

# Adaptive Live Broadcasting for Highly-Demanded Videos

Hung-Chang Yang<sup>1</sup>, Hsiang-Fu Yu<sup>1</sup>, Li-Ming Tseng<sup>1</sup>, Yi-Ming Chen<sup>2</sup>

[cyht@dslab.csie.ncu.edu.tw](mailto:cyht@dslab.csie.ncu.edu.tw), [yu@dslab.csie.ncu.edu.tw](mailto:yu@dslab.csie.ncu.edu.tw), [tsenglm@csie.ncu.edu.tw](mailto:tsenglm@csie.ncu.edu.tw),  
[cym@im.mgt.ncu.edu.tw](mailto:cym@im.mgt.ncu.edu.tw)

<sup>1</sup>Distributed System Laboratory, Department of Computer Science and Information Engineering, National Central University, Jung-Li, Taiwan 32054, R.O.C.

<sup>2</sup>Department of Information Management, National Central University, Jung-Li, Taiwan 32054, R.O.C.

## Abstract

With the growth of broadband networks, the Video-on-Demand (VoD) becomes realistic. Many significant broadcasting schemes are proposed to reduce the bandwidth requirements for stored popular videos, but they cannot be used to support live video broadcast perfectly.

Herein, we propose a new broadcasting scheme, called *Adaptive Live Broadcasting (ALB)* scheme, which supports live video broadcasting and performs well over a wide range of request arrival rates. From our analysis and comparison, we find that our ALB scheme is suitable to broadcast live video. It has several significant advantages: (1). It has the shortest maximum waiting time with fixed channels. (2). It has the least maximum I/O transfer requirements with fixed maximum waiting time at client end. Finally, a simulation is employed to evaluate several live broadcasting schemes, such as UD, ST, AFB and ALB. The results reveal our ALB scheme consumes the least server bandwidth.

**Keywords:** Adaptive Live Broadcasting Scheme, Network Bandwidth Scheduling, Popular Video Service, Video-on-Demand (VoD)

## 1 Introduction

With the growth of broadband networks, the Video-on-Demand (VoD) [13] becomes realistic. Many studies start investigating VoD. One of the important areas is to explore how to distribute the top ten or twenty so-called “hot” videos more efficiently. Broadcasting is a promising solution. It transfers each video according to a fixed schedule and consumes constant bandwidth regardless of the number of

requests for that video. That is, the number of users watching a given video is independent of their bandwidth requirements. A basic broadcasting scheme is the batch scheme [1]. The batch scheme delays the users’ requests for a certain amount of time and serves these requests in batch so that the bandwidth consumption is saved. However, the batch scheme still requires quite large bandwidth for a hot video. For example, a film lasts 120 minutes. If each request for the film has to be served within 10 minutes, we need to allocate 12 (120/10) video channels.

Suppose the set-top-box (STB) at the client end can buffer portions of the playing video on disk. With the STB, many significant broadcasting schemes were proposed, such as fast broadcasting (FB) [4, 8], pagoda broadcasting (PB) [10], new pagoda broadcasting (NPB) [11], recursive frequency splitting (RFS) [3], staircase broadcasting (SB) [6] and harmonic broadcasting (HB) [5, 7]. These schemes divide a video into multiple equal-size segments and distribute these segments through several independent data streams. As well, they require the STB to receive all segments from the streams when the user starts watching the video. The broadcasting schemes substantially reduce the bandwidth requirements for hot videos. For example, with the FB, a video server allocates 4 video streams for a 120-minute video, then its waiting time is less than 8 minutes. Both the bandwidth consumption and waiting time of the fast broadcasting are superior to those of the batch scheme.

In the real world, some history events are very hot, for example, Comet Shoemaker-Levy collision with Jupiter, thousands of people attempt to connect to Internet to watch the video immediately. Such actions easily produce the network congestion. However, most of these schemes, such as PB, NPB, RFS, SB and

HB, can not broadcast such hot live videos and alleviate the congestion. In order to overcome this obstacle, this paper proposed the Adaptive Live Broadcasting (ALB) scheme, which supports the live video broadcasting. The simulation results indicate that ALB has shorter waiting time and less bandwidth requirements among 4 live broadcasting schemes.

The rest of this paper is organized as follows. Next section introduces related work. Section 3 describes the ALB. Section 4 shows analysis and comparison. Section 5 presents the simulation results. Conclusions are finally made in Section 6.

## 2 Related work

### 2.1 The requirements of live video broadcasting

Initially, we analyze three important requirements for live video broadcasting that differs from the stored video broadcasting as following.

- R1. *The total data transfer rate can not be larger than the media production rate.* In the case of live broadcasting, the new media is produced at constant speed such that the broadcasting schemes that always transfer data at a higher rate than media production rate can not support live broadcasting.
- R2. *The live video segment can not be transferred preemptively until the video segment is produced.* The scenes of a live video are captured and broadcasted with video progress; the broadcasting schemes can not transmit the posterior and unavailable segments of live video in advance.
- R3. *The broadcasting scheme has to tolerate the varying length of live videos.* People always wish that a live video is held according to the schedule; however, in the real world, the live video often ends either early or late, rarely on time. Most broadcasting schemes suppose that the video's length is known and fixed. In the case of early ending, the broadcasting schemes simply free the allocated channels, or repeat the last or blank video segments. Hence, the viewer watching the video is not affected. In the case of late ending, the broadcasting schemes require additional bandwidth to handle the situation of video elongation.

### 2.2 The schemes regarding live broadcasting

#### 2.2.1 New Pagoda Broadcasting scheme (NPB)

The NPB [11] employs *rectangular matrix allocation* method to distribute the segment-to-stream mapping. The mapping is optimal when each segment  $S_i$  can be broadcasted exactly once every  $i$  slots. Accordingly, the NPB broadcasts each segment  $S_i$  once every  $i$  slots as possible as it can.

Figure 1 depicts the NPB with 4 streams. It is able to transmit 26 segments and guarantees that viewing delay will not exceed 4 minutes 37 seconds for a 2-hour video.

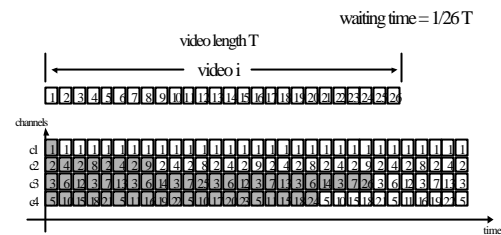


Figure 1 The NPB scheme with 4 streams.

When we attempt to apply NPB to the live video broadcasting, we find it fail to the requirements R1 and R2. For example, broadcasting a video with 3 streams as shown in Figure 2. The segments  $S_2$  and  $S_3$  are unavailable at slot 1, and the segments  $S_4$  and  $S_6$  are unavailable at slot 2, and so on.

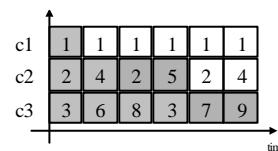


Figure 2 The NPB scheme with 3 streams.

If we add an additional live stream and delay some segments distribution, the NPB can broadcast live video. As well, we called it *Live NPB* scheme as shown in Figure 3.

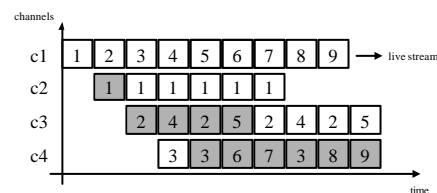


Figure 3 The Live NPB scheme with 4 streams.

### 2.2.2 Recursive Frequency Splitting scheme (RFS)

By using a more complex segment-to-stream mapping, the RFS [6] provides smaller waiting time than the NPB scheme when the number of streams is larger than 4. Figure 4 depicts the RFS with 4 streams.

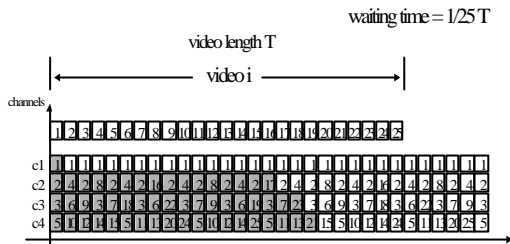


Figure 4 The RFS scheme with 4 streams

As earlier, the RFS fails to the requirements R1 and R2. We must add an additional live stream and delay some segments distribution to support live video broadcasting, called *Live RFS* scheme.

### 2.2.3 Fast Broadcasting scheme (FB)

The FB scheme [4, 8] reduces the bandwidth requirement in the logarithmic order of maximum waiting time. It partitions the video into  $2^k - 1$  segments  $S_1$  to  $S_{2^k - 1}$  and the stream  $j$ , where  $1 \leq j \leq k$ , transmits segments  $S_{2^{j-1}}$  to  $S_{2^j - 1}$  as indicated in Figure 5.

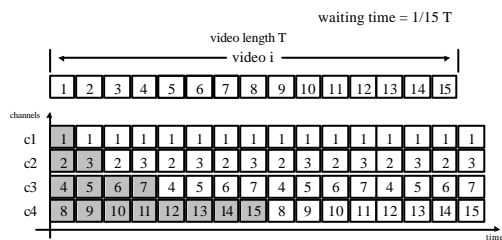
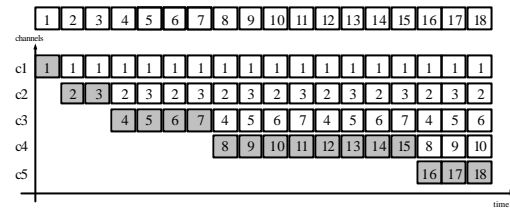
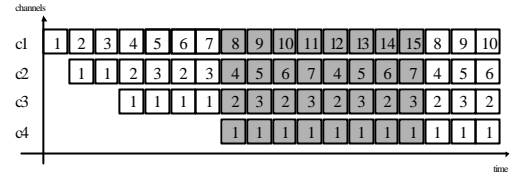


Figure 5 Illustration of channel allocation for FB.

The FB scheme can directly support live video broadcasting with slight modification as shown in Figure 6. In Figure 6(a), the FB scheme can support live video broadcasting by delaying the distribution of segments. Figure 6(b) is an illustration of *Live FB* scheme.



(a)



(b)

Figure 6 The Live FB scheme

### 2.2.4 Adaptive Fast Broadcasting scheme (AFB)

The disadvantage of the FB is it can not dynamically allocate bandwidth even though no request arrival. To overcome this obstacle, the AFB scheme [9] dynamically allocates the bandwidth according to the users' requests.

The AFB scheme can also support live video broadcasting, because it is based upon FB. An example for AFB scheme is shown as Figure 7. Assuming  $N=15$  and there are 3 requests.

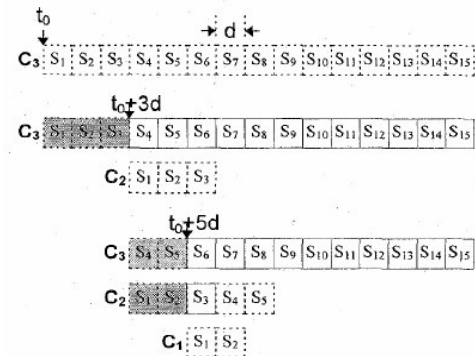


Figure 7 The AFB scheme.

### 2.2.5 Universal Distribution scheme (UD)

The universal distribution scheme [12] is a dynamic broadcasting scheme based upon the FB scheme. In Figure 8, a first request is arrival at slot 0, and it is following by two other requests arriving at slots 3 and 4 respectively. Not that the segment  $S_2$  and  $S_3$  are allocated to slots 4 and 5, but that they are allocated to slots 5 and 6. Hence the segment  $S_2$  and  $S_3$  can be shared with the second and third request.

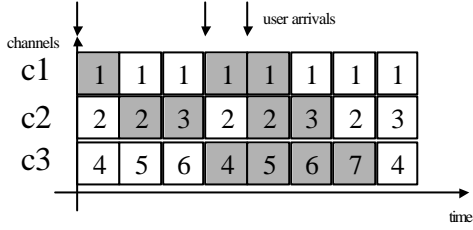


Figure 8 The UD scheme with 3 requests.

Since the UD is based upon FB, it also supports live video broadcasting, as shown in Figure 9, and we call it *Live UD* scheme.

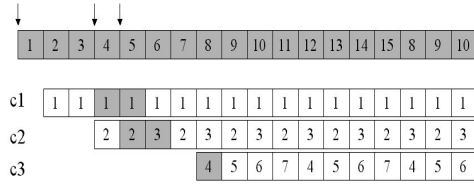


Figure 9 The Live UD scheme

### 2.2.6 Stream Tapping scheme (ST)

The ST [2] supposes clients have a small buffer on their STB. The buffer allows them to “tap” into streams of data on the video server originally created for other clients, and then store the data until they are needed. In the best case, clients can get most of their data from existing streams, which greatly reduce the duration of their own stream. The ST scheme can support live video broadcasting.

## 3 Adaptive Live Broadcasting scheme

In this section, we propose a new broadcasting scheme, called *adaptive live broadcasting* (ALB) scheme, to support live video broadcasting.

Before getting into the detail of our algorithm, we give the following necessary lemmas. Assume the number of segments of live video is  $n$  and the number of requests is  $m$ .

#### Lemma 1:

Each segment  $S_i$ ,  $1 \leq i \leq n$ , must be broadcasted at least once on one of the  $k$  channels in every continuous  $i$  time slots.

#### Proof:

Suppose that a user starts playing the video at time slot  $j$ . Then the user will consume segment  $S_i$  at time slot  $j+i-1$ . This implies that  $S_i$  must be broadcasted on one of the channels at time slot  $j+i-1$ , or has been broadcast on one of the channels during slots  $j, j+1 \dots j+i-2$ . This proves that the condition given in the lemma is a sufficient condition. Next, we prove that this is

also a necessary condition. If  $S_i$  has not been sent on the aforementioned time slots, the user will experience an interruption at time slot  $j+i-1$ .

#### Lemma 2:

To support live video broadcasting, each segment  $S_i$ ,  $1 \leq i \leq n$ , must be broadcasted in time slot  $i$  at first time.

#### Proof:

Due to we cannot pre-fetch the live video and store it in the disk of VoD server beforehand. Each segment  $S_i$  must be broadcasted in time slot  $i$  at first time.

#### Lemma 3:

The necessary recasting segments  $N_i$  for the request  $R_i$  arrived at the time slot  $T_i$  are  $BS_{T_i} - BS_{T_i-1}$ , where  $1 \leq i \leq m$  and  $BS_{T_i}$  are the segments broadcasted from time slot 0 to time slot  $T_i$ .

In Figure 10, when the second user enters into the session at 9<sup>th</sup> slot, the video server has to recast the segment  $N_2 = BS_9 - BS_8 = \{S_1, S_2, S_3, S_7, S_8 \text{ and } S_9\}$  according to lemma 3. In addition, when the fourth user enters into the session at 14<sup>th</sup> slot, the video server has to recast the segments  $N_4 = BS_{14} - BS_{13} = \{S_1, S_2, S_3, S_4, S_6, S_7, S_8, S_{12}, S_{13} \text{ and } S_{14}\}$ . And so on.

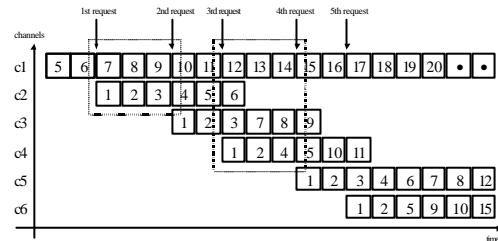


Figure 10 The necessary recasting segments when the user enters into the session.

Herein, our adaptive live broadcasting scheme is introduced. In order to satisfy Lemma 1, we put the segment  $S_i$  every  $i$  time slots. If there is no free slot at that position, the segment  $S_i$  is put into the previous time slot until it can be. To satisfy Lemma 2, we use a stream, called live stream, to broadcast the segments  $S_1, S_2, S_3 \dots S_n$ . Figure 11 illustrates the segment to stream mapping.

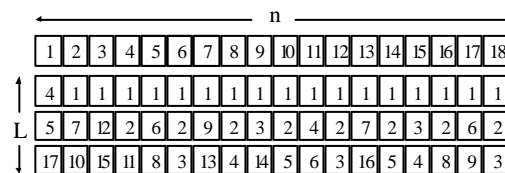


Figure 11 The segment to stream mapping.

Figure 12 presents the entire algorithm of the adaptive live broadcasting scheme.

**Assumptions:**

- the number of segments is  $n$
- $L$  indicates the number of total channels minus 1
- slot  $k$  contains  $m_k$  segments

**Algorithm:**

```

initialize all  $m_k$  to 0
for  $i := 1$  to  $n$  do

    if  $n \bmod i$  equal 0 then  $max = \left\lceil \frac{n}{i} \right\rceil + 1$ 

    else  $max = \left\lceil \frac{n}{i} \right\rceil$ 

    for  $j := 2$  to  $max$  do
         $p = i * j$ 
        while  $m_p \leq L$  do
             $p = p - 1$ 
        end while
        if  $p > n$  then  $p = p - n$ 
        schedule  $S_i$  in slot  $p$ 
         $m_p = m_p + 1$ 
    end for loop
end for loop

```

Figure 12 The entire ALB's algorithm.

Up to now, our proposed scheme can satisfy the requirements R1 and R2. In the following, we propose two approaches to the requirement R3.

The first discards the exceeding part of the video. The video is like the hour's news of CNN. The older news would be discarded, and the newest news would be added. Our adaptive live broadcasting scheme can broadcast the hour's news in a period, and broadcast the newer news by discarding to broadcast the older news at the next period, as shown in Figure 13. The period has seven time slots. In the first hour, we broadcast the news 1 to 7. In second hour, we broadcast news 2 to 8 by discarding the older news 1 and adding the newer news 8.

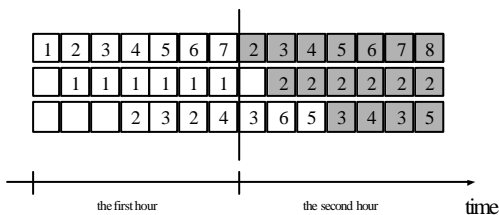


Figure 13 An example of broadcasting hour's news.

The second allocates additional channels to transfer the unpredictable prolonged segments of live video as shown in Figure 14.

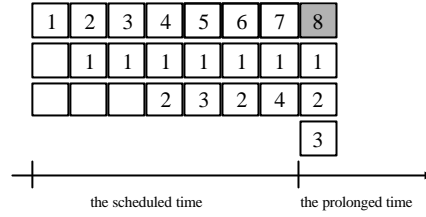


Figure 14 Broadcasting unpredictable prolonged segments by allocating additional channels.

To efficiently play videos by ALB, the user's request are served on demand by the following principles:

- 1) We exploit the live stream as the main stream and only recast the necessary segments when the user enters into the session.
- 2) The later user can share the segments that are recasted to the previous users.
- 3) The necessary recasting segments  $S_i$  delay  $i$  time slots to broadcast as possible as it can when the user enters into session.

Figure 15 illustrates the ALB's playing. The first channel is the live stream, and there are 4 users enter into the session. When the user enters into the session at 2<sup>nd</sup> time slot, the video server has to recast the segments  $S_1$  and  $S_2$ . When the user enters into the session at 3<sup>rd</sup> time slot, the video server has to recast the segments  $S_1$  and  $S_3$ , because the  $S_2$  has been recasted. When the user enters into the session at 9<sup>th</sup> time slot, the video server has to recast the segments  $S_1, S_2, S_3 \dots S_9$ . As indicated in Figure 15, the segment  $S_5$  is recasted at 14<sup>th</sup> time slot not 10<sup>th</sup> time slot. This is because distribution must be delayed as possible as it can to increase the probability of sharing the bandwidth. Due to serving the user's request on demand, the ALB scheme requires less bandwidth.

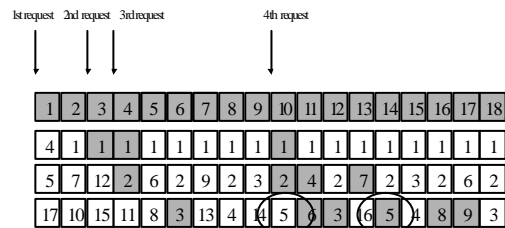


Figure 15 To efficiently play video by ALB.

## 4 Analysis and comparison

At first, we derive the maximum segments with fixed channels, and then compare the found results from the ALB and some existing schemes. We also analyze the user waiting time and disk rate transfer requirement.

### 4.1 Maximum segments with fixed channels

To maximize the bandwidth utilization, we need to obtain maximum sharing of each recasted segment. In order to achieve the goal, the scheme has better to delay the segments distribution as long as possible. According to Lemma 1, if a transmission schedule starting at slot  $i+1$  cannot share its  $j$ -th segment  $S_j$  with any previous transmission schedule, the schedule will attempt to put the segment  $S_j$  in slot  $i+j$ . Thus the first segment must be scheduled at least once every slots, and second segment must be schedule at least once every two slots, and so on. Therefore, we can distribute each segment on demand according to its minimum frequency. The algorithm is shown in Figure 16.

#### Assumptions:

new video request arrives during slot  $i$

#### Algorithm:

```

for  $j := 1$  to  $i$  do
    search slots  $i+1$  to  $i+j$  for an already
    scheduled segment of  $S_j$ 
    if not found then
        schedule segment  $S_j$  in slot  $i+j$ 
    end if
end for loop

```

Figure 16 The algorithm of minimum frequency scheduling.

In the worst case, there is at least one user at each time slot, the segments that user required are plotted in Figure 17.

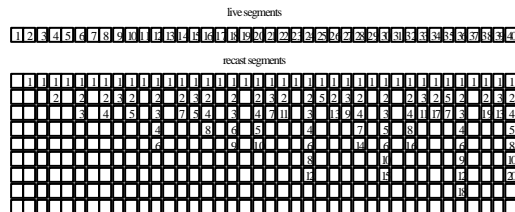


Figure 17 The segments that user required in the worst case.

In addition, we can calculate the maximum number of segments,  $n$ , with fixing channels,  $c$ , by the following:

$$\sum_{i=1}^{n-1} (\text{number of factors of } i) + n \leq n * c$$

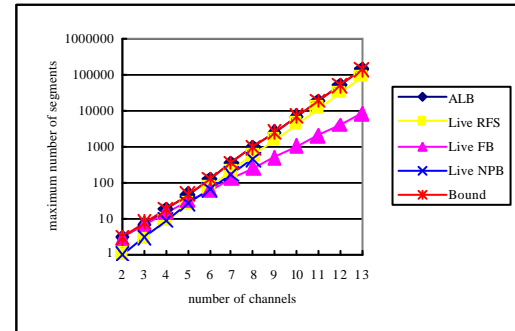
Next, we show how to calculate the maximum segments of the ALB scheme. The minimum segments to be broadcasted is  $\sum_{i=1}^n \left\lceil \frac{n}{i} \right\rceil$ , since the segment  $S_i$  must be broadcasted every  $i$  time slots. Furthermore, we consider that all segments whose index is the factor of value  $n$  cannot be schedule in slot  $n$  due to the fixed channels. Therefore, the up-bound can be calculated by the following:

$$\sum_{i=1}^n \left\lceil \frac{n}{i} \right\rceil + [(\text{number of factors of } n) - c] \leq c * n$$

Finally, we derive the maximum segments with fixed channels from Live RFS, Live NPB, Live FB and ALB schemes. Figure 18 lists the result, where its first row indicates the number of channels. We can find that the ALB scheme outperforms all other schemes.

Channel	2	3	4	5	6	7	8	9	10	11	12	13
Bound	3	8	18	48	130	350	936	2550	6952	18876	51300	139464
ALB	3	7	18	47	129	349	925	2549	6951	18875	51299	139463
Live RFS	1	3	9	25	73	201	565	1522	4284	11637	31677	86428
Live FB	3	7	15	31	63	127	255	511	1023	2047	4095	8191
Live NPB	1	3	9	26	66	172	442	NA	N/A	NA	NA	NA

(a)



(b)

Figure 18 The maximum segments in different schemes.

## 4.2 Waiting time vs. bandwidth allocation

Suppose the client end has enough disk space to buffer portions of the playing video on disk. When we just miss a segment  $S_j$  of a requested video, the maximum waiting time will equal to the access time of  $S_j$ . The length of the video is  $D$ , which is equally divided into  $N$  data segments. Therefore, using the ALB scheme, the maximum waiting time to access a broadcast video is  $\frac{D}{N}$ . Figure 19 shows that

maximum waiting time vs. network bandwidth allocation of ALB, Live RFS, Live FB and Live NPB schemes. For example, when the number of channels is 13, the waiting time we get using ALB scheme is 17 times shorter than the waiting time obtained using Live FB scheme.

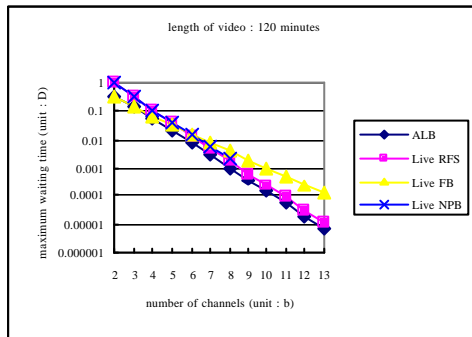


Figure 19 Maximum waiting time vs. network bandwidth allocation.

In the Figure 20, we compare the maximum waiting time in minutes with network bandwidth. We can see that our ALB scheme has the shortest waiting time. For the same waiting time requirement, the ALB scheme needs the least bandwidth. For example, suppose there is a video with length  $D=100$  minutes. If the maximum waiting time must be within 1 minute  $=0.01D$ , the ALB scheme needs about 6 channels. For the same condition, the Live RFS scheme needs about 7 channels, Live FB scheme needs about 8 channels and Live NPB needs 7 channels.

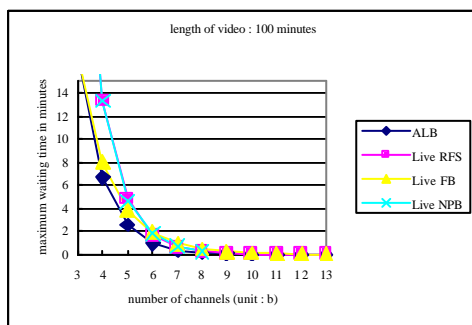


Figure 20 Maximum waiting time in minutes vs. bandwidth allocation.

## 4.3 Disk transfer rate requirements at client end

At client end, we will write the input video data into disk as it needs to be buffered. When we need to consume the data, we need to read the data from disk. The disk transfer (input/output) rate requirements are the sum of the read requirement and write requirement. According to the client buffer requirements, we find that the maximum disk I/O rate requirements will occur during we read a segment from disk and write the input data from channels  $\{C_0, C_2, C_3, \dots, C_{b-1}\}$  to the disk as shown in Figure 21. When the user who enters into the session at 4<sup>th</sup> time slot needs to playback the segment  $S_2$ , he/she needs to receive and write the input segment  $S_{11}, S_4$  and  $S_6$  from channels  $\{C_0, C_2, C_3\}$  simultaneously. Hence, the disk transfer rate requirements of the ALB scheme are  $b^*b$ .

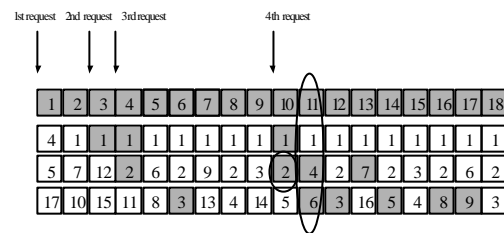


Figure 21 The example of the maximum disk I/O rate requirements.

Figure 22 shows the disk transfer rate requirement for the maximum waiting time of the ALB, Live RFS, Live FB and Live NPB schemes. We can see that our ALB scheme needs the least disk transfer rate.

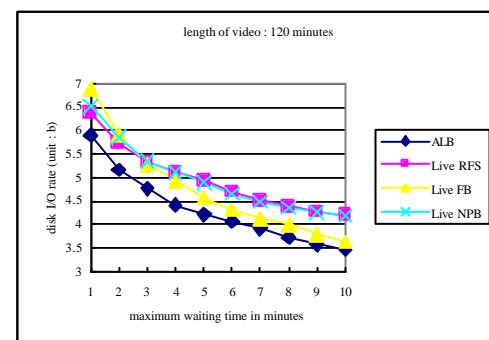


Figure 22 Disk transfer rate requirements for maximum waiting time.

## 5 Simulation

To evaluate the performance of the ALB scheme, we wrote a simple simulation program. Assume that the time of user's request for a particular video were distributed according to exponential distribution,  $f(x) = \lambda * e^{-\lambda x}$ ,  $x \geq 0, \lambda > 0$ . This is because where the majority of users watch the live video on time and the latecomers decreases with the time. We assumed a video lasts 127 minutes, which is close to the average duration of a feature video. We partitioned the video into 127 segments, as it would simplify the comparison with the UD and AFB. Figure 23 displays the bandwidth requirements for the UD, ST with unlimited extra tapping and unlimited client buffer, AFB and ALB with the value of  $\lambda$  is 0.14 and the number of users is form 1 to 100. The ALB outperforms ST, UD and AFB when the request arrival users great than 15 users. ST scheme performs slightly better than ALB only when request arrival users less than 15 users.

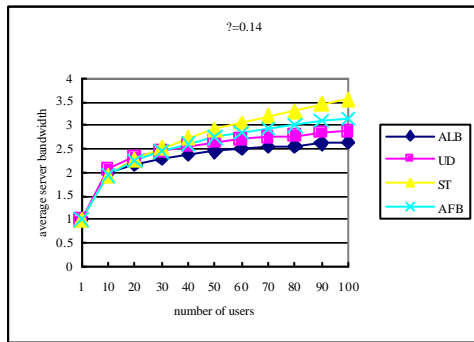


Figure 23 The bandwidth comparison of ALB, UD, ST and AFB with the value of  $\lambda$  is 0.14 and the number of users is from 1 to 100.

Figure 24 and 25 show the bandwidth requirement with the number of users is from 1 to 1000 and form 1 to 10000 respectively. We can find that our ALB scheme still outperforms ST, UD and AFB schemes.

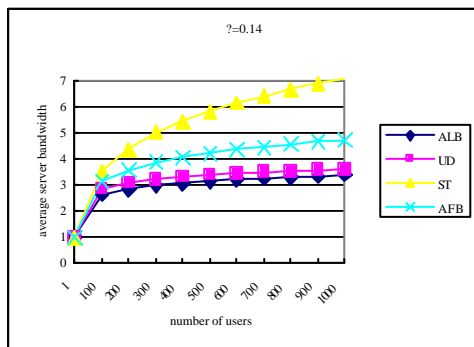


Figure 24 The bandwidth comparison of ALB, UD, ST and AFB with the value of  $\lambda$  is 0.14 and the number of users is from 1 to 1000.

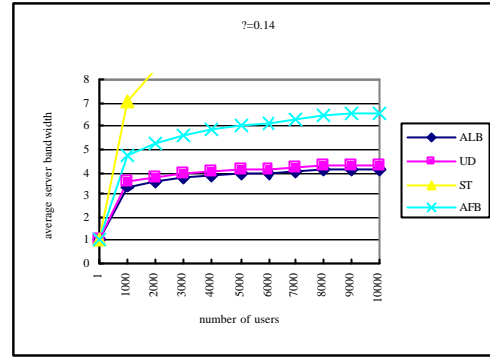


Figure 25 The bandwidth comparison of ALB, UD, ST and AFB with the value of  $\lambda$  is 0.14 and the number of users is form 1 to 10000.

Figure 26 depicts the impact of the value of  $\lambda$  on the ALB scheme. The bandwidth requirement for ALB scheme decreases as the value of  $\lambda$  increases. This phenomenon indicates that the requests arrive at begin of the live video when the value of  $\lambda$  is bigger. Thus the required bandwidth is less.

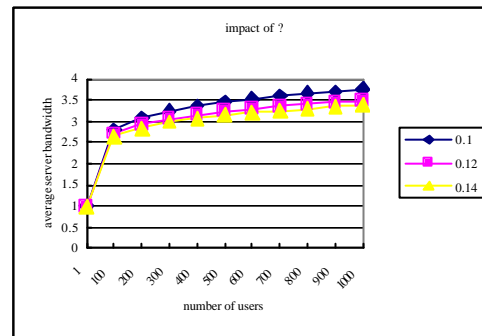


Figure 26 The impact of  $\lambda$  on the ALB scheme.

## 6 Conclusions

With the growth of broadband networks, the Video-on-Demand (VoD) becomes realistic. One of the important areas is to explore how to distribute the popular videos more efficiently. Many significant broadcasting schemes are proposed to reduce the bandwidth requirements for stored popular videos, but they cannot be used to support live video broadcast perfectly.

Herein, we propose a new broadcasting scheme, called *Adaptive Live Broadcasting (ALB)* scheme, which supports live video broadcasting and performs well over a wide range of request arrival rates. From our analysis and comparison, we find that our ALB scheme is suitable to broadcast live video. It has several significant advantages: (1). It has the shortest maximum waiting time with fixed channels. (2).



It has the least maximum I/O transfer requirements with fixed maximum waiting time at client end. Finally, a simulation is employed to evaluate several live broadcasting schemes, such as UD, ST, AFB and ALB. The results reveal our ALB scheme consumes the least server bandwidth.

## Reference

- [1] T. Chiueh and C. Lu, "A periodic broadcasting approach to video-on-demand service", *SPIE*, vol. 2615, pp. 162-169, Oct 1995.
- [2] S. W. Carter and D. D. E. Long, "Improving video-on-demand server efficiency through stream tapping", *Proc. 5<sup>th</sup> Int. Conf. On Computer Communications and Networks*, pp. 200-207, Sep 1997.
- [3] Chi-He Chang, Jang-Ping Sheu and Yu-Chee Tseng, "A Recursive Frequency-Splitting Scheme for Broadcasting Hot Videos in VOD Service", *M. S. Thesis*, Department of Computer Science and Information Engineering, National Central University, Chung-Li, Taiwan, R.O.C., Jun 2000.
- [4] L-S. Juhn and L-M. Tseng, "Fast broadcasting for hot video access", *RTCSA'97: the proceedings of the 4<sup>th</sup> international workshop on real-time computing systems and applications*, pp.237-243, Oct 1997.
- [5] L-S. Juhn and L.M. Tseng, "Harmonic broadcasting for video-on-demand service", *IEEE Trans. Broadcasting*, vol. 43, no.3, pp. 268-271, Sep 1997.
- [6] L-S. Juhn and L-M. Tseng, "Staircase data broadcasting and receiving scheme for hot video service", *IEEE Trans. Consumer Electronics*, vol. 43, no.4, pp. 1110-1117, Nov 1997.
- [7] L-S. Juhn and L.M. Tseng, "Enhanced harmonic data broadcasting and receiving scheme for popular video service", *IEEE Trans. Consumer Electronics*, vol. 44, no. 2, pp. 343-346, May 1998.
- [8] L-S. Juhn and L.M. Tseng, "Fast data broadcasting and receiving scheme for popular video service", *IEEE Trans. Broadcasting*, vol. 44, no. 1, pp. 100-105, Mar 1998.
- [9] L-S. Juhn and L-M. Tseng, "Adaptive fast data broadcasting scheme for video-on-demand service", *IEEE Trans. Broadcasting*, vol. 44, no. 2, pp. 182-185, Jun 1998.
- [10] J-F. Paris, S. W. Carter and D. D. E. Long, "A Hybrid broadcasting protocol for video on demand", *Proc. 1999 Multimedia Computing and Networking Conference*, San Jose, CA, pp. 317-326, Jan 1999.
- [11] J-F. Paris, "A simple low bandwidth broadcasting protocol for video on demand", *Proc. 7<sup>th</sup> Int. Conf. On Computer Communications and Networks*, pp. 690-697, Oct 1999.
- [12] J-F. Paris, S. W. Carter and D. D. E. Long, "A universal distribution protocol for video-on-demand", *Proc. 1<sup>st</sup> Int. Conf. On Multimedia and Expo 2000*, vol. 1, pp. 49-52, Jul 2000.
- [13] V. Rangan, H. Vin and S. Ramanathan, "Designing an on-demand multimedia service", *IEEE Communications Magazine*, vol. 30, no. 7, pp. 56-65, Jul 1992.