A Simple Low-Bandwidth Broadcasting Protocol for Video-on-Demand

Jehan-François Pâris
Department of Computer Science
University of Houston
Houston, TX 77204-3475

paris@cs.uh.edu

Abstract

Broadcasting protocols can reduce the cost of video-on-demand services by using much less bandwidth to transmit videos that are simultaneously watched by many viewers. Unfortunately, the most efficient broadcasting protocols are also the most difficult to implement because they allocate a multitude of very low bandwidth streams to each video.

We present a protocol that uses between three and seven high-bandwidth streams per video and never requires more than five percent more bandwidth than any other protocol to guarantee a given average waiting time.

Keywords: video-on-demand, broadcasting protocols, pyramid broadcasting, harmonic broadcasting, pagoda broadcasting.

I. INTRODUCTION

Ten years ago, most multimedia pundits were predicting that video-on-demand (VOD) [Wong88] would radically change our home entertainment habits. What happened was quite different: none of the companies that invested in VOD were able to come up with a single successful commercial system. The best explanation for this lack of success is that video-on-demand is still too costly to compete with either video rentals or pay-per-view television. The main culprit for this situation is the high bandwidth requirements of VOD service. These requirements result in high communication costs and, more importantly, demand video servers that can handle unusually high I/O traffic rates.

Broadcasting is one of several techniques that aim at reducing these bandwidth requirements [Wong88]. It only applies to videos that are likely to be simultaneously watched by many viewers. Rather than responding to individual requests, broadcasting uses a proactive approach and executes frequent rebroadcasts of these "hot" videos. The bandwidth savings can be considerable since the top ten or twenty most popular videos are likely to account for over forty percent of the total demand [Dan94, Dan96].

The past three years have seen the development of many efficient broadcasting protocols starting with Viswanathan and Imielinski's *Pyramid Broadcasting* protocol [Visw96]. All these protocols assume that the client STB has enough local storage to store up to 55 percent of the video. As a result, the later parts of the video can be broadcast either less frequently or at a lower bandwidth.

If we attempt to rank these broadcasting protocols according to the bandwidth they require to guarantee a given average delay, we quickly notice that protocols that allocate a multitude of very low bandwidth streams to each video require much less bandwidth than protocols using a few high-bandwidth streams shared by all videos. For instance, the *Quasi-Harmonic Broadcasting* protocol (QHB) [Pâr98] requires slightly less than five times the video consumption rate to guarantee an average viewing delay of 45 seconds for a two-hour video. Thus the equivalent of 100 conventional video streams would be enough to service all the customers wanting to watch any of the top 20 videos.

To achieve this result, the QHB protocol needs 82 data streams to each video. A server broadcasting the top twenty videos would then have to manage 1640 parallel data streams whose bandwidths would vary between 1 and 1/82 times the video consumption rate. Managing such a large number of independent data streams is likely to be a daunting task. The only existing alternative was to use a protocol partitioning the videos to be broadcast into segments of increasing lengths and using many fewer data streams. Unfortunately, these protocols also require much more bandwidth to achieve a given maximum delay.

We presented a few months ago a hybrid protocol that partitioned each video into a large number of small segments and used time-division multiplexing to ensure that each segment was broadcast at the appropriate bandwidth [Pâr99a]. While our *Pagoda Broadcasting* protocol requires much less total bandwidth than other protocols that use a small number of high-bandwidth streams per video, it still needs significantly more bandwidth than the QHB protocol. The *New Pagoda Broadcasting* protocol we are presenting in this paper uses a more sophisticated segment-to-stream mapping to further reduce its bandwidth requirements. As a result, its bandwidth requirements fall within 5 percent of the bandwidth requirements of the QHB protocol.

II. VIDEO BROADCASTING PROTOCOLS

The simplest video broadcasting protocol is *staggered broadcasting* [Dan96]. It requires a fairly large number of channels per video to achieve a reasonable waiting time. Consider, for instance, a video that lasts two hours, which happens to be close to the average duration of a feature movie. Guaranteeing a maximum waiting time of 10 minutes would require starting a new instance of the video every 10 minutes and a total of 12 channels.

Many more efficient protocols have been proposed. All these protocols divide each video into *segments* that are simultaneously broadcast on separate data streams. One of these streams transmits nothing but the first segment of the video. The other streams transmit the remaining segments at lower bandwidths. When customers want to watch a video, they first wait for the beginning of the first segment on the first stream. While they are watching that segment, their set-top box (STB) starts to download enough data from the other streams so that it will be able to play each segment of the video in turn.

All these protocols can be subdivided into two groups. Protocols in the first group are all based on Viswanathan and Imielinski's *Pyramid Broadcasting* protocol [Vis96]. They include Aggarwal, Wolf and Yu's *Permutation-Based Pyramid Broadcasting* protocol [Agg96] and Hua and Sheu's *Skyscraper Broadcasting* protocol [Hua97]. These three protocols subdivide each video *j* to be broadcast into *K* segments S_i^M of increasing sizes. The entire bandwidth dedicated to the *M* videos to be broadcast is divided into *K* logical streams of equal bandwidth. Each stream is allocated a set of segments to broadcast so that stream *i* will broadcast segments S_i^1 to S_i^M in turn.

While these protocols require much less bandwidth than staggered broadcasting to guarantee the same maximum waiting time, they cannot match the performance of the protocols based on the harmonic broadcasting protocol [Juh97, Pâr98], which we will discuss in more detail.

Harmonic Broadcasting (HB) divides a video into *n* equally sized segments. Each segment S_i, for $1 \le i \le n$, is broadcast repeatedly on its own data stream with a bandwidth b/i, where *b* is the consumption rate of the video. When customers order a video, their STB waits for the start of an instance of S_1 and then begins receiving data from every stream for the video.

The total bandwidth required to broadcast the *n* segments is thus given by

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

where *H*(*n*) is the harmonic number of *n*.

Let *d* represent the amount of time it takes for a client to consume a single segment of the video. Since the first segment is broadcast with that periodicity, *d* is given by

$$d = S_1/b = D/n$$

and is also the maximum amount of time a client should wait before viewing its request.

HB has one major flaw: It does not always deliver all data on time unless the client always waits an extra slot of time before consuming data. Hence the true delay is $2d$ instead of *d* [Pâr98]. Several variants of HB do not impose this extra waiting time, among which *Cautious Harmonic Broadcasting* (CHB) and *Quasi-Harmonic Broadcasting* (QHB) [Pâr98]. CHB guarantees on time delivery of all segments by slightly increasing the bandwidths at which it broadcasts segments S_3 to S_n. QHB uses a more complex scheme than CHB but requires almost no extra bandwidth.

Even though the total bandwidth requirements for HB and its variants are quite small, the multitude of streams these protocols involve complicates the task of the STBs and the servers. *Pagoda Broadcasting* (PB) [Pâr99a] avoids this problem by broadcasting less frequently later segments instead of lowering their bandwidth. For example, a segment mapping using three streams would be:

First Stream	S_1	S_1	S_1	S_1	S_1	S_1
Second Stream	S_2	S_4	S_2	S_5	S_2	S_4
Third Stream	S_3	S_6	S_8	S_3	S_7	S_9

Thus the client would have to wait at most 14 minutes for a two-hour video. This is 3 minutes longer than what QHB would allow, but servers and clients have to manage much fewer streams. More generally, PB can broadcast

$$4(5^{k-1}) - 1$$

distinct segments with 2*k* streams and

$$2(5^{k}) - 1$$

segments with 2*k*+1 streams. The maximum waiting time for a video of duration *D* broadcast over *n* streams is thus given by

$$d = D / [2 \times 5^{((n-1)/2)}]$$

for *n* odd, and

$$d = D / [4 \times 5^{((n-2)/2)}]$$

for *n* even.

Figure 1 compares the bandwidth requirements of Pagoda Broadcasting with those of Pyramid Broadcasting [Visw97], Skyscraper Broadcasting with a maximum width of 52 [Hua97] and Quasi-Harmonic Broadcasting with more than 16 subsegments. To eliminate the factor *D* representing the duration of the video, the maximum waiting times on the x-axis are expressed as percentages of the video lengths. All bandwidths are expressed in multiples of the video consumption rate *b*. As one can see, Pagoda Broadcasting performs much better than Pyramid Broadcasting and Skyscraper Broadcasting but is outperformed by Quasi-Harmonic Broadcasting.

III. The New Pagoda Broadcasting Protocol

Two factors explain why Pagoda Broadcasting (PB) outperforms Pyramid Broadcasting and Skyscraper Broadcasting. First, PB uses fixed-size segments instead of the increasing-size segments favored by the two other protocols. Second, it allocates its data streams in pairs consisting of an even and an odd stream so that segments that do not fit in stream 2*k* can be moved to stream 2*k*+1. The result is that the frequency at which the video data are broadcast will not significantly exceed the minimum frequency at which the data need to be broadcast to guarantee on time delivery of all data.

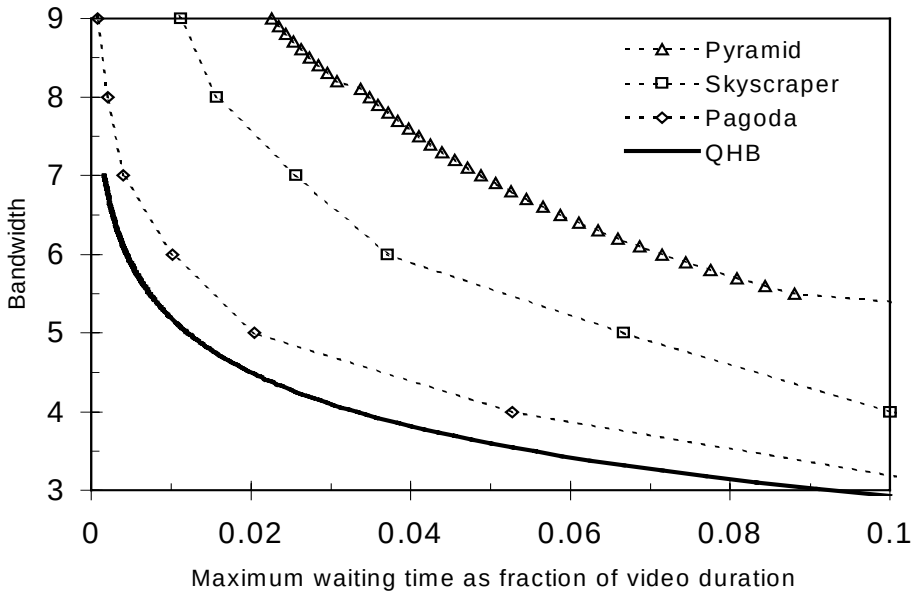

Figure 1: Compared bandwidth requirements of Pagoda Broadcasting, Pyramid Broadcasting, Skyscraper Broadcasting and Quasi-Harmonic Broadcasting.

Several further optimizations are still possible. First we should generalize the so-called *rectangular matrix* segment-to-slot mapping that was introduced to increase the number of segments that could fit in the last stream of a PB broadcast with an even number of streams.

Let us consider the case of a video being broadcast using four streams. Table 1 contains the segment to stream mapping obtained with the original PB protocol whith stream 4 containing segments S_{10} to S_{19} endlessly repeated once every 10 slots.

Define a slot as the time interval it takes to transmit one segment. Let us now consider sets of five consecutive slots and assume they constitute rows of a large matrix representing the segment-to-slot mapping. The current segment-to-slot mapping for stream 4 could be represented by:

S_{10}	S_{11}	S_{12}	S_{13}	S_{14}
S_{15}	S_{16}	S_{17}	S_{18}	S_{19}

Rather than allocating the slots to the segments using the row major order, we could use the column major order:

S_{10}	S_{12}	S_{14}	S_{16}	S_{19}
S_{11}	S_{13}	S_{15}	S_{17}	S_{20}
...	...	...	S_{18}	S_{21}

Under the new mapping, only segments S_{10} to S_{15} continue to be broadcast once every 10 slots. Segments S_{16} to S_{19} are now broadcast once every 15 slots, which saves enough bandwidth to add two additional segments, namely, segments S_{20} and S_{21} and increase the number of segments that can be broadcast from 19 to 21.

This rectangular matrix allocation method will be at the base of our *New Pagoda Broadcasting* (NPB) protocol. We will use it for *all data streams* rather than just for the last one. In addition, the mapping process will consider all available slots in all data streams rather than looking at pairs of streams. The whole process will be much less formal than that of the original PB protocol but will lead to segment-to-slot mappings that will be much closer to the ideal mapping where each segment S_i would be broadcast exactly once every i slots.

Consider first the case of a video being broadcast using four streams. As always stream 1 will continuously rebroadcast segment S_1. In our matrix representation, we will denote this as

S_1

In the original PB protocol, stream 2 was alternating between transmitting segment S_2 once every two slots and either segment S_4 or S_5 once every 4 slots. In the NPB protocol, S_5 is moved to stream 4 and replaced by the two alternating segments S_8 and S_9. The segment to stream mapping of stream 2 is thus

S_2	S_4
...	S_8
...	S_4
...	S_9

Table 1: Segment to stream mapping obtained with the original PB protocol for four data streams

First Stream	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1
Second Stream	S_2	S_4	S_2	S_5	S_2	S_4	S_2	S_5	S_2	S_4
Third Stream	S_3	S_6	S_8	S_3	S_7	S_9	S_3	S_6	S_8	S_3
Fourth Stream	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}	S_{17}	S_{18}	S_{19}

This segment-to-slot mapping is more economical than the first one because we use 1/4 of the stream bandwidth to transmit two segments requiring respectively 1/8 and 1/9 of that bandwidth rather than a single segment requiring 1/5 of it.

Let us now consider the segment-to-slot mapping of stream 3:

S_3	S_6	S_{12}
...	S_7	S_{13}
...	...	S_{14}
...	...	S_{25}
...	...	S_{12}
...	...	S_{13}
...	...	S_{14}
...	...	S_{26}

Note that segments S_{10} and S_{11} are not mapped into stream 3: since they need to be broadcast at periodicities respectively equal to 1/(10*d*) and (1/11*d*), they will fit better in stream 4, which already contains segment S_5. Thanks to our rectangular matrix mapping we are able to broadcast segment S_3 once every 3 slots, segment S_6 and S_7 once every 6 slots, segment S_{12} to S_{14} once every 12 slots as well as segments S_{25} and S_{26} once every 24 slots.

Stream 4 will contain all segments that did not fit in any of the first three streams:

S_5	S_{10}	S_{15}	S_{18}	S_{21}
...	S_{11}	to	to	to
...	...	S_{17}	S_{20}	S_{24}

Segment S_5 will be broadcast once every 5 slots, segment S_{10} and S_{11} once every 10 slots, segment S_{15} to S_{20} once every 15 slots and segments S_{21} to S_{24} once every 20 slots. Our NPB protocol will thus be able to transmit 26 segments with 4 data streams, that is 23.8 percent more segments than the improved PB protocol and 36.8 percent more than the original PB protocol. This means that allocating to a video a bandwidth equal to four times the video consumption rate will guarantee that the viewing delay will never exceed 4 minutes 37 seconds for a 2 hour video.

We will not detail the segment-to-slot mappings for 5, 6 and 7 data streams. For instance, in the segment-to-slot mapping for 5 streams:

a) stream 1 will continuously rebroadcast segment S_1;

b) stream 2 will broadcast segment S_2 once every 2 slots, segment S_4 once every 4 slots, and segments S_8 and S_9 once every 8 slots;

c) stream 3 will broadcast segment S_3 once every 3 slots, segment S_6 once every 6 slots, segments S_{12} and S_{13} once every 12 slots, and segments S_{15} to S_{19} once every 15 slots;

d) stream 4 will broadcast segment S_5 once every 5 slots, segment S_{10} and S_{11} once every 10 slots, segments S_{20} to S_{27} once every 20 slots, and segments S_{30} to S_{35} once every 30 slots;

e) stream 5 will broadcast segment S_7 once every 7 slots, segment S_{14} once every 14 slots, segments S_{28} and S_{29} once every 28 slots, segments S_{36} to S_{45} once every 35 slots, segments S_{46} to S_{51} once every 42 slots, segments S_{52} to S_{58} once every 49 slots, and segments S_{59} to S_{66} once every 30 slots.

Allocating to a video a bandwidth equal to five times the video consumption rate will guarantee that the viewing delay will never exceed 110 seconds for a 2-hour video. This is 25.75 percent less than with the PB protocol.

Even smaller viewing delays can be achieved by increasing the total bandwidth allocated to each. For instance, bringing the total bandwidth to six times the video consumption rate would allow partitioning each video into 172 segments and guaranteeing a maximum viewing delay of 42 seconds for a two-hour video. If this were not small enough, allocating seven times the video consumption rate would allow to partition each video into 442 segments. This would guarantee that no customer would have to wait more than 17 seconds for a two-hour video and provide an average delay of 8.14 seconds.

Figure 2 compares the bandwidth requirements of the New Pagoda Broadcasting protocol with those of the original Pagoda Broadcasting protocol, Cautious Harmonic Broadcasting (CHB) and Quasi-Harmonic Broadcasting with more than 16 subsegments (QHB). As before, the maximum waiting times on the x-axis are expressed as percentages of the video lengths and all bandwidths are expressed in multiples of the video consumption rate *b*.

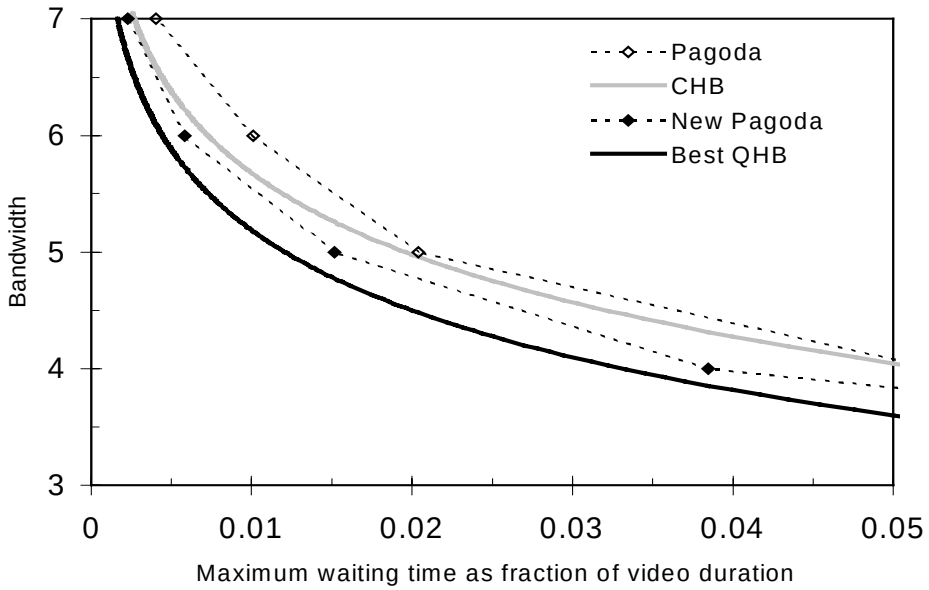

Figure 2: Compared bandwidth requirements of the New Pagoda Broadcasting protocol, Pagoda Broadcasting, Cautious Harmonic Broadcasting and Quasi-Harmonic Broadcasting.

As one can see, NPB performs much better than the original Pagoda Broadcasting protocol and the Cautious Harmonic Broadcasting protocol. Consider for instance the maximum waiting times guaranteed by the three protocols when the total bandwidth allocated to each video varies between four and seven times its consumption rate. This constitutes a reasonable range of waiting delays varying between a maximum of 4 minutes 37 seconds and a minimum of 17 seconds. In that ranges, the maximum waiting delays guaranteed by NPB are on the average 29 percent lower than those achieved by the original Pagoda Broadcasting protocol and 21 percent lower than those achieved by the CHB protocol.

One may wonder at this time if even lower maximum delays could not be achieved by even more complex segment-to-slot mappings. Let us just observe that any segment-to-slot mapping would have to ensure that each segment S_i is broadcast at least once every i slots. Hence, the maximum number n_{max} of segments that could be transmitted using m data streams of bandwidth b will have to satisfy the inequality

$$\sum_{i=1}^{n_{max}} \frac{b}{i} \leq mb$$

In other words, no segment to block mapping would allow any variant of our protocol to use less bandwidth than the QHB protocol to guarantee a given maximum viewing delay. Going back to Figure 2, we can see that the bandwidth required by our NPB protocol to guarantee a maximum viewing delay remains within five percent of that of the QHB protocol when the total bandwidth allocated to each video varies between four and seven times its consumption rate. While better variants of our protocol still remain possible, none of them will ever provide any significant performance improvement.

IV. Open Problems

Two important problems are not successfully resolved by existing video broadcasting protocols and still remain open for further investigation. These are how to provide the VOD users with VCR-like controls and how to handle compressed video signal.

A. Providing VCR-like controls

A common limitation of nearly all VOD broadcasting protocols is that they require the viewers to watch each video in sequence as in a theater. They do not provide controls allowing the viewers to move fast forward or backward as when watching a videocassette on a VCR. The only exception to this rule is staggered broadcasting, which can allow viewers to jump backward and forward but only from one data stream to another.

Implementing "fast reverse," that is, the equivalent of a VCR rewind control requires additional storage space on the STB disk drive to keep the portions of the video that have been already viewed rather than discarding them. The evolution of technology favors this solution as disk drive capacities have been doubling every year for the last three years. Implementing fast forward is still to be resolved as it would allow the viewers to access any part of the video in a nearly random fashion and destroy all the assumptions on which efficient VOD broadcasting protocols are built.

B. Handling compressed video

Nearly all existing video broadcasting protocols assume that the videos will have a fixed bandwidth corresponding to a fixed video consumption rate. This assumption is not correct because the server will broadcast compressed videos whose bandwidth requirements depend on the rate at which the images being displayed change [Gar94, Ber95]. For instance, daytime action scenes and cartoons will require more band-

width than slower moving scenes and night scenes. To ensure jitter-free delivery of video in a system allocating a fixed bandwidth to each video, we would thus have to set the video broadcasting bandwidth to the maximum bit rate required by the fastest moments of the fastest paced scenes of the video. As a result, a significant fraction of the bandwidth could remain unused most of the time.

The only video broadcasting protocol that avoids this drawback is the Variable Bandwidth Harmonic Broadcasting protocol [Pâr99b]. As it is based on the Cautious Harmonic Broadcasting protocol, it requires a fairly large number of low bandwidth streams to achieve a reasonable maximum delay. Adapting the NPB protocol to handle compressed video in an efficient manner is not a trivial task because the segment transmission times would not be equal anymore to their viewing times.

V. Conclusions

One of the most promising approaches to reduce the cost of video-on-demand services is to broadcast continuously the most frequently requested videos. Unfortunately the best existing broadcasting protocols all use a very large number of very low bandwidth streams for each video.

We have presented a broadcasting protocol that requires at most six percent more bandwidth than the best harmonic broadcasting protocol while only using between three and seven data streams per video. Our *New Pagoda Broadcasting* protocol partitions each video into fixed-size segments whose duration is equal to the maximum waiting time. It maps these segments into data streams of equal bandwidth and uses time-division multiplexing to ensure that successive segments of a given video are broadcast at the proper decreasing frequencies. It offers all the advantages of the best harmonic broadcasting protocols but none of their drawbacks.

References

[Agg96] C. C. Aggarwal, J. L. Wolf, and P. S. Yu. A permutation-based pyramid broadcasting scheme for video-on-demand systems. *Proc. Int Conference on Multimedia Computing and Systems*, pages 118–126, June 1996.

[Ber95] J. Beran, R. Sherman, M. Taqqu, and W. Willinger. Long-range dependence in variable bit-rate video traffic. *IEEE Trans. on Communications*, 43: 1566–1579, 1995.

[Dan94] A. Dan, D. Sitaram, and P. Shahabuddin. Scheduling policies for an on-demand video server with batching. *Proc. 1994 ACM Multimedia Conference*, pages 15–23, Oct. 1994.

[Dan96] A. Dan, D. Sitaram, and P. Shahabuddin. Dynamic batching policies for an on-demand video server. *Multimedia Systems*, 4(3):112–121, June 1996.

[Gar94] M. Garrett and W. Willinger. Analysis, modeling and generation of self-similar VBR video traffic. *Proc. ACM SIGCOMM '94 Conference*, Aug. 1994, pages 269–280.

[Hua97] K. A. Hua and S. Sheu. Skyscraper broadcasting: a new broadcasting scheme for metropolitan video-on-demand systems. *Proc. ACM SIGCOMM '97 Conference*, Sept. 1997, pages 89–100.

[Juh97] L. Juhn and L. Tseng. Harmonic broadcasting for video-on-demand service. *IEEE Transactions on Broadcasting*, 43(3):268–271, Sept. 1997.

[Pâr98] 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, July 1998.

[Pâr99a] J.-F. Pâris, S. W. Carter and D. D. E. Long, A Hybrid broadcasting protocol for video on demand. *Proc. 1999 Multimedia Computing and Networking Conference*, San Jose, CA, January 1999, pp. 317–326.

[Pâr99b] J.-F. Pâris, A Broadcasting protocol for compressed video. *Proc. Euromedia '99*, Munich, Germany, April 1999, pp. 78–84.

[Visw96] S. Viswanathan and T. Imielinski. Metropolitan area video-on-demand service using pyramid broadcasting. *Multimedia Systems*, 4(4):197–208, Aug. 1996.

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