Combining Pay-Per-View and Video-on-Demand Services

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[1]

Department of Computer Science
Jack Baskin School of Engineering
University of California
Santa Cruz, CA 95064

{carter, darrell}@cse.ucsc.edu

Abstract

Most efforts aimed at reducing the costs of video-on-demand services have focussed on reducing the cost of distributing the top ten to twenty videos by broadcasting them in a periodic fashion rather than waiting for individual requests. Unfortunately nearly all existing VOD broadcasting protocols require client set-top boxes (STB) to include enough local storage to store up to 55 percent of each video being viewed. Here we present a novel VOD broadcasting protocol that does not make that demand. Our Dual Broadcasting protocol can accommodate clients who do not have any storage device in their STB while providing a much lower maximum waiting time to customers whose STB includes a disk drive. We also discuss two possible extensions to this new protocol. One of them is aimed at reducing the bandwidth requirements of the protocol while the other extends the functionality of the VOD service by providing reverse and fast forward controls.

Keywords: video-on-demand, pay per view, broadcasting protocols, harmonic broadcasting, pagoda broadcasting.

1. Introduction

After more than ten years of investigations, video-on-demand (VOD) [9] has yet to succeed on the marketplace. This situation has a simple explanation: VOD services are expensive to provide since their customers can select both the videos they want to watch and the time at which they want to watch them. As a result, VOD cannot compete on a price basis with cheaper, more established alternatives such as videocassette rentals and pay-per-view television (PPV).

Most efforts aimed at reducing the cost of VOD services have focused on reducing the bandwidth necessary for distributing the top ten or twenty most popular videos. The savings that can be achieved are considerable since these so-called “hot” videos are likely to be responsible for over forty percent of the total demand [2, 3]. One of the most promising approaches is to schedule repeated broadcasts of these “hot” videos rather than waiting for individual requests. This technique is known as *video broadcasting* [9].

The simplest broadcasting protocol is *staggered broadcasting*: it consists of retransmitting the same video on several distinct channels at equal time intervals. The major disadvantage of this approach is the number of channels per video required to achieve a reasonable waiting time. Several more efficient protocols have also been proposed [1, 4–8]. Some of these protocols require fewer than four channels to guarantee a maximum waiting time of five minutes for a two-hour video.

Even so, broadcasting protocols have two limitations. First, customers who want to watch a video may have to wait, say, between two and fifteen minutes for the next scheduled broadcast of the video. Second, the most efficient broadcasting protocols all require set-top boxes capable of storing as much as 55 percent of each video being watched. With the current state of the storage technology, this implies that the STB must have a local disk.

Requiring a hard drive in each STB will significantly increase their cost. Little relief can be expected in the near future from the current evolution of disk technology since the base price for the cheapest hard drives has remained very stable during the last three years. This raises the issue of who should pay for these upgraded STBs. Selling them at their true cost is likely to diminish the initial customer base for VOD services. On the other hand, distributing STB’s at a subsidized

[1] This research was supported by the Office of Naval Research under Grant N00014-92-J-1807, and by the National Science Foundation under Grant CCR-970434.

price would increase the capital costs of providing VOD services. One could almost say that the net effect of using an efficient video broadcasting protocol is a mere transfer of capital costs from the VOD servers to the customer STB.

We propose a solution to this dilemma, namely, a video broadcasting protocol that requires much lower bandwidth than staggered broadcasting but can nevertheless accommodate diskless STBs. To achieve these two contradictory objectives, our protocol will work in the following manner. Each video to be distributed will be allocated a fixed number of broadcasting streams. Some of these streams will continuously rebroadcast the video in staggered fashion. The remaining streams will rebroadcast more frequently the first minutes of the video for the sole use of the customers whose STB includes a disk drive. Our protocol will thus provide at the same time two different services, namely, a PPV service to the customers that have a diskless STB and a VOD service to the customers with a disk drive in their STB.

The major motivation for our new *Dual Broadcasting* protocol is that it will be much cheaper to use a single protocol to provide both enhanced PPV and VOD services than to use distinct protocols for the two services. The additional flexibility gained by combining the two services will also be demonstrated by the two extensions we will propose. Our first extension is aimed at reducing the bandwidth requirements of the protocol. As we will see, the continuous retransmission of the first minutes of each video will consume an inordinate amount of bandwidth allocated to the VOD service. We could save this bandwidth by requiring the STBs of the VOD customers to snoop on the PPV streams providing the enhanced PPV service and to store the first few minutes of each video. A second extension extends the functionality of the VOD service by providing rewind and fast forward controls.

The remainder of the paper is organized as follows. Section 2 discusses some relevant video broadcasting protocols. Section 3 introduces our new protocol and compares its bandwidth requirements to those of the other broadcasting protocols. Section 4 discusses two possible extensions. Finally, Section 5 has our conclusions.

2. Video Broadcasting Protocols

The simplest video broadcasting protocol is *staggered broadcasting* [3]. 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 for 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 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 [8]. They include Aggarwal, Wolf and Yu's *Permutation-Based Pyramid Broadcasting* protocol [1] and Hua and Sheu's *Skyscraper Broadcasting* protocol [4]. These three protocols subdivide each video j to be broadcast into K segments S_i^j 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 [5, 6], 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 n^{th} harmonic number.

Table 1: Segment to slot mapping for Dual Broadcasting with two VOD channels

Current PPV Stream	S_1	S_2	S_3	S_4	S_5	S_6	S_7
First VOD Stream	S_3	S_1	S_1	S_1	S_1	S_1	S_1
Second VOD Stream	S_4	S_5	S_6	S_2	S_2	S_3	S_2

Since the first segment is broadcast at a bandwidth equal to the video consumption rate *b*, the maximum amount of time customers will have to wait before viewing a video is given by the duration *d* of that first segment.

HB offers two major limitations. First, 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 two slots instead of one [6].

Several variants of HB do not impose the extra waiting time [6]. *Cautious Harmonic Broadcasting* (CHB) broadcasts the video in a similar fashion as HB. The first stream broadcasts S_1 repeatedly as HB did, but the second stream alternates between broadcasting S_2 and S_3 at bandwidth *b*. Then the remaining *n*-3 streams broadcast segments S_4 to S_n in such a way that stream *i* will transmit segment S_{i+1} at bandwidth b/i. Hence segments S_3 to S_n are transmitted at a higher bandwidth than in the original HB protocol. *Quasi-harmonic Broadcasting* (QHB) uses a more complex scheme but requires almost no extra bandwidth.

A second limitation of HB and its variants is that they require a fairly large number of independent data streams. Even though their total bandwidth requirements are quite small, the mere number of these streams complicates the task of the STBs and the servers. *Pagoda Broadcasting* [7] avoids this problem by broadcasting less frequently later segments instead of lowering their bandwidth. For example, a segment mapping using three streams would be:

S_1	S_1	S_1	S_1	S_1	S_1
S_2	S_4	S_2	S_5	S_2	S_4
S_3	S_6	S_8	S_3	S_7	S_9

Since the video can be partitioned into 9 segments, the client would have to wait at most $120/9 = 14$ minutes for a two-hour video. This is a few minutes longer than what QHB would allow, but it requires the client to manage fewer streams. More generally, Pagoda Broadcasting can broadcast $4(5^{k-1}) - 1$ distinct segments with $2k$ streams and $2(5^k) - 1$ segments with $2k+1$ streams. The maximum waiting time for a video of duration *D* broadcast over *n* streams is thus given by

$$d = D / [2\times5^{((n-1)/2)}]$$

for *n* odd, and

$$d = D / [4\times5^{((n-2)/2)}]$$

for *n* even.

3. The Dual Broadcasting Protocol

Developing a protocol that can handle clients with and without local storage is trivial; simply dividing the VOD server's available bandwidth between staggered broadcasting and one of the other protocols from the previous section would work. The problem is that naively combining two protocols in this way means the protocols might duplicate work and thus waste bandwidth. The Dual Broadcasting protocol allows clients with local storage to use bandwidth allocated for users without local storage, and so no such duplication takes place.

The Dual Broadcasting protocol works as follows. For each video to be broadcast *k* data streams are set aside for clients without local storage (PPV streams) and *l* streams are set aside for clients with local storage (VOD streams). The PPV streams use staggered broadcasting; that is, if the duration of the video is *D*, then a new instance of the video is started every D/k minutes and the maximum delay for a client using these streams is $d_{PPV} = D/k$.

Table 2: Segment to slot mapping for Dual Broadcasting with three VOD channels.
The dash (“–”) denotes an empty slot.

Current PPV Stream	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}	S_{17}
1st VOD Stream	S_2	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1
2nd VOD Stream	S_3	S_4	S_7	S_2	S_{14}	S_2	S_8	S_2	S_{16}	S_2	S_6	S_2	S_7	S_2	S_3	S_2	S_6
3rd VOD Stream	S_{10}	S_5	S_{11}	S_{12}	S_{13}	S_3	S_{15}	S_4	S_3	S_5	S_4	S_3	–	S_5	S_4	S_8	S_9

For clients with local storage, the Dual Broadcasting protocol uses a strategy similar to Pagoda Broadcasting, but since the clients can always receive at least $D - d_{PPV}$ minutes of the video from a PPV stream, only the first d_{PPV} minutes of the video needs to be broken up into segments. If there are n_{VOD} segments, S_1 to $S_{n_{VOD}}$, then the maximum delay for a client using the l VOD streams is

$$d_{VOD} = \frac{d_{PPV}}{n_{VOD}} = \frac{D}{kn_{VOD}}$$

If the same amount of bandwidth were used without considering the PPV streams, the delay would be D / n_{VOD} or k times greater than d_{VOD}.

Define a slot as the time interval it takes to transmit one segment. Observe that each PPV stream broadcasts segments S_1 to $S_{n_{VOD}}$ in order every D time units. Collectively, the k PPV segments repeat these segments every d_{PPV} time units, that is, every n_{VOD} slots. The Dual Broadcasting protocol will take advantage of these broadcasts to reduce the number of times each of these segments will be repeated in the VOD streams. Hence we will use these segments when mapping segments to slots in the VOD streams. Consider, for example, the very simple case when there is only one VOD stream ($l = 1$). Then, for each block of d_{PPV} minutes, we have

Current PPV Stream	S_1	S_2	S_3
VOD Stream	S_2	S_1	S_1

and $n_{VOD} = 3$. Segment S_1 is repeated in every slot, segment S_2 is repeated at least every two slots and segment S_3 is repeated every three slots. Adding to the k PPV streams a single VOD stream will thus reduce the maximum waiting time for the VOD clients d_{VOD} to one third of the maximum waiting time for the PPV clients. This is much better than what we could have achieved using a separate VOD broadcasting protocol because no VOD broadcasting protocol with one stream can achieve a maximum delay less than the duration of the video D without using more than b units of bandwidth.

The same approach can be followed with more than one VOD stream. Two VOD streams would allow us to partition the d_{PPV} first minutes of the video into 7 segments using the segment to stream mapping of Table 1. This reduces the maximum waiting time for the VOD clients d_{VOD} to one seventh of the maximum waiting time for the PPV clients. Here too we can observe that segment S_i for $i = 1, \ldots, 7$ is repeated at least once every i slots.

Adding a third VOD stream would allow us to partition the d_{PPV} first minutes of the video into 17 segments using the segment to stream mapping represented in Table 2

One may wonder at this stage what is the maximum number of segments that can be packed in the PPV streams and, say, l VOD streams. To derive an upper bound for this quantity, let us observe that the PPV streams and l VOD streams give us l+1 streams to map our n_{VOD} segments into. As we observed before, each segment S_i for $i = 1, \ldots, n_{VOD}$ must be repeated at least once every i slots. Since we want the mapping to repeat itself without alterations, this means that each group of n_{VOD} consecutive slots should contain at least

$$\left\lceil \frac{n_{VOD}}{i} \right\rceil$$

copies of segment S_i. Since there is a total of $(l+1)n_{VOD}$ slots available, each segment to stream mapping must satisfy the inequality

$$\sum_{i=1}^{n_{VOD}} \left\lceil \frac{n_{VOD}}{i} \right\rceil \le (l+1)n_{VOD}.$$

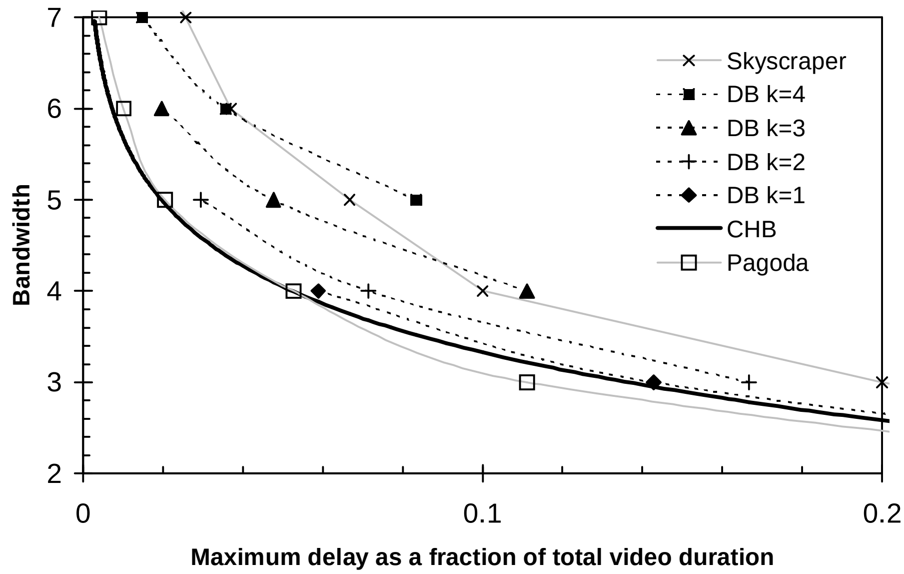

Figure 1: Bandwidth requirements of the Dual Broadcasting protocol for various numbers of PPV streams.

The above inequality correctly predicts that the maximum number of segments that can be broadcast using the PPV streams and one VOD stream is 3 since

$$\sum_{i=1}^{4}\left\lceil \frac{4}{i} \right\rceil = 4+2+2+1 > 2\times 4$$

On the other hand, it predicts that 8 segments could be broadcast using the PPV streams and two VOD streams while no such mapping exists.

Figure 1 shows the bandwidth versus client waiting time curves for Cautious Harmonic Broadcasting (CHB), Pagoda Broadcasting, Skyscraper Broadcasting and our Dual Broadcasting protocol (DB) with between one and four PPV streams. 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 on Figure 1, the bandwidth requirements of our Dual Broadcasting protocol remain very close to those of the CHB and Pagoda Broadcasting protocols as long as the number of PPV streams *k* remains less than three. Larger values of *k* reduce the maximum waiting time for the customers of the PPV service but increase the total cost of the service. Even then the protocol remains competitive with Skyscraper Broadcasting, which is known to be the best of all Pyramid-based broadcasting protocols.

One last factor of the performance of a VOD broadcasting protocol is its maximum disk storage requirements. Most protocols require enough free space on the STB disk drive to store between 40 and 60 percent of the total duration of each video. Our Dual Broadcasting protocol will require less free space as we will never have to store more than the first d_{PPV} minutes of the video. Increasing the number *k* of PPV streams above four could make it feasible to replace the STB hard drive by a sufficiently large random-access memory in some not too distant future.

4. Possible Extensions

To illustrate the additional flexibility gained by combining the two services, we sketch two possible extensions to our Dual Broadcasting protocol. Our first extension is aimed at reducing the bandwidth requirements of the protocol by requiring the STBs of the VOD customers to snoop on the PPV streams and preload from them the first segment of each video being broadcast. A second extension extends the functionality of the VOD service by providing reverse and fast forward controls.

4.1 Reducing bandwidth consumption through snooping

Looking back at all three segment to stream mappings in the previous section, one can see that one of the *l* VOD streams is almost entirely dedicated to the continuous retransmission of the first segment of the

Table 3: Segment to slot mapping for Dual Broadcasting with snooping and one VOD channel.

Current PPV Stream	S_1	S_2	S_3	S_4	S_5	S_6
VOD Stream	S_3	S_4	S_5	S_2	S_3	S_2

video. Hence eliminating the need to retransmit this first segment would save us an entire data stream, that is, b units of bandwidth. One way to achieve this goal would be to let the STBs of the VOD customers snoop on the PPV streams and preload from them the first segment of each video. The scheme would require extra space on the STB disk drives and a tighter coordination between the customer STBs and the service providers. It would work better if these service providers broadcast the top five to ten "hot" videos rather than a much wider range of videos.

Having eliminated the need to dedicate a data stream to the continuous rebroadcasting of segment S_1, one single VOD stream would allow us to partition the d_{PPV} first minutes of the video into 6 segments using the segment to stream mapping represented on Table 3. As a result, the maximum waiting time for the VOD clients d_{VOD} is reduced to one sixth of the maximum waiting time for the PPV clients. Similarly, two VOD streams would allow us to partition the d_{PPV} first minutes of the video into 16 segments using the segment to stream mapping of Table 4.

4.2 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 only 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 more difficult 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. The situation is different for our Dual Broadcasting protocol thanks to the existence of the k PPV streams. If there are enough of these streams, any jump forward would leave the viewer not too far from what is being currently broadcast on one of these streams and the missing information will be on the average equal to one half of the staggering interval d_{PPV}. Assuming that not too many viewers may want to use this new fast forward feature, we could send the missing information *on demand* to the customer STB.

5. Conclusions

One of the main reasons explaining the failure of video-on-demand (VOD) services in the marketplace is the high cost of providing these services. Most efforts aimed at reducing these costs have focussed on reducing the cost of distributing the top ten to twenty videos by broadcasting them in a periodic fashion rather than waiting for individual requests. Unfortunately, nearly all existing VOD broadcasting protocols require client set-top boxes (STB) to include enough local storage to store up to 55 percent of each video being viewed.

We have presented a novel VOD broadcasting protocol that does not make that demand. Our *Dual Broadcasting* protocol can accommodate clients who do not have any storage device in their STB while providing a much lower maximum waiting time to customers whose STB includes a disk drive. Despite this additional flexibility, the bandwidth requirements of our new protocol remain comparable to that of Cautious Harmonic Broadcasting and Pagoda Broadcasting, two of the best VOD broadcasting protocols.

We have also presented two possible extensions to our Dual Broadcasting protocol. Our first extension is aimed at reducing the bandwidth requirements of the protocol by letting the STBs that include a disk drive preload the first few minutes of each video. A second extension extends the functionality of the VOD service by providing move backward and fast forward controls.

Table 4: Segment to slot mapping for Dual Broadcasting with snooping and two VOD channels.

Current PPV Stream	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}
1st VOD Stream	S_6	S_9	S_4	S_2	S_{10}	S_2	S_{12}	S_2	S_{14}	S_2	S_6	S_2	S_{15}	S_2	S_4	S_2
2nd VOD Stream	S_{10}	S_3	S_5	S_7	S_{11}	S_3	S_{13}	S_4	S_3	S_5	S_4	S_3	S_7	S_5	S_3	S_8

References

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

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

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

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

[6] J.-F. Pâris, S. W. Carter, and D. D. E. Long. "Efficient broadcasting protocols for video-on-demand." *Proceedings of the 6th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems* (MASCOTS '98), Montréal, Canada, July 1998, pages 127–132.

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

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

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