

A CHANNEL-BASED HEURISTIC DISTRIBUTION PROTOCOL FOR VIDEO-ON-DEMAND

Qiong Zhang

Jehan-François Pâris¹

Department of Computer Science
University of Houston
Houston, TX 77204-3010

Abstract

We present a new dynamic broadcasting protocol improving upon existing protocols in two ways. First, our *channel-based heuristic distribution* protocol uses a heuristic segment allocation scheme for the k broadcasting channels assigned to each video. Second, it can require all customers to wait for a minimum waiting time before watching the video of their choice. As a result, our protocol performs as well as the best reactive distribution protocols at low request arrival rates and as well as the best broadcasting protocols at high request arrival rates.

1. INTRODUCTION

Broadcasting protocols provide the most cost-effective solution for distributing popular videos on demand to large customer bases. As a result, they offer the best hope for the successful deployment of metropolitan video-on-demand (VOD) services.

Broadcasting protocols anticipate customer demand and distribute the various segments of each video according to a deterministic schedule. The bandwidth savings can be considerable since the top ten to twenty most popular videos are likely to account for over forty percent of the total demand [4].

The first truly efficient broadcasting protocol was Viswanathan and Imielinski's *pyramid broadcasting* protocol [10]. Pyramid broadcasting and all broadcasting protocols it inspired assume that customers will receive their videos through a set-top box (STB) capable of (a) simultaneously receiving data from several video channels and (b) storing in a local buffer up to 60 percent of each video being watched. As a result, the best broadcasting protocols only require six times the video consumption rate to achieve a maximum waiting time of 32 seconds for a two-hour video [9].

Broadcasting protocols have nevertheless one major drawback: they are poorly suited for distributing less popular videos, say, videos that are requested less than ten times an hour [3, 8]. This would not be a problem if the frequency of requests for any given video were not likely to vary widely

with the time of the day. For instance, child-oriented fare will always be in higher demand during the day and early evening hours than at night. Conversely, videos appealing to older viewers are likely to follow an opposite pattern.

The best solution to this problem is offered by *dynamic broadcasting* protocols. These protocols operate like conventional broadcasting protocols but keep track of user requests, which allows them to skip segment transmissions that are not needed by any user. As a result, they require much less bandwidth when requests become less frequent. Two examples of this approach are the *universal distribution* (UD) protocol [8] and the *dynamic heuristic broadcasting* protocol [3]. While both protocols perform satisfactorily at low to medium request arrival rates, their bandwidth requirements at high request arrival rates still exceed those of fixed-delay broadcasting protocols, such as the fixed-delay pagoda broadcasting protocol [9].

The *channel-based heuristic distribution* (CBHD) protocol we present does not suffer from this limitation. It performs as well as the best reactive protocols at low to medium requests arrival rates and as well as the fixed-delay pagoda broadcasting protocol at high request arrival rate.

2. PREVIOUS WORK

For brevity sake, we will focus our discussion on the distribution protocols that are directly relevant to our work. These include the fast broadcasting protocol [7], on which the original universal distribution protocol was based.

Juhn and Tseng's *fast broadcasting* (FB) protocol is one of the most intuitive broadcasting protocols. It allocates to each video k channels whose bandwidths are all equal to the video consumption rate b . It then partitions each video into $2^k - 1$ segments S_1 to S_{2^k-1} of equal duration d . As Figure 1 indicates, the first channel continuously rebroadcasts segment S_1 , the second channel transmits segments S_2 and S_3 , and the third channel transmits segments S_4 to S_7 . More generally, channel j with $1 \leq j \leq k$ transmits segments $S_{2^{j-1}}$ to S_{2^j-1} in such a way that each segment is repeated once every $2^{j-1}d$ time units.

The *fixed-delay pagoda broadcasting* (FDPB) protocol [9] is a more recent protocol that requires all users to wait for a fixed delay before watching the video they have selected. The protocol uses this delay to reduce the bandwidth

¹ Both authors were supported in part by the Texas Advanced Research Program under grant 003652-0124-1999 and the National Science Foundation under grant CCR-9988390.

First Channel	S ₁	S ₁	S ₁	S ₁
Second Channel	S ₂	S ₃	S ₂	S ₃
Third Channel	S ₄	S ₅	S ₆	S ₇

Figure 1. The first three channels for fast broadcasting

First Channel	First Subchannel	S ₁ and S ₂
	Second Subchannel	S ₃ to S ₅
Third Channel	First Subchannel	S ₆ to S ₈
	Second Subchannel	S ₉ to S ₁₂
	Third Subchannel	S ₁₃ to S ₁₇
Third Channel	First Subchannel	S ₁₈ to S ₂₁
	Second Subchannel	S ₂₂ to S ₂₆
	Third Subchannel	S ₂₇ to S ₃₂
	Fourth Subchannel	S ₃₃ to S ₃₉
	Fifth Subchannel	S ₄₀ to S ₄₇

Figure 2. The first three channels for the FDPB protocol with a delay of four segments

required to transmit the first minutes of each video. As shown on Figure 2, each channel is subdivided into a variable number of subchannels. Each of these subchannels has a different periodicity, which allows the FDPB protocol to pack more segments per channel. As a result, the FDPB protocol achieves lower customer delays than the FB protocol with the same amount of bandwidth.

The *universal distribution* (UD) protocol [8] is a dynamic broadcasting protocol based upon the FB protocol. Segments are transmitted only on demand, which saves a considerable amount of bandwidth when request arrival rates do not exceed 60 requests per hour. Above 200 requests per hour, all channels become saturated and the UD reverts to a conventional FB protocol.

The *dynamic heuristic broadcasting* (DHB) protocol [3] is not based on any static broadcasting protocol. In particular, segments are not assigned to specific channels. The protocol schedules all segments on demand and relies on a bandwidth smoothing heuristics to limit, but not eliminate bandwidth peaks.

3. THE CBHD PROTOCOL

All the existing dynamic broadcasting protocols have their own limitations. While the UD protocol performs as well as the best reactive protocols at low request arrival rates, its performance at high request arrival rates is less satisfactory as it is outperformed there by the best conventional broadcasting protocol [8]. The DHB protocol has better average

bandwidth requirements than the UD protocol but experiences sudden bandwidth peaks that cannot be eliminated. In addition, its average bandwidth requirements at high request arrival rates are still higher than those of the FDPB protocol. Protocols like stream tapping [2], patching [1], HMSM [5] and selective catching [6] provide true instant access to the videos but never reach a bandwidth plateau at high request arrival rates.

Our new *channel-based heuristic distribution* (CBHD) protocol incorporates features from the UD, DHB and FDPB protocols. Like the UD protocol, the CBHD protocol assigns to each video a fixed number k of channels whose bandwidths are equal to the video consumption rate b . Each video is also partitioned into n segments of equal duration d . As a result, each channel will be divided into *slots* of the same duration d . Depending on the request arrival patterns, these slots can be either free or occupied by an instance of a segment.

The CBHD protocol guarantees that customers requesting a video will never wait more than md time units, where m is a positive integer. Whenever $m > 1$, the protocol also requires that all customers wait at least $(m-1)d$ time units. As a result, the average customer waiting time is $(m-1/2)d$ time units.

There are two advantages in adopting this policy. First, customer STBs can begin receiving data from the video server before the customer starts watching the video and this was found to reduce the bandwidth required to transmit the first minutes of each video [9]. Second, a small predictable delay provides a more consistent quality of service to the customer than a random delay.

The first segment of the video is segment S_1 . To ensure that S_1 is received on time, we need to have at least one instance of S_1 in any of the m slots following the arrival of a request for the video. More generally, segment S_j will be received on time if there is at least one instance of S_j in any of the $j+m-1$ slots following the arrival of a request.

Consider now what happens when the request arrival rate for a video becomes so high that the video server always receives one request per slot. Even under these conditions, the video server can satisfy all user requests by scheduling one instance of segment S_j every $j+m-1$ slots.

We will use this property to decide how many segments can be assigned to each channel. Since segment S_1 never needs to appear more than once every m slots, the video server can satisfy all user requests by scheduling one instance S_1 every m slots. As successive segments need to be repeated less and less frequently, we can safely assign m segments to the first channel. Hence the first segment to be broadcast by the second channel will be segment S_{m+1} . Since this segment never needs to appear more than once every $2m$ slots, we can assign segments S_{m+1} to S_{3m} to that channel. More generally, channel i will be assigned segments $S_{(2^{i-1}-1)m+1}$ to $S_{(2^i-1)m}$. Allocating k channels to a video will allow us to partition a video into $(2^k-1)m$ segments and achieve a the video.

Assumptions:

video is assigned k channels and partitioned into $(2^k - 1)m$ segments of equal duration d
channel i with $1 \leq i \leq k$ transmits *on demand* segments $S_{(2^{i-1}-1)m+1}$ to $S_{(2^i-1)m}$
all customers wait for at least md time units
new video request arrives during slot s

Algorithm:

```

for  $i := 1$  to  $k$  do
  for  $j := (2^{i-1}-1)m+1$  to  $(2^i-1)m$  do
    search for a previous instance of segment  $S_j$  in slots  $s+1$  to  $s+j+m-1$  of channel  $i$ 
    if not found then
      find furthest free slot  $f$  of channel  $i$  such that  $s+1 \leq f \leq s+j+m-1$ 
      schedule a new instance of segment  $S_j$  in slot  $f$ 
    end if
  end for loop
end for loop

```

Figure 3. The channel-based heuristic distribution protocol.

This result is not different of that achieved by the UD protocol. What sets the two algorithms apart, is how they assign segments to slots within each of these channels. The UD protocol used an allocation method that reverted to the FB protocol at high request arrival rates [8]. The CBHD protocol uses a simpler heuristics that always leaves free slots in most channels. Hence the same waiting times can be achieved with a lower average bandwidth.

Figure 3 describes this heuristics. Assume that a request arrives at the video server during slot s . The server will first review its current segment schedule for all video segments. If it finds an instance of a given segment S_j already scheduled in one of the next $j+m-1$ slots, it does not schedule any new instance of S_j . Otherwise, it scans the channel to which S_j was assigned and searches for the best free slot in which it can schedule a new instance of S_j . This slot must be no later than $j+m-1$ slots ahead to ensure the continuity of the video viewing process. The server picks the furthest free slot f in the interval $[s+1, s+j+m-1]$ and schedule a new instance of S_j in that slot.

Consider now a channel i broadcasting segments $S_{(2^{i-1}-1)m+1}$ to $S_{(2^i-1)m}$. Let us further assume that the channel is operating at its saturation point due to a very high arrival rate. As we mentioned earlier, continuity of viewing requires each segment S_j to be broadcast once every $j+m-1$ slots. The minimum average slot occupancy at saturation point for that channel will then be given by

$$\sum_{j=(2^{i-1}-1)m+1}^{(2^i-1)m} \frac{1}{j} = H((2^i-1)m) - H((2^{i-1}-1)m) < 1$$

where $H(n)$ is the n th harmonic number. Observed average slot occupancies will always be high because the lack of free slots will result in some segments being scheduled more frequently than they have to. Consider for instance the case of the second channel having to transmit segments S_2 and S_3 . The minimum average slot occupancy at saturation point for that channel is given by

$$\sum_{j=2}^3 \frac{1}{j} = H(3) - H(1) = 0.8333$$

In practice, we observe that the all channel slots are occupied as the channel continuously repeats the pattern

$$S_3 \ S_2 \ S_2 \ S_3 \ S_2 \ S_2 \ \dots$$

4. PERFORMANCE EVALUATION

Figure 4 compares the bandwidth requirements of the CBHD protocol (with $m = 1$) with those of four other video distribution protocols for arrival rates varying between one and one thousand customers per hour. We assumed that the server was broadcasting a two-hour video partitioned into 127 segments, which would correspond to a maximum customer waiting time of slightly less than a minute. Request arrivals were simulated by a Poisson process. All bandwidths are expressed in “channels,” that is, in multiples of the video consumption rate.

The protocols against which we compared the bandwidth requirements of the CBHD protocol are:

- stream tapping*, which has the lowest bandwidth requirements of all reactive protocols imposing no delay on customer requests (including *patching*),
- a *universal distribution* (UD) protocol partitioning the video into 127 segments,
- a *dynamic hybrid broadcasting* (DHB) protocol, also partitioning the video into 127 segments, and
- a *fixed-delay pagoda broadcasting* (FDPB) protocol partitioning the video into 8128 segments and requiring customers to wait for the duration of 64 segments to achieve a waiting time equal to $1/127$ of the video duration.

As one can see, the CBHD protocol performs much better than the original UD protocol at medium to high arrival rates. It requires slightly more bandwidth than the DHB

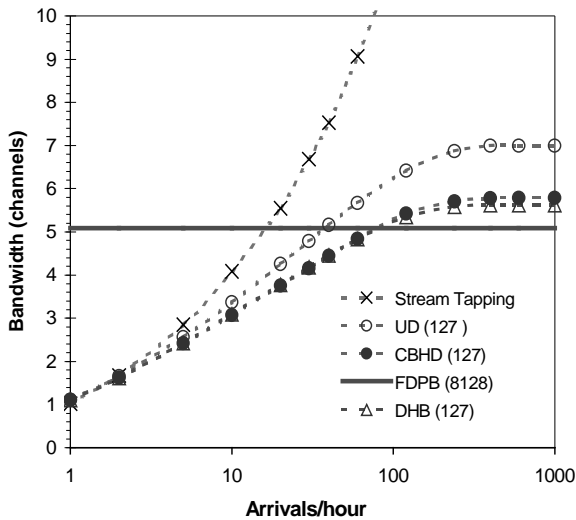


Figure 4. Compared bandwidth requirements of the CBHD, UD, DHB and FDPB protocols.

protocol, which should not be surprising because the DHB protocol uses a much less constrained segment-to-slot mapping. This slight edge is more than compensated by the higher peak bandwidth requirements of the DHB protocol, namely 8 channels instead of 7 for the CBHD protocol.

The much better performance of the FDPB protocol can be easily explained: FDPB is a fixed-delay protocol that requires the customer STB to accept video data as the customer waits for the video while the CBHD protocol only does it when $m > 1$.

Figure 5 compares the performance of the FDPB and the CBHD protocols with the same delay coefficient $m = 64$. Both protocols now partition the video into 8128 segments to achieve a maximum waiting time still equal to $1/127$ of the video duration. As one can see, the CBHD protocol has now lower average bandwidth requirements than the FDPB protocol. Its single drawback is its higher peak bandwidth requirements, 7 channels instead of 6 channels for the FDPB protocol.

Similar results were obtained for waiting times varying between $1/32$ and $1/256$ of the duration of the video, that is between slightly less than four minutes and slightly less than 30 seconds for a two hour video. Space considerations prevented us from including them in this paper.

We also found that there was little motivation for selecting larger values of m . While quadrupling m from $m = 1$ to $m = 4$ always reduced the average bandwidth requirements of the CBHD protocol by at least 10 percent, quadrupling it from $m = 16$ to $m = 64$ rarely reduced the same requirements by more than 2 percent.

As most video distribution protocols, the CBHD protocol imposes storage and bandwidth requirements on the customer STB. It should be able to receive data at k times the video consumption rate and may have to store locally the whole contents of the last channel.

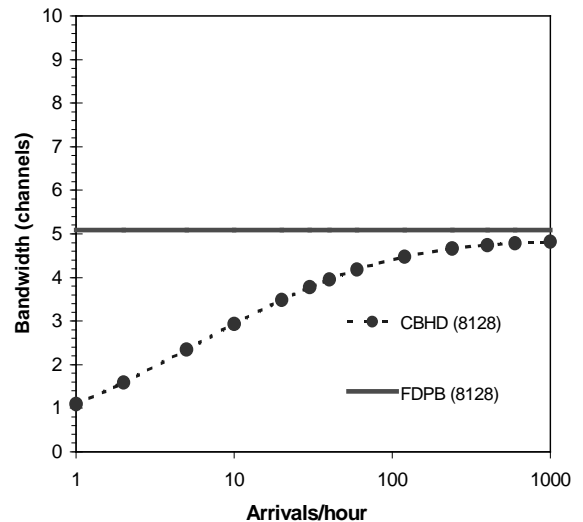


Figure 5. Compared bandwidth requirements of the CBHD and the FDPB protocols for $m = 64$ and 8128 segments.

5. CONCLUSIONS

We have presented a dynamic broadcasting protocol that improves upon the original *universal distribution* protocol by using a heuristic segment allocation scheme for the k channels assigned to each video and imposing a minimum waiting time on all customer requests. As a result, our *channel-based heuristic distribution* protocol performs as well as the best reactive distribution protocols at low request arrival rates and as well as the best broadcasting protocols at high request arrival rates.

REFERENCES

- [1] Cai, Y., K.Hua and K.Vu. Optimizing patching performance. *Proc. ACM/SPIE Multimedia Computing and Networking Conf.*, Jan. 1999.
- [2] Carter, S. W. and D. D. E. Long. Improving video-on-demand server efficiency through stream tapping. *Proc. 6th Int. Conf. on Computer Communications and Networks*, pp. 200-207, Sep. 1997.
- [3] Carter, S. R., J.-F. Pâris, S. Mohan, and D. D. E Long. A dynamic heuristic broadcasting protocol for video-on-demand, *Proc. 21st Int. Conf. on Distributed Computing Systems*, pp. 657-664, Apr. 2001.
- [4] Dan, A., D. Sitaram, and P. Shahabuddin. Dynamic batching policies for an on-demand video server. *ACM Multimedia Systems Journal*, 4(3):112-121, June 1996.
- [5] Eager, D. L., M. K. Vernon and J. Zahorjan. Minimizing bandwidth requirements for on-demand data delivery. *Proc. 5th Int. Workshop on Advances in Multimedia Information Systems*, Oct. 1999.
- [6] Gao, L., Z.-L Zhang and D. Towsley. Catching and selective catching: efficient latency reduction techniques for delivering continuous multimedia streams. *Proc. 1999 ACM Multimedia Conf.*, pp. 203-206, Nov. 1999.
- [7] Juhn, L. and L. Tseng. Fast data broadcasting and receiving scheme for popular video service. *IEEE Trans. on Broadcasting*, 44(1):100-105, Mar. 1998.
- [8] Pâris, J.-F., S. W. Carter and D. D. E. Long. A universal distribution protocol for video-on-demand. *Proc. Int. Conf. on Multimedia and Expo 2000*, Vol. 1, pp. 49-52, July 2000.
- [9] Pâris, J.-F. A fixed-delay broadcasting protocol for video-on-demand. *Proc. 10th Int. Conf. on Computer Communications and Networks*, pp. 418-423, Oct. 2001.
- [10] Viswanathan, S. and T. Imielinski. Metropolitan area video-on-demand service using pyramid broadcasting. *ACM Multimedia Systems Journal*, 4(4):197-208, 1996.