On the Design of Efficient Video-on-Demand Broadcast Schedules

Ailan Hu   Ioanis Nikolaidis*   Peter van Beek
Department of Computing Science
University of Alberta
Edmonton, Alberta
Canada T6G 2H1
{ahu,yannis,vanbeek}@cs.ualberta.ca

Abstract

In order to address the scalability problems of Video–on–Demand systems, several periodic broadcast schemes have been proposed that partition a video into segments and repetitively broadcast each segment on a separate channel. A new scheme is presented for the bandwidth–efficient periodic broadcast of video. The proposed scheme determines the segment sizes and their corresponding channel bandwidths as a result of a non–linear optimization problem which minimizes the total required bandwidth for the broadcast. The new scheme outperforms the existing schemes in terms of bandwidth demands while it also decouples the playout latency from the number of available channels. Further analysis reveals that its asymptotic bandwidth requirements exactly match the asymptotic bandwidth requirements reported for Poly-Harmonic Broadcasting.

1. Introduction

In order to solve the scalability problems of Video-on-Demand (VoD) systems for frequently requested ("hot" set) videos, several *periodic broadcast* schemes have been proposed [1, 2, 3, 4, 5, 6]. The common feature of all proposed schemes is that low *playout latency* (i.e., delay between "tuning–in" to a video broadcast and the point where the playout of the video can begin) is achieved by partitioning the video files into a number, n, of segments. Each segment is repeatedly broadcast on a different channel. The playout latency depends on how frequently the *first* segment is broadcast. Subsequent segments are guaranteed to be received in advance of their playout. The guarantee is established by the particular timing relation between the broadcast of the different segments and by the segment download policy used by the receiver. The exact broadcast timing relation and download policy is different for each scheme.

Note that there exist several ways to allocate channels out of the available physical transmission capacity, e.g. by using time–division multiplexing. A study of the particular bandwidth allocation scheme for the individual channels is outside the scope of this paper and is assumed to be implemented by a suitable scheduler. In the following, a channel will simply represent a certain allocated bandwidth. Note also that buffering at the receiver is necessary because the playout of a segment can overlap with the reception of future segments. Given the proliferation of mass storage devices, such as disks, capable of tens of gigabytes of storage, we will assume that the receiver storage is not a constraint in the development of a broadcast scheme. Potentially, an entire video can be stored even in its entirety at the receiver prior to its playout.

Summarizing, three features can be identified in each proposed periodic broadcast scheme: (a) the bandwidth, b_i, allocated to each channel/segment, i, (b) the size the segments, S_i, relative to the video file size, S, and, (c) the particular broadcast timing relation and receiver download policy.

Existing schemes can be classified into two groups. In the first group belong schemes that assume equal bandwidth (EB) for each channel and different segment sizes. Pyramid Broadcasting (PB), Permutation Pyramid Broadcasting (PPB) and Skyscraper Broadcasting (SB) belong to EB. In the second group belong schemes that assume equal segment (ES) sizes but different bandwidth for each channel. Cautious Harmonic Broadcasting (CHB), Quasi-Harmonic Broadcasting (QHB) and Poly-Harmonic Broadcasting (PHB) belong to ES.

From the existing results, it appears that bandwidth efficiency is a feature of ES schemes. Namely, CHB and QHB demonstrate a very slowly increasing bandwidth demand for increasing number of channels. PHB, requires the least total bandwidth of all. Remarkably, it can be shown that,

*This works was supported in part by NSERC under grant OGP0194424.

in the asymptotic case (infinite number of channels), PHB's bandwidth demand converges to $\ln\left(\frac{S}{w}+1\right)$ times the consumption[1] bandwidth of the video, where S is the video duration and w is the playout latency. Incidentally, PHB is the only scheme where the download policy is to start downloading immediately upon "tune–in" all of the channels allocated to a video, whereas in all other protocols the download can start only from the beginning of a segment.

As a result of the previous work in this area, several question arise: Are the EB schemes by definition inefficient because of their choice to assign equal bandwidth to each channel? If not, how should the bandwidth assigned to each channel be calculated? Is PHB's converging bandwidth demands a result of its ES nature or of the fact that the downloading of the video information starts immediately at the tune–in instant, i.e., because of its download policy? Overall, can we remove the a-priori design choices for assigning bandwidth or segment sizes and view them as the solution to an optimization problem with the objective of producing a broadcast scheme that minimizes the bandwidth demands?

This paper addresses the above questions. Namely, it proposes the construction of a bandwidth–efficient EB scheme which outperforms the existing schemes and which demonstrates the same asymptotic bandwidth demands as PHB. The new results provide strong evidence that it is the receiver download policy that provides to PHB its particular bandwidth efficiency rather than its ES nature. Moreover, because of its construction process, the proposed scheme is capable of providing *any* arbitrary playout latency for *any* arbitrary restriction on the number of channels. In this sense, it is the first scheme to our knowledge which successfully decouples the playout latency from the available number of channels.

In this paper we first establish the particular temporal relationship of the transmitted segments. The temporal relationship is the result of several design considerations arising from the study of previously proposed schemes and of PHB in particular. Subsequently, the exact segment durations and channel bandwidths are determined as a result of an optimization procedure. We select as objective the minimization of the server bandwidth required for the broadcast of a single video.

The remaining of the paper is structured as follows: In section 2, we provide an overview of the existing schemes and comment on their performance. In section 3, we introduce a new scheme and describe the optimization procedure behind its construction. In section 4, we derive the properties of the new scheme. Finally, in section 5 we summarize our findings and indicate directions for improvement and topics for future research.

[1] *Consumption bandwidth*, b, is the rate at which the data for a CBR video are played out. All bandwidth values used in this paper are expressed in units of the consumption bandwidth, i.e., $b = 1$.

2. Previous Work

Figure 1 summarizes the construction of four of the most representative broadcast schemes. The duration of the segments in PB follow a geometric series with parameter α. Because the transmission bandwidth is the same for all segments, the larger segments repeat less frequently than the shorter ones. α is chosen such that the playout is uninterrupted. PB is defined in [1] as operating on a set of M videos that share the server bandwidth. Although M is allowed to be 1, when $M \geq 2$, the construction of PB implies that all M videos are of approximately the same duration. The playout latency reported in Figure 1 is the best latency, i.e., when $M = 1$.

In PB, α influences the bandwidth efficiency, and although its optimal value (for minimum bandwidth) is equal to e, it is not always realizable because of α's dependence on the integer quantities n and M. However, the primary problem of PB is the excessive use of bandwidth which is due to the simplistic nature of the receiver which receives concurrently only two segments (the one being played out and the next one to be played). A well known improvement to PB is the Permutation Pyramid Broadcasting (PPB) [2] scheme.

PPB multiplexes each segment with itself, creating logical sub-channels out of each channel. The same segment is broadcast on each sub-channel with certain phase delay from one sub-channel to the next. With this approach, PPB gives the opportunity to receive a segment earlier than would have been possible with PB. However, when implementing PPB, the value of n needs to be constrained. The constraint results in limited improvement to the client latency once a certain bandwidth is exceeded. That is, PB can improve the latency exponentially with more available bandwidth while PPB can do so only up to a limit.

A different approach to the construction of the variable segment durations is proposed in Skyscraper Broadcasting (SB) [3]. The function that is used to recursively "grow" the segment durations is producing smaller segments than the geometric sequence of PB. Nevertheless, the function used provides sufficient duration to guarantee that the playout, once it starts, will not be interrupted. Moreover, in SB, a maximum limit W on the segment durations is enforced. The maximum bandwidth and buffer space requirements are thus controlled by W. SB provides a bigger improvement than PPB, because PPB maintains the same (large) segment sizes of PB. However, a common feature of all EB schemes (PB, PPB and SB) is that they couple the playout latency to the number of segments, n. That is, because n is an integer, arbitrary playout latency values cannot be achieved.

HB [4] was the first member of the ES family of broadcast schemes. HB was revised to the CHB scheme and subsequently resulted in QHB. The important feature of the

PB [1]	SB [3]	QHB [5]	PHB [6]
$b_i = \alpha,$ $S_i = \begin{cases} \frac{S(\alpha-1)}{(\alpha^n-1)} & i=1, \\ \alpha S_{i-1} & i \geq 2, \end{cases}$ $w = \frac{S_1}{\alpha}.$	$b_i = 1,$ $S_i = \begin{cases} \frac{S}{\sum_{j=1}^{n} \min(f(j),W)} & i=1, \\ f(i)S_1 & i \geq 2, \end{cases}$ $f(i) = \begin{cases} 1 & i=1, \\ 2 & i=2,3, \\ 2f(i-1)+1 & i \bmod 4 = 0, \\ f(i-1) & i \bmod 4 = 1, \\ 2f(i-1)+2 & i \bmod 4 = 2, \\ f(i-1) & i \bmod 4 = 3, \end{cases}$ $w = S_1.$	$b_i = \begin{cases} 1 & i=1, \\ \frac{m}{im-1} & i \geq 2 \end{cases}$ $S_i = \frac{S}{n},$ $w = S_1.$	$b_i = \frac{1}{m+i-1},$ $S_i = \frac{S}{n},$ $w = mS_1.$

Figure 1. Construction rules and playout latency of periodic broadcast schemes.

HB, CHB and QHB is that the server bandwidth demands follow a very slowly increasing function as the number of segments increases. The increase of the segments allows the decrease of the playout latency, which is again coupled to the specific value of the number of segments. One of the problems associated with ES schemes is that they require the link schedulers to be able to allocate any fractional amount of bandwidth to individual channels. QHB's approach is to re-partition each segment into m sub-segments. The repartition allows a more flexible timing relation between the segment broadcasts (since the sub–segment becomes the new unit of information received and played–out).

Interestingly, the decreased bandwidth demands of QHB are the result of the more flexible timing relation of the segments. The example of QHB is taken to the extreme with PHB. According to PHB, a segment can be loaded starting at any point in time during its transmission. Hence, by downloading from a channel for as much time as the duration of the corresponding segment (where the duration is calculated at the corresponding channel's bandwidth) it is guaranteed that all the information of the segment are downloaded and available.

In all existing schemes, lower latencies require larger values of n. Thus, a possible restriction stems from the limits imposed by the system design, e.g., hardware restrictions, on the number of segments, n. For higher values of n, the EB schemes are problematic due to the linear increase of the required bandwidth. However, the ES schemes hold promising results, since their required bandwidth is a slowly growing function with respect to n. The question one may pose at this point, is whether it is possible for EB schemes to also exhibit slowly growing bandwidth for increasing n, since they are, in a sense, "duals" of the ES schemes. In the following sections, such a bandwidth–efficient EB scheme is in fact constructed.

3. The Construction of GEBB

The properties of PHB suggest that in order to improve the playout latency, no extra bandwidth may be necessary but, rather, a more "intelligent" way to partition the bandwidth to the individual channels. If we develop this observation further, we realize that it would be possible to keep the total bandwidth the same and improve the playout latency by assigning different sizes to different segments (which subsequently results in different per–channel bandwidths, even if their sum remains the same). Another side of the same observation is that if such a scheme is constructed, then for a given playout latency, there must exist a *limit* for the necessary bandwidth by changing (namely, increasing) the number of channels. PHB's asymptotic bandwidth demand is a direct evidence that such a limit exists. Does this limit apply to PHB only? We will see that the limit is the same for the scheme we construct next.

The complication that arises while trying to implement an arbitrary assignment of segment sizes or channel bandwidths is that the timing relations of the segment broadcasts cannot be maintained for most of the proposed schemes. In fact, it cannot be maintained in *all* schemes with the remarkable exception of PHB. PHB operates in a "greedy" fashion, i.e., receive as much of the data as possible from *all* of the channels immediately after tune-in, and ceases receiving a segment immediately before playing out the corresponding segment. It is therefore possible to use the same timing restriction of PHB and develop a scheme that observes the

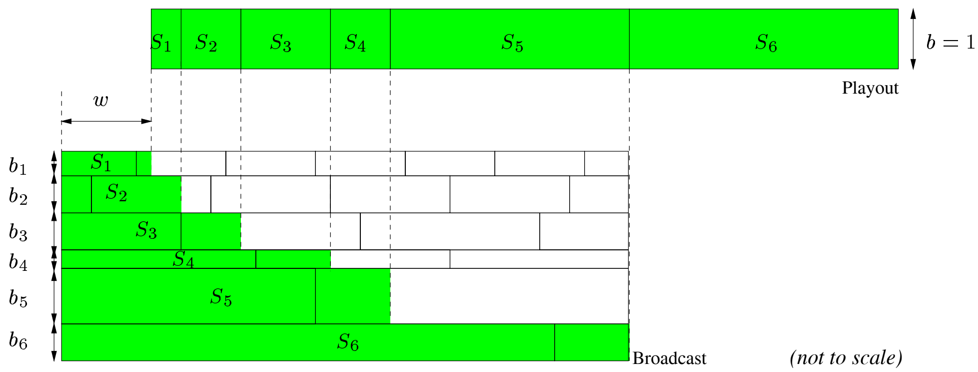

Figure 2. Example broadcast and playout according to GEBB. The shaded area represents data received and played out by the client.

timing relation as presented in the example of Figure 2.

We pursue an analytical derivation of the segment durations and their corresponding channel bandwidths with the objective of minimizing the total server bandwidth required to broadcast a specific video. The optimization problem can be formally stated as:

$$\text{minimize} \sum_{i=1}^{n} b_i. \tag{1}$$

subject to,

$$b_i \left(w + \sum_{j=1}^{i-1} S_j \right) = S_i \quad i = 1, 2, \ldots, n, \tag{2}$$

$$\sum_{i=1}^{n} S_i = S, \tag{3}$$

$$S_i > 0, b_i > 0 \quad i = 1, 2, \ldots, n. \tag{4}$$

The condition represented by equation (2) ensures that segment $i + 1$ is *completely received* at the exact timepoint when the playout of segment i terminates. Thus the segments are available always and exactly on–time for their playout. Note that another option (not covered in this paper) is to allow a segment to be only partially received when its consumption starts.

The above non-linear optimization problem can be solved using the Lagrange Multiplier method. However, a simpler approach is to derive the quantities $(1 + b_i)$ which can be used to simplify the problem:

$$(1 + b_i) = \frac{w + \sum_{j=1}^{i} S_j}{w + \sum_{j=1}^{i-1} S_j} \quad i = 1, 2, \ldots, n. \tag{5}$$

Multiplying all n of the $(1 + b_i)$ quantities produces:

$$\prod_{i=1}^{n} (1 + b_i) = \frac{w + \sum_{j=1}^{n} S_j}{w} = \frac{S}{w} + 1 \tag{6}$$

Thus, we have determined that the product of the n variables $(1 + b_i)$ is a constant, $\frac{S}{w} + 1$. However, note that the minimization objective (1) is equivalent to the minimization of the same objective, plus a constant, namely:

$$\text{minimize} \sum_{i=1}^{K} b_i \equiv \text{minimize} \sum_{i=1}^{K} (1 + b_i) \tag{7}$$

Combined, equations (6) and (7) form a familiar and widely known optimization problem (frequently posed as a puzzle). Namely, that of determining the value of n positive numbers that minimizes their sum given that their product is constant. The answer is that the minimization of the sum is achieved only when all the variables take the same value. Let us represent this optimal value as $(1 + b^\star)$. Consequently,

$$(1 + b^\star)^n = \frac{w + S}{w} \tag{8}$$

hence,

$$b_i = b^\star = \sqrt[n]{\frac{S}{w} + 1} - 1 \quad i = 1, 2, \ldots, n. \tag{9}$$

Remarkably, our first conclusion, after relaxing the assumptions of EB and ES, is that the required bandwidth is minimized when the channel bandwidths are equal. That is, the optimal solution belongs to the EB family. Further manipulation of equation (2), given the optimal value of b_i and subject to the condition of equation (2) produces:

$$S_i = w b^\star \left(1 + b^\star\right)^{i-1} \quad i = 1, 2, \ldots n \tag{10}$$

GEBB
$b_i = b^\star = \sqrt[n]{\frac{S}{w}+1} - 1,$ $S_i = wb^\star(1+b^\star)^{i-1}.$

Figure 3. Construction rules of GEBB.

The resulting values of S_i verify another previously held intuition, that of using a geometric series to form the segment durations. This approach was used in PB and PPB. However, we determine that the parameter of the geometric series is drastically different from the ones proposed so far. In particular, the parameter is influenced by the number of segments as well as from the ratio of playout latency over the total video duration ($\frac{w}{S}$).

Because of its construction rules, we will call the new scheme the Greedy Equal-Bandwidth Broadcasting (GEBB) scheme. The summary of its construction rules is illustrated on Figure 3. Note that the playout latency, w, is a parameter which influences the calculation of the segment sizes and bandwidths and is not dependent on any other variable. In other words, a valuable feature of the proposed scheme is that w, is *decoupled* from any other system parameter and can be assigned any value desired.

4. The Properties of *GEBB*

The total server bandwidth necessary by GEBB for a particular video is $nb^\star$. Given that the $b^\star$ is dependent on n, it is worthwhile to explore the extent to which an increasing n, can improve the total bandwidth demands. It is straightforward to show that:

$$\lim_{n\to\infty} n\left(\sqrt[n]{\frac{S}{w}+1} - 1\right) = \ln\left(\frac{S}{w}+1\right) \tag{11}$$

The limit is the exact same as the one reported for PHB [6]. The limit can be better illustrated by observing Figure 4. Given the timing constraints the rate at which data is received starts at its maximum possible value at time 0 (the time when the client "tunes-in" to the broadcast stream). For the first w time units it does not consume any of the data. At time w the entire first segment has been accumulated and its playout can commence. Subsequent segments arrive in time for their playout.

Let us define a function $t(B)$ (depicted on Figure 4) which denotes the time at which the data are downloaded by the receiver at a rate B. Since $n \to \infty$, $t(B)$ can be approximated by a continuous function. The "greedy" nature of the clients indicates that $t(B)$ is non-decreasing. Furthermore, integration of any horizontal region, R, in the broadcast area in Figure 4 should corresponds to a region Q in the playout area, i.e., $t \cdot dB \approx dt$. When dB is infinitesimally small, $U \to 0$ and:

$$t\,dB = dt \;\;\Rightarrow\;\; t(B)\,dB = dt(B)$$

hence, $t(B)$ must be an exponential function: $t(B) = ce^B$ where the constant c is determined by the boundary condition $t(0) = w$. thus:

$$t(B) = we^B$$

The minimum bandwidth, B_0, is determined indirectly by the following:

$$\int_0^{B_0} t(B)dB = S$$

which, solving for B_0, produces,

$$B_0 = \ln\left(\frac{S}{w}+1\right)$$

the exact limit we derived earlier.

Regarding the storage requirements of the client, we note that (for relatively small w compared to S) in the first w time units, $nwb^\star$ data are accumulated before the beginning of playout. Subsequently, consumption at the rate of $b = 1$ can start. As long as the consumption rate is smaller than the rate at which received data are downloaded, the buffer contents increases. Since the change in the accumulation rate occurs at the boundaries between time segments, we can express the minimum buffer demand of the scheme as:

$$nwb^\star + \sum_{j=1}^{l} S_j\left((n-j)b^\star - 1\right) \tag{12}$$

whereby l is defined as the last time segment for which $((n-l)b^\star - 1) > 0$. By solving the corresponding equation for l we determine that:

$$l = \lfloor n - \frac{1}{b^\star} \rfloor \tag{13}$$

A better insight is gained by observing the asymptotic behavior of the buffer demand as $n \to \infty$. In Figure 4, the rectangular area defined by the shaded area corresponds to the contents of the client buffer at time $t(B)$ for $B < B_0$ where $B_0 = \sum_{i=1}^{\infty} b_i = \ln\frac{S}{w} + 1$. The shaded area changes with changing B and it is given by:

$$t(B)(B_0 - B) \tag{14}$$

by locating the point where the derivative of the equation 14 is zero and given the fact that $t(B) > 0$ in the $[0, B_0]$ range, we find the value of B which maximizes the area of

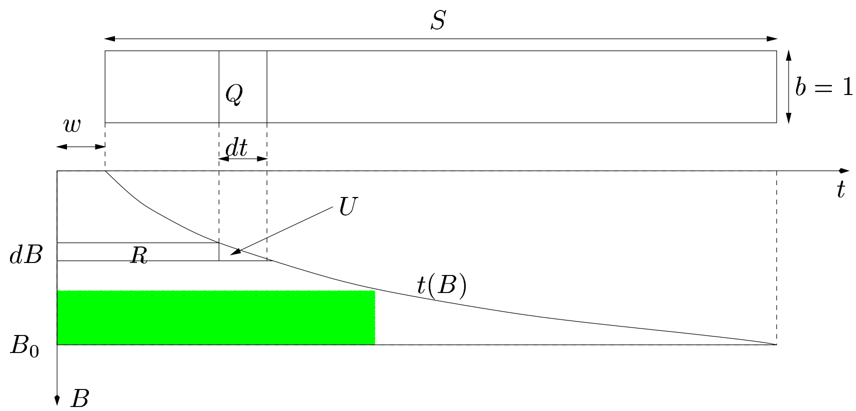

Figure 4. Illustration of the time function $t(B)$ for $n \to \infty$.

the rectangle is $\ln\left(\frac{S}{w}+1\right)-1$ which corresponds to buffer contents

$$\frac{S}{e}+\frac{w}{e} \tag{15}$$

Since we consider a w much smaller than S, the term $\frac{S}{e}$ dominates the asymptotic storage requirements and it corresponds to approximately 37 % of the entire movie.

The client I/O bandwidth requirements are the largest at the beginning of the video reception. It would appear that the I/O bandwidth is equal to the consumption rate, plus the receive bandwidth (which is equal to the server bandwidth). However, since there is no consumption in the first w units, the I/O bandwidth demand is either the initial bandwidth used immediately after "tune–in", or the bandwidth necessary immediately after time w, at which time it includes the playout bandwidth $b = 1$ necessary to retrieve the first segment from storage for playout. Which one of the two quantities is larger depends on whether $b^\star > 1$ or not. Thus, for completeness the client I/O bandwidth is:

$$\max(b^\star,1)+(n-1)b^\star = \begin{cases} nb^\star & \frac{S}{w} \geq 2^n-1 \\ 1+(n-1)b^\star & \frac{S}{w} < 2^n-1 \end{cases} \tag{16}$$

4.1. Comparative Study

Given that PB, PPB and SB require bandwidth linear to n, it would appear that GEBB is outperforming all of them. However, as n increases in PB, PPB, and SB, the playout latency of the clients is also improving. Thus, to make a fair assessment while restricting the number of free parameters, we will use as a comparison basis the fraction of the playout latency over the video duration which we will call the *latency fraction* and denote with w/S. Realistically, the latency fraction will likely take values in the range from 0.01 to almost 1. A small value indicates a small tolerable playout latency. A value of 1 represents a viewer that is willing to wait as much time as the actual video duration. Note that the value of this ratio can be much larger than 1, for servicing clients that may not even wish to view a video in real–time but store it completely for later playout. In the presented comparisons of GEBB with EB and ES schemes, we focuses on the bandwidth demand and the corresponding value of the number of segments, n, that provides the same latency fraction for GEBB and the other schemes.

Figure 5 compares GEBB to PB and SB. PB is unable to capture latency fractions other than the ones where n can be an integer, thus only the two points on the PB line in Figure 5 are attainable for PB. Furthermore, for the same n, PB always requires more bandwidth than GEBB. PB cannot use less than α bandwidth, while, as shown, GEBB can reduce its bandwidth without any bound, for large latency fractions. We should underline that the PB depicted in this figure represents the *ideal* case for PB where $M = 1$ and α is set to the optimal value e. The problems of SB are similar to PB, in the sense of quantized latency fraction values. However, there is an added pathology. A larger number of segments, n, is necessary in order for SB to approach the same latency fraction as PB does. Hence, SB requires an even larger number of segments to reach the performance of GEBB, compared to PB.

Figure 6 compares GEBB to QHB and PHB. The bandwidth efficiency of QHB and PHB is much better (observe that the y–axis range has been restricted to the range [1,5] to zoom into the differences). However, GEBB outperforms both of them with respect to the segments necessary. Because of its rigid timing relation, QHB cannot exploit fully the existence of additional channels, when they are available. As a result, in order to reach the same bandwidth efficiency as GEBB, it requires far more segments than GEBB. For example, the intersection of the lines QHB $(m = 4)$ and GEBB $(n = 8)$ on Figure 6 is located at a point pro-

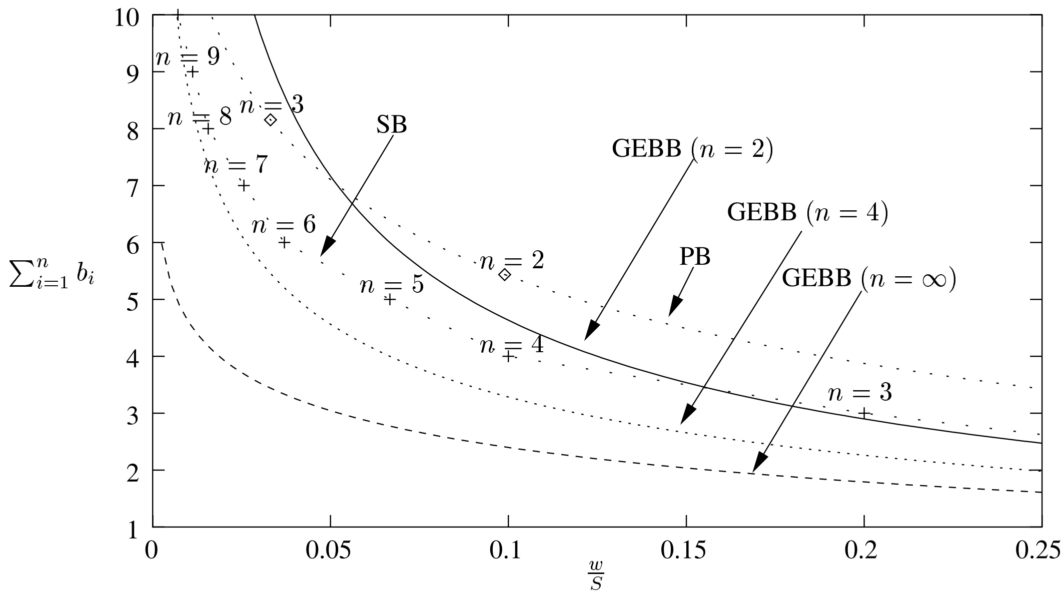

Figure 5. Comparison of bandwidth requirements for PB ($\alpha = e$), SB and GEBB.

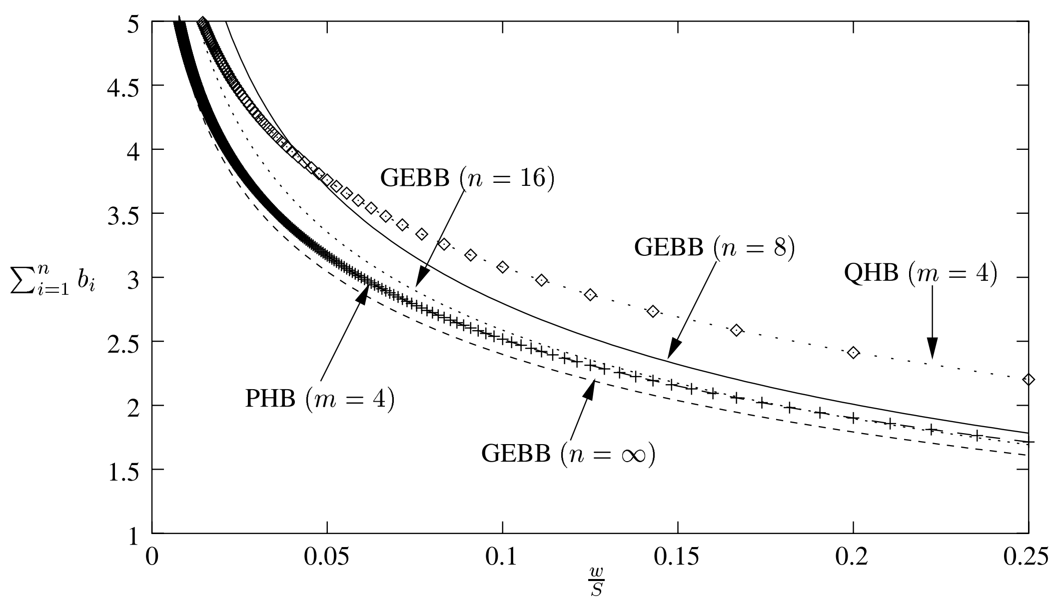

Figure 6. Comparison of bandwidth requirements for QHB ($m = 4$), PHB ($m = 4$) and GEBB.

viding a latency fraction of approximately 0.04 but requires that the QHB uses approximately 25 channels, compared to the eight channels of GEBB. Similarly, the QHB line intersects with GEBB ($n = 16$) for approximately $n = 100$ QHB channels. Note that by increasing m, QHB will not outperform GEBB but will converge to the bandwidth value which is equal to the sum of the harmonic series up to its n–th term [5].

The comparison of GEBB to PHB brings the very interesting observation that PHB's performance is essentially close to optimal. GEBB and PHB agree in the asymptotic sense for $n \to \infty$ but the fixed size segments of PHB result in more segments being necessary to achieve the same playout latency. For example, the intersection of the lines PHB ($m = 4$) and GEBB ($n = 16$) on Figure 6 is located at a latency fraction of approximately 0.2 which corresponds to a value of $n = 20$ for PHB. PHB's bandwidth demands can improve with multiplying m by a factor. However, in order to maintain the same playout latency, n must be multiplied by the same factor that m is multiplied. Therefore, in PHB, improved latency comes again at the cost of increased number of segments. Finally, note that even though the situation is better compared to EB schemes, QHB and PHB still couple the playout latency to the number of segments (on Figure 6, only the points marked on the QHB and PHB lines are defined by the corresponding scheme, the dots and dashes are shown only to provide visual continuity).

5. Conclusions

In this paper we have explored the principles behind the two families of protocols for periodic broadcasting in VoD systems, namely, the ones that descend from Pyramid Broadcasting and the ones that descend from Harmonic Broadcasting. We attempted to pick the most flexible scheme in terms of timing relations and to formulate it in a way that allows the construction of a bandwidth–efficient broadcast scheme. We were able to produce a well-defined optimization problem which was solved exactly. The solution of this problem allows the construction of periodic broadcast schedules that can attain any prescribed client latency with the minimum possible bandwidth. A comparison with pre-existing schemes demonstrated the advantage of the scheme, with respect to bandwidth demands, as well as with respect to implementation complexity (smaller number of segments, n).

We are currently extending the presented scheme to VBR video periodic broadcast. Another direction for future research is the inclusion of client constraints to the optimization problem. We are also exploring avenues for the problem of splitting available bandwidth capacities to "hot" and "cold" movie sets and determining the best combination of broadcast/multicast scheduling algorithm that can be used to administer these two sets.

References

[1] S. Viswanathan and T. Imielinski, "Pyramid broadcasting for video on demand service," *IEEE Multimedia Computing and Networking Conference,* pp. 66-77, San Jose, California, 1995.

[2] C. C. Aggarwal, J. L. Wolf, and P. S. Yu, "A permutation-based pyramid broadcasting scheme for video-on-demand systems," *Proc. of the IEEE Int'l conf. on Multimedia Systems '96*, Hiroshima, Japan, June 1996.

[3] 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.

[4] L. Juhn and L. Tseng, "Harmonic broadcasting for video-on-demand service," *IEEE Transactions on Broadcasting*, vol. 43, no. 3, pp. 268-271, September 1997.

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

[6] J.-F. Paris, S. W. Carter, and D. D. E. Long, "A low bandwidth broadcasting protocol for video on demand," *Proc. 7th Int. Conference on Computer Communications and Networks (IC3N'98)*, pp. 690–697, October 1998.