pyramid broadcasting is optimal among all online adaptive broadcasting algorithms: in a strong sense, the number of channels cannot be smaller than v, unless v is as large as $\sim \lg n$.

Theorem 7. *Consider any online adaptive broadcasting algorithm that at time t uses $c(t)$ physical channels to serve $v(t)$ current viewers and for which $c(t) \leq v(t)$ at all times t. Then there is a sequence of requests such that, for all $v \in \{1, 2, \ldots, \lg n - \lg \lg n\}$, there is a time t when $v(t) = v$ and $c(t) = v(t)$.*

Proof. Consider the sequence of requests at times 0, $\frac{1}{2}n$, $\frac{3}{4}n$, $\frac{7}{8}n$, $\frac{15}{16}n$, ..., $(1 - 1/2^i)n$, ... In this sequence of requests, we claim that no re-use of common segments between different viewers is possible in the time interval $[0, n)$. Consider the ith viewer, who arrives and starts recording at time $(1 - 1/2^i)n$. In the time interval $[(1 - 1/2^i)n, n)$, the ith viewer needs to be sent the first $n/2^i - 1$ segments of the movie. (The -1 term is because the viewer waits for the first time unit (segment), and only starts watching at time $(1 - 1/2^i)n + 1$.) But all previously arriving viewers must have already been sent those segments before time $(1 - 1/2^i)n$, because by construction they have already watched them by that time. Therefore, no segments that are watched by any viewer in the time interval $[0, n)$ can have their transmissions shared between viewers.

Now define the *buffer amount* of a viewer to be the the amount of time that each viewer is "ahead", i.e., the amount of time that a viewer could wait before needing its own rate-1 broadcast until the end of the movie (which is at least time n). Because there is no re-use between viewers, we maintain the invariant that, if there are v current viewers, then the total buffer amount of all viewers is at most v. A buffer amount of 1 for each viewer is easy to achieve, by having each viewer record for the one unit of wait time on its own rate-1 channel. It is also possible to "transfer" a buffer amount from one viewer to another, by partially using one viewer's channel to send part of another viewer's needed segment, but this operation never strictly increases the total buffer amount.

In the time interval $[(1 - 1/2^{v-1})n, (1 - 1/2^v)n)$, there are exactly v active viewers, and each viewer needs to watch $(1/2^{v-1} - 1/2^v)n$ segments, except for one viewer who watches one fewer segment. Viewers might during this time "use up" their buffer amount, by using their channels for the benefit of other viewers, catching up to real time. However, this can only decrease the resource requirement during this time interval by up to v, so the total resource requirement is still at least $v(1/2^{v-1} - 1/2^v)n - v - 1$. On the other hand, if there are c physical channels in use during this time interval when exactly v viewers are active, then the maximum bandwidth usable in this time interval, $c(1/2^{v-1} - 1/2^v)n$, must be at least the resource requirement. Thus, $c \geq v - (v - 1)/((1/2^{v-1} - 1/2^v)n) = v - 2^v(v - 1)/n$.

Because c measures physical channels, c is integral, so the bound on c in fact implies $c \geq v$ provided the error $2^v(v - 1)/n$ is less than 1. If $v \leq \lg n - \lg\lg n$, then $2^v(v-1) = (n/\lg n)(\lg n - \lg\lg n - 1) < n$. Therefore, for any v in this range (as claimed in the theorem), we need as many physical channels as viewers. $\square$

Adaptive pyramid broadcasting inherits the simplicity-of-implementation properties that have made pyramid broadcasting popular: not only is the algorithm integral on segments, it is integral on chunks, always broadcasting entire segments from beginning to end in real time.

5 Greedy Broadcasting and Offline Scheduling

Suppose we have a fixed amount of available bandwidth, and our goal is to satisfy as many viewers as possible. The natural greedy algorithm is to send the segments of a movie that are required soonest, as soon as there is available bandwidth. We imagine a *wavefront* sweeping through the requests in the order that they are needed. The *front time* must always remain ahead of real time, or in the worst case equal to real time. If the front time ever falls behind real time, some viewer will not be satisfied. The greedy algorithm is suboptimal in the following sense:

Theorem 8. *There is a sequence of requests for a single movie that is satisfiable within a fixed available bandwidth but for which the greedy algorithm fails to find a satisfactory broadcast schedule.*

Theorem 9. *There is a family of request sequences for two movies that is satisfiable offline within a fixed available bandwidth, but which can force any online scheduling algorithm to fail.*

6 Conclusions and Open Questions

We introduced the concept of adaptive broadcasting schedules which gracefully adjust to varying numbers of viewers. We measured the performance of three new algorithms under two metrics inspired by realistic bandwidth cost considerations. In particular, we showed that adaptive harmonic broadcasting is optimal

up to lower-order terms under total amount of data transmitted, and that adaptive pyramid broadcasting achieves optimal maximum channel use at the cost of a constant factor penalty on the total amount of data transmitted. All the algorithms generalize to multiple different-length movies being watched by different numbers of viewers, and the same worst-case optimality results carry over.

We also showed that any online algorithm might fail to satisfy a given bandwidth requirement that is satisfiable offline for a two-movie schedule. One open question is to determine the best competitive ratio on the fixed bandwidth bound achievable by online broadcasting schedules versus the offline optimal.

References

[AB+96] A. Albanese, J. Blörnet, J. Edmonds, M. Luby and M. Sudan. Priority encoding transmission. *IEEE Trans. Inform. Theory*, 42(6):1737–1744, 1996.

[B-NL02] A. Bar-Noy and R. E. Ladner. Windows scheduling problems for broadcast systems. *Proc. 13th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 433–442, 2002.

[CP01] S. R. Carter, J.-F. Paris, S. Mohan and D. D. E. Long. A dynamic heuristic broadcasting protocol for video-on-demand. *Proc. 21st International Conference on Distributed Computing Systems*, pages 657–664, 2001.

[DSS94] A. Dan, D. Sitaram, and P. Shahabuddin. Dynamic batching policies for an on-demand video server. *ACM Multimedia Systems*, 4(3):112–121, 1996.

[ES02] L. Engebretsen and M. Sudan. Harmonic broadcasting is bandwidth-optimal assuming constant bit rate *Proc. 13th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 431–432, 2002.

[EVZ00] D. L. Eager, M. K. Vernon and J. Zahorjan. Bandwidth skimming: a technique for cost-effective video-on-demand. *Proc. IS&T/SPIE Conference on Multimedia Computing and Networking (MMCN)*, pages 206-215, 2000.

[EVZ01] D. L. Eager, M. K. Vernon and J. Zahorjan. Minimizing bandwidth requirements for on-demand data delivery. *IEEE Transactions on Knowledge and Data Engineering*, 3(5):742–757, 2001.

[JT97] L. Juhn and L. Tseng. Harmonic broadcasting for video-on-demand service. *IEEE Transactions on Broadcasting*, 43(3):268–271, 1997.

[JT98] L. Juhn and L. Tseng. Fast data broadcasting and receiving scheme for popular video service. *IEEE Trans. on Broadcasting*, 44(1):100–105, 1998.

[ME+01] A. Mahanti, D. L. Eager, M. K. Vernon and D. Sundaram-Stukel. Scalable on-demand media streaming with packet loss recovery. *Proc. 2001 ACM Conf. on Applications, Technologies, Architectures and Protocols for Computer Communications (SIGCOMM'01)*, pp. 97–108, 2001.

[PCL98a] J.-F. Pâris, S. W. Carter and D. D. E. Long. Efficient broadcasting protocols for video on demand. *Proc. 6th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pages 127–132, 1998.

[PCL98b] J.-F. Pâris, S. W. Carter and D. D. E. Long. A low bandwidth broadcasting protocol for video on demand. *Proc. 7th International Conference on Computer Communications and Networks*, pages 690–697, 1998.

[VI96] S. Viswanathan and T. Imielinski. Metropolitan area video-on-demand service using pyramid broadcasting. *Multimedia Systems*, 4(4):197–208, 1996.

[Won88] J. W. Wong. Broadcast delivery. *Proc. of the IEEE*, 76(12):1566–1577, 1988.