International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:8, 2007

A Broadcasting Strategy for Interactive Video-on-Demand Services

Yu-Wei Chen, and Li-Ren Han

***Abstract*—**In this paper, we employ the approach of linear programming to propose a new interactive broadcast method. In our method, a film S is divided into n equal parts and broadcast via k channels. The user simultaneously downloads these segments from k channels into the user's set-top-box (STB) and plays them in order. Our method assumes that the initial p segments will not have fast-forwarding capabilities. Every time the user wants to initiate d times fast-forwarding, according to our broadcasting strategy, the necessary segments already saved in the user's STB or are just download on time for playing. The proposed broadcasting strategy not only allows the user to pause and rewind, but also to fast-forward.

***Keywords*—**Broadcasting, Near Video-on-Demand (VOD), Linear Programming, Video-Cassette-Recorder (VCR) Functions, Waiting Time.

I. Introduction

THE most typical service of Video-On-Demand (VOD) is when the customer request service of watching film, then establishes a dedicated stream to user. Users can carry out the video-cassette-recorder (VCR) functions (for example fast forward, stop, forward, rewind, pause, playing…etc.). This system is so-called true VOD system.

The main problem of true VOD system lies in the requirement for extremely large bandwidth for the transmission of sizable information. For reducing the high bandwidth requirement, many researchers proposed various alternative methods. The broadcasting methods [1]-[9] are well-known approach for relieving the stress of the high demand on bandwidth.

The broadcasting approach works by first dividing a film into several segments, then allowing the server to play them repeatedly in a few specified channels. Should a user wish to view the film, the maximum waiting time will be the length of one such segment. The user must also have enough buffer space to store the downloaded segments for continuous playback. This system is also known as the Near VOD system.

The broadcasting method sacrifices some VCR functions, especially fast forward function, for saving the bandwidth of network. According to the definition of VOD, we think that the VCR function providing is also the important factor for consideration in design. Of course, the buffer size is also an important aspect when providing VCR capabilities; the bigger the buffer, the more segments of the film can be stored, implying the increased ability to support more VCR functions.

In recent years, an increasing number of experts and scholars have gone into this area of research [10]-[13] use the broadcasting as a basis, proposing ways to support VCR functions more completely to achieve a more interactive VOD system. References to these related research motivated the intention of our research: to propose a new broadcasting method that not only reduces the users' waiting time, but also support more complete VCR functions, especially fast-forwarding.

In this paper, we propose a new broadcasting strategy for more complete VCR functions to be executed. The feature is to provide sufficient space for pre-storage concept. To achieve this, the user will need a relatively larger buffer. Our method makes the same assumption as [11]-[13]: that the capacity of harddisks will increase and prices will become more affordable, enough to support the space needed to save this type of interactive VOD system. Thus we think that in the Near VOD system, shrinking waiting time and the provision of VCR capabilities are more important than the size of the buffer, and become the two most important factors.

In this paper, we employ the approach of linear programming to propose a new broadcast method with support for VCR functions. In our approach, a film S is divided into n equal parts and broadcast via k channels. The user simultaneously downloads the segments from k channels into the set-top-box (STB) and plays them in order. The initial p segments will not have fast-forwarding capabilities. With respect to films available on the market, commercials and statements of copyright will always appear in the initial parts. Of course, in view of the providers' business considerations, not wanting the front parts of the film to be skipped but to be watched are a logical assumption. Every time the user wants to initiate d times fast-forwarding, according to our broadcasting strategy, the necessary segments already saved in the user's STB or are just download on time for playing. The proposed broadcasting strategy not only allows the user to pause and rewind, but also to fast-forward.

This paper is organized as follows. In Section 2, we briefly introduce the related works. In Section 3, we propose and analyze the new method based on the approach of linear programming for supporting more complete VCR functions. Finally, some conclusions are given in Section 4.

Manuscript received October 19, 2007. This work was supported in part by NSC93-2213-E-156-003.

Y. W. Chen is with Department of Computer and Information Science, Aletheia University, No. 32, Chen-Li Street, Tamsui, Taipei, 25103 Taiwan (corresponding author to provide phone: 886-2-26212121 Ext. 5222; fax: 886-2-26260518; e-mail: ywchen@email.au.edu.tw).

Li-Ren Han, was with Graduate School of Management Science, Aletheia University, No. 32, Chen-Li Street, Tamsui, Taipei, 25103 Taiwan.

II. THE RELATED WORKS

In this section, we introduce four methods of interactive broadcasting methods: Providing VCR Function with Active Buffer Management Scheme [10], Interactive Pagoda Broadcasting [11], VCR-Oriented Periodic Broadcasting Scheme (VPB) [12], and Broadcasting Scheme with Supporting VCR functions [13].

■ Providing VCR Functions with Active Buffer Management Scheme [10]

Assume we want to play a film of length L through k=4 channels. As Figure 1 shows, this film will be divided into four segments $\{S_0, S_1, S_2, S_3\}$; when time t_0 starts, the whole film will repeatedly transmit in order on channel C_0. After 1/4 of the film's running time, the same film will also be transmitted in order on channel C_1. After 2/4 of the film's running time, the same film will also be transmitted in order on channel C_2. Finally, after 3/4 of the film's running time, the same film will be transmitted in order on channel C_3. We can observe from Fig. 1 that in any one time period, all four segments of the film are playing on four separate channels.

Due to the number of segments being the same as channels in this broadcasting strategy and broadcasting in the same time span, the buffer is not needed for skipping to and playing a certain part of the film. When user wants to skip from time t to film segment S_j, the channel which the segment is playing on must be found. Only then can the jump VCR function be made. Thus we can make use of the equation provided by [11] to know segment S_j playing on channel C_x at time t, $x = ((\lfloor (t - t_0)/(L/k) \rfloor - j) \bmod k$. The equation is as follows:

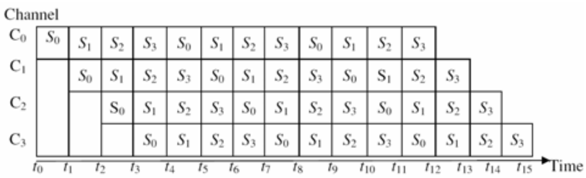

Fig. 1 Providing VCR functions with active buffer management scheme

■ Interactive Pagoda Broadcasting Scheme (IPB) [11]

This method mainly uses the Pagoda broadcasting method to achieve support for VCR functions. The method assumes the bulk of users have not watched the film yet, thus the frequency of commands for pause and rewind will be bigger than fast-forward. On the basis that storage mediums like harddisks are increasingly affordable, that the user already has enough temporary storage to store the whole film is also assumed. This then implies the user only has to read for their own STB instead of downloading from the server after activating pause and rewind functions. This means the server only uploads information when the user activates the fast-forward function. If the user needs to fast-forward, a request will be sent to the server; this request will let the server know which segment the user lacks and allocate an extra stream to send the missing segments to the user. Also, if the user is unable to receive the played segment in time after the fast-forward command has been sent to the server, the segment will be missed. Although this method has a shorter waiting time, extra bandwidth is still needed to support the specified VCR function like fast-forwarding. The fast-forwarding function ceases to be available when the bandwidth is fully utilized.

■ VCR-Oriented Periodic Broadcasting Scheme (VPB) [12]

Kwon and Yeom [12] proposed a VCR-oriented periodic broadcasting (VPB), which is based on the stagger method as well. The basic concept of the VPB is described as follows. Consider the total bandwidth for transmitting a film is n-times the rate of use of the film during normal playback and the largest speed of fast-forward/fast-backward is d-times where d is an integer. In the VPB, the bandwidth of each channel is n/d times the rate of use of the film during normal playback. Every n/d successive segments are broadcast on each channel. Since the VPB broadcasts all the segments every one time slot, it guarantees a service latency of one time slot. Fig. 2 shows the VPB when n=4 and d=2. Two successive segments S_0 and S_1 (S_2 and S_3) are broadcast on each time slot $[t_i, t_{i+1}]$.

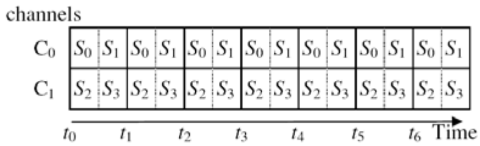

Fig. 2 A VCR-oriented periodic broadcasting with two-times fast-forwarding/fast-backwarding

■ Broadcasting Scheme with Supporting VCR functions [13]

In [13], a film S with length L is split into n equal-length segments and played through k channels. The user simultaneously downloads the segments from k channels into STB and plays them in order. The initial p segments in the film S will not have fast-forwarding capabilities. Define n_i as the number of segments played in C_i. The basic concept of [13] is described as follows. The initial i channels ($0 \le i \le \lfloor \log_2 p \rfloor$), every C_i needs to be delegated 2^i segments. Segment S_z, $z = p + \left\lfloor \left(\sum_{j=0}^{i-1} n_j - p + 1 \right) \right\rfloor$ will play on C_i for $i > \lfloor \log_2 p \rfloor$. Since the initial p segments cannot support fast-forwarding but the rest can support d times that, S_z only needs every $p + \left\lfloor \left(\sum_{j=0}^{i-1} n_j - p + 1 \right) / d \right\rfloor$ times of segment playing time to ensure smooth playback. Chen *et al*. [13] deduce the equation for n_i as follows:

$$n_i = \begin{cases} 2^i \,, & 0 \le i \le \lfloor \log_2 p \rfloor \\ p + \left\lfloor \left(\left(\sum_{j=0}^{i-1} n_j \right) - p + 1 \right) / d \right\rfloor , & \lfloor \log_2 p \rfloor < i \le k - 1 \end{cases} \quad (1)$$

They add up all segments played on all k channels to get the total n number of segments, i.e., $n=\sum_{i=0}^{k-1} n_i$. The longest waiting time is $T=L/n$.

III. Proposition of Broadcasting Method

This section comprises of three subsections. The first subsection defines the symbols used. The second subsection presents the proposed interactive broadcasting strategy. The last subsection shows a comparison.

A. Definition of Symbols

For arbitrary broadcasting method, the customers need to have the enough buffers and capacity of receiving multi-channels such that they can play most VCR functions, but without fast-forward function [13]. Therefore, this research will aim at to fast-forward to describe, but the rest VCR functions then no longer mention.

In our method, the film S is first divided into n equal lengths $\{S_1, S_2, \ldots, S_n\}$ played on k channels; the value of n will be obtained in the second subsections of this section. The bandwidth of each channel is the same as the consumption rate for playing a film. In addition, we assume that the user can simultaneously receive contents from k channels.

The symbols used in our broadcasting method are defined as follows:

Number of channels: k channels.
Symbol of channel: C_j for $1 \leq j \leq k$.
The total number of segments of film: n segments.
The segment of film: S_i for $1 \leq i \leq n$.
The length of film: L.
The longest waiting time: $T= L/ n$.
Supporting several times to fast forward: d times fast-forward.
The number of segments without fast forward: p.
The time interval of one time slot: $[t_i, t_{i+1}]$.
The length of one time slot: t.

Since before watching segment S_i, the first i-1 segments S_1, S_2, …, S_{i-2} and S_{i-1} must be watched, segment S_i must appear at least once every $B(i)$, $B(i)=i$ if $1 \leq i \leq p$ and $B(i)=\lfloor p+(i-p)/d \rfloor$ if $i>p$, time slots to ensure continuous playback.

B. Proposed Broadcasting Method

The second subsection presents a new interactive broadcasting method, based on the approach of linear program. Only given four parameters p, d, and k, our method can deduce out the "the initial p segments of film without fast forward but the rest segments supporting fast forward." The users can watch the complete film and play VCR functions at any time. Let us first consider the following linear program:

Maximize $\displaystyle\sum_{i=1}^{n}\sum_{j=1}^{k} x_{ij}$

Subject to

$$\sum_{j=1}^{k} x_{ij} = 1, \ \forall\, i = 1,2,\ldots,n$$

$$\sum_{i=1}^{n} x_{ij} \times \frac{1}{B(i)} \leq 1, \forall\, 1 \leq j \leq k$$

$$x_{ij} = \begin{cases} 1, & \text{if segment } S_i \text{ is broadcast on } C_j \\ 0, & \text{otherwise} \end{cases}$$

$$B(i) = \begin{cases} i, & 1 \leq i \leq p \\ p+(i-p)/d, & i > p \end{cases}$$

$$i = 1,2,\ldots,n, \ \ j = 1,2,\ldots,k \ .$$

According to the above-mentioned linear program, we can determine which segments could be broadcast on C_j for $1 \leq j \leq k$.

For example, consider the case of p=2, d=2, k=4. Applying the above-mentioned linear program, we can deduce out the "the initial two (p=2) segments without fast forward but the rest segments supporting two-times (d=2) fast forward." The maximum number of segments divided is 13. Table I shows the necessary bandwidth in channel of the 13 segments and which channel S_i is broadcast.

In the example of the case of p=2, d=2, and k=4, Fig. 3 illustrates the schedule of 13 segments on four channels.

In what follows, we introduce how to broadcast the segments on the same channel. We apply the channel C_3 to explain the concept. In Table 1, segments S_2, S_4, and S_{11} must appear at least once every 2, 3, and 6.5 time slots, respectively. It implies that segments S_2, S_4, and S_{11} must occupy at least 0.5, 0.34, and 0.16 of one channel bandwidth, respectively. Therefore, each time slot is partitioned as three subslots with lengths 0.5, 0.34, and 0.16 of one time slot for broadcasting S_2, S_4, and S_{11}, respectively. Fig. 4 illustrates how segments S_2, S_4, and S_{11} are broadcasted on channel C_3. The number, e.g, 1-16%, means the corresponding portion of one segment. The number "97-12%" in time slot 7 means the portions 97-100% and 1-12%.

C. Analysis and Comparison

In Table II, the total number of film segments in [13] and ours are clearly shown when number of channels k is from one to ten, S_1 to S_3 cannot support fast-forwarding but the rest can support two times that. Table II shows the results of our proposed method, as the number of channels increase, the total number of film segments that can be divided increases as well and the longest waiting time becomes shorter. But when the speed of fast-forwarding required by the user increases, the number of film segments that can be divided decreases.

time slot	1	2	3	4
C_1	S_1	S_1	S_1	S_1
C_2	Part of $\{S_3, S_6, S_8, S_{12}\}$	Part of $\{S_3, S_6, S_8, S_{12}\}$	Part of $\{S_3, S_6, S_8, S_{12}\}$	Part of $\{S_3, S_6, S_8, S_{12}\}$
C_3	Part of $\{S_2, S_4, S_{11}\}$	Part of $\{S_2, S_4, S_{11}\}$	Part of $\{S_2, S_4, S_{11}\}$	Part of $\{S_2, S_4, S_{11}\}$
C_4	Part of $\{S_5, S_7, S_9, S_{10}, S_{13}\}$	Part of $\{S_5, S_7, S_9, S_{10}, S_{13}\}$	Part of $\{S_5, S_7, S_9, S_{10}, S_{13}\}$	Part of $\{S_5, S_7, S_9, S_{10}, S_{13}\}$

Fig. 3 The schedule of 13 segments on four channels in the case of *p*=2, *d*=2, and *k*=4

Time slot 1			Time slot 2			Time slot 3			Time slot 4			Time slot 5		
S_2 1-50%	S_4 1-34%	S_{11} 1-16%	S_2 51-100%	S_4 35-78%	S_{11} 17-32%	S_2 1-50%	S_4 79-100%	S_{11} 33-48%	S_2 51-100%	S_4 1-34%	S_{11} 49-64%	S_2 1-50%	S_4 35-78%	S_{11} 65-80%

Time slot 6			Time slot 7			Time slot 8			Time slot 9			Time slot 10		
S_2 51-100%	S_4 79-100%	S_{11} 81-96%	S_2 1-50%	S_4 1-34%	S_{11} 97-12%	S_2 51-100%	S_4 35-78%	S_{11} 13-28%	S_2 1-50%	S_4 79-100%	S_{11} 29-44%	S_2 51-100%	S_4 1-34%	S_{11} 45-60%

Fig. 4 Broadcasts S_2, S_4, and S_{11} on channel *C*3 in the case of *p*=2, *d*=2, and *k*=4

TABLE I
THE RESULTS OF APPLYING LINEAR PROGRAM ON THE CASE OF *p*=2, *d*=2, AND *k*=4

Segments	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}	S_{13}
$B(i)$	1	2	2.5	3	3.5	4	4.5	5	5.5	6	6.5	7	7.5
$1/B(i)$	1	0.5	0.4	0.34	0.286	0.25	0.223	0.2	0.182	0.167	0.16	0.15	0.134
Which channel S_i is broadcast	C_1	C_3	C_2	C_3	C_4	C_2	C_4	C_2	C_4	C_4	C_3	C_2	C_4

TABLE II
A COMPARISON OF THE TOTAL NUMBER OF SEGMENTS BETWEEN [13] AND OURS
WHEN S_1 TO S_3 CANNOT SUPPORT FAST-FORWARDING
BUT THE REST C AN SUPPORT TWO TIMES THAT

Segments that Cannot Fast-Forward / *k* (channels)	S_1 [13]	S_1 Ours	S_1, S_2 [13]	S_1, S_2 Ours	S_1, S_2, S_3 [13]	S_1, S_2, S_3 Ours
1	1	1	1	1	1	1
2	2	3	3	3	3	3
3	4	5	6	6	6	8
4	7	9	10	13	11	14
5	11	15	16	23	18	26
6	17	28	25	38	29	46
7	26	47	39	66	45	81
8	40	79	60	111	69	136
9	61	132	91	186	105	227
10	92	222	138	311	159	379

IV. Conclusion

In this paper, we have proposed a new broadcasting method for interactive VOD services, based on the approach of linear program. Our method has shorter waiting time for customers than the previous method [13] while providing the same VCR functions. When the user wants to initiate fast-forwarding, according to our broadcasting strategy, the required segments were saved in the user's STB or those are being continuously downloaded from channels. Consequently, the user can play the portions of the film needed in time when fast-forwarding is initiated.

References

[1] C. C. Aggarwal, J.L. Wolf, and P.S. Yu, A Permutation-Based Pyramid Broadcasting Scheme for Video-on-Demand Systems, In IEEE Proceedings of the International Conference on Multimedia Computing and Systems, (Jun 1996) 118-126.
[2] T. Chiueh and C. Lu, A Periodic Broadcasting Approach to Video-on-Demand Service, International Society for Optical Engineering, 2615 (Oct 1995) 162-169.
[3] L. Gao, J. Kurose, and D. Towsley, Efficient Schemes for Broadcasting Popular Videos, In International Workshop on Network and Operating Systems Support for Digital Audio and Video, (Aug. 1998) 317-329.
[4] K.A. Hua and S. Sheu, Skyscraper broadcasting: A New Broadcasting Scheme for Metropolitan Video-on-Demand Systems, In ACM SIGCOMM'97, 27 (Sept. 1997) 89-100.
[5] L.-S. Juhn and L.-M. Tseng, Enhanced Harmonic Data Broadcasting and Receiving Scheme for Popular Video Service, IEEE Transactions on Consumer Electronics, 44 (May 1998) 343-346.
[6] L.-S. Juhn and L.-M. Tseng, Fast Data Broadcasting and Receiving Scheme for Popular Video Service, IEEE Transactions on Broadcasting, 44 (Mar 1998) 100-105.
[7] J.-F. Paris, S.-W. Carter, and D.-D. Long, A Hybrid Broadcasting Protocol for Video on Demand, In Multimedia Computing and Networking, (1999) 317-326.
[8] Y.-C. Tseng, M.-H. Yang, and C.-H. Chang, A Recursive Frequency-Splitting Scheme for Broadcasting Hot Videos in VOD Service, IEEE Trans. on Communications, 50 (Aug. 2002) 1348-1355.
[9] Y. W. Chen, An Enhanced Recursive Frequency Splitting Broadcasting Algorithm for Near Video-on-Demand Services, Information Processing Letters, 92 (2004) 299-302.
[10] Z. Fei, I. Kamel, S. Mukherjee, and M. H. Ammar, Providing Interactive Functions for Staggered Multicast Near Video-On-Demand Systems, In Proceedings of the IEEE International Conference on Multimedia Computing and Systems, 2 (1999) 949-953.
[11] J.-F. Paris, An Interactive Broadcasting Protocol for Video-on-Demand, Proceedings of the 20th IEEE International Performance, Computing, and Communications Conference (IPCCC 2001), Phoenix, AZ, Apr., (2001) 347-353.
[12] J. B. Kwon and H. Y. Yeom, VCR-oriented Video Broadcasting for Near Video-on-Demand Services, IEEE Trans. on Consumer Electronics, 49 (2003) 1106-1113.
[13] Y. W. Chen and Y. T. Lee, A Broadcasting Scheme with Supporting VCR Functions for Near Video-on-Demand Systems, Computer Communications, 29 (2006) 257-267.