Seamless Workload Adaptive Broadcast

Yang Guo, Lixin Gao†, Don Towsley, and Subhabrata Sen‡

Computer Science Department
University of Massachusetts
Amherst MA 01003
{yguo, towsley}@cs.umass.edu

†ECE Department
University of Massachusetts
Amherst MA 01003
lgao@ecs.umass.edu

‡AT&T Labs-Research
Florham Park, NJ
sen@research.att.com

Abstract

The high-bandwidth requirements and long-lived characteristics of digital video make transmission bandwidth usage a key limiting factor in the widespread streaming of such content over the Internet. A challenging problem is to develop bandwidth-efficient techniques for delivering popular videos to a large, asynchronous client population with time-varying demand characteristics. In this paper we propose *seamless workload adaptive broadcast* to address above issues. Seamless workload adaptive broadcast is based on Fibonacci Periodic Broadcast (FPB) scheme. By introducing a feedback control loop into the FPB scheme, and enhancing the FPB scheme by techniques such as parsimonious transmission and instantaneous playback, the seamless workload adaptive broadcast provides instantaneous or near-instantaneous playback services and it can adapt to workload changes seamlessly. Furthermore, the FPB scheme proposed in this paper is bandwidth efficient and has the periodic seamless channel transition property.

I. Introduction

The high-bandwidth requirements and long-lived characteristics of digital video make transmission bandwidth usage a key limiting factor in the widespread streaming of such content over the Internet. For high-demand content, clients asynchronously issue requests to receive their chosen media streams. In addition, the demand for a particular video can vary over time, due to time-of-day (week) effects, changing popularity, etc. A challenging problem is to develop bandwidth-efficient techniques for delivering popular videos to such a large, asynchronous client population with time-varying demand characteristics. In this paper we report on the design and evaluation of such a delivery scheme.

Various techniques have been developed to reduce server and network bandwidth associated with delivering a popular video to asynchronous clients, by allowing multiple clients to receive all, or part of, a single transmission. Periodic Broadcast (PB) schemes [1–4] divide a video object into multiple segments, and continuously broadcast the segments on a set of transmission channels. Using a constant number of channels, PB can provide streaming video with a pre-determined playback startup delay to an arbitrary number of clients. Other proposed techniques, such as patching [5–7] and stream merging [8–10], are not as bandwidth efficient as PB when the client arrival rate is high.

Research on Periodic Broadcast has focused on further improving its efficiency – to reduce the server bandwidth requirement with a pre-determined playback delay while keeping the client side resource requirement, such as clients' receiving bandwidth or workahead buffer size, low. However the Periodic Broadcast schemes proposed so far exhibit following drawbacks:

- *Workload insensitivity*. A Periodic Broadcast scheme is essentially an open loop scheme that doesn't adapt to changing workload. PB transmits all segments and uses the same amount of bandwidth regardless of the demand for the video. PB is designed to serve popular videos. However in reality the videos' popularity changes over time. Furthermore, there are cold and lukewarm videos, whose popularity often cannot be determined in advance.
- *Delayed playback*. Clients in Periodic Broadcast experience a playback delay, which can be significant if the number of channels used is small.

A broadcast scheme that can adapt to the workload and offer instantaneous, or near-instantaneous playback is desirable. In this paper, we propose *seamless workload adaptive broadcast* to address above issues. Seamless workload adaptive broadcast is based on the *workload adaptive broadcast architecture* and the *Fibonacci PB (FPB)*. A workload adaptive broadcast architecture is centered around a PB scheme, with the addition of following techniques.

- *Parsimonious transmission*. The server sends out segments only when a client is prepared to receive them.
- *Workload adaption*. Addition of a control loop into Periodic Broadcast helps Periodic Broadcast adapt to the workload. The server collects client arrival information, and then dynamically adjusts the number of channels used in PB to minimize the overall bandwidth usage.
- *Instantaneous playback*. This technique enables the instantaneous or near-instantaneous playback in workload adaptive broadcast.

We introduce the Fibonacci Period Broadcast (FPB) scheme, which is especially suitable for workload adaptive broadcast architecture. Here Fibonacci refers to the fact that the sizes of the video clip delivered over the channels are proportional to the Fibonacci numbers. A PB exhibits the seamless transition property if it can periodically change the number of channels without disrupting existing clients' reception and without any additional channels. FPB exhibits the *seamless transition property* and is as bandwidth efficient as other similar PB schemes. Seamless transition property ensures the possibility to adjust the number of server channels smoothly. Thus FPB is especially suitable for workload adaptive broadcast. We also derive a recursive formula to calculate the average bandwidth usage, which can be used to determine when to add or remove server channels.

Simulation study shows that FPB is more bandwidth efficient than other PB schemes with the same client side network bandwidth requirement. Parsimonious FPB saves network bandwidth when the client request rate is low. Finally, we show that the seamless workload-adaptive scheme serves the video with changing popularity well. The bandwidth usage in seamless workload-adaptive broadcast scheme is proportional to the workload.

The remainder of paper is organized as follows. In Section 2, we describe the architecture of workload adaptive broadcast. In Section 3, we present the seamless workload-adaptive scheme. Section 4 is dedicated to performance evaluation. Conclusions and future work are included in Section 5.

II. Workload Adaptive Broadcast Architecture

In this section, we describe the workload adaptive broadcast architecture. Workload adaptive broadcast architecture is centered around a Periodic Broadcast scheme coupled with additional features such as parsimonious transmission, dynamic channel adjustment, and instantaneous or near-instantaneous playback. These features enable it to perform well even under the changing workloads.

Fig. 1 depicts the architecture of the workload adaptive broadcast. The server consists of two components: a modified PB scheduler and a workload adaptor. Below we describe them respectively.

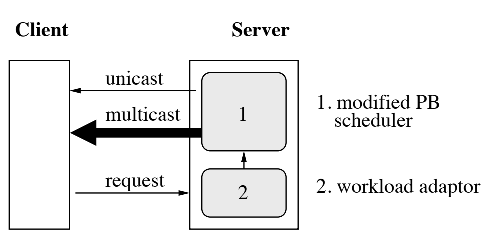

Fig. 1. Workload adaptive broadcast architecture

- **Modified PB scheduler**. This component serves the video using a modified PB scheme that (1) provide instantaneous playback services; and (2) save the network bandwidth by only transmitting segments that are needed by clients (parsimonious transmission).

To illustrate how instantaneous playback is achieved, suppose a video clip is divided into equal size segments. Fig. 2 illustrates an example where the video is divided into 11 equal size segments. When a client arrives, segment A is unicast to the client immediately to enable instantaneous playback. The client then starts to receive the data from the modified PB scheme, which use one more channel for segment B. Since segments A and B are of the same size, the client can start to receive segment B while playing back segment A. Moreover, since segment B and segment 1 are of the same size, they can be received sequentially. The above device helps to achieve the instantaneous playback while the client still listens to the same number of server channels as before. For instance, in Fig. 2, the client 1 receives segment A immediately, and then receives shadowed segment B, and backslashed segment 1, etc. from multicast channels.

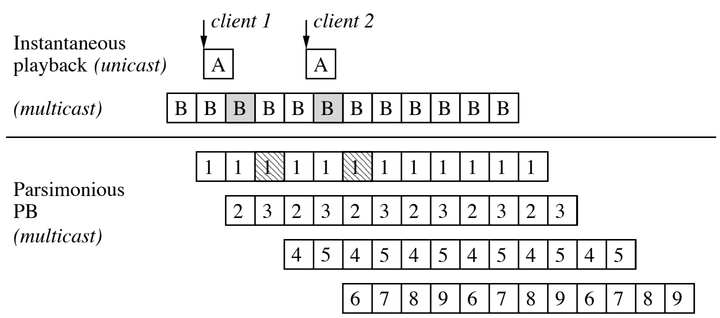

Fig. 2. Modified PB scheme

Considering the network's propagation delay and client side's buffering delay, a short delay on the order of seconds can be treated as near-instantaneous playback. If a small playback delay is allowed, the component can batch the requests and multicast the segment A, which can further reduce the bandwidth usage. In the following discussion, we always assume the instantaneous playback is required. The workload adaptive broadcast architecture applies to both instantaneous and near-instantaneous playback scheme.

• **Workload adaptor**. The workload adaptor collects the client arrival rate information, and determine the number of channels required by the modified PB scheduler in order to minimize the overall bandwidth usage.

We use an exponential smoothing algorithm to estimate the average client request rate. The number of arrivals is collected periodically. Denote by $\hat{\lambda}_n$ as the arrival rate after the n-th update period, then

$$\hat{\lambda}_{n+1} = \omega\hat{\lambda}_n + (1-\omega)A_n/\Delta. \tag{1}$$

where A_n is the number of arrivals during the n-th period and Δ is the period length. The weight, ω, and the update rate, $1/\Delta$, determine how quickly the average arrival rate converges to the current arrival rate.

The number of channels used by the parsimonious PB in modified PB scheduler, denoted by K, is determined by the average arrival rate. Let $B(K, L, \lambda)$ be the average bandwidth usage of workload adaptive broadcast where K is the number of channels allocated and L is the length of video clip. The workload adaptor chooses the number of channels, K^*, so as to minimize the overall average bandwidth usage, i.e., $K^* = \text{argmin}_K B(K, L, \lambda)$. If a change in number of channels is necessary, the adaptor asks the modified PB scheduler to do the change accordingly. During the transition period, it is desirable that clients not be disrupted and the service of other videos not be affected. In the next section, we will introduce the FPB scheme, which provides the property of smooth channel transition.

III. Seamless workload adaptive broadcast

In this section we first present FPB, an efficient PB scheme that exhibits the seamless transition property. We then describe the seamless workload adaptive broadcast using the FPB.

A. Fibonacci Periodic Broadcast (FPB)

The FPB scheme has the segmentation series of $1, 2, 3, 5, 8, \ldots$, the Fibonacci series excludes the first two numbers, 0 and 1. Channel $n, n = 1, 2, 3, \ldots$, is responsible for delivering consecutive F_n segments to clients, where F_n is defined as

$$\begin{cases} F_0 = F_1 = 1 \\ F_{n+2} = F_{n+1} + F_n, \quad (n = 1, 2, 3, \cdots) \end{cases} \tag{2}$$

Below we describe the server's transmission schedule and client's reception schedule respectively.

A.1 Server's broadcasting schedule

Suppose the FPB uses K channels to transmit a video clip of length L. The n-th channel, $1 \leq n \leq K$, is responsible for delivering F_n segments to clients, from segment $\sum_{i=1}^{n-1} F_i + 1$ to segment $\sum_{i=1}^{n} F_i$. We use $[\sum_{i=1}^{n-1} F_i + 1,\ \sum_{i=1}^{n} F_i]$ to represent these consecutive segments. The FPB scheme consists of a start rule, a repeat rule, and a transmission pattern within a period.

Start rule. The transimission of channel 1 starts first. The n-th channel starts transmission after the $(n-1)$-th channel completes the transmission of F_{n-1} segments.

Repeat rule. In FPB scheme, each channel repeats its transmission pattern once every F_K segments. We call F_K the *period* of the FPB scheme.

Transmission pattern within a period. The first channel repeatedly sends out the first segment F_K times. For channel $n, n = 2, 3, \cdots, K-1$, the transmission pattern is illustrated in Fig. 3. Channel n first transmits F_n segments, segment $[\sum_{i=1}^{n-1} F_i + 1,\ \sum_{i=1}^{n} F_i]$. It then transmits $K - n$ batches of segments, where batch i, $i = 1, 2, \cdots, K-n$, consists of F_{N-2+i} segments. Since $F_n + \sum_{i=n-1}^{K-2} F_i = F_K$, the period is F_K. Each batch repeats the segments that have been transmitted from the beginning of this period. Finally, the last channel, channel K, sends out the last F_K segments of the clip sequentially.

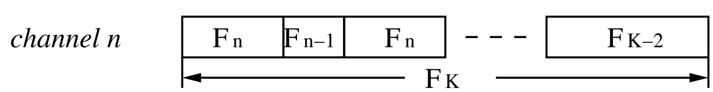

Fig. 3. Transmitting Pattern of Channel n (one period)

Figure 4 gives an example of FPB using six channels. The video clip is divided into $\sum_{i=1}^{6} F_i = 32$ segments. The period is $F_6 = 13$ segments. The transmission pattern is as described above. For instance, the third channel is responsible for transmitting segment $\{4, 5, 6\}$ to clients. It starts by sending out segment $\{4, 5, 6\}$, and then three batches, $\{4, 5\}$, $\{4, 5, 6\}$, and $\{4, 5, 6, 4, 5\}$, respectively. Each batch repeats the segments that has been transmitted from the beginning of this period. We call the collection of one period of K channels a *K-channel cluster* of FPB scheme. All K-channel clusters are identical and independent of each other. In fact clients that start to receive segments within a cluster only fetch the data from the same cluster. Therefore it suffices to describe the client's reception schedule in one cluster.

A.2 Client's reception schedule

The cluster exhibits recursive structure. For instance, the 6-channel cluster in Fig. 4 consists of a 5-channel cluster and a 4-channel cluster. The 5-channel and 4-channels are further subdivided into clusters.

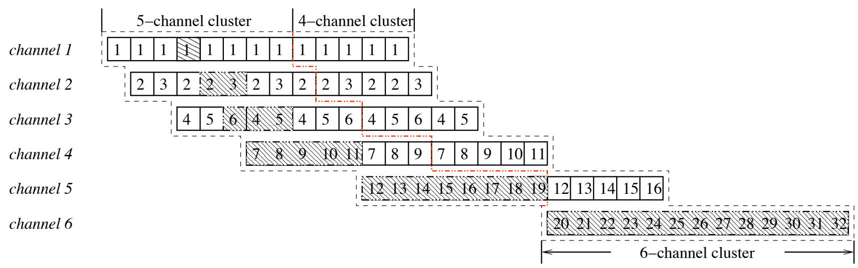

Fig. 4. A 6-channel Cluster and Its Sub-clusters in FPB

We explore the cluster's recursive structure in the FPB scheme, and present an algorithm that generates the client's reception schedule.

Reception schedule in 1-channel cluster. This is a trivial case. The client receives the first segment immediately.

Reception schedule in 2-channel cluster. Denote by T the starting time of the cluster, and by P the arrival time of the client. We use the segment length as the time unit. All clients arriving during a segment will be batched and served together at the starting time of the next segment. Hence we use the starting time of next segment as the arrival time of these clients.

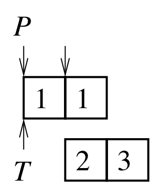

Fig. 5. A 2-channel Cluster

If $P = T$, clients receive the first segment 1, and continue to receive segment 2 and 3 from channel 2. If $P = T + 1$, clients receive the second segment 1, and simultaneously receives segment 2. Segment 3 is received after segment 2. In both cases, clients listen to at most two channels simultaneously.

Reception schedule in K-channel cluster **($K > 2$)**. Above we have shown that there is a valid reception schedule for a 1-channel and 2-channel cluster where clients listen to at most 2 channels simultaneously. Suppose there is a valid reception schedule for a $(K - 1)$-channel cluster and a $(K - 2)$-channel cluster. In the following we show that there exists valid reception schedule for a K-channel cluster. By induction, a valid reception schedule exists for an arbitrary cluster.

A K-channel cluster has two sub-clusters, a $(K - 1)$-channel cluster and a $(K - 2)$-channel cluster. If a client arrives during the first F_{K-1} segments, it receives the segments associated with the first $K - 1$ channels according to the reception schedule for the $(K-1)$-channel cluster. Once these have been received, the client receives the F_K segments associated with the K-th channel. Since the reception of segments from the K-th channel occurs after the reception from the first $K - 1$ channels finishes, clients listen to at most 2 channels.

If a client arrives during the F_{K-2} segments, it receives the segments associated with the $K - 2$ channels according to the reception schedule for the $(K - 2)$-channel cluster. Once these have been received, the client try to receive segments from $(K - 1)$-th and K-th channel, without violating the "listening to at most 2 channels" rule. The client can start to listen to the $(K-1)$-th channel once the $(K-3)$-th channel finishes the transmission at time $T + F_{K-2} - 2 + F_K$, where T is the starting time of the cluster. It can be shown that clients must receive all F_{K-1} segments from channel $K - 1$. The completing time of the $(K - 2)$-th channel coincide with the starting time of the K-th channel, thus the clients are able to fetch the last F_K segments from the K-th channel.

Pseudo-code for generating the client reception schedule is included in the Appendix [11]. As an example, suppose the client starts at 4th segment in channel 1 (backslashed segment in Fig. 4). Since it falls into a 5-channel cluster, the client receives 13 segments, $[20,\ 32]$, from channel 6. Within the 5-channel cluster, the 4th segment belongs to the 4-channel cluster, thus it receives 8 segments, $[12,\ 19]$, from channel 5. Within this 4-channel cluster, the 4th segment belongs to the later 2-channel cluster, instead of the leading 3-channel cluster. Thus it receives 5 segments, $[7,\ 11]$, from channel 4, and 3 segments from channel 3, in the order of segment $6, 4$, and 5. Since the 4th segment is the first segment in a 2-channel cluster, the client obtains the first segment immediately, and segments 2 and 3 in the following slot. The segments received by this client are marked as backslashed segments in Fig. 4. Note that if clients start the reception from the first segment in a cluster, they can receive the entire video listening to one channel at a time and no client-side buffer is required.

A.3 Seamless transition property

FPB exhibits the seamless channel transition property. Assume that it uses a fixed number, say K, of channels, and the newly assigned number of channels is K'. During the channel transition period, we require that (1) the clients already starting their service not experience any disruption during the transition; (2) the newly arrived clients use the FPB scheme with K' channels; (3) the total number of channels used during the transition period is no larger than $\max\{K, K'\}$. We call a transition satisfying the above conditions as a *seamless transition*.

A naive solution is to allocate another set of K' channels for newly arrived clients. The previous K channels are held until all old clients are served. The solution requires $K + K'$ channels and wastes the bandwidth during the transition period. Moreover, if the server supports multiple video clips, the channel transition can lead to a resource deadlock problem. We state the following result with the proof included in the Appendix [11].

Theorem 1: The seamless transition can be achieved at a cluster boundary in FPB scheme.

B. Seamless Workload Adaptive Broadcast

Seamless workload adaptive broadcast uses parsimonious FPB scheme in the modified PB scheduler. The channel transition is made at the boundary of each cluster when necessary. Below we first describe how to calculate the average bandwidth usage in seamless workload adaptive broadcast. We then show that the workload adaptor in seamless workload adaptive broadcast can do a simple table lookup to decide whether a channel transition is necessary. If it is necessary, the workload adaptor performs transition at the boundary of the cluster, which make the transition smooth.

Since two extra segments are needed to provide instantaneous playback (see Fig. 2), the segment size in seamless workload-adaptive broadcast, $g'(K, L)$, is

$$g'(K, L) \quad = \quad \frac{L}{\sum_{i=1}^{K} F_i + 2}. \tag{3}$$

On average there are $\lambda g'(K, L)$ arrivals during a segment. For each arrival, the modified PB scheduler transmits segment A, and, hence, the average number of transmitted segments from the modified PB scheduler in a period is $\lambda g'(K, L) F_K$. Denote by p the probability of an arrival in a segment. Suppose the client arrival process is Poisson with arrival rate λ. We have $p = 1 - e^{-\lambda g'(K,L)}$ for all segments because of the memoryless property of Poisson process. For the PB component, an average number of

pF_K copies of segment B are transmitted, Denote by $C(K,p)$ the average number of segments transmitted during a K-channel cluster by Parsimonious FPB. The average total number of segments transmitted is $\lambda F_K g^{'}(K,L) + C(K,p) + pF_K$. Hence the average number of busy channels, $B(K,\lambda,L)$, is

$$B(K,L,\lambda) = \lambda g^{'}(K,L) + p + \frac{C(K,p)}{F_K}. \tag{4}$$

We would like to choose a value of K, K^*, that minimizes the average number of busy channels, i.e., $K^* = \text{argmin} B(K,L,\lambda)$.

Theorem 2: In parsimonious FPB scheme, the average number of segments transmitted in a K-channel cluster, $C(K,p)$, satisfies the recursion

$$\begin{aligned} C(K,p) &= C(K-1,p) + C(K-2,p) + (1-p)^{F_{K-1}}(1-(1-p)^{F_{K-2}})F_{K-3} \\ &\quad +(1-(1-p)^{F_{K-2}})F_{K-2} + (1-(1-p)^{F_K})F_K \end{aligned} \tag{5}$$

for $K \geq 3$, where $C(1,p) = p$, and $C(2,p) = 6p - 2p^2$.

The proof is included in the Appendix [11]. No closed-form solution of K^* exists. Fig. 6 shows the average number of busy channels as a function of number of channels, K, for different client arrival rates. As the arrival rate increases and the number of channels increases, the average number of busy channels decreases substantially. However there exists a number of channels above which the improvement rapidly decreases.

Fig. 7 plots the optimal number of channels as the normalized workload varies. The optimal number of channels is defined as the least number of channels such that the bandwidth usage is within 1% of the minimum bandwidth usage. Here the normalized workload is the product of the client request arrival rate and the video length. The curve exhibits the staircase shape. We can determine the range of normalized workloads over which the same number of channels are required and represent it in a table. The workload adaptor can choose the number of server channels from the table to reduce bandwidth usage while maintaining its own schedule as well as clients' reception schedule as simple as possible. If a certain playback delay can be treated as near-instantaneous playback, the modified PB scheduler can batch the requests and multicast the segment A. In the following section, however, we assume instantaneous playback is desired. We expect a similar result when small playback delay is allowed.

IV. Performance Evaluation

We evaluate the seamless workload-adaptive broadcast scheme from the following three perspectives: (1) how FPB compares with other PB schemes; (2) whether and how much parsimonious FPB can save server bandwidth when the client request rate is low; and (3) how seamless workload adaptive broadcast adapts to changing video popularity. We show that FPB is more efficient than other popular PB schemes with the same client side network bandwidth requirement. Parsimonious FPB scheme uses less bandwidth when the client request rate is slow. To evaluate the seamless workload adaptive broadcast, we use a workload whose rate changes dramatically throughout a day. Simulation results show that seamless workload adaptive broadcast adapts to the changing workload nicely.

• **Comparison of PB schemes**. Fig. 8 illustrates the server bandwidth requirement (number of channels) vs. the startup delay represented in fraction of video length. We compare FPB with dynamic skyscraper, skyscraper, and GDB3, which all require clients listen to 2 channels simultaneously. For the same amount of playback delay, FPB uses less bandwidth than other schemes.

• **Efficiency of Parsimonious FPB**. Fig. 9 shows the average number of busy channels vs. the client request rate. Three curves corresponds to the cases where $7, 10$, and 20 server channels are used, respectively. As the client request rate increases, the average number of busy channels also increases. The reason behind

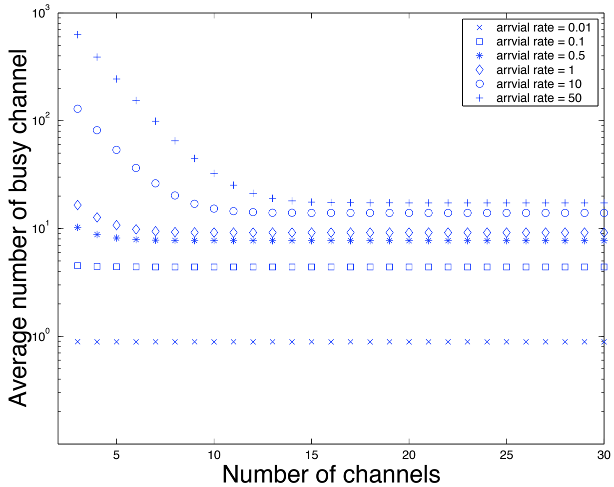

Fig. 6. Average Number of Busy Channels vs. Number of Channels with Different Client Workload

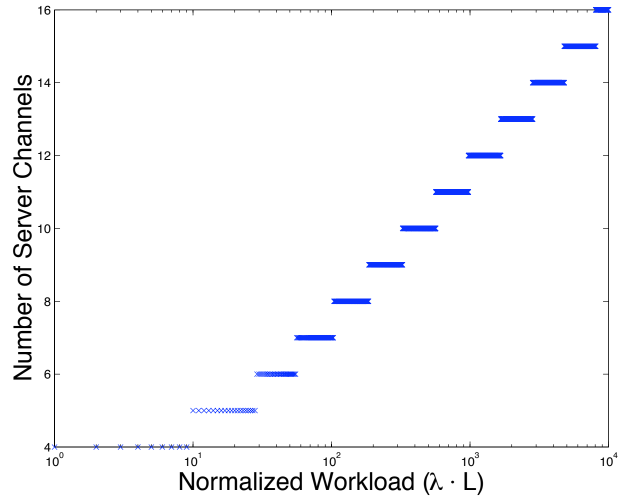

Fig. 7. Optimal Number of Server Channels vs. Normalized Workload

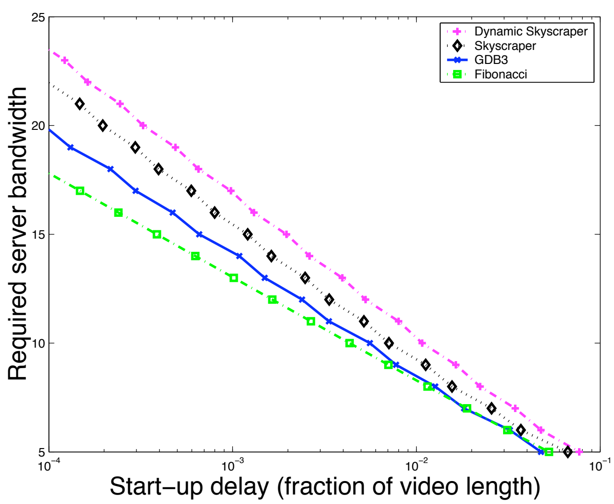

Fig. 8. Comparison of PB schemes (requiring clients listen to 2 channels simultaneously)

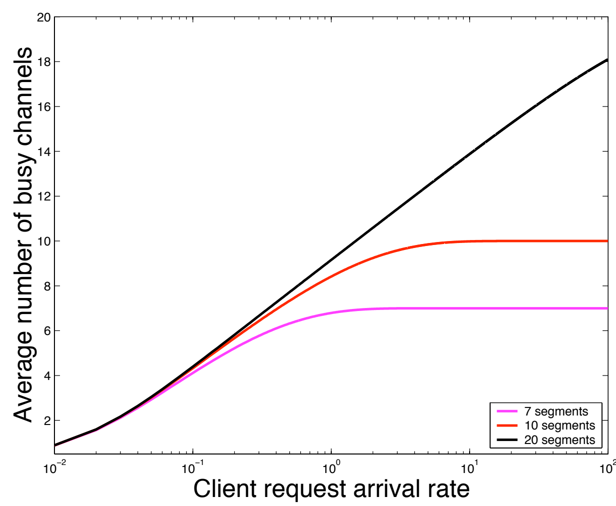

Fig. 9. Average Number of Busy Channels vs. Client Arrival Rate in Parsimonious FPB

this is that more segments are transmitted as more requests arrive. Eventually, all segments need to be sent out, and all channels assigned to the PB scheme are used. Thus the number of busy channels reaches the number of channels assigned to the PB scheme.

• **Performance of seamless workload adaptive broadcast**. Finally we investigate the performance of the seamless workload adaptive scheme. Fig. 10 depicts the client arrival rate during 24-hour period. The arrival process is Poisson with a time varying rate. During peak hour (from 10am to 4pm), the rate is around 15 requests/min, while during off-peak hour, the rate is around 0.3 requests/min. The dotted line is the estimated client arrival rate from the workload estimator. The exponential smoothing average algorithm (formula (1)) is used to keep track of the client arrival rate. In this experiment, w is set to be 0.1. The average client arrival rate is updated once every minute. We can see that the workload estimator does a nice job keeping track of the actual arrival rate, filtering out the short-term rate change.

The seamless workload adaptive broadcast choose the number of channels used in FPB component based on the estimated client arrival rate to minimize the required bandwidth to serve the clients. Fig. 11 shows

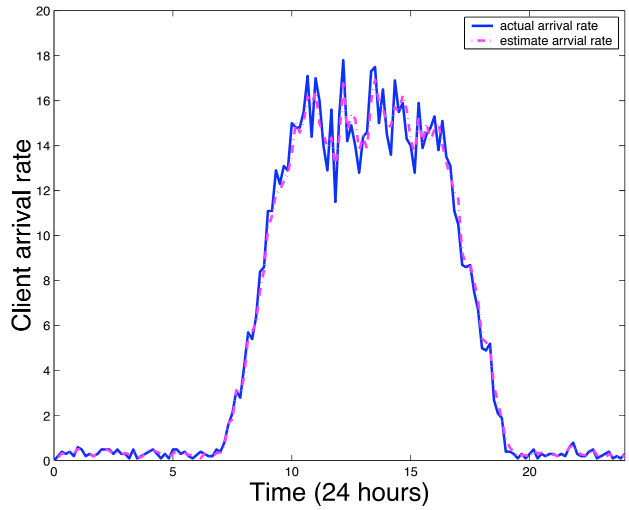

Fig. 10. Client arrival rate vs. time of the day

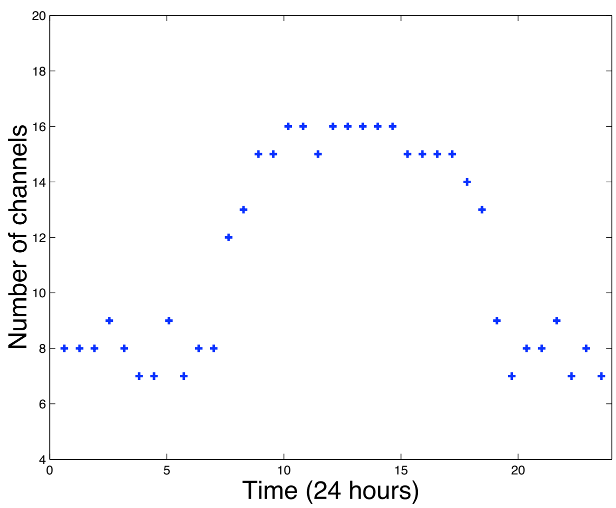

Fig. 11. Number of channels used in FPB vs. time of the day

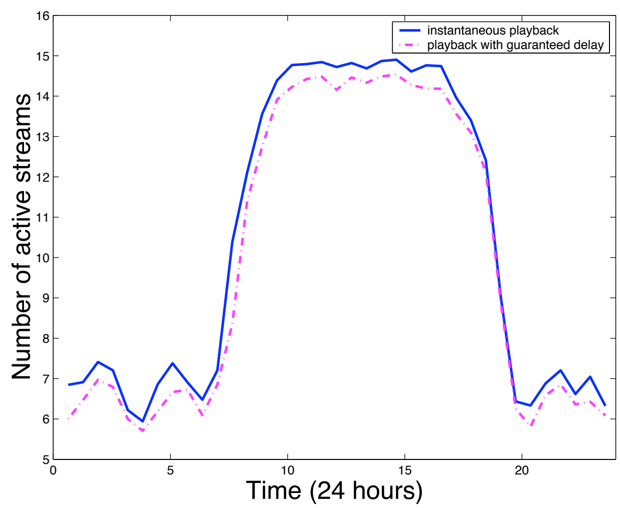

Fig. 12. Number of active streams vs. time of the day

the number of channels chosen at different times during the day. More channels are used when the client arrival rate is high. The shape of the curve resembles the client arrival rate process.

Fig. 12 depicts the number of active channels (server bandwidth usage) over the time. The bandwidth usage in seamless workload adaptive broadcast is proportional to the workload. The seamless workload adaptive broadcast adapts to the workload by adjusting the number of channels used in the PB component. The dashed line in the figure is the bandwidth consumed by the PB component alone (not including the bandwidth used for instantaneous playback). The difference between the solid and dashed lines is the bandwidth required to provide instantaneous playback. There are more arrivals during the peak hour. The seamless workload adaptive broadcast increases the number of channels used in the peak hour, thus decreasing the size of segment. The result is that the bandwidth used to enable the instantaneous playback does not increase dramatically, which keep the overall bandwidth usage low.

V. Conclusions and Future Work

[12] and [4] also propose the PB scheme that has the Fibonacci series as segmentation series. From a server bandwidth efficiency point of view they are equivalent to the FPB scheme proposed in this paper. However both the server-side broadcasting and the client-side reception schedules are quite different. The FPB scheme exhibits the seamless channel transition property which is essential for seamless workload adaptive broadcast. Also the FPB scheme can support clients with few resources by scheduling them at the beginning of a cluster.

The Seamless channel transition property was first studied in [13], where channel transitions are possible at the boundary of each segment. However the scheme proposed in [13] requires the client to be able to listen to the same number of channels as that used by the server.

In this paper we propose the workload adaptive broadcast architecture and seamless workload adaptive broadcast based on FPB scheme to provide VoD service to a large, asynchronous client population with time-varying workload. By introducing the feedback control loop into the PB scheme, and enhancing the PB scheme by the techniques such as parsimonious transmission and instantaneous playback technique, the seamless workload adaptive broadcast provides instantaneous or near-instantaneous playback service and it can adapt to workload changes seamlessly. Simulation experiments show that the required bandwidth is proportional to the client request rate. The FPB scheme proposed in this paper is bandwidth efficient and has the seamless channel transition property.

Future research can proceed along several avenues. Workload adaptive broadcast is a framework that can be used by many PB schemes. We would like to explore the possibility of using other PB scheme in workload adaptive broadcast architecture. Secondly, it will be interesting to compare the performance with other delivery schemes such as patching and stream merging schemes. Finally, implementation of seamless workload adaptive broadcast in the test-bed can further help us to evaluate the scheme in a practical setting.

References

[1] K. Hua and S. Sheu, "Skyscraper broadcasting: A new broadcasting scheme for metropolitan video-on-demand systems," in *Proc. ACM SIGCOMM*, September 1997.

[2] L. Gao, D. Towsley, and J. Kurose, "Efficient schemes for broadcasting popular videos," in *Proc. Inter. Workshop on Network and Operating System Support for Digital Audio and Video*, July 1998.

[3] A. Hu, "Video-on-demand broadcasting protocols: A comprehensive study," in *Proc. IEEE INFOCOM*, April 2001.

[4] A. Mahanti, D. L. Eager, M. K. Vernon, and D. Sundaram-Stukel, "Scalable on-demand media streaming with packet loss recovery," in *Proc. SIGCOMM 2001*, August 2001.

[5] S. Carter and D. Long, "Improving video-on-demand server efficiency through stream tapping," in *Proc. International Conference on Computer Communications and Networks*, 1997.

[6] K. Hua, Y. Cai, and S. Sheu, "Patching: A multicast technique for true video-on-demand services," in *Proc. ACM Multimedia*, September 1998.

[7] L. Gao and D. Towsley, "Supplying instantaneous video-on-demand services using controlled multicast," in *Proc. IEEE International Conference on Multimedia Computing and Systems*, 1999.

[8] D. Eager, M. Vernon, and J. Zahorjan, "Bandwidth skimming: A technique for cost-effective video-on-demand," in *Proc. SPIE/ACM Conference on Multimedia Computing and Networking*, January 2000.

[9] A. Bar-Noy and R. E. Ladner, "Competitive on-line stream merging algorithm for media-on-demand," in *Proc. of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, January 2001.

[10] E. C. Jr., P. Jelenkovic, and P. Momcilovic, "The dyadic stream merging algorithm," in *Proc. of Web Caching and Content Distribution*, June 2001.

[11] Y. Guo, L. Gao, D. Towsley, and S. Sen, "Seamless workload adaptive broadcast," tech. rep., Department of Computer Science, University of Massachusetts Amherst. http://www-net.cs.umass.edu/ yguo/seamless.ps, 2001.

[12] J.-F. Paris and D. Long, "Limiting the receiving bandwidth of broadcasting protocols for video-on-demand," in *Proceedings of the Euromedia 2000 Conference*, May 2000.

[13] Y. Tseng, C. Hsieh, M. Yang, W. Liao, and J. Sheu, "Data broadcasting and seamless channel transition for highly-demanded videos," in *IEEE INFOCOM 2000*, pp. 727–736, March 2000.

APPENDIX

Note: The Appendix is included in the current submission for the reviewers' perusal and will be excluded from the final version of the paper.

Proof of theorem 1: Suppose the FPB uses K channels first, and then attempts to transit to $K^{'}$ channels at time t, the boundary of a cluster. Let L be the length of a video clip, and $g(K, L)$ the size of a segment in FPB using K channels. Since $\sum_{i=1}^{K} F_i = F_{K+2} - 2$, We have

$$g(K, L) = L/(F_{K+2} - 2). \tag{6}$$

Denote by $t_k, 1 \leq k \leq K$, the time when the channel k in old cluster becomes available, and $s_k, 1 \leq k \leq K^{'}$, the time when the server starts to use channel k in new cluster. According to starting rule, we have

$$t_k = t + (F_{k+1} - 2)g(K), \quad 1 \leq k \leq K. \tag{7}$$

and

$$s_k = t + (F_{k+1} - 2)g(K^{'}), \quad 1 < k \leq K^{'}. \tag{8}$$

We call vector $[t_1 \; t_2 \; \cdots t_K]$ the switch-out vector and vector $[t_1 \; t_2 \; \cdots, t^{'}_K]$ the switch-in vector. In order to achieve the seamless transition, the cluster using $K^{'}$ channels must not overlap with the cluster using K channels. Below we prove that the seamless transition is true for both $K^{'} < K$ case and for $K^{'} > K$ case.

(1) $K^{'} < K$ (see Fig. 14). Since $K^{'} < K, g(K^{'}) > g(K)$. Thus $s_i \geq t_i$ for all $i, 1 \leq i \leq K^{'}$. Hence there is no overlap between two clusters and seamless transition can be achieved in this scenario.

(2) $K^{'} > K$ (see Fig. 15). Let $N = K^{'} - K$. Since $K^{'} > K$, the first N channels in $K^{'}$-channel cluster can use N idle channels. Therefore it is sufficient to prove

$$s_{i+N} \geq t_i. \tag{9}$$

for all $i, 1 \leq i \leq K$. Substitution of Equation(6), (7), and (8) into (9) yields

$$\begin{aligned} t + (F_{i+N+1} - 2)\frac{L}{F_{K'+2} - 2} \quad &\geq \quad t + (F_{i+1} - 2)\frac{L}{F_{K+2} - 2} \\ \frac{F_{K+2} - 2}{F_{K+N+2} - 2} \quad &\geq \quad \frac{F_{i+1} - 2}{F_{i+N+1} - 2} \end{aligned} \tag{10}$$

where $N \geq 1, 1 \leq i \leq K$.

Lemma 1: $\{\frac{F_j - 2}{F_{j+1} - 2}, \;\; j \geq 2\}$ is a monotonically non-decreasing sequence.

Proof (sketch): It is equivalent to prove,

$$\frac{F_{j+1} - 2}{F_{j+2} - 2} \geq \frac{F_j - 2}{F_{j+1} - 2} \tag{11}$$

for all $j \geq 2$. Since Fibonacci number F_j is

$$F_j = \frac{1}{\sqrt{5}}\Big[(\frac{1 + \sqrt{5}}{2})^{j-1} - (\frac{1 - \sqrt{5}}{2})^{j-1}\Big]. \tag{12}$$

We can verify Equation (11) by substituting F_j into Equation (12). The details are omitted for the brevity of the expression. ∎

Now we are ready to prove Equation (10).

$$\frac{F_{K+2} - 2}{F_{K+N+2} - 2} = \frac{F_{K+2} - 2}{F_{K+3} - 2} \cdot \frac{F_{K+3} - 2}{F_{K+4} - 2} \cdots \cdot \frac{F_{K+N+1} - 2}{F_{K+N+2} - 2}$$

Apply Lemma 1, we have

$$\begin{aligned} \frac{F_{K+2} - 2}{F_{K+N+2} - 2} \quad &\geq \quad \frac{F_{i+1} - 2}{F_{i+2} - 2} \cdot \frac{F_{i+2} - 2}{F_{i+3} - 2} \cdots \cdot \frac{F_{i+N} - 2}{F_{i+N+1} - 2} \\ &= \quad \frac{F_{i+1} - 2}{F_{i+N+1} - 2}. \end{aligned}$$

```
schedule(T, P, K)
if (K ≤ 2)
    if ( K = 1)
        receive 1 segment from 1st channel at P;
    else if (K = 2 & P = T)
        receive 1 segment from 1st channel at P;
        receive 2 segments from 2nd channel starting at P + 1;
    else if (K = 2 & P = T + 1)
        receive 1 segment from 1st channel at P;
        receive 2 segments from 2nd channel starting at P;
    return;
else
    if (P − T ≤ F_{K−1} )
        schedule(T,P, K − 1);
    else if (P − T > F_{K−1} )
        schedule(T + F_{K−1}, P, K − 2);
        receive F_{K−1} segments from (K − 1)-th channel,
        starting at T + F_{K−2} − 2 + F_K;
    receive F_K segments from K-th channel, starting at T + F_{K+1} − 2;
    return;
```

Fig. 13. Pseudo-code for generating reception schedule for client arrives at P. The cluster starts at T and there are K channels in total

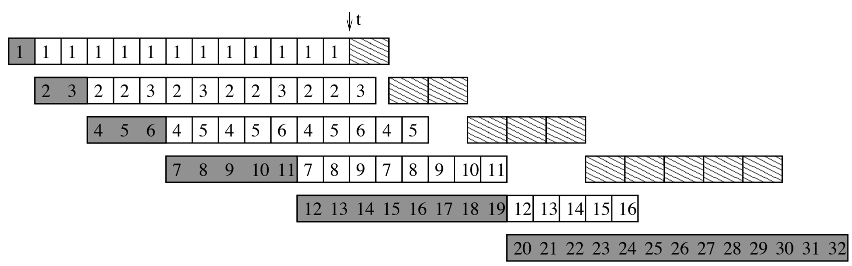

Fig. 14. Seamless Transition in FPB ($K^{'} < K$)

This completes the proof. ∎

Proof of theorem 2:
It is easy to see $C(1,p) = p$. For $C(2,p)$, there are four possible scenarios: (1) no arrival, (2) arrivals in both segments, (3) arrivals in first segment but no arrival in second segment, and (4) no arrivals in first segment but arrivals in second segment. We can calculate the probability of above four scenarios, and the corresponding number of segments needs to be transmitted in each scenario.

$$\begin{aligned} C(2,p) &= 4p^2 + 3p(1-p) + 3(1-p)p, \\ &= 6p - 2p^2. \end{aligned}$$

For $K \geq 3$, as described in Section III-A, K-channel cluster has two sub-clusters, $(K-1)$-channel cluster and $(K-2)$-channel cluster. This explains the first two terms on the right side of Equation (5).

According to the algorithm generating the reception schedule (Fig. 13), these two cluster are independent of each other except the last F_{K-3} segments in $(K-1)$-th channel of $(K-1)$-channel cluster (for instance, 3 backslashed segments in channel 5 in Fig. 16). The arrivals in second sub-cluster also need use these segments. Therefore if no arrivals exist in the first sub-cluster but there are arrivals in the second sub-cluster, these segments needs to be sent out. The probability that the above scenario occurs is $(1-p)^{F_{K-1}}(1-(1-p)^{F_{K-2}})$, and F_{K-3} segments needs to be transmitted. This interprets the third term on the right side of Equation (5).

The fourth term at the right hand side of Equation (5) is for the last F_{K-2} segments in $(K-2)$-th channel, which are solely used by the arrivals belonging to second sub-cluster. The probability that there is at least one arrival in this $(K-2)$-channel cluster is $1-(1-p)^{F_{K-2}}$.

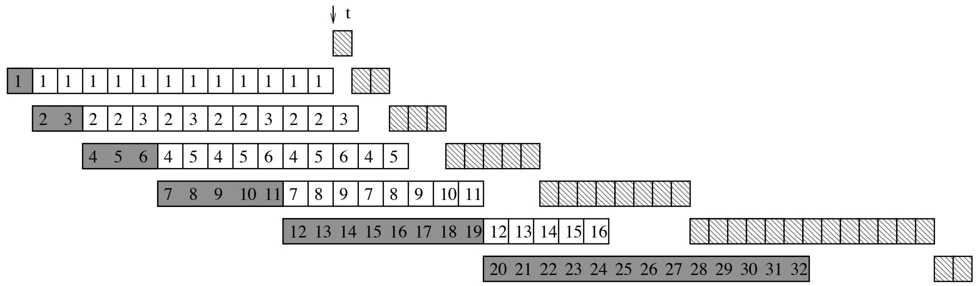

Fig. 15. Seamless Transition in FPB ($K^{'} > K$)

Finally, the F_K segments in K-th channel are shared by all arrivals in this cluster. The probability that there is at least one arrival within the cluster is $1 - (1 - p)^{F_K}$. This explains the last term at the right side of equation (5). ∎

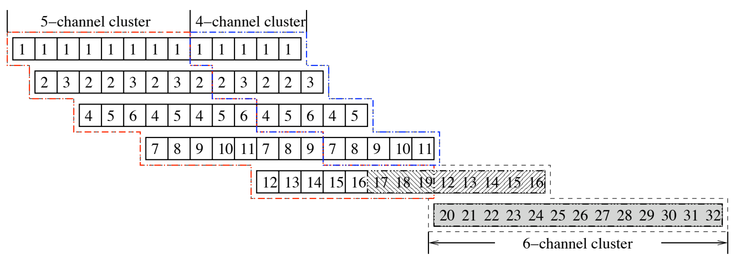

Fig. 16. A 6-channel Cluster and Its Two Sub-clusters: 5-channel Cluster and 4-channel Cluster.