# Smooth Workload Adaptive Broadcast [1]

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

‡Networking Research
AT&T Labs-Research
Florham Park, NJ 07932
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** ***smooth workload adaptive broadcast*** **to address the above issues. A key component of our scheme is Flexible Periodic Broadcast (FPB). By introducing a feedback control loop into FPB, and enhancing FPB using techniques such as parsimonious transmission, smooth workload adaptive broadcast provides instantaneous or near-instantaneous playback services and can smoothly adapt to workload changes. Furthermore, FPB, as proposed in this paper, is bandwidth efficient and exhibits the periodic smooth 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, a large number of 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 exhibiting 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 [1–5] divide a CBR video object into multiple segments, and continuously broadcast the segments on a set of IP multicast channels. Using a constant number of channels, PB can provide streaming video with a predetermined playback startup delay to an arbitrary number of clients. Other proposed techniques, such as patching and stream merging [6–11], are not as bandwidth efficient as PB when the client arrival rate is high.

Research on Periodic Broadcast has focused on 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 work-ahead buffer size, low. However Periodic Broadcast schemes proposed so far exhibit the following drawbacks:

- *Workload insensitivity*. A Periodic Broadcast scheme is essentially an open loop scheme that does not adapt to changing workload demands. 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 video popularity changes over time. Furthermore, the popularity of videos 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.

It is desirable to design a broadcast scheme that can adapt to the dynamically changing workload and offer instantaneous, or near-instantaneous playback. In this paper, we propose a technique called *smooth workload adaptive broadcast* to address the above issues. This scheme consists of two main components: *the workload adaptive broadcast architecture* and *Flexible PB (FPB)*. The workload adaptive broadcast architecture is centered around an arbitrary PB scheme, with the addition of following techniques.

- *Parsimonious transmission*. The server transmits a segment only if it is required by at least one client.
- *Workload adaption*. Addition of a control loop 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 instantaneous or near-instantaneous playback in workload adaptive broadcast.

We introduce the Flexible Period Broadcast (FPB) scheme, which is especially suitable for a workload adaptive broadcast architecture. A PB scheme exhibits the *smooth transition property* if it can periodically change the number of channels without disrupting existing clients' reception and without requiring any additional channels. FPB exhibits the smooth transition property and is as bandwidth efficient. The smooth transition property provides the op-

---

[1] This research was supported in part by the National Science Foundation under NSF grants EIA-0080119, ANI-9973092, ANI-9977635, ANI-0196039, ANI-0196038, and ANI-9875513. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

portunity to adjust the number of server channels seamlessly. 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 studies show 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 smooth workload-adaptive scheme adapts well to the changing request rate.

The remainder of the paper is organized as follows. In Section II, we describe the architecture of workload adaptive broadcast. In Section III, we present the flexible periodic broadcast (FPB). Section IV is dedicated to the smooth workload-adaptive broadcast scheme. Section V includes the performance evaluation. Finally, related work and conclusions are presented in Section VI and VII.

## II. WORKLOAD ADAPTIVE BROADCAST ARCHITECTURE

In this section, we describe the workload adaptive broadcast architecture. This architecture is centered around an arbitrary 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 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.

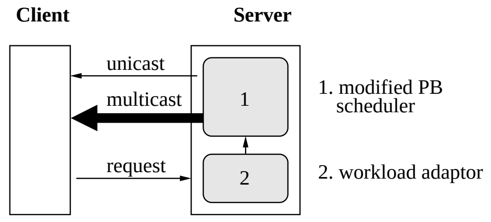


Fig. 1. Workload adaptive broadcast architecture

- **Modified PB scheduler**. Periodic Broadcast (PB) schemes divide a video into equal size segments, and continuously broadcast the segments on a set of transmission channels. Suppose a PB scheme uses $K$ channels, and channel $n$ is responsible to deliver $F_n$ segments. We denote $\{Fn\}_{n=1}^{K}$ to be the *segmentation series* of this PB scheme.

Modified PB scheduler modifies a PB scheme to provide instantaneous playback service without increasing client side bandwidth requirement, and to save the network bandwidth by only transmitting segments that are needed by clients (parsimonious transmission). We denote a PB scheme using parsimonious transmission as *parsimonious PB*.

To illustrate how instantaneous playback is achieved, suppose a video clip is divided into equal size segments. All but the first two segments (segment $A$ and $B$) are transmitted using the parsimonious PB scheme (see Fig. 2). When a client arrives, segment $A$ is unicast to the client immediately. The modified PB scheme uses 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 allows clients to achieve the instantaneous playback while listening 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 shaded segment 1, etc. from multicast channels.

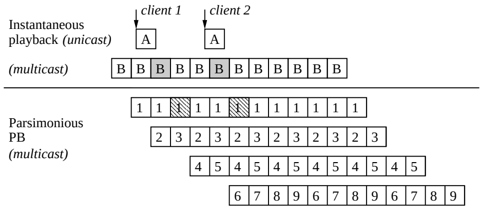


Fig. 2. Modified PB scheme

Note that if a small playback delay is acceptable, or the bandwidth is limited and many clients have to be delayed or rejected, the component can batch the requests and multicast segment A, which can further reduce the bandwidth usage. In the following discussion, we always assume that instantaneous playback is provided. The workload adaptive broadcast architecture applies to both instantaneous and near-instantaneous playback schemes.

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

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

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

where $A_n$ is the number of arrivals during the $n$-th period and $\Delta$ is the period length. The weight, $\omega$, 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 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 the 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 notifies the modified PB scheduler to make the change. 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. FLEXIBLE PERIODIC BROADCAST (FPB)

The Flexible PB scheme (FPB) has the segmentation series $1, 2, 3, 5, 8, \ldots$. This corresponds to the Fibonacci series with the first two numbers, 0 and 1, excluded. Channel $n, n = 1, 2, 3, \ldots$, is responsible for delivering consecutive $F_n$ segments to clients, where $F_n$ is defined as

$$\begin{cases} F_0 = F_1 = 1 \\ F_{n+2} = F_{n+1} + F_n, \quad (n = 0, 1, 2, 3, \cdots) \end{cases} \tag{2}$$

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

### A. *Server's broadcasting schedule*

Suppose that FPB scheme 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 schedule within a period.

**Start rule**. The transmission 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**. Each channel repeats its transmission schedule once every $F_K$ segments. We call $F_K$ the *period* of the FPB scheme.

**Transmission schedule 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 schedule comprises $K - n + 1$ batches of segments, where the first batch contains $F_n$ segments, and batch $i$, $2 \leq i \leq K{-}n{+}1$, contains $F_{n+i-3}$ segments, as illustrated in Fig. 3. Note that the total number of segments in these batches is $F_K$, $F_n + \sum_{i=n-1}^{K-2} F_i = F_K$.

Now we introduce the segments that are transmitted in each batch. In the first batch, the server sends out segment $[\sum_{i=1}^{n-1} F_i{+}1, \sum_{i=1}^{n} F_i]$. In the second batch, the segments that are the same as the leading $F_{n-1}$ segments in the first batch are transmitted. Batch $i$, $3 \leq i \leq K - n + 1$, consists of segments contained in the previous $i - 2$ batches, from batch 1 to batch $i - 2$.

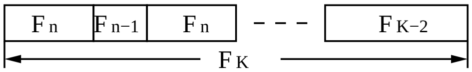


Fig. 3. Transmitting Pattern of Channel $n$ (one period)

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 $\{4, 5, 6\}$, followed by three batches, $\{4, 5\}$, $\{4, 5, 6\}$, and $\{4, 5, 6, 4, 5\}$, respectively. 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.

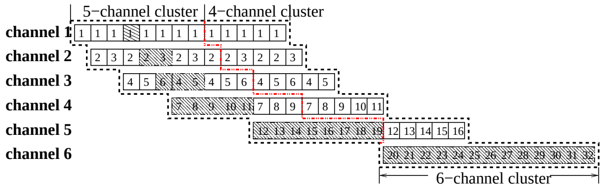


Fig. 4. A 6-channel Cluster and Its Sub-clusters in FPB

### B. *Client's reception schedule*

The cluster exhibits a 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. 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 start 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 the next segment as the arrival time of these clients.

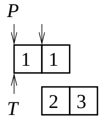


Fig. 5. A 2-channel Cluster

If $P = T$, clients receive the first instance of segment 1, and continue to receive segment 2 and 3 from channel 2. If $P = T + 1$, clients receive the second instance of 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 a 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 completes, 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 tries to receive segments from the $(K-1)$-th and $K$-th channels, 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 completion time of the $(K-2)$-th channel coincides 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 provided in Fig. 6, where $T$ is the cluster start time, $P$ is the client arrival time, and $K$ is the total number of channels used. As an example, suppose the client starts at the time of the 4th segment in channel 1 (backslashed segment in Fig. 4). Since it falls within 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.

### C. *Smooth transition property*

FPB exhibits the smooth channel transition property and can flexibly adjust the number of channels used. 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 a *smooth 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.

```
Schedule($T$, $P$,$K$)
if ($K \leq 2$)
    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 ($P - T \leq F_{K-1}$ )
        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. 6. Pseudo-code for generating reception schedule for client arrives at $P$. The cluster starts at $T$ and there are $K$ channels in total

*Theorem 1:* A smooth transition can be achieved at a cluster boundary in FPB scheme. ∎

## IV. SMOOTH WORKLOAD ADAPTIVE BROADCAST

Smooth workload adaptive broadcast uses the parsimonious FPB in the modified PB scheduler in its architecture (Fig. 1). 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 smooth workload adaptive broadcast. This is used to create a table of bandwidth usages indexed by the normalized workload, the product of video request rate and video length. The need for a channel transition is determined from a table lookup. If necessary, the workload adaptor performs a transition at the boundary of the cluster, leading to a smooth transition.

Denote by $g^{'}(K, L)$ the segment size in smooth workload-adaptive broadcast. Since two extra segments are needed to provide instantaneous playback (see Fig. 2), we have

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

We now focuses on the average number of busy channels in the modified FPB assuming the client arrival process is

Poisson with arrival rate $\lambda$. Since clusters in FPB are identical and independent of each other, we focus on a single $K$-channel cluster.

On average there are $\lambda g^{'}(K, L)$ arrivals during a segment. For each arrival, the modified PB scheduler transmits segment $A$. Hence, the average number of copies of segment $A$ transmitted in a cluster is $\lambda g^{'}(K, L) F_K$.

Denote by $p$ the probability of an arrival in a segment, $p = 1 - e^{-\lambda g^{'}(K,L)}$ because of the memoryless property of Poisson process. Hence an average number of $pF_K$ copies of segment $B$ are transmitted in a cluster. 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 $\lambda F_K g^{'}(K, L) + C(K, p) + pF_K$. Since the length of a $K$-channel cluster is $F_K$, the average number of busy channels, $B(K, \lambda, L)$, is

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

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

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

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

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

No closed-form expression for $K^*$ exists. We numerically compute $B(K, L, \lambda)$ for $1 \leq K \leq 30$ using Formula (3), (4) and (5), and select $K^*$ that minimizes the average number of busy channels. For Fibonacci series, the sum $\sum_{n=1}^{n=K} F_n$ grows exponentially as $K$ increases. Hence the number of segments that a video needs to be divided into increases exponentially. When $K = 30$, a 100 minute long video will be divided into $\sum_{n=1}^{n=30} F_n = 2,178,309$ segments. Therefore we think it is sufficient to only consider $1 \leq K \leq 30$.

Fig. 7 plots the optimal number of channels as a function of normalized workload. The optimal number of channels is defined to be 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 a 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.

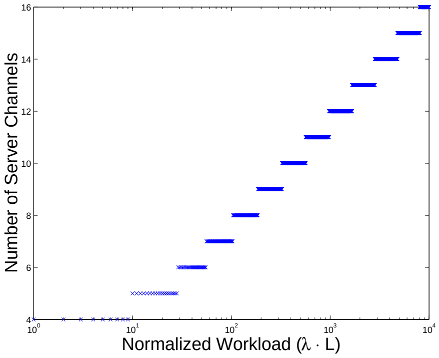


Fig. 7. Optimal Number of Server Channels vs. Normalized Workload

## V. PERFORMANCE EVALUATION

We evaluate the smooth 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 smooth 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 low. To evaluate the smooth workload adaptive broadcast, we use a workload whose rate changes dramatically throughout a day. Simulation results show that smooth workload adaptive broadcast adapts nicely to the changing workload.

- **Comparison of PB schemes**. Fig. 8 illustrates the server bandwidth requirement (number of channels) vs. the startup delay represented in fraction of video length. For instance, if the length of video is 100 mins and the startup delay is 0.001 of video length, it takes clients at most 6 seconds to start the playback. We compare FPB with dynamic skyscraper, skyscraper, and GDB3, which all require clients listen to 2 channels simultaneously. FPB uses less bandwidth than other schemes, particularly for short startup delays.
- **Efficiency of Parsimonious FPB**. Fig. 9 shows the average number of busy channels vs. client request rate (requests/min). The length of video is 100 minutes. 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 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 smooth workload adaptive broadcast**. Finally we investigate the performance of the smooth

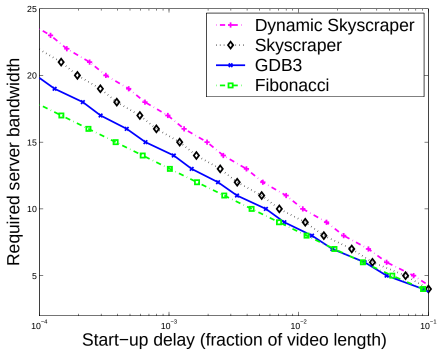


Fig. 8. Comparison of PB schemes (requiring clients listen to 2 channels simultaneously)

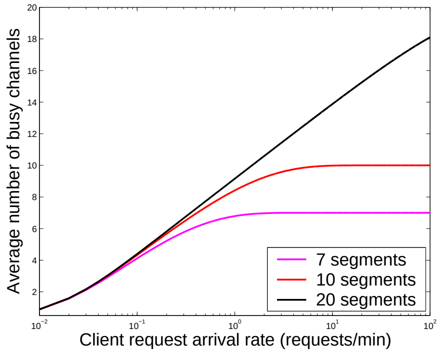
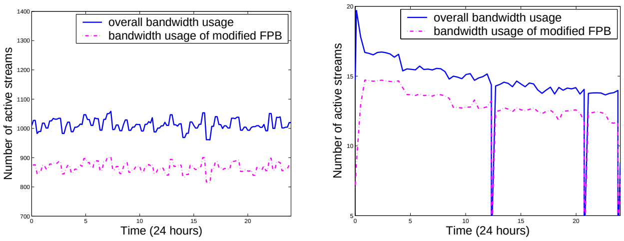


Fig. 9. Average Number of Busy Channels vs. Client Arrival Rate in Parsimonious FPB

workload adaptive scheme. Fig. 10(a) depicts the client arrival rate during 24-hour period. The arrival process is Poisson with a time varying rate. During peak hours (from 10am to 4pm), the rate is around 15 requests/min, while during off-peak hours, 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 smooth workload adaptive broadcast chooses the number of channels used in FPB based on the estimated client arrival rate to minimize the required bandwidth to serve the clients. Fig. 10(b) shows 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. 10(c) depicts the number of active channels (server bandwidth usage) over time. The bandwidth usage in smooth workload adaptive broadcast is proportional to the workload. Smooth 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. It is interesting to notice that the bandwidth usage for instant playback remains low even though there are more arrivals during the peak hour. Recall that server needs to transmit segment $A$ to each client and bandwidth usage should go up. The smooth workload adaptive broadcast increases the number of channels used in the peak hour, thus decreasing the size of a segment. The result is that the bandwidth used to enable the instantaneous playback does not increase dramatically; thus the overall bandwidth usage remains low.



(a) Aggregate bandwidth usage (b) Bandwidth usage for a single video

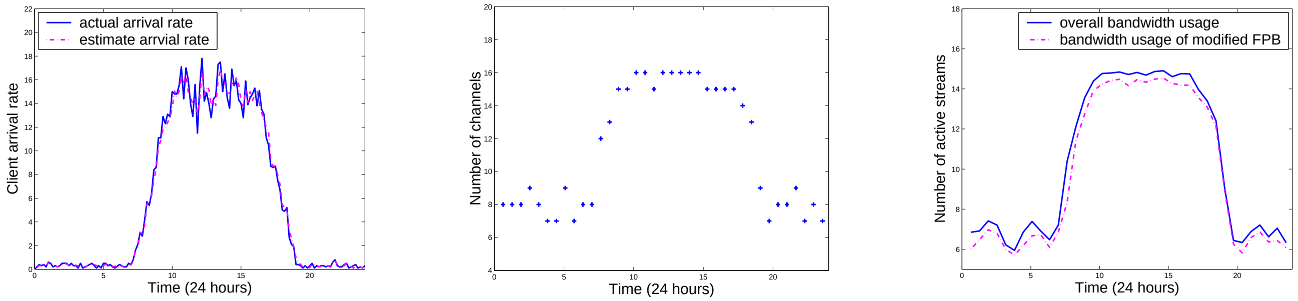
Fig. 11. Performance of the smooth workload adaptive scheme: multiple video case

We further evaluate the performance of smooth workload adaptive broadcast when it supports a collection of 100 videos of equal length. The aggregate client request rate for these videos is a constant, set to 500 requests/min. The video popularity obeys the zipf distribution, i.e., the $i$-th most popular video attracts a fraction of requests that is proportional to $1/i^{1-\alpha}$, where $\alpha$ is set to 0.271. We further assume that a video's popularity rank changes by one every 4 hours, e.g., the $i$-th most popular video become the $i+1$-th most popular video, and least popular video becomes the most popular one. Our purpose is to examine whether the smooth workload adaptive broadcast can smoothly handle the popularity change of many videos at the same time.

Fig. 11(a) depicts the aggregate bandwidth usage for a period of 24 hours. One observes that the number of active streams oscillates around its average however it never deviates more than 10% from its average. Fig. 11(b) depicts the bandwidth usage for a single video. Note that the bandwidth usage decreases every 4 hours, corresponding to the popularity change. The bandwidth usage leaps at time 0 due to the dramatic popularity change this video is experiencing - its popularity rank changes from the last to the first. However the smooth workload adaptive broadcast quickly adapts to such change.

## VI. RELATED WORK

Several works [12–14] in the past addressed the problem of changing workload adaption. [12] proposes to use a PB scheme to serve popular videos and dynamically change the number of channels assigned to a video based on the level of demand. This PB scheme possesses the smooth channel transition property; thus the channel change can be smoothly performed. However, unlike FPB scheme, the



(a) Client arrival rate (b) Number of channels used in FPB (c) Number of active streams

Fig. 10. Performance of the smooth workload adaptive scheme: a single video case

scheme in [12] requires the client to be able to listen to all the channels simultaneously. More importantly, a PB scheme without using parsimonious transmission becomes inefficient when a video turns unpopular. Also a playback delay is experienced by every client.

Both [13] and [14] use variations of the *batching* technique to tackle the problem of changing workload adaption. Batching serves multiple requests with a single multicast stream. [13] introduces rate-based channel allocation scheduling into batching to account for the changing workload. Since batching doesn't allow clients to prefetch data, it is not as bandwidth efficient as PB. [14] proposes a hybrid scheme that is a combination of batching and PB. Popular videos are handled by PB, while batching is used for less popular videos. The popularity of the videos is periodically re-evaluated to determine the group of videos that are served by PB. It is challenging to determine whether a video should be served by PB or by batching, and both schemes require clients to wait for a period of time before being served. In contrast, smooth workload adaptive broadcast provides a unified approach that performs well for both popular and less popular videos, and offers instantaneously playback if necessary.

PB schemes in [15] and [4] also use the Fibonacci series as segmentation series and have comparable bandwidth usage as FPB proposed in this paper. However the schemes differ from FPB in terms of both the server-side broadcasting and the client-side reception schedules are different, and neither of these PB schemes exhibit the smooth channel transition property. Also our FPB scheme can support clients with limited access bandwidth and/or limited buffering by scheduling them at the beginning of a cluster.

## VII. Conclusions and Future Work

In this paper we present a workload adaptive broadcast architecture and smooth workload adaptive broadcast based on FPB 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 smooth workload adaptive broadcast provides instantaneous or near-instantaneous playback service and can adapt nicely to workload changes. 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 smooth 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. In addition, implementation of smooth 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 *ACM Multimedia System Journal*, January 2002.

[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. Sen, L. Gao, and D. Towsley, "Frame-based periodic broadcast and fundamental resource tradeoffs," in *Proc. IEEE International Performance Computing and Communications Conference*, April 2001.

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

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

[8] L. Gao and D. Towsley, "Threshold-based multicast for continuous media delivery," in *IEEE Transactions on Multimedia*, December 2001.

[9] 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.

[10] A. Bar-Noy, G. Goshi, R. E. Ladner, and K. Tam, "Comparison of stream merging algorithms for media-on-demand," in *Proc. SPIE/ACM Conference on Multimedia Computing and Networking*, January 2002.

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

[12] 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.

[13] K. Almeroth, A. Dan, D. Sitaram, and W. Tetzlaff, "Long term resource allocation in video delivery systems," in *Proc. IEEE INFOCOM*, April 1997.

[14] J. Oh, K. A. Hua, and K. Vu, "An adaptive video multicast scheme for varying workloads," *to appear in ACM/Springer Multimedia Systems*, 2002.

[15] 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.

## APPENDIX

*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

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

and

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

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 smooth transition, the cluster using $K'$ channels must not overlap with the cluster using $K$ channels. Below we prove that the smooth transition is true for both $K' < K$ case and for $K' > K$ case.

(1) $K' < K$ (see Fig. 12). Since $K' < K$, $g(K',L) > g(K,L)$. Thus $s_i \geq t_i$ for all $i$, $1 \leq i \leq K'$. Hence there is no overlap between two clusters and smooth transition can be achieved in this scenario.

(2) $K' > K$ (see Fig. 13). 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

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

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:* 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 $F_j$ can represented as

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

We can verify Equation (11) by substituting $F_j$ into Equation (12). ∎

Now we are ready to prove Equation (10).

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

Apply Lemma 1, we have

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

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, $K$-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. 6), these two clusters 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. 14). 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 need 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 $1-(1-p)^{F_{K-2}}$.

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). ∎

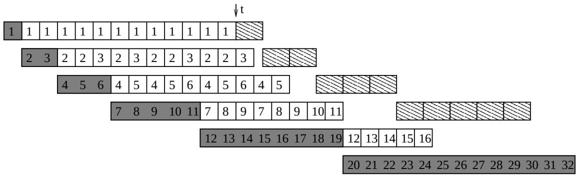


Fig. 12. Smooth Transition in FPB ($K' < K$)

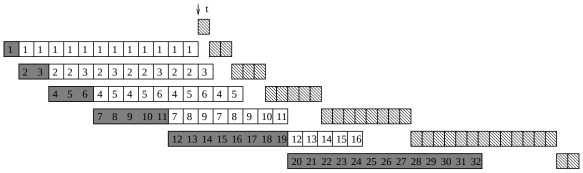


Fig. 13. Smooth Transition in FPB ($K' > K$)

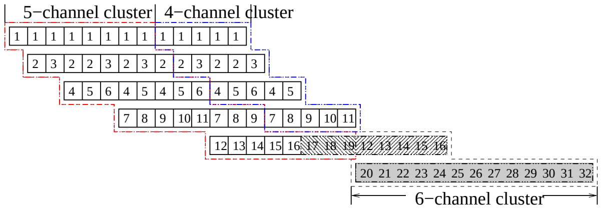


Fig. 14. A 6-channel Cluster and Its Two Sub-clusters: 5-channel Cluster and 4-channel Cluster.