Title

Efficient broadcasting protocols for video on demand

Authors

Paris, J-F
Carter, SW
Long, DDE

Publication Date

1998

DOI

10.1109/mascot.1998.693685

Efficient Broadcasting Protocols for Video on Demand

Jehan-François Pâris[†]
Department of Computer Science
University of Houston
Houston, TX 77204-3475
paris@cs.uh.edu

Steven W. Carter Darrell D. E. Long[‡]
Department of Computer Science
Baskin School of Engineering
University of California, Santa Cruz
Santa Cruz, CA 95064
{carter, darrell}@cse.ucsc.edu

Abstract

Broadcasting protocols can improve the efficiency of video on demand services by reducing the bandwidth required to transmit videos that are simultaneously watched by many viewers. We present here two broadcasting protocols that achieve nearly the same low bandwidth as the best extant broadcasting protocol while guaranteeing a lower maximum access time. Our first protocol, cautious harmonic broadcasting, requires somewhat more bandwidth than our second protocol, quasi-harmonic broadcasting, but is also much simpler to implement.

Keywords: video on demand, video broadcasting.

1 Introduction

Video on demand (VOD) proposes to provide cable television subscribers with the possibility of watching the video of their choice at the time of their choice, as if they were watching a rented video cassette. Despite the attractiveness of the concept, so far VOD has not been a commercial success because the technology is still very expensive and its potential users are unwilling to pay much more for a VOD selection than they are used to paying for a video cassette rental.

Broadcasting is one of several techniques that aim at reducing the cost of VOD [1]. It is clearly not a panacea as it only applies to videos that are likely to be watched by many viewers. Even so, the savings that can be achieved are nevertheless considerable, as it is often the case that 55 percent of the demand is for the 20 most popular videos [2,3]. A naive broadcasting strategy would simply consist of retransmitting the same video on n distinct channels at equal time intervals $d = D/n$, where D is the duration of the video being broadcast. The major problem with this approach is the number of channels per video required to achieve a reasonable access time. Assuming an average video duration of 120 minutes, 24 channels per video would be required to guarantee that no customer would ever have to wait more than 5 minutes.

Viswanathan and Imielinski [4] have proposed a better solution that assumes that the clients can receive and store some segments of a video while watching other segments. Their *pyramid broadcasting protocol* has been followed by several more recent proposals among which we should mention Aggarwal, Wolf and Yu's *permutation-based pyramid broadcasting protocol* [5], Hua and Sheu's *skyscraper broadcasting protocol* [6] and Juhn and Tseng's *harmonic broadcasting protocol* [7].

Of all these protocols, harmonic broadcasting is the one promising the lowest bandwidth cost to achieve a given access time. However, this excellent performance is marred by the fact that harmonic broadcasting sometimes fails to provide actual on-time delivery of all frames. Juhn [8] suggested that this problem could be fixed by slightly delaying the moment at which the clients start consuming the video. While this extra buffering solves the problem, it also very significantly increases the video access time and reduces the competitive advantage of harmonic broadcasting over its rivals.

We present here two variants of the harmonic protocol that do not impose on their end-users any additional delay before they can begin viewing the video they ordered. The first protocol, *cautious harmonic broadcasting*, entirely avoids the problem by guaranteeing that all frames of any video segment transmitted at a reduced bandwidth will be effectively present in the client memory before it starts playing that segment. The second protocol, *quasi-harmonic broadcasting*, modifies the organization of the segments to ensure on-time delivery of all frames under all conditions. While more complex than either harmonic broadcasting or cautious harmonic broadcasting, quasi-harmonic broadcasting offers a guaranteed on-time delivery of all frames without any significant bandwidth penalty.

2 Harmonic Broadcasting

Harmonic broadcasting (HB) [7] breaks a video into n equally-sized *segments*. If the length of the video is D and the consumption rate of the video is b, then $S = Db$ is the size of the video, and each segment S_i, for $1 \leq i \leq n$, has size S/n. HB then dedicates n streams for the video, and each stream i repeatedly shows segment S_i with band-

[†] On sabbatical leave at the Department of Computer Science, University of California, Santa Cruz.

[‡] This research was supported by the Office of Naval Research under Grant N00014–92–J–1807.

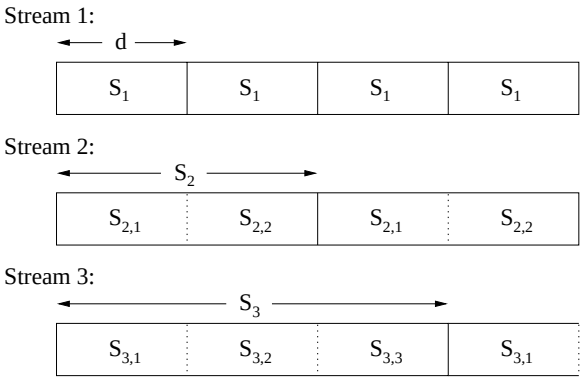

Figure 1: An illustration of the first three channels for a video under harmonic broadcasting.

width b/i (see Figure 1).

The client must wait for the beginning of an instance of segment S_1 before it can start receiving (and viewing) data. Once it starts receiving segment S_1, it will also start receiving every other stream dedicated for the video. That means the client and the VOD server must be able to support a bandwidth of

$$B_{HB} = \sum_{i=1}^{n} \frac{b}{i} = b \sum_{i=1}^{n} \frac{1}{i} = bH(n)$$

where $H(n)$ is the harmonic number of n. $H(n)$ grows very slowly, so even for large values of n, B_{HB} is likely to be less than six times the data consumption rate b.

We define a *slot* as the amount of time it takes for the client to consume a single segment of the video, and we represent this time by d. Thus, we have

$$d = \frac{S_1}{b} = \frac{D}{n}$$

and note that d is also the maximum amount of time a client must wait before receiving data.

We also define a *subsegment* as the part of a segment the client receives during a slot. Since the first stream uses bandwidth equal to the consumption rate, it only has a single subsegment, the segment itself. Every other stream i will divide the segment into i equal subsegments, $S_{i,1}$, $S_{i,2}, \ldots, S_{i,i}$.

Note that since the client will be receiving video data out of order, it will require local buffer space within the customer set-top box. The buffer space needs to be large enough to hold slightly less than 40 percent of the size of the video [7]. Using an example of MPEG-1 encoding and a two-hour video, the buffer would need to be approximately 500 megabytes in size.

The main idea behind HB is that when the client is ready to consume segment S_i, it will have been receiving data for $i-1$ slots of time. So the client will have $i-1$ of the i subsegments of segment S_i in its buffer, and it will receive the last subsegment while it is consuming the full segment. It is at this point that HB breaks down.

Claim 1 *Harmonic broadcasting does not always deliver all data on time.*

Proof:

Refer to Figure 1 and consider a client who arrives just in time to begin receiving the second instance of segment S_1. Call the time when the client begins receiving data t_0.

At time $t_0 + d$, the client will be ready to consume segment S_2 and it will have one subsegment of the segment, $S_{2,2}$, in its buffer. However, it will require all of the data from subsegment $S_{2,1}$ by time $t_0 + 3d/2$ but it will not receive it until time $t_0 + 2d$.

□

Since the problem with HB involves data not being at the client in time, a straightforward solution is to have the client simply delay a certain amount of time before consuming data. Clearly, a delay of one slot would work; the client would then have all i subsegments of S_i before consuming it. As we will see, the minimum required delay is not much lower.

Claim 2 *Harmonic broadcasting requires the client to wait for $(n-1)d/n$ units of time.*

Proof:

As before, assume that the client begins receiving data at time t_0. Now consider any stream i.

The worst case for the VOD server occurs when the client receives the first subsegment of stream i last. This is because the client will require all of the data from that subsegment before the data from any of the other subsegments.

The client can consume the first subsegment while it is receiving the subsegment as long as all of the data from it will arrive at the client in time. Since the data from the first subsegment will not finish arriving at the client until time $t_0 + id$, and since it takes time d/i to consume the subsegment, the client cannot start consuming the subsegment until time $t_0 + (i-1)d + (i-1)d/i$. Thus the client must delay by $(i-1)d/i$ units of time in order to receive all data from stream i on time.

Therefore, in order for the client to guarantee that it will receive all data from all streams on time, it must delay consumption by

$$\max_{1 \le i \le n} \left\{ \frac{(i-1)d}{i} \right\} = \frac{(n-1)d}{n}$$

units of time.

□

Since n will be large, the client will essentially have to wait an entire slot before consuming data.

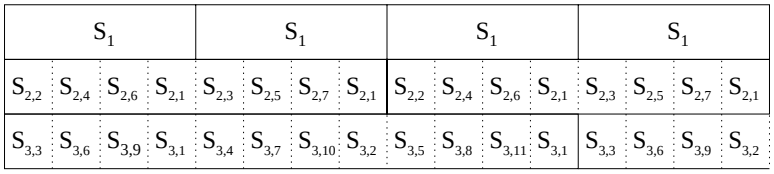

Figure 2: An illustration of the first three streams for a video under quasi-harmonic broadcasting when $m = 4$.

3 Our Solution

The two protocols we present trade some additional bandwidth for the guarantee that (a) all frames will always arrive on time and (b) the client will never have to wait any extra time after the beginning of an instance of segment S_1.

3.1 Cautious Harmonic Broadcasting

Cautious harmonic broadcasting (CHB) avoids frame delay problems by adopting a more conservative stream allocation policy. As in harmonic broadcasting, stream 1 transmits the first segment of the video S_1 at full bandwidth. The second stream is allocated differently: it transmits alternately segments S_2 and S_3 at full bandwidth. Streams 3 to $n-1$ respectively transmit segments S_4 to S_n at decreasing bandwidths $b_i = b/i$ for $i = 3, \ldots, n-1$.

Late frame deliveries cannot occur while the first three segments of the video are consumed because these three segments are broadcast at full bandwidth. They cannot either occur for any of the $n-3$ remaining segments because the entire contents of segment i will be retransmitted every $i-1$ subsegments and will thus be available to the client before it starts consuming S_i. The total bandwidth B_{CHB} required by the CHB protocol is given by:

$$B_{CHB} = 2b + \sum_{i=3}^{n-1} \frac{b}{i} = \frac{b}{2} + bH(n-1).$$

That is, $b(1/2 - 1/n)$ units of bandwidth more than the original HB protocol. Given that the term in $1/n$ will quickly become negligible for large values of n, the extra bandwidth required by the CHB protocol is very close to $b/2$ or half the consumption rate of the video.

3.2 Quasi-Harmonic Broadcasting

CHB guarantees that all subsegments of a segment will arrive at the client before the client starts to consume the segment. If we were to allow the client to consume data from a segment while it is receiving data for the segment, then we can improve upon the protocol. We take this approach with *quasi-harmonic broadcasting*.

Quasi-harmonic broadcasting (QHB), like the other harmonic protocols, divides each video into n equal segments and broadcasts the first segment repeatedly on the first channel. But then each segment i, for $1 < i \leq n$, is broken into $im - 1$ *fragments* for some parameter m, and the client will receive m fragments from each channel per time slot. If we divide each time slot into m equally sized *subslots*, then the client will receive a single fragment during each subslot.

The key to QHB is in how the fragments are laid out. Consider some channel i. The last subslot of each time slot is used to transmit the first $i-1$ fragments of S_i in order. The rest of the subslots transmit the other $i(m-1)$ fragments such that the k^{th} subslot of slot j is used to transmit fragment $(ik + j - 1) \bmod i(m-1) + i$ (see Figure 2).

This mapping can be better understood using an example. Consider for instance the first three segments of a video and assume that $m = 4$. As seen on Figure 2, the first segment of the program occupies a single slot and will be broadcast unchanged.

The second segment consists of seven fragments that occupy two slots, each comprising four subslots. The first fragment of the segment, that is fragment $S_{2,1}$, is broadcast in the last subslots of both slots, while the six remaining subslots respectively contain fragments $S_{2,2}$ to $S_{2,7}$: fragment $S_{2,i}$ will occupy subslot $i/2 + 1$ of slot $i \bmod 2 + 1$. Thus the client will always have in memory two fragments of segment S_2 before it starts consuming the contents of that segment. One of these two fragments will necessarily be fragment $S_{2,1}$ while the other one could either be $S_{2,2}$ or $S_{2,3}$. We need only to worry about the case where $S_{2,2}$ is the missing fragment because it will have to be consumed before the client has finished downloading it from the server. Observe however that the client will start consuming $S_{2,2}$ after it has ended consuming fragment $S_{2,1}$, that is, after one seventh of the total slot transmission time has elapsed. During that time, it will have already downloaded four sevenths of $S_{2,2}$. The remaining three sevenths will arrive before they are actually needed given that they will be transmitted at one half of the normal rate.

The third segment is subdivided into eleven fragments that occupy three slots each comprising four subslots. Fragment $S_{3,1}$ is repeatedly broadcast in the last subslot of each odd slot while fragment $S_{3,2}$ is repeatedly broadcast in the last subslot of each odd slot. The ten remaining subslots contain fragments $S_{3,3}$ to $S_{3,12}$ and fragment $S_{3,i}$ will occupy subslot $i/3 + 1$ of slot $i \bmod 3 + 1$. Thus the client will always have in memory eight fragments of segment S_3 before it starts consuming the contents of that segment. Two of these two fragments will necessarily be fragments $S_{3,1}$ and $S_{3,2}$ while the three missing fragments could be any of the nine remaining fragments. The case when $S_{3,3}$ is the first missing segment is the only one that we have to consider here because it is the only one where a fragment will have to be consumed before the client has finished downloading it from the server. Observe however that the client will start consuming $S_{3,3}$ after it has ended

consuming both $S_{3,1}$ and $S_{3,2}$, that is after two elevenths of the total slot time has elapsed. During that time, it will have already downloaded eight elevenths of $S_{3,3}$. The remaining three elevenths will arrive before they are actually needed given that $S_{3,3}$ is transmitted at one third of the normal rate.

Understanding the fragment to segment mapping will also help us computing the total bandwidth afforded by the protocol. The essential difference between QHB and the original HB protocol is that QHB partitions each segment into $im-1$ fragments to be broadcast over im subslots while HB effectively allocates one fragment to each subslot. QHB then uses the remaining subslot to broadcast a redundant copy of the first fragment of each segment to guarantee that the consumer will always have in memory the first $i-1$ fragments of each segment before it starts consuming the contents of that segment.

As a result of this redundancy, each subslot of stream i will now have to broadcast $1/(im-1)$ of segment S_i instead of $1/im$. Since slightly more information will have to be transmitted in the same time interval as before, this will result in an increase of the required bandwidth. While the original harmonic broadcasting protocol transmitted segment S_i at a bandwidth $b_i = b/i$, our protocol requires a bandwidth b'_i equal to

$$b'_i = \begin{cases} b & \text{if } i = 1 \\ \frac{bm}{im-1} & \text{otherwise} \end{cases}.$$

The additional bandwidth per stream is given by:

$$\delta b_i = \frac{bm}{im-1} - \frac{b}{i} = \frac{b}{i(im-1)}$$

for all $i > 1$.

As a result, the bandwidth required by our protocol to transmit the i^{th} segment of a video can be made arbitrarily close to the i^{th} term of the harmonic series by increasing arbitrarily the number of subslots for each slot and, hence, the number of fragments per segment. We therefore decided to call our protocol the quasi-harmonic broadcasting protocol.

The total bandwidth B_{QHB} required by our quasi-harmonic broadcasting protocol is given by:

$$B_{QHB} = b + \sum_{i=2}^{n} \frac{bm}{im-1} = bH(n) + \sum_{i=2}^{n} \frac{b}{i(im-1)}.$$

The series

$$\sum_{i=2}^{n} \frac{b}{i(im-1)}$$

converges for all values of m and has a closed form:

$$b(1 - \gamma - \Psi(2 - \frac{1}{m}))$$

where Ψ is the digamma function and γ is Euler's constant. Although unwieldy, this expression provides an upper bound for the additional bandwidth required by the QHB protocol. Observing that $\Psi(2) = 1 - \gamma$, we can verify that this additional bandwidth becomes equal to zero when the number of subslots per slot m goes to infinity.

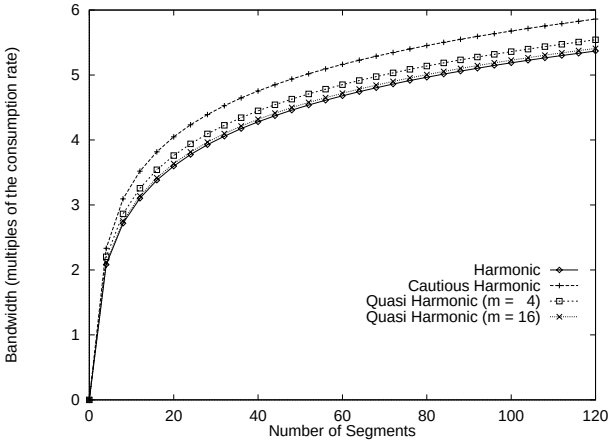

Figure 3: Bandwidth versus the number of segments for the three harmonic protocols.

Figure 3 displays the bandwidth requirements of harmonic broadcasting, cautious harmonic broadcasting and quasi-harmonic broadcasting for videos requiring between 1 and 120 segments and selected values of m. To eliminate the factor b representing the bandwidth of a standard full speed channel, all quantities on the y-axis are expressed in standard channels, that is, taking the bandwidth of a standard channel as unit of measurement. As one can see, the bandwidth requirements of HB and QHB become virtually indistinguishable as soon as there are more than, say, sixteen subslots per channel.

It may not be obvious at this point that QHB transmits all of the data to the client on time, so we will demonstrate it.

Claim 3 *Quasi-harmonic broadcasting delivers all video data on time.*

Proof:

Consider any channel i, and suppose the client started receiving data at time t_0.

The client will start to consume S_i at time $t_0 + (i-1)d$. That means the client will already have $(i-1)m$ of the $im-1$ fragments of S_i in its buffer. In particular, it will have the first $i-1$ fragments since they appear in every consecutive series of $i-1$ time slots.

The client must receive all of the first i fragments of S_i by time $t_0 + (i-1)d + d/m$. If the client has not yet already received the i^{th} fragment, then it will receive that fragment in the first subslot of the current time slot. That subslot will end at time $t_0 + (i-1)d + d/m$, and thus the client will receive the first i fragments on time.

For the next i fragments, the client will once again already have at least $i-1$ of the fragments

in its buffer, and it can receive the last fragment during the current subslot.

This argument repeats for the remainder of the segment, until the last subslot of the current time slot, during which the client only needs to receive $i-1$ fragments, all of which are already in its buffer.

□

3.3 Discussion

Our comparison between the three protocols is actually unfair to both cautious harmonic broadcasting and quasi-harmonic broadcasting because they are compared against a protocol that provides a much higher access time for the same segment duration.

As we showed earlier, the harmonic broadcasting protocol cannot deliver all frames on time unless the moment at which the clients start consuming the video is delayed by at least $(n-1)d/n$. Delaying the beginning of the video by almost one slot has however the major disadvantage of doubling the maximum access time and tripling the average access time as the customer will now have to wait between one and two slots before beginning to view the video. The same maximum access time and a lower average access time could have been obtained by using either one of our new protocols and doubling the segment size, thus reducing by a factor of two the number of segments required to broadcast a given video. This in turn would result in sufficient bandwidth savings to make our protocols more efficient than the original harmonic broadcasting protocol.

Consider first the case of the cautious harmonic broadcasting protocol. Doubling the segment size would result in the elimination of the last $n/2$ streams for a total bandwidth savings equal to:

$$\sum_{i=\frac{n}{2}+1}^{n} \frac{b}{i} = b(H(n) - H(\frac{n}{2}))$$

To show that the savings achieved would exceed the bandwidth overhead of the cautious harmonic broadcasting, let us observe that:

$$\sum_{i=\frac{n}{2}+1}^{n} \frac{b}{i} > \sum_{i=\frac{n}{2}+1}^{n} \frac{b}{n} = \frac{b}{2},$$

while the bandwidth overhead of the cautious broadcasting protocol was given by $b(1/2 - 2/n) < \frac{1}{2}$. Hence the cautious harmonic broadcasting protocol can provide the same maximum access time as the harmonic broadcasting protocol—and a lower mean access time—at a lower bandwidth cost.

The comparison is even more favorable to the quasi-harmonic protocol. Reducing by a factor of two the number of segments would result in a total bandwidth savings equal to:

$$\sum_{i=\frac{n}{2}+1}^{n} \frac{bm}{im-1} > \sum_{i=\frac{n}{2}+1}^{n} \frac{b}{i} > \frac{b}{2}$$

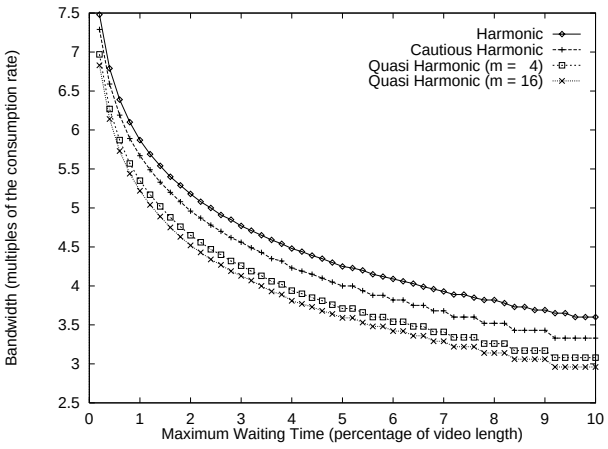

Figure 4: Bandwidth versus maximum access times for the three harmonic protocols.

while the bandwidth overhead of the quasi-harmonic protocol:

$$\sum_{i=\frac{n}{2}+1}^{n} \frac{b}{i(im-1)} > b(H(n) - H(\frac{n}{2}))$$

can be made arbitrarily small by selecting a sufficiently large value of m. As a result, the quasi-harmonic broadcasting protocol can provide the same maximum access time as the harmonic broadcasting protocol—and a lower mean access time—while using at least $b/2$ less bandwidth.

Figure 4 displays the bandwidth needed by each of the three harmonic broadcasting protocols to guarantee a given maximum access time. To eliminate the factor D representing the length of the video, the maximum waiting times on the x-axis are expressed as percentages of the video length. As in Figure 3, all quantities on the y-axis are expressed in standard channels, that is, taking the bandwidth of a standard channel as unit of measurement. As one can see, cautious harmonic broadcasting and quasi-harmonic broadcasting always require a lower bandwidth than harmonic broadcasting to guarantee the same maximum access time. The savings are quite significant, especially for quasi-harmonic broadcasting with 16 subslots, which requires between 0.64 and 0.68 standard channels less than harmonic broadcasting to guarantee the same maximum access time. In other words, quasi-harmonic broadcasting requires between 8 and 19 percent less bandwidth than harmonic broadcasting to provide the same quality of service.

We would have obtained even better figures if we had selected the mean access time rather than the maximum access time as our performance index. We believe however that the maximum access time is a better performance index because end-users tend to be mostly concerned by never having to wait too long for the services they request.

4 Related Work

The first protocol to break a video into segments and display those segments repeatedly on different channels was pyramid broadcasting [4]. With this protocol the n segments are not the same size; each segment i, for $1 < i \leq n$, is α^{i-1} times as large as the first segment, where α is a value close to the base of natural logarithms e. Pyramid broadcasting also groups the videos together on each channel, so channel i, for example, repeatedly shows the i^{th} segment of each video.

As with harmonic broadcasting, the client must wait for the first segment of the video it wants to consume. Once it starts receiving (and possibly consuming) data from one segment, it waits for the earliest opportunity to receive data from the next segment as well, and it receives the entire video in this pipelined fashion.

Given a bandwidth B times the consumption rate of the videos and M videos to broadcast, pyramid broadcasting calculates n to be $\lfloor B/(Me) \rfloor$ and it dedicates a bandwidth of B/n for each of the n channels. Since n grows with B, the maximum waiting time for the client improves exponentially with B. However, since the data is pipelined, the client will have to support a high bandwidth, and since the last (and largest) segment of the video will have to be stored in the client's buffer, the buffer will have to be large enough to hold nearly 80 percent of the video.

Permutation-based pyramid broadcasting [5] solves some of the problems of pyramid broadcasting in that it eliminates the pipelining of data to the client, and it removes the grouping of multiple videos per channel. But it also constrains n to be between two and seven, and so the client waiting times will not improve exponentially beyond a certain bandwidth.

Skyscraper broadcasting [6] limits the "width" of each segment, and so its segments, if stacked one on top of the other, would form a skyscraper shape rather than the pyramid shape of the segments from pyramid broadcasting. Skyscraper broadcasting also uses a more slowly-growing function to calculate its segment sizes, allowing for smaller buffer sizes on the clients, and it reduces the bandwidth clients must support.

Figure 5 shows the bandwidth versus client waiting times for the three protocols described in this section as well as for harmonic broadcasting and cautious harmonic broadcasting. We used the "unconstrained" version of permutation-based pyramid broadcasting [5] and skyscraper broadcasting with a maximum width of 52 [6].

5 Conclusions

Video broadcasting protocols can improve the efficiency of video on demand services by reducing the bandwidth required to transmit videos that are simultaneously watched by many viewers. One of the newest broadcasting protocols to be proposed, harmonic broadcasting, requires much less bandwidth than other broadcasting protocols to guarantee the same maximum access time. We found that the harmonic broadcasting protocol could not ensure on-time delivery of all frames of a given video unless the actual viewing of the video is delayed.

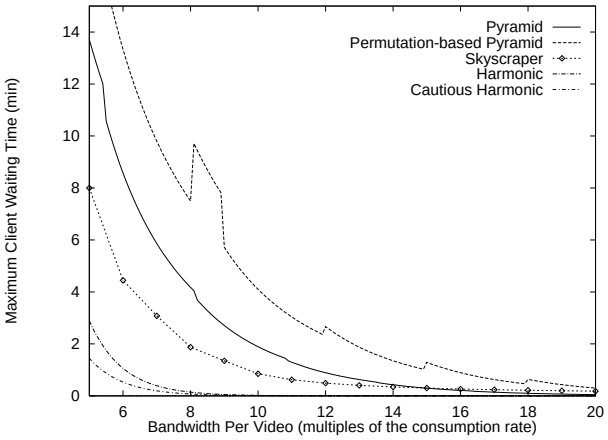

Figure 5: How harmonic broadcasting compares to other broadcasting protocols.

We have presented two broadcasting protocols that achieve nearly the same low bandwidth as harmonic broadcasting without imposing any additional delay on their end-users. As a result, both protocols require less bandwidth than harmonic broadcasting to achieve the same maximum access time. Our first protocol, cautious harmonic broadcasting, requires somewhat more bandwidth than our second protocol, quasi-harmonic broadcasting, but is also much simpler to implement. Selecting between them should be the result of evaluating a trade-off between bandwidth capacity and protocol complexity.

References

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

[2] A. Dan, D. Sitaram, and P. Shahabuddin. Scheduling policies for an on-demand video server with batching. *ACM Multimedia*, pp. 15–23, San Francisco, CA, October 1994.

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

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

[5] C. C. Aggarwal, J. L. Wolf, and P. S. Yu. A permutation-based pyramid broadcasting scheme for video-on-demand systems. *Proc. International Conference on Multimedia Computing and Systems*, pp. 118–26, Hiroshima, Japan, June 1996.

[6] K. A. Hua and S. Sheu. Skyscraper Broadcasting: a new broadcasting scheme for metropolitan video-on-demand systems. *SIGCOMM 97*, pp. 89–100, Cannes, France, September 1997.

[7] L.-S. Juhn and L.-M. Tseng. Harmonic broadcasting for video-on-demand service. *IEEE Transasctions on Broadcasting*, 43(3):268–71, September 1997.

[8] L.-S. Juhn. *Private correspondence*, October 1997.