Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2012, Article ID 717538, 7 pages
doi:10.1155/2012/717538

Research Article

An Efficient Periodic Broadcasting with Small Latency and Buffer Demand for Near Video on Demand

Ying-Nan Chen and Li-Ming Tseng

Department of Computer Science and Information Engineering, National Central University, Chung-Li 32054, Taiwan

Correspondence should be addressed to Ying-Nan Chen, ynchen@dslab.csie.ncu.edu.tw

Received 31 January 2012; Accepted 3 February 2012

Academic Editor: Hsiang-Fu Yu

Broadcasting Protocols can efficiently transmit videos that simultaneously shared by clients with partitioning the videos into segments. Many studies focus on decreasing clients' waiting time, such as the fixed-delay pagoda broadcasting (FDPB) and the harmonic broadcasting schemes. However, limited-capability client devices such as PDAs and set-top boxes (STBs) suffer from storing a significant fraction of each video while it is being watched. How to reduce clients' buffer demands is thus an important issue. Related works include the staircase broadcasting (SB), the reverse fast broadcasting (RFB), and the hybrid broadcasting (HyB) schemes. This work improves FDPB to save client buffering space as well as waiting time. In comparison with SB, RFB, and HyB, the improved FDPB scheme can yield the smallest waiting time under the same buffer requirements.

1. Introduction

How to efficiently maintain the exhausted bandwidth with the growth in the number of clients is an important issue of VOD deployment. Dan et al. [1] presented that 80 percent of the demand is for a few number (10 or 20) of very popular videos. One way to broadcast a popular video is to partition the video into segments, which are transmitted on several channels currently and periodically. The approach (called periodic broadcasting [2]) lets multiple users share channels and thus obtains high bandwidth utilization. One of the channels only broadcasts the first segment in real time. The other channels transmit the remaining segments. When clients want to watch a video, they wait for the beginning of the first segment on the first channel. While clients start watching the video, their set-top boxes (STBs) or computers still download and buffer unreceived segments from the channels to enable them to play the video continuously.

The staggered broadcasting [3] scheme treats a complete video as a single segment and then transmits it on each channel at different start times. The fast broadcasting (FB) scheme [4] improves segment partitioning and arrangement to yield shorter service latency. The harmonic broadcasting (HB) scheme [5] initially partitions a video into equally sized segments, which are further divided into smaller subsegments according to the harmonic series. HB can yield the minimum waiting time [6]; however, its implementation is difficult due to the multitude of broadcasting channels [7]. The recursive frequency-splitting (RFS) [7] scheme broadcasts a segment as close to its frequency as possible to achieve a near-minimal waiting time. The study [8] focuses on reducing the computation complexity of RFS. Paris [9] proposed a fixed-delay pagoda broadcasting (FDPB) scheme that required clients to wait for a small-fixed delay before watching the selected videos.

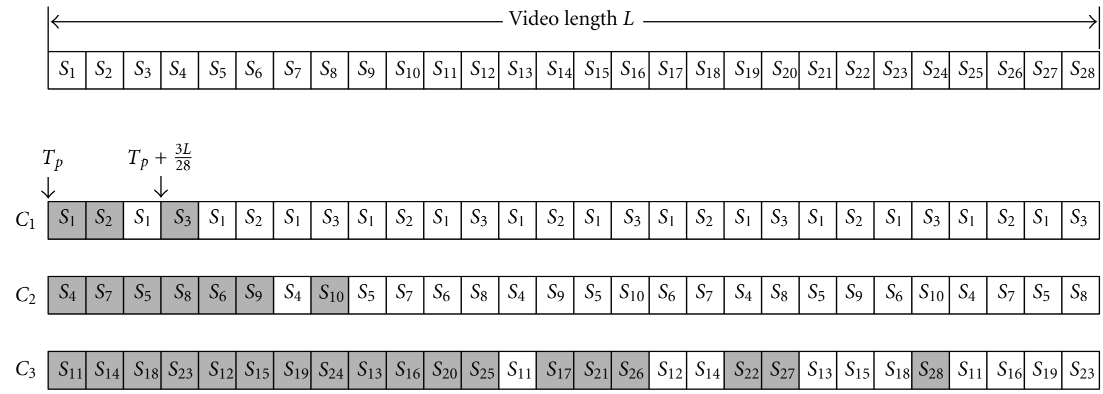
The staircase broadcasting (SB) scheme [10] requires a client to buffer only 25% of a playing video. In modifying the FB scheme, the reverse fast broadcasting (RFB) scheme [11] also buffers 25% of video size, merely half of what is required by the FB scheme. By combining RFS and RFB the hybrid broadcasting scheme (HyB) [12] yields small client buffering space and waiting time. The study in [13] proposed a generalized reverse sequence-based model to reduce their client buffer requirements. This work aims at improving FDPB to reduce required playback latency and buffering space. We prove the applicability of the improved FDPB, and compare

Stream 1		
Substream	1	2
First segment	1	2
Last segment	1	3

Stream 2		
Substream	1	2
First segment	4	7
Last segment	6	10

Stream 3				
Substream	1	2	3	4
First segment	11	14	18	23
Last segment	13	17	22	28

(a) The segment allocation on three channels



(b) The segment broadcasting on three channels

Figure 1: Illustration of channel allocation for the FDPB-3 scheme.

its performance with that of several existing approaches. Given a bandwidth of 5 channels, the new scheme reduces the broadcast latency by as much as 24% when compared to HyB and 34.5% when compared to RFB and SB. The buffer requirements for these schemes are about 25% of video size.

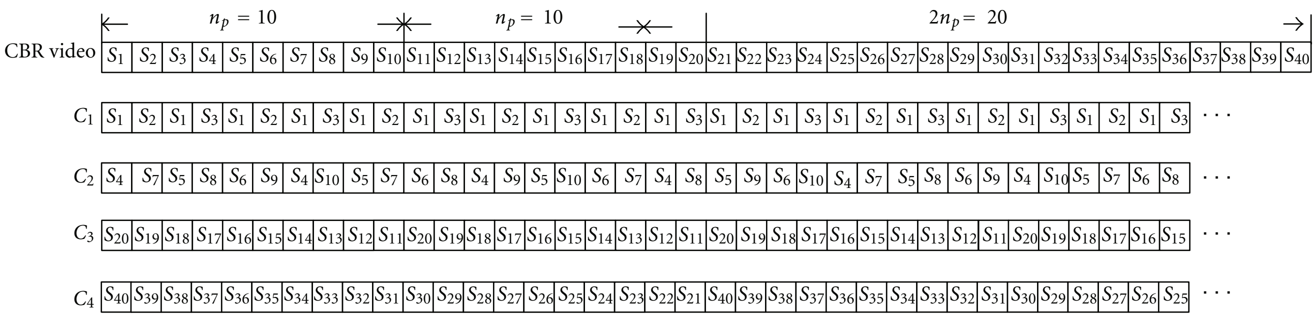
The rest of this paper is organized as follows. Section 2 introduces FDPB and proposes its improvement. Performance analysis and comparison are in Section 3. We make brief conclusions in Section 4.

2. Proposed Scheme

This section briefly reviews FDPB and then presents the improvement.

2.1. Fixed-Delay Pagoda Broadcasting (FDPB) Scheme. To help explain the improved FDPB, we first review the FDPB scheme in the literature [9]. Suppose that a video of duration L is broadcast over k channels. The scheme partitions each video into m equal-size segments of duration $l = L/m$. These segments will be broadcast at different frequencies over the k channels. The FDPB-p scheme requires customers to wait for a fixed time interval $w = pl$, where $p \in$ int, $p \geq 1$. Segment S_i is thus transmitted at least once every $p + i - 1$ slots. The FDPB-p scheme partitions each channel C_j into $c_j = \sqrt{p+i-1}$ subchannels in such a way that slot j of channel C_j belongs to its subchannel j. FDPB then maps segments into subchannels in a strict sequential fashion.

Figure 1 illustrates the video partition and segment arrangement of FDPB-3. Channel C_1 is partitioned into two subchannels because $\sqrt{3+1-1}$ is close to 2. Let S_i be the ith segment. Since $p = 3$, segment S_1 needs to be broadcast at least once every three slots and is assigned to subchannel 1. The first segment of subchannel 2 is segment S_2, which appears at least once every $3 + 2 - 1 = 4$ slots. We thus map two segments into subchannel 2, and channel C_1 transmits three segments (i.e., S_1 to S_3). The first segment of channel C_2 is segment S_4, which is broadcast at least once every $3 + 4 - 1 = 6$ slots. Since 6 is close to $4 = 2^2$, channel C_2 is partitioned into two subchannels. The first subchannel periodically transmits segments S_4 to S_6, and the second subchannel broadcasts segments S_7 to S_{10}. The first segment of channel C_3 is segment S_{11}, which is broadcast at least once every $3 + 11 - 1 = 13$ slots. Similarly, since 13 approaches to $16 = 4^2$, channel C_3 is partitioned into four subchannels. The first subchannel sends segments S_{11} to S_{13}, the second subchannel sends segments S_{14} to S_{17}, the third subchannel sends segments S_{18} to S_{22}, and the fourth subchannel sends S_{23} to S_{28}. Hence, channel C_3 broadcasts 18 segments. Figure 1(b) demonstrates how to broadcast the segments on three channels and how to download the segments colored gray. Suppose that a client starts receiving the segments at time T_p. This figure shows that the client begins to watch the video at time $T_p + 3L/28$.



An example, where $k = 4$ and $n_p = 10$

Figure 2: Segment transmission by RFDPB-3.

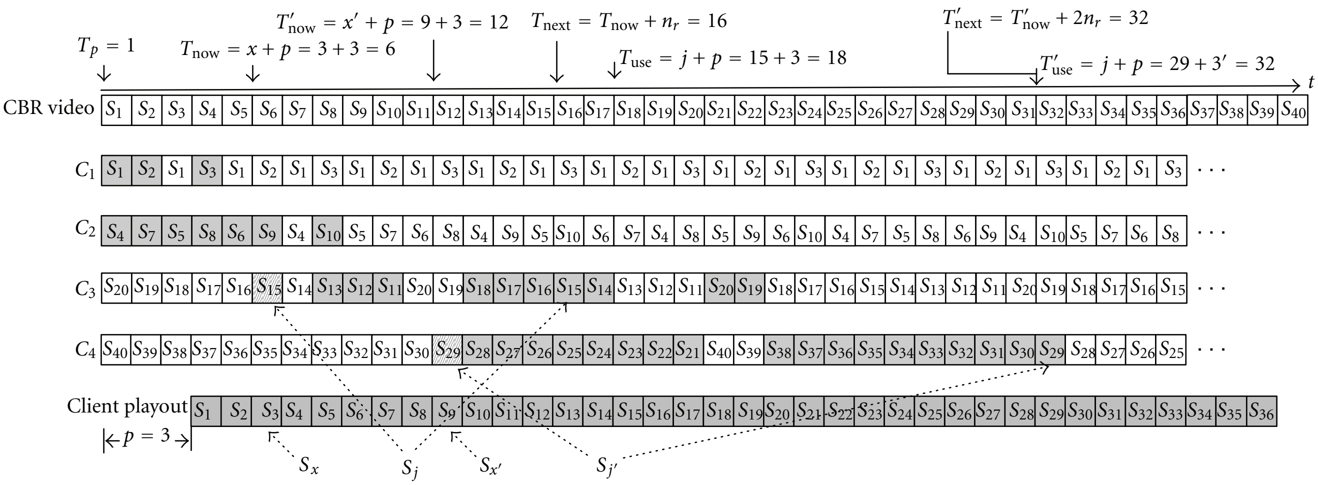


Figure 3: An illustration of segment downloading by RFDPB-3, where $k = 4$, $n_p = 10$, and $N = 40$.

2.2. Improvement of FDPB. By integrating the RFB with FDPB, this work designs the RFB-FDPB (RFDPB) scheme, which gains small low buffer requirements as well as waiting time.

2.2.1. Segment Delivery on the Server Side. Let k be the number of allocated channels. The data transmission rate of each channel is assumed to equal the playback rate b. An RFDPB server broadcasts segments according to the following steps.

(1) The largest number of segments that FDPB-p can transmit using $k-2$ channels is assumed to equal n_p. The RFDPB-p scheme then equally divides a constant-bit-rate (CBR) video into N segments, denoted by $S_1, S_2, \ldots, S_N$ in sequence, where $N = 4n_p$. Let L be the length of the video. The length of a segment, l, thus equals L/N.

(2) Channels C_1 to C_{k-2} periodically send segments S_1 to S_{n_p} according to FDPB-p.

(3) Segments S_{2n_p} to S_{n_p+1} are transmitted on channel C_{k-1}.

(4) Segments S_{4n_p} to S_{2n_p+1} are broadcast on channel C_k.

Figure 2 illustrates the segment delivery by RFDPB-3, where $k = 4$. The server broadcasts segments S_1 to S_{10} on channels C_1 to C_2 by FDPB-3 because FDPB-3 can transmit 10 segments using two channels, as indicated in Figure 1. Due to $n_p = 10$, RFDPB periodically broadcasts segments S_{20} to S_{11} and S_{40} to S_{21} on channels C_3 and C_4.

2.2.2. Segment Reception on the Client Side. Suppose that a client has enough buffers to store portions of a playing video. Let l be a basic time unit during video playing. We also let T_p be the time that the client starts downloading the video and be the origin (i.e., the first time unit) of the time axis throughout the paper. The following steps are involved in playing a video at the client.

(1) The client receives the segments on channels C_1 to C_{k-2} immediately when they are available on networks. Figure 3 further demonstrates the segment downloading, where the segments downloaded by a client are gray.

(2) The segment downloading on channel C_{k-1} is as follows. Suppose that the client first sees segment S_j on channel C_{k-1} at time T_{now} and next segment

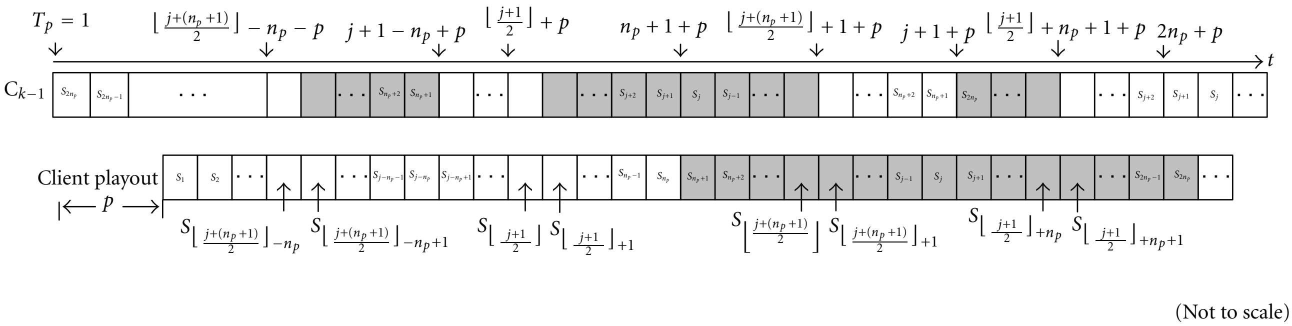


Figure 4: The segment downloading on channel C_{k-1} under RFDPB-p.

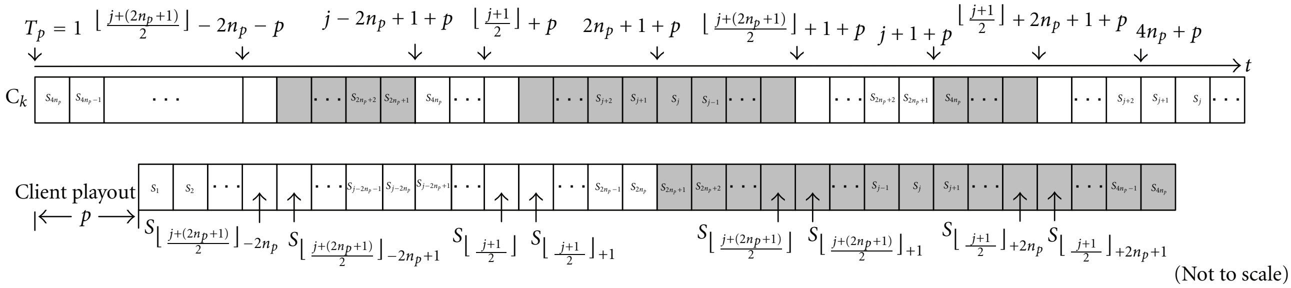


Figure 5: The segment downloading on channel C_k under RFDPB-p.

S_j at time T_{next}. The client is also assumed to play segments S_x and S_j at time T_{now} and T_{use}, respectively. The work in [11, 12] indicates that if $T_{\text{next}} \leq T_{\text{use}}$, the client does not need to receive segment S_j at time T_{now} and actually performs the downloading at time T_{next}, without causing playing interruption. FDPB further reveals that a client has to wait for a fix time unit p to play received segments. Thus, $T_{\text{use}} = j + p$th time unit, $T_{\text{now}} = x + p$th time unit, and $T_{\text{next}} = T_{\text{now}} + n_p$. Substituting these equations into $T_{\text{next}} \leq T_{\text{use}}$, we obtain

$$x + n_p \leq j. \tag{1}$$

If this inequality holds, a client does not receive segment S_j at time T_{now}; otherwise, it immediately downloads it. Suppose that the client first sees segment S_{15} with only diagonal lines on channel C_3 at the 6th time unit, as indicated in Figure 3. Due to $p = 3$, $x = 3$, $j = 15$, and $n_p = 10$, $T_{\text{now}} = x + p = 6$, $T_{\text{use}} = j + p = 18$, and $T_{\text{next}} = T_{\text{now}} + n_p = 16$. Clearly, it is unnecessary to download segment S_{15} at this time unit because inequality (1), $x + n_p = 13 \leq j = 15$, holds. The result also reflects that the client does not have to receive the segment when $T_{\text{next}} = 16 \leq T_{\text{use}} = 18$. When next segment S_{15} colored gray in Figure 3 arrives at the 16th time unit, the inequality is no more true, $13 + 10 > 15$, and the client thus downloads it.

(3) The segments on channel C_k are received in the similar way, as also indicated in Figure 3. Suppose that the client first sees segment $S_{j'}$ on channel C_k at time T'_{now}, and next segment $S_{j'}$ at time T'_{next}. The client is also assumed to play segments $S_{x'}$ and $S_{j'}$ at time T'_{now} and T'_{use}. Similarly, if $T'_{\text{next}} \leq T'_{\text{use}}$, the client does not receive segment $S_{j'}$. Substituting $T'_{use} = j' + p$th time unit, $T'_{\text{now}} = x' + p$th time unit, and $T'_{\text{next}} = T'_{\text{now}} + 2n_p$ into inequality $T'_{\text{next}} \leq T'_{\text{use}}$ obtains

$$x' + 2n_p \leq j'. \tag{2}$$

If the inequality is true, the client skips the downloading of segment $S_{j'}$. Suppose that the client first sees segment S_{29} with only diagonal lines on channel C_4 at the 12th time unit, as indicated in Figure 3. Due to $p = 3$, $x' = 9$, $j' = 29$, and $n_p = 10$, $T'_{\text{now}} = x' + p = 12$, $T'_{\text{use}} = j' + p = 32$, and $T'_{\text{next}} = T'_{\text{now}} + 2n_p = 32$. Clearly, it is unnecessary to download segment S_{29} at this time unit because inequality (2), $x' + 2n_p = 29 \leq j' = 29$, is true. The result also reflects that the client does not have to receive the segment when $T'_{\text{next}} = 32 \leq T'_{\text{use}} = 32$. When next segment S_{29} colored gray in Figure 3 arrives at the 32nd time unit, the inequality is no more true, $29 + 20 > 29$, and the client thus downloads it.

(4) The client plays the video in the order of $S_1, S_2, \ldots, S_N$ at time $T_p + p$.

(5) Stop loading data from networks when all the segments have been received.

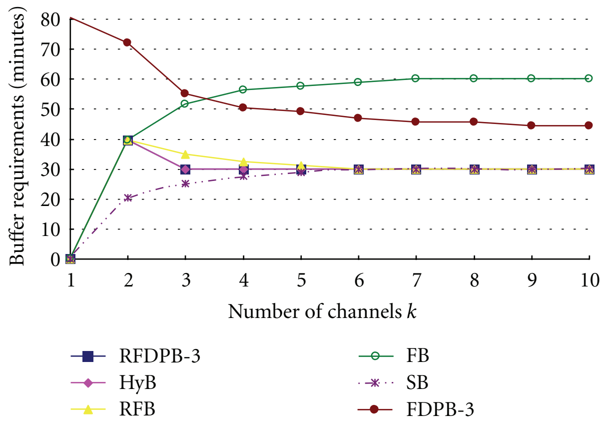


Figure 6: Comparison of the maximum buffer requirements in the number of minutes of a 120-minute video.

Table 1: The maximum buffering space required by different schemes in the percentage of video size using k channels.

k	1	2	3	4	5	6	7	8	9	10
RFDPB-3	0	33	25	25	25	25	25	25	25	25
HyB	0	33	25	25	25	25	25	25	25	25
RFB	0	33	29	27	26	25	25	25	25	25
FB	0	33	43	47	48	49	50	50	50	50
SB	0	17	21	23	24	25	25	25	25	25
FDPB-3	67	60	46	42	41	39	38	38	37	37

2.2.3. Workable Verification. This section describes that the RFDPB scheme guarantees continuous playing on the client side. Because the segment broadcasting on channels C_1 to C_{k-2} is based on FDPB [9], RFDPB is workable for these channels. We next prove that the segment arrangements on channels C_{k-1} and C_k also ensure a client to continuously play a video. FDPB further indicates that a video server must broadcast segment S_j at least once in every $j+p-1$ time units to keep on-time data delivery on the client side. For the RFDPB scheme, a server broadcasts segment S_j on channel C_{k-1} in every n_p time units, where $n_p+1 \le j \le 2n_p$. Because $j+p-1 > n_p$, an RFDPB client can receive video segments on channel C_{k-1} on time. Similarly, the RFDPB scheme requires a server to transmit segment $S_{j'}$ on channel C_k in every $2n_p$ time units, where $2n_p+1 \le j' \le 4n_p$. Because $j'+p-1 > 2n_p$, the segment delivery on this channel is also on time. Accordingly, RFDPB ensures continuous video playing at the client.

3. Analysis and Comparison

Before investigating entire buffer requirements for RFDPB, this paper analyzes the segment downloading on channel C_{k-1} first. For RFDPB-p, a client plays the segments received from this channel during time units n_p+1+p to $2n_p+p$. The possible time to download the segments is during the pth to $2n_p+p$th time units. Suppose that a client sees segment S_j at the n_p+1+pth time unit. Inequality (1) makes a complete segment-downloading diagram for channel C_{k-1}, as indicated in Figure 4, where the segments downloaded by the client are colored gray. The figure shows that the client does not continuously download segments, and some segments are skipped. The segments are not downloaded during the first to $\lfloor (j+(n_p+1))/2 \rfloor - n_p + p$th time units, during the $j+1-n_p+p$th to $\lfloor (j+1)/2 \rfloor + p$th time units, during the $\lfloor (j+(n_p+1))/2 \rfloor + 1 + p$th to $j+p$th time units, and during the $\lfloor (j+1)/2 \rfloor + n_p + 1 + p$th to $2n_p+p$th time units, respectively. Such results reflect that the client can delay downloading these segments by n_p time units, due to (1). Similarly, (2) also makes a complete segment-downloading diagram for channel C_k, as indicated in Figure 6, where a client is assumed to see segment S_j at the $2n_p+1+p$th time unit.

The RFDPB scheme uses the same methodology as HyB, combining a nonharmonic scheme and the RFB to reduce clients' waiting time and buffering spaces (Figure 5). For segment broadcasting on channels C_1 to C_{k-2} using the nonharmonic scheme, channels C_{k-1} and C_k broadcast the other segments according RFB. Yu's work [12] has proved HyB only buffer 25% of video size for $k \ge 3$, where k is the number of broadcasting channels. The proof of the maximum number of segments buffered by a client for RFDPB is similar to that of HyB, so this paper neglects it. Therefore, let $B(k)$ be the maximum number of segments buffered by a client, where k is the number of broadcasting channels. For RFDPB,

$$\begin{aligned} B(1) &= 0 \\ B(2) &= 1 \\ B(k) &= n_p, \quad k \ge 3. \end{aligned} \tag{3}$$

Table 2: The maximum numbers of segments, N, offered by different schemes.

k	1	2	3	4	5	6	7	8	9	10
FDPB-6	7	25	71	186	485	1286	3425	9195	24790	67054
RFDPB-6	1	3	28	100	284	744	1940	5144	13700	36780
HyB	1	3	4	12	36	100	292	804	2260	6088
FB (RFB)	1	3	7	15	31	63	127	255	511	1023
SB	1	3	7	15	31	63	127	255	511	1023

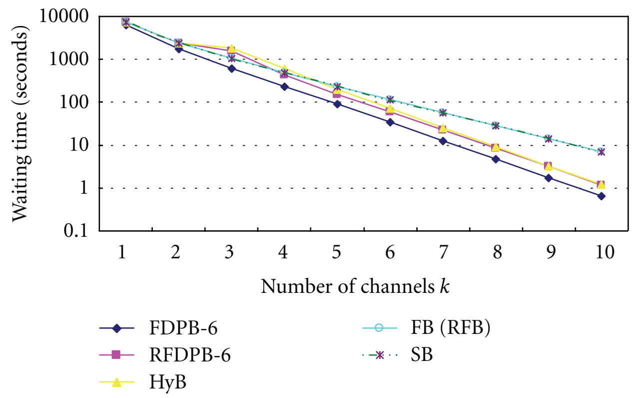


Figure 7: The maximum waiting time incurred on new clients at different numbers of channels (L = 120 minutes).

Table 3: The waiting time (seconds), offered by RFDPB-9, RFDPB-6, HyB, and RFDPB-3 schemes.

k	1	2	3	4	5	6	7	8	9	10
RFDPB-9	7200	2400	1350	385.71	139.66	52.6	19.9	7.47	2.79	1.04
RFDPB-6	7200	2400	1542.86	432	152.11	58.06	22.27	8.4	3.15	1.17
HyB	7200	2400	1800	600	200	72	24.66	8.96	3.19	1.18
RFDPB-3	7200	2400	1800	540	192.86	72.97	28.57	11.07	4.19	1.58

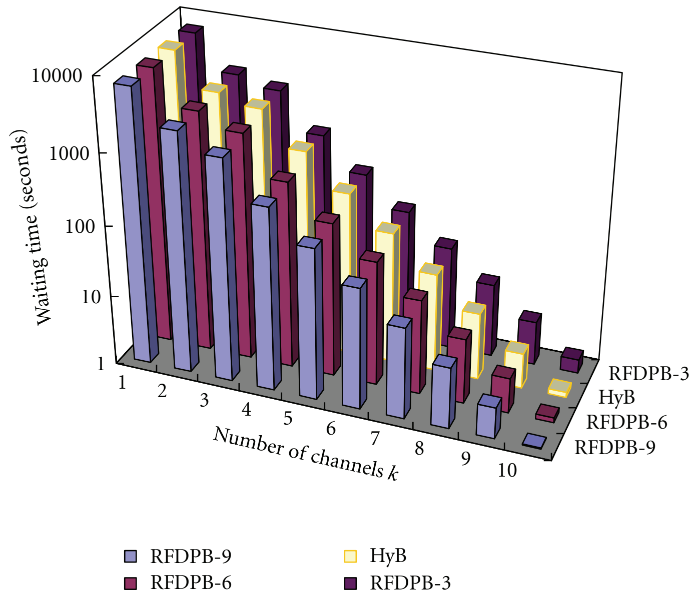


Figure 8: The maximum waiting time (RFDPB and HyB) incurred on new clients at different numbers of channels (L = 120 minutes).

The previous analysis indicates that RFDPB buffers at most $n_r/N = 25\%$ of video size. Table 1 lists the comparison of maximum buffering spaces required by RFDPB-3, FB, SB, RFB, FDPB-3, and HyB. RFDPB-3 reduces the buffering space up to 32.4% when compared to FDPB-3 and up to 50% when compared to FB and has the same good result with the other schemes. Given a video of 120 minutes, Figure 6 shows the buffer requirements at different channels. To understand how well the RFDPB scheme performs on clients waiting time, this work calculates the values of N offered by RFDPB-6, SB, FB, RFB, FDPB-6, and HyB given different numbers of server channels, as listed in Table 2. The inverse of N offered by each scheme reflects the waiting time for a new client to start his/her VOD service. Figure 7, which is drawn in a logarithmic scale, shows that RFDPB-6 performs close to FDPB-6 stably. For $k \geq 4$, the RFDPB-6 outperforms all the schemes, except the FDPB-6. To understand the relationship between RFDPB and HyB schemes, this work calculates clients' waiting time of RFDPB-9, RFDPB-6, HyB, and RFDPB-3, as listed in Table 3. Figure 8 shows that, for $p \geq 6$, RFDPB-p outperforms HyB. The previous comparisons clearly indicate that RFDPB exhibits a good tradeoff between client buffering spaces and waiting time.

4. Conclusions

This paper presents an improved version of FDPB for efficient periodic broadcasting of popular videos. The proposed scheme takes advantage of the FDPB and the RFB schemes to obtain small waiting time and low buffer demand. Through mathematical analysis, we prove the applicability of this scheme by demonstrating that client playback continuity is guaranteed. Given a bandwidth of 5 channels, the new scheme reduces the broadcast latency by as much as 24% when compared to HyB and 34.5% when compared to RFB and SB. The buffer requirements for these schemes are about 25% of video size.

References

[1] A. Dan, D. Sitaram, and P. Shahabuddin, "Dynamic batching policies for an on-demand video server," *Multimedia Systems*, vol. 4, no. 3, pp. 112–121, 1996.

[2] D. Saparilla, K. W. Ross, and M. Reisslein, "Periodic broadcasting with VBR-encoded video," in *Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societie (INFOCOM '99)*, pp. 464–471, March 1999.

[3] K. C. Almeroth and M. H. Ammar, "The use of multicast delivery to provide a scalable and interactive video-on-demand service," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 6, pp. 1110–1122, 1996.

[4] L-S. Juhn and L. M. Tseng, "Fast data broadcasting and receiving scheme for popular video service," *IEEE Transactions on Broadcasting*, vol. 44, no. 1, pp. 100–105, 1998.

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

[6] Z.-Y. Yang, L. S. Juhn, and L. M. Tseng, "On optimal broadcasting scheme for popular video service," *IEEE Transactions on Broadcasting*, vol. 45, no. 3, pp. 318–322, 1999.

[7] Y.–C. Tseng, M. H. Yang, and C. H. Chang, "A recursive frequency-splitting scheme for broadcasting hot videos in VOD service," *IEEE Transactions on Communications*, vol. 50, no. 8, pp. 1348–1355, 2002.

[8] J.-P. Sheu, H. L. Wang, C. H. Chang, and Y. C. Tseng, "A fast video-on-demand broadcasting scheme for popular videos," *IEEE Transactions on Broadcasting*, vol. 50, no. 2, pp. 120–125, 2004.

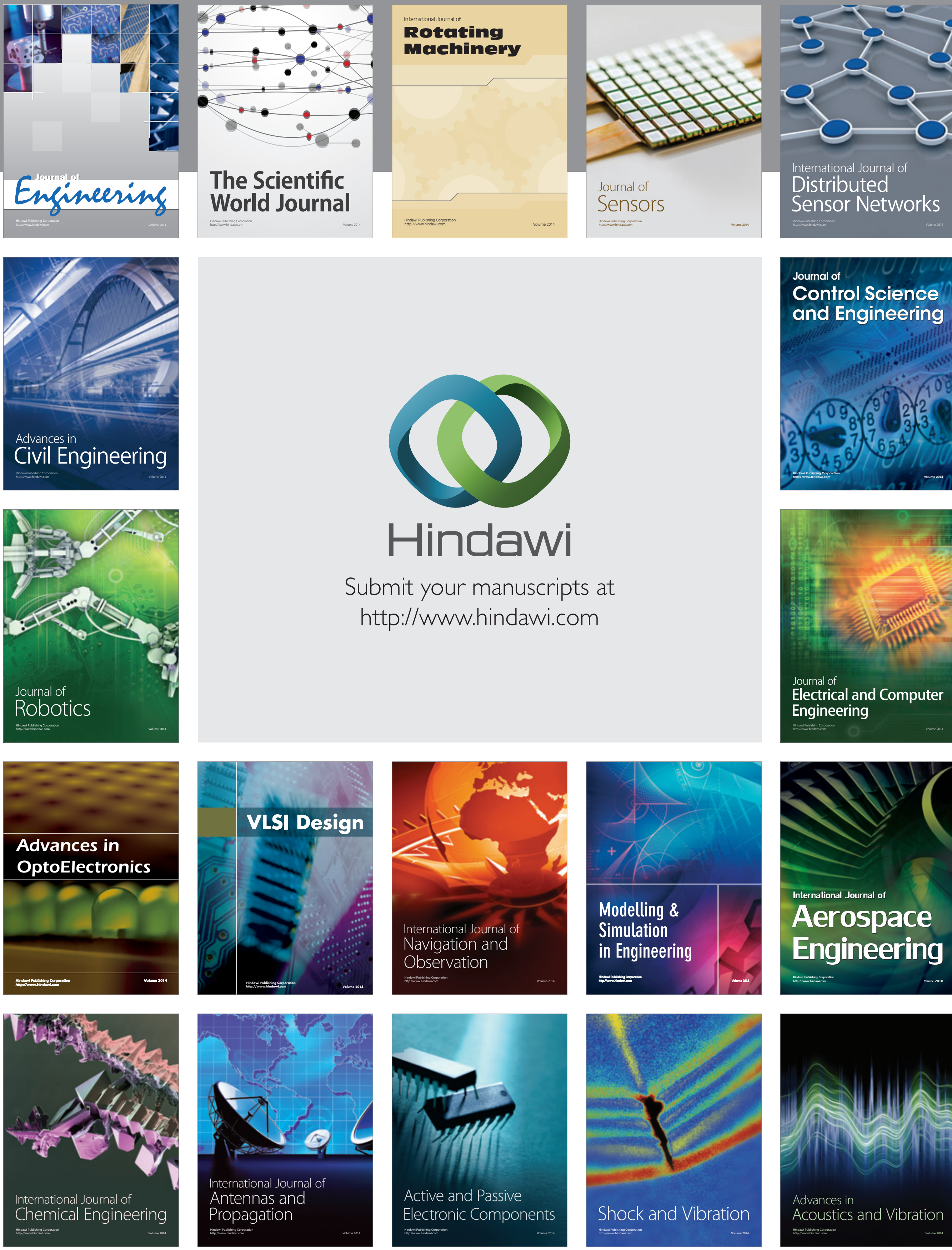
[9] J. F. Paris, "A fixed-delay broadcasting protocol for video-on-demand," in *Proceedings of the 10th International Conference on Computer Communications and Networks (ICCCN '01)*, pp. 418–423, Scottsdale, Ariz, USA, October 2001.

[10] L.-S. Juhn and L. M. Tseng, "Staircase data broadcasting and receiving scheme for hot video service," *IEEE Transactions on Consumer Electronics*, vol. 43, no. 4, pp. 1110–1117, 1997.

[11] H.-F. Yu, H. C. Yang, and L. M. Tseng, "Reverse Fast Broadcasting (RFB) for video-on-demand applications," *IEEE Transactions on Broadcasting*, vol. 53, no. 1, pp. 103–110, 2007.

[12] H.-F. Yu, "Hybrid broadcasting with small buffer demand and waiting time for video-on-demand applications," *IEEE Transactions on Broadcasting*, vol. 54, no. 2, Article ID 4433980, pp. 304–311, 2008.

[13] H.-F. Yu, P. H. Ho, and H. C. Yang, "Generalized Sequence-Based and Reverse Sequence-Based Models for Broadcasting Hot Videos," *IEEE Transactions on Multimedia*, 2008.


Journal of Engineering
The Scientific World Journal
International Journal of Rotating Machinery
Journal of Sensors
International Journal of Distributed Sensor Networks
Advances in Civil Engineering
Hindawi
Submit your manuscripts at http://www.hindawi.com
Journal of Control Science and Engineering
Journal of Robotics
Journal of Electrical and Computer Engineering
Advances in OptoElectronics
VLSI Design
International Journal of Navigation and Observation
Modelling & Simulation in Engineering
International Journal of Aerospace Engineering
International Journal of Chemical Engineering
International Journal of Antennas and Propagation
Active and Passive Electronic Components
Shock and Vibration
Advances in Acoustics and Vibration