A Novel Broadcasting Scheme for Considering the Limitation of STB Capability

Yu-Wei Chen[1] and Chui-Chang Chiu

Department of Computer and Information Science
Aletheia University
NO. 32, Chen-Li Street, Tamsui, Taipei, 25103 Taiwan
ywchen@email.au.edu.tw

Abstract*—This paper presents a novel broadcasting method for considering the limitation of STB capability. Assume Y channels can be used to transmit one video but STB at the user's end can only simultaneously receive and process R channels ($Y \geq R \geq 1$). After completion of receiving video's segments transmitted on one channel, the STB is switched to a new channel to receive the other segments. Our broadcasting method can ensure proper playback without breaks between any two sequential segments.*

Keywords: Broadcasting, Video on Demand (VOD), Set Top Box (STB)

1. 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, i.e., fast forward, stop, forward, rewind, pause, playing…etc. Such a system is so-called true VOD system.

The true VOD system's main problem lies in the requirement for extremely large bandwidth for the transmission of sizable information. For the sake of the solving the problem of the need for high bandwidth to VOD, lots of scholars bring up an alternative plan, broadcasting [1]-[16].

The broadcasting approach works by first dividing the film into several segments, then allowing the server to repeatedly play them 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. Due to the usage of different methods to divide the film into different number of segments, there will be a difference in waiting time. The user must also have enough buffer space to store the downloaded segments for continuous playback to occur. This system is also known as the Near VOD system.

Previous broadcasting methods, fast broadcasting [6], pagoda [10], new pagoda [11], recursive frequency-splitting (RFS) [15], made the same assumption that the client's STB can receive and process the segments from all channels. This motivates us to propose a new broadcasting method.

In this paper, we present a novel broadcasting method for considering the limitation of STB capability. Assume Y channels can be used to transmit one video but STB at the user's end can only simultaneously receive and process R channels ($Y \geq R \geq 1$). After completion of receiving video's segments transmitted on one channel, the STB is switched to a new channel to receive the other segments. Our broadcasting method can ensure proper playback without breaks between any two sequential segments.

This paper is organized as follows. In Section 2, we briefly introduce the related works. In Section 3, we present the proposed broadcasting method. Finally, some conclusions are addressed in Section 4.

2. The Related Works

2.1. Fixed-Length Segment-Scheduling Problem

Besides staggered broadcasting method [2], the other methods fast broadcasting [6], pagoda [10], new pagoda [11], recursive frequency-splitting (RFS) [15], are all problems of Fixed-Length Segment-Scheduling. In such a problem, the film S is divided into n equal segments $\{S_1, S_2, \ldots, S_n\}$ and played back through Y channels. These methods [6][10][11][15] assume that each channel's bandwidth is just sufficient to fit the usage rate of the film during normal playback and that the user can receive and process segments from Y

[1]**Corresponding author. Tel.: +886-2-26212121 Ext. 5222;**
Fax: +886-2-26205236.
E-mailaddress:ywchen@email.au.edu.tw,
cyw533@ms23.hinet.net
Support by NSC92-2213-E-156-003

channels simultaneously. Since before watching film segment S_i the first i-1 segments S_1, S_2, …, S_{i-2}, and S_{i-1} have to be watched, segment S_i has to at least appear once in the time every i segments is played to ensure proper playback without breaks in between. Therefore segment S_i at least needs to use up $1/i$ of channel bandwidth. Under the condition of uninterrupted playback, the largest number of segments that can be divided is n, and the upper bound of n satisfies:

$$\frac{1}{1}+\frac{1}{2}+...+\frac{1}{n} \le Y < \frac{1}{1}+\frac{1}{2}+...+\frac{1}{n+1} \quad \text{...... (1)}$$

2.2. Staggered Broadcasting [2]

Assume we want to play a film of time length L through Y channels. In the initial time, the whole film will repeatedly transmit on C_0. After the time $i \cdot L/Y$, $1 \le i \le Y-1$, the same whole film will also be transmitted on C_i periodically. The maximum waiting time is L/Y. Figure.1 shows the scheduling of stagger method for Y=3.

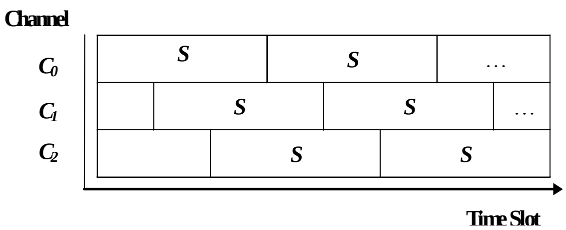


Fig. 1. Staggered Broadcasting.

2.3. Fast broadcasting [6]

The basic concept of fast broadcasting method is described as follows. Considering Y channels {C_0, C_1, …, C_{y-1}} available, a film is first partitioned into 2^y-1 segments {S_1, S_2, …, S_{2^y-2}, S_{2^y-1}}. Segment S_1 will repeatedly transmit on C_1. The 2^i segments {S_{2^i}, S_{2^i+1}, …, $S_{2^{i+1}-1}$} repeatedly transmit in order on C_{i+1} for $1 \le i \le y-1$. Figure.2 shows the scheduling of fast broadcasting method for Y=3.

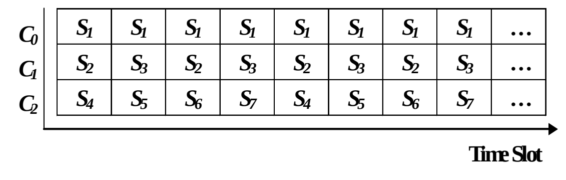


Fig. 2. Fast broadcasting.

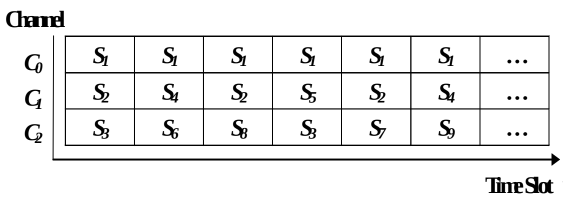
2.4. Pagoda Broadcasting [10]

The basic concept of pagoda broadcasting method is described as follows. Considering y channels {C_0, C_1 …, C_{Y-1}} available, a film is first partitioned into n segments {S_1, S_2, …, S_n} where $n = 4(5^{(Y/2)-1})-1$ if Y is even and $n = 2(5^{\lfloor Y/2 \rfloor})$ if Y is odd. Segment S_1 will repeatedly transmit on C_0. Let q=2r-1 for $r \ge 1$ and the index of S_z is smaller than those of the other segments broadcasted on C_q. The z/2 segments {S_z, S_{z+1}, …, $S_{3z/2-1}$} repeatedly transmit in order on the odd slots on C_q. The z segments {S_{2z}, S_{2z+2}, S_{2z+4}, …, S_{3z-2}, S_{2z+1}, S_{2z+3}, S_{2z+5}, …, S_{3z-1}} repeatedly transmit in order on the even slots on C_q.

After transmitting segments on C_q, the following introduces the segments scheduling on C_{q+1}. The z/2 segments {$S_{3z/2}$, $S_{3z/2+1}$, …, S_{2z-1}} repeatedly transmit in order on the slots 3i+1, $0 \le i \le z-1$, on C_{q+1}. The z segments {S_{3z}, S_{3z+2}, S_{3z+4}, …, S_{4z-4}, S_{4z-2}, S_{3z+1}, S_{3z+3}, …, S_{4z-3}, S_{4z-1}} repeatedly transmit in order on the slots 3i+2 on C_{q+1}. The z segments {S_{4z}, S_{4z+2}, S_{4z+4}, …, S_{5z-4}, S_{5z-2}, S_{4z+1}, S_{4z+3}, …, S_{5z-3}, S_{5z-1}} repeatedly transmit in order on the slots 3i+3 on C_{q+1}.

If Y is odd, the pagoda method is finished. Otherwise, if Y is even, the z segments {S_z, S_{z+1}, S_{z+2}, …, S_{2z-1}} repeatedly transmit in order on C_{Y-1}.

Fig.3 and Fig.4 show the Pagoda scheme's scheduling for 3 and 4 channels, respectively.



F

Fig. 3. Pagoda scheme's scheduling for 3 channels.

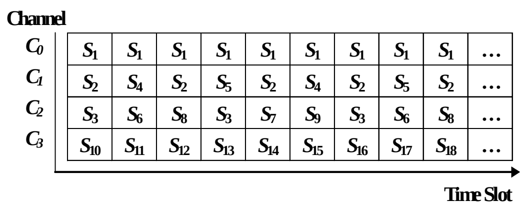


Fig. 4. Pagoda scheme's scheduling for 4 channels.

2.5. Recursive Frequency Splitting（RFS）[15]

Tseng, Yang, and Chang first define the slot sequence SS(C_i, p, q) as an infinite sequence of time slots [p, p+q, p+2q, …] belonging to channel C_i, beginning at slot p, and repeating infinitely with a period of q slots, where C_i is one of the Y channels, $p \ge 0$ is an integer, and $q \ge 1$ is an integer, $0 \le p \le q-1$.

Initially, let POOL = {SS(C_1, 0, 1), SS(C_2, 0, 1), SS(C_3, 0, 1), …, SS(C_Y, 0, 1)} denote the set of free channels and let j be the index of segment. The initial value of j is 1. Second, pick a slot sequence SS(C_i, p, q) with the smallest value of j mod q form POOL such that $q \leq j$. Let POOL = POOL − {SS(C_i, p, q)}. Third, split SS(C_i, p, q) into {SS(C_i, p, αq), SS(C_i, $p+q$, αq), SS(C_i, $p+2q$, αq), …SS(C_i, $p+(\alpha-1)q$, αq)} where $\alpha = \lfloor j/q \rfloor$. Segment S_j is broadcasted on the slots in SS(C_i, p, αq). Do the union POOL = POOL ∪ { SS(C_i, $p+xq$, αq) | $1 \leq x \leq \alpha-1$}. If POOL is not empty, then increase i by one and go to the second phase. Otherwise, terminate this process and output the value of i. Figure.5 illustrates the result of RFS algorithm with four channels.

Channel																				
C_0	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1
C_1	S_2	S_4	S_2	S_8	S_2	S_4	S_2	S_{16}	S_2	S_4	S_2	S_8	S_2	S_4	S_2	S_{17}	S_2	S_4	S_2	S_8
C_2	S_3	S_6	S_9	S_3	S_7	S_{18}	S_3	S_6	S_{22}	S_3	S_7	S_9	S_3	S_6	S_{19}	S_3	S_7	S_{23}	S_3	S_6
C_3	S_5	S_{18}	S_{12}	S_{14}	S_{15}	S_5	S_{11}	S_{13}	S_{28}	S_{24}	S_5	S_{10}	S_{12}	S_{14}	S_{25}	S_5	S_{11}	S_{13}	S_{21}	S_{15}

Time Slot

Fig. 5. Recursive frequency-splitting Scheme's scheduling for 4 channels.

3. Broadcast Method which only Comprises Ability to Simultaneously Receive and Process Fixed Bandwidth

We first use an example to explain the fundamental idea behind the Broadcasting method. Assume there are a total of 4 channels {C_0,C_1,C_2,C_3} (Y=4) to transmit a video, but STB can only simultaneously receive and process 3 channels (R=3). Video segments from 3 channels {C_0,C_1,C_2} will be received from the beginning. After video transmitted from C_0 has been completely received, video information will be received from C_3, after which STB only needs to receive the missing video segments from C_1,C_2,C_3. To allocate video for transmission via 4 channels, we first cut a video S into S_1,S_2,S_3…S_{21}. Why the video can be cut into 21 segments will be further elaborated later.

Channel																				
C_0	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1	S_1
C_1	S_2	S_4	S_2	S_5	S_2	S_4	S_2	S_5	S_2	S_4	S_2	S_5	S_2	S_4	S_2	S_5	S_2	S_4	S_2	S_5
C_2	S_3	S_6	S_8	S_3	S_7	S_9	S_3	S_6	S_8	S_3	S_7	S_9	S_3	S_6	S_8	S_3	S_7	S_9	S_3	S_6
C_3	S_{10}	S_{13}	S_{17}	S_{11}	S_{14}	S_{18}	S_{12}	S_{15}	S_{19}	S_{10}	S_{16}	S_{20}	S_{11}	S_{13}	S_{21}	S_{12}	S_{14}	S_{17}	S_{10}	S_{15}

Time Slot

Fig. 6. A novel broadcasting scheme.

The first 3 channels in Fig. 6 has allocation methods obtaining the same results with Pagoda and RFS, and it can be observed that currently only S_1 is transmitted in C_0, thus the cycle of C_0 is 1, while in C_1 there is one circulation for every 4 Time Slots S_2, S_4, S_2, S_5 thus the cycle on C_1 is 4. One circulation in C_2 is S_3, S_6, S_8, S_3, S_7, S_9 thus the cycle of C_2 is 6. The last video segment transmitted in the first 3 Channels is S_9, thus transmits segments S_{10}~S_{21}. Since STB can only receive 3 Channels at one time, receiving C_3 has to wait till receiving of C_0 is completed before switching to receiving C_3, which means one time slot will be delayed, therefore S_{10} has to appear at least once every 9 times and not 10 times in order to ensure smooth broadcast.

Due to S_{10} appearing at least once every 9 Time Slots, and 9=3*3(using the smallest divisible factor), therefore we can, according to time division, split this channel into 3 subchannels {C_3^1 、 C_3^2 、 C_3^3}, where each subchannel takes up 1/3 channel bandwith, first we analyse C_3^1, the starting segment to be transmitted by C_3^1 is S_{10}, and S_{10} appears at least once every 9 time slots, since S_{11}, S_{12} can also appear at least once every 9 time slots, we can repeatedly transmit a total of 3 segments S_{10}, S_{11}, S_{12} in C_3^1; the beginning Segment of C_3^2 is S_{13}, because S_{13} has to appear at least once every 12 time slots, we thus can repeatedly transmit the 4 Segments S_{13}, S_{14}, S_{15}, S_{16} in C_3^2 ; the beginning segment in C_3^3 is S_{17}, and S_{17} can be taken to appear at least once every 15 time slots, thus can repeatedly transmit the 5 segments S_{17}, S_{18}, S_{19}, S_{20}, S_{21} in C_3^3; this is mainly because S_{17} appears at least once every 16 times, and $\lfloor 16/3 \rfloor$=5 means a total of 5 segments including S_{17} can be transmitted in C_3^3. Looking at Fig.7, making use of our method we can place the segments in C_3 in the following manner: (represents C_3^1 represents C_3^2 represents C_3^3)

S_{10}	S_{13}	S_{17}	S_{11}	S_{14}	S_{18}	S_{12}	S_{15}	S_{19}	S_{10}	S_{16}	S_{20}	S_{11}	S_{13}	S_{21}

Fig. 7. Segments on C_3.

In the above we have already explained the fundamental ideas behind our method using an example. When using 4 channels to transmit video segments while the processing capability of STB at the client's end can at most only simultaneously process 3 channels, we can cut the video into 21 Segments.

Before we introduce below our Broadcasting calculations, we first define a few symbols; let d_i be the total number of Time Slots in one cycle, in this one cycle all Segments transmitted in C_i will appear at least once. Therefore d_0, d_1, d_2 are 1, 4, 6 respectively. The calculations where d_k for $k \geq 3$ will be elaborated in the calculations. D_k represents how many Time Slots has be to waited before receiving video segments in C_k, and since 3 Channels can be received at the same time, D_0, D_1, D_2 are all 0，$D_k = \sum_{i=0}^{k/3-1} d_{l+3i}$, for $k \geq 3$ and $l=k$ mod 3. Let S_x be the beginning video segment transmitted in C_k, because D_k time slots will be delayed, thus S_x has to transmit at least once every x-D_k Time Slots. We will first reduce x-D_k to its lowest divisible factors, then according to the allocation methods mentioned in the previous page to calculate how many Segments have to be cut. The following is the calculation steps for cutting Segments:

Broadcasting Algorithm

Step 1：There are a total of Y channels, allocate S_1~S_{x-1} the first R channels, where the allocation methods of segments in the first R channels we use RFS [17] to proceed.

Step 2：Let segments to be sent by C_k ($k \geq 1$) after S_x ($x \geq 2$), due to S_x appearing at least once in x-D_k slots.

Step 3：Calculate the lowest divisible factor of x-D_k, assume $P_1 \times P_2 \times \ldots \times P_Z$ ($Z>1$) let $W = P_1 \times P_2 \times \ldots \times P_{Z-1}$ ($Z>1$) for $Z>1$. When $Z=1$, $W=1$.

Step 4：Break up C_k according to time-division into W subchannels (C_k^1, C_k^2, …, C_k^W) with identical bandwidth .

Step 5：$\alpha=x-D_k$

For($g=1$；$g \leq W$；g++)
{
$v=\alpha/W$
if ($g=W$) $d_k=v*W$
Let S_x, S_{x+1}, …, S_{x+v-1} be distributed onto C_k^g .
$\alpha=x+v-D_k$
$x=x+v$
}

From Step 5 in the Algorithm we can see that whenever a channel C_k ($k \geq 1$) and which segments to be sent are determined, the corresponding d_k value can be calculated. Based on the proposed Broadcasting Algorithm, Table.1 and Fig.8 lists the corresponding relationship between the number of channels and cut segments:

Table .1. Corresponding relationship between the number of channels and cut segments:
X: Number of channels,
Y: Simultaneously receiving channels

Y \ X	4	5	6	7	8	9	10
3	21	46	87	191	427	948	2205
4	25	64	118	219	482	1155	2731
5	25	73	146	277	627	1520	3728
6	25	73	201	476	937	2254	4737
7	25	73	201	562	1264	2513	6170

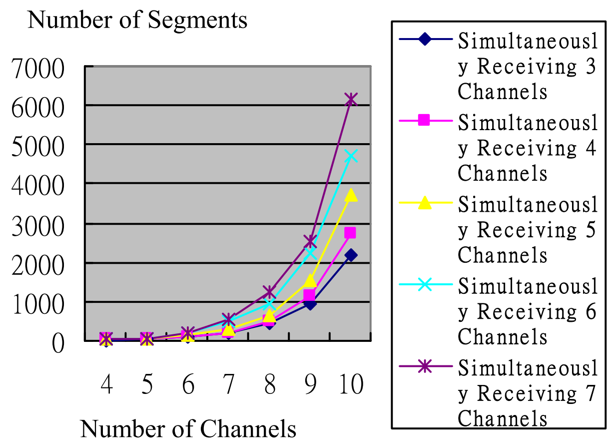


Fig. 8. Line graph showing cut segments.

4. Conclusion

This paper has proposed a novel broadcasting method for considering the limitation of STB capability. Our method can satisfy users whose STBs can not simultaneously receive and process all used channels

Reference

[1] C. C. Aggarwal, J. L. Wolf, and P. S. Yu, “A permutation-based pyramid broadcasting scheme for video-on-demand systems,” *IEEE Proceedings of the International Conference on Multimedia Computing and Systems (ICMCS)*, pp. 118-126, Jun 1996.

[2] T. Chiueh and C. Lu, “A periodic broadcasting approach to video-on-demand service,” *Proc. SPIE 95, Integration Issues in Large Commercial Media Delivery Systems*, Philadelphia, Pennsylvania, 1995.

[3] Eager, D. L. and M. K. Vernon, “Dynamic skyscraper broadcast for video-on-demand,” *Proceedings of the 5th International Workshop on Advances in Multimedia Information Systems*, pp. 18-32, Sep. 1998.

[4] L. Gao, J. Kurose, and D. Towsley, “Efficient schemes

for broadcasting popular videos," *International Workshop on Network and Operating Systems Support for Digital Audio and Video*, pp. 317-329, Aug. 1998.

[5] K. A. Hua and S. Sheu, "Skyscraper broadcasting: A new broadcasting scheme for metropolitan video-on-demand systems," *ACM SIGCOMM'97*, Cannes, France, Sept. 1997.

[6] 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, Mar 1998.

[7] 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, Sept. 1997.

[8] 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, Nov 1997.

[9] L.-S. Juhn and L.-M. Tseng, "Enhanced harmonic data broadcasting and receiving scheme for popular video service," *IEEE Transactions on Consumer Electronics*, vol. 44, no. 2, pp. 343-346, May 1998.

[10] J. -F. Pâris, S.-W. Carter, and D.-D.Long, "A hybrid broadcasting protocol for video on demand," *Proceedings of the IS&T/SPIE Conference on Multimedia Computing and Networking (MMCN '99)*, pp. 317-326, 1999.

[11] J. -F. Pâris, "A simple low-bandwidth broadcasting protocol for video-on-demand," *Proceedings of the 8-th International Conference on Computer Communications and Networks (IC3N '99)*, pp. 118-123, 1999

[12] J. -F. Pâris, "An interactive broadcasting protocol for video-on-demand," *Proceedings of the 20th IEEE International Performance, Computing, and Communications Conference（IPCCC 2001）*, pp. 347-353, Phoenix, AZ, Apr. 2001.

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

[14] S. Viswanathan and T. Imielinski, "Metropolitan area video-on-demand service using pyramid broadcasting," *IEEE Multimedia Systems*, vol. 4, no. 4, pp. 197-208, 1996.

[15] M. -H. Yang, C. -H. Chang, and Y. -C. Tseng, "A Recursive Frequency-Splitting scheme for broadcasting hot videos in VOD service," *IEEE Trans. on Communications*, vol. 50, no. 8, pp. 1348-1355, 2002.

[16] 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.

[17] Kuo-Tsang Huang, "The Multiple Videos Broadcasting Scheme for Video on Demand," Graduate school of Mathematical Sciences, Aletheia University, June 2003