Scheduling Methods Based on Data Division for Continuous Media Data Broadcast

Tomoki Yoshihisa
Grad. School of Information Science and Technology, Osaka University
yosihisa@ist.osaka-u.ac.jp

Masahiko Tsukamoto
Grad. School of Information Science and Technology, Osaka University
tuka@ist.osaka-u.ac.jp

Shojiro Nishio
Grad. School of Information Science and Technology, Osaka University
nishio@ist.osaka-u.ac.jp

***Abstract*— Recently, various schemes have arisen to broadcast continuous media data such as audio and video. Some of them have focused on reducing clients' waiting time under the continuity condition, i.e., to play data completely without any interruptions. These schemes usually employ multiple channels to broadcast continuous media data. However, receivers of many existing broadcast systems such as those equipped with wireless LAN or Bluetooth cannot receive data from multiple channels concurrently. In this paper, we propose and evaluate a scheduling scheme to reduce the waiting time of clients with a single channel.**

I. Introduction

Recently, broadcasting continuous media data such as audio and video has become increasingly popular [1–5]. In general broadcast systems, a server broadcasts data repetitively, and clients selectively receive broadcast data which may be desired by a user. In such systems, the user orders his/her client to receive a data, then the client waits until the data is broadcast. Hence, the schedule of the broadcast data at the server side has a great effect on clients' waiting time.

Meanwhile, it is important for users to play data without any interruptions. Previous studies have aimed at reducing the waiting time of clients to start playing data under this *continuity conditions* and this is achieved by dividing data into several segments and broadcasting these segments via multiple channels. However, with multiple channels, clients are forced to receive data from more than one channel concurrently, while the server has to control multiple channels. Hence, the broadcasting mechanism becomes more complicated than that of broadcasting data with a single channel. Furthermore, typical popular commercial wireless LAN transceivers, such as Bluetooth [6] or set-top-boxes, cannot always concurrently receive data from multiple channels.

In this paper, we propose a scheduling scheme to reduce the waiting time of clients with a single channel. In our scheme, assuming that users can play data continuously, data is divided into several segments to produce an effective schedule. By broadcasting the first segment frequently, the waiting time is reduced. In this scheme, the number of segments and the broadcast schedule affect the waiting time. We show the schedule that gives the minimum average waiting time, in which data is divided into two segments of equal size. Where data is divided into more than two segments, we propose and evaluate a heuristic scheduling method. The remainder of this paper is organized as follows. Section II explains our basic idea. We show how broadcasting precedent segments with a single channel frequently reduce the waiting time of clients. Our assumed system environment and our scheme are explained in Section III. In Section IV, for the case of dividing data into two segments, we show the schedule that gives the minimum average waiting time. Section V explains our heuristic scheduling method for the case of dividing data into more than two segments. Section VI remarks related works, and finally, we conclude the paper in Section VII.

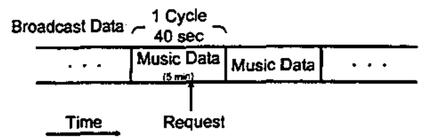

Fig. 1. A simple repetition scheme for a continuous media data broadcast.

II. Basic Idea

In this section, first, we show the average waiting time of a simple repetition scheme, and following that, we outline our scheme's basic idea.

Suppose that it takes 40 seconds to broadcast a 5-minute continuous media data. Note that this case is suitable for broadcasting a 5-minute MP3 encoded music data (the data size: 5 Mbytes) on 1-Mbps satellite or Bluetooth systems. In the case of broadcasting the data repeatedly, as shown in Fig. 1 (the *simple repetition scheme*), a user who requests the data to his/her client should wait 20 seconds on average until the client starts playing it. Note that the data is streaming media data, i.e., the client can start playing the data immediately after receiving it.

To explain our basic idea, we divide the data into two 2.5-minute segments. It takes 20 seconds to broadcast each segment. Let S_1 indicate the first segment and S_2 the remaining segment. The server broadcasts these segments in the order of $S_1S_1S_2$ as shown in Fig. 2. Accordingly, the broadcast cycle becomes 60 seconds.

Let T_i ($i = 1, 2, \cdots$) indicate the time slot. We assume that the broadcast cycle starts at T_1. That is, S_1 is broadcast at T_1 and T_2, S_2 is broadcast at T_3 in sequence. The clients' average waiting time can be calculated as follows.

0-7803-7978-0/03/$17.00

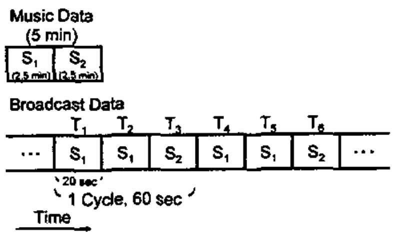

Fig. 2. A simple example to explain our basic idea.

- When requesting the data in T_1:
  The client receives S_1 broadcast at T_2 and can store S_2 broadcast at T_3 while it plays S_1. The average waiting time in this case is 10 seconds.
- When requesting the data in T_2:
  The client receives S_1 broadcast at T_4 and receives S_2 broadcast at T_3 or T_6 as soon as it finishes playing S_1. The average waiting time in this case is 30 seconds.
- When requesting the data in T_3:
  The client receives S_1 broadcast at T_4 and receives S_2 broadcast at T_6 as soon as it finishes playing S_1. The average waiting time in this case is 10 seconds.

As a result, the average waiting time is 17 seconds. This is 15% shorter than that of the simple repetition scheme.

In this case, the average waiting time of the broadcast schedule $S_1S_1S_1S_1S_1S_1S_1S_2$ is 12.5 seconds. This is 38% shorter than that of the simple repetition scheme.

Thus, by dividing data into several segments and the server broadcasts precedent segments frequently, the average waiting time is reduced.

III. Assumed System Environment and Our Scheme

We assume that the time needed to broadcast continuous media data is shorter than the playing time. Let D indicate the playing time of data and D' the time needed to broadcast it. We define the *playback ratio* a by the following equation:

$$a = \frac{D}{D'}. \tag{1}$$

For example, when data is encoded by MPEG Audio Layer 3 (MP3), the data size of a 5-minute audio file is approximately 5 Mbytes. Hence, it takes 40 seconds to broadcast the data on a 1-Mbps network. Hence, the playback ratio a is $300/40 = 7.5$.

A. Our Assumed System Environment

- A client is not able to receive data from multiple channels concurrently, i.e., the client can only receive data from a single channel.
- A server broadcasts one item of continuous media data with a single channel.
- A client who starts playing data has to be able to play the data without any interruptions, i.e., the continuity condition must be satisfied.
- D' is shorter than D. The playback ratio a (≥ 1) is constant.

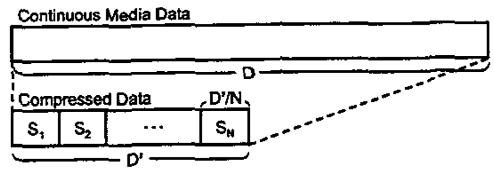

Fig. 3. A division of a data.

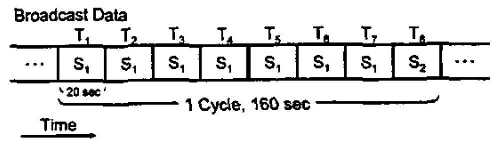

Fig. 4. A broadcast schedule of the division-based broadcasting scheme.

- A client has a buffer that can receive and store data while the client plays the data. The buffer is large enough to store the data.

B. Our Proposed Scheme

We propose a scheduling scheme, the *division-based broadcasting scheme*. In this scheme, an item of broadcast data is divided into N segments S_i ($i = 1, \cdots, N$) of equal size. The necessary time to broadcast a segment is D'/N (Fig. 3). The server broadcasts c segments in a broadcast cycle. Fig. 4 shows an example of the division-based broadcasting scheme for $c = 6$, $N = 3$.

One important problem relates to how we can produce a broadcast schedule that effectively reduces the waiting time. For example, if it takes 1 minute to broadcast a 5-minute item of MP3 data ($a = 5$, $D = 5$ min) and the data is divided into two segments ($N = 2$), and 4 or 5 segments are broadcast in a broadcast cycle ($c = 4, 5$), we can suppose the broadcast schedules shown in Table I. Since $S_1S_2S_1S_2$ is regarded as $c = 2$, this broadcast schedule is not shown in the table. The minimum average waiting time of Table I is 21 minutes and the broadcast schedule is $S_1S_1S_1S_1S_2$; thus, for any given a, c, and N, the average waiting time is calculated. However, since these values can be given arbitrarily and the number of possible broadcast schedules is infinite, it is not possible to find the broadcast schedule which gives the minimum average waiting time in all schedules when using search-based methods. Hence, in Section IV, for the case of dividing data into two segments of equal size, we present a schedule that gives the minimum average waiting time. In Section V we propose a heuristic scheduling method for dividing data into more than two segments, since many combinations of scheduling segments makes the analysis difficult.

IV. Dividing Data into Two Segments

Here we analyze the division of data into two segments of equal size. In this case, we find the schedule that gives the minimum average waiting time.

TABLE I
BROADCAST SCHEDULES WHICH INCLUDE 4 OR 5 SEGMENTS IN ONE BROADCAST CYCLE ($a = 5$, $D = 5$ MIN, $N = 2$).

$c = 4$		$c = 5$	
Broadcast Schedule	Average Waiting Time (sec)	Broadcast Schedule	Average Waiting Time (sec)
$S_1S_1S_1S_2$	23	$S_1S_1S_1S_1S_2$	21
$S_1S_1S_2S_2$	38	$S_1S_1S_1S_2S_2$	33
$S_1S_2S_2S_2$	60	$S_1S_1S_2S_1S_2$	27
		$S_1S_1S_2S_2S_2$	51
		$S_1S_2S_1S_2S_2$	33
		$S_1S_2S_2S_2S_2$	75

Let α denote the integral part of a, namely $\alpha = \lfloor a \rfloor$. We define A by the following equation:

$$A = (\alpha + 1)(a - \alpha) - \alpha. \tag{2}$$

The following statements give the broadcast schedule that allows the minimum average waiting time:

- For the case where $A \leq 0$:

$$\overbrace{S_1 \cdots S_1}^{\alpha} S_2, \tag{3}$$

and the average waiting time is given by

$$\frac{\alpha + 3}{4a(\alpha + 1)} D. \tag{4}$$

- For the case where $A \geq 0$:

$$\overbrace{S_1 S_1 \cdots S_1}^{\alpha+1} S_2, \tag{5}$$

and the average waiting time is given by

$$\frac{3\alpha - 2a + 6}{2a(\alpha + 2)} D. \tag{6}$$

V. Dividing Data into More than Two Segments

We propose a heuristic scheduling method of the division based broadcasting scheme, the *segment insertion method*, for dividing data into more than two segments.

A. Segment Insertion Method

The broadcast schedule A is produced according to the following process:

1) Let A be empty.
2) Divide the continuous media data into N segments $(S_1, \cdots, S_N)$ of equal size.
3) Add n_N pieces of S_N to A.
4) First, let $i = N - 1$ and add n_i pieces of S_i to the left side of each segment included in the schedule A.
5) Decreasing i, repeat step 2 until $i = 1$.

With the segment insertion method, the number of pieces to be added, n_i $(i = 1, \cdots, N)$, is given by the following equations:

- For the case where $i = 1$ and $N = 2$:

$$n_1 = \lfloor a \rfloor. \tag{7}$$

- For the case where $i = 1$ and $N = 3, 4, \cdots$:

$$n_1 = \left\lfloor \frac{(N-2)a + 1}{2^{N-2}} - 1 \right\rfloor. \tag{8}$$

- For the case where $i = 2, \cdots, N$:

$$n_i = 1. \tag{9}$$

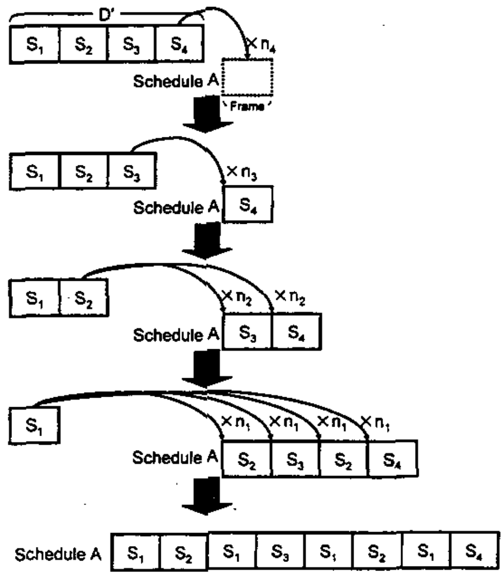

Fig. 5. A process of the segment insertion method ($a = 4$ and $N = 4$).

Fig. 5 shows a process of the segment insertion method for $a = 4$. The average waiting time of the segment insertion method is given by the following equation:

$$W = \frac{D}{aN}\left(\frac{1}{2} + \frac{1}{n_1 + 1}\right). \tag{10}$$

Note that, Eq. (10), the average waiting time of the segment insertion method depends on n_1; i.e., the higher n_1 is, the shorter the average waiting time becomes.

B. Comparison with the Simple Repetition Scheme

Fig. 6 shows the average waiting time of the segment insertion method. Since the average waiting time is proportional to the playing time D, the vertical axis is the average waiting time divided by D, while the horizontal axis is the playback ratio a. In addition, 'SI ($N = i$)' ($i = 2, 3, 4$) represents the average waiting time of the segment insertion method when the case that the data is divided into i segments, and 'N=i (c=$i, \cdots, 10$)' is the minimum average waiting time of all schedules including i to 10 segments in one broadcast cycle. Moreover, 'simple' means the average waiting time of the simple repetition scheme explained in Section II.

In the case of broadcasting an MP3-encoded 5-minute audio data with a 1-Mbps wireless broadcast systems such as satellite or bluetooth systems, $a = 7.5$, as mentioned in Section III. Here, although the average waiting time of the simple

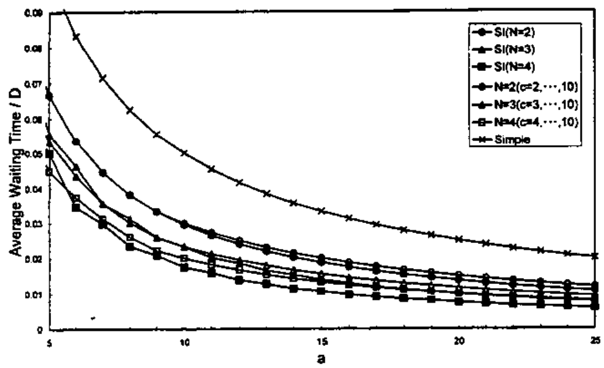

Fig. 6. The average waiting time of the segment insertion method.

repetition scheme is 20 seconds, that of the 4-division segment insertion method is 8 seconds. This is 60 % shorter.

VI. Related Work

To reduce the waiting time under the continuity condition, various strategies have been studied so far.

In pyramid broadcasting (PB) [2], data, the quantity of which is M, is divided into K segments, and K channels are used. Let D_i denote the playing time of the i^{th} segment. The data is divided so that D_i is $\alpha^{i-1}D_1$, where, α is 1 or more. The server broadcasts the i^{th} segment of each data using the i^{th} channel. Since the time required to broadcast precedent segments is shorter, the first segment is broadcast at frequent intervals. Hence, clients have many chances to receive the first segment, and their waiting time is reduced. In this scheme, however, the greater K becomes, the bigger the data size of the i^{th} segment is; therefore, the buffer capacity requirements increase.

Skyscraper broadcasting (SB) [3] also considers this problem. Accordingly, a segment with a longer playing time than D_1W is divided so that its playing time is D_1W. By modifying W, buffer requirements and the average waiting time are given arbitrarily.

Dynamic skyscraper broadcasting has also been proposed to deal with this problem [4]. This scheme is based on the SB scheme, and it dynamically changes broadcast data, and can improve the performance of SB when clients infrequently request data.

A permutation-based pyramid broadcasting [5] divides each channel into p sub-channels, and a server broadcasts the i^{th} segment using p sub-channels. The beginning of the broadcast cycle of each sub-channel is shifted, and as a result, the waiting times are improved compared to those of PB. Although the paper suggests time division multiplexing which employs a single channel, the paper does not describe a detailed technique for reducing clients' waiting time via a single channel.

VII. Conclusion and Future Work

In this paper, we proposed a division-based broadcasting scheme to reduce the waiting time of clients with a single channel. In our strategy, assuming that clients can play data continuously, the data is divided into several segments to produce an effective schedule. By broadcasting the first segment frequently, clients' waiting time is reduced. Thus, in this scheme, the number of segments and the broadcast schedule affect the waiting time.

For the case of dividing data into two segments, we also presented a schedule which gives the minimum average waiting time. The number of the first segment S_1 included in the broadcast schedule is determined by the ratio of the data playing time and the time needed to broadcast the segment, whereas the last segment S_2 is included only once. When the server broadcasts an MP3-encoded continuous media data using a 1-Mbps satellite system, the average waiting time is reduced by about 38% compared to the simple repetition scheme, which broadcasts data without division. Hence, dividing the data into only two segments is rather realistic.

For the case of dividing data into more than two segments, we proposed and evaluated the segment insertion method. As a result of analyzing the average waiting time of the segment insertion method, we found that the average waiting time is always shorter than or equal to that of the simple repetition scheme. The greater the number of segments N is, the shorter the average waiting time becomes.

In future, we plan to study the delivery of multiple data with a single channel and other effective heuristics.

Acknowledgement

This research was supported in part by "The 21st Century Center of Excellence Program" of the Ministry of Education, Culture, Sports, Science and Technology, Japan, and Special Coordination Funds for Promoting Science and Technology of the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

[1] G. B. Horn and J. E. Rasmussen, "A Scalable and Reliable Paradigm for Media on Demand," IEEE Computer, Vol. 34, No. 9, pp. 40–45, September 2001.

[2] S. Viswanathan and T. Imilelinski, "Pyramid Broadcasting for Video on Demand Service," in Proceedings of the SPIE Multimedia Computing and Networking Conference (MMCN '95), pp. 66–77, February 1995.

[3] K. A. Hua and S Sheu, "Skyscraper Broadcasting: A New Broadcasting Scheme for Metropolitan Video-on-Demand Systems," in Proceedings of the ACM SIGCOMM, pp. 89–100, September 1997.

[4] D. L. Eager and M. K. Vernon, "Dynamic Skyscraper Broadcasts for Video-on-Demand," in Proceedings of the 4th International Workshop on Multimedia Information Systems (MIS '98), pp. 18–32, September 1998.

[5] C. C. Aggarwal, J. L. Wolf, and P. S. Yu, "A Permutation-Based Pyramid Broadcasting Scheme for Video-on-Demand System," in Proceedings of the IEEE International Conference on Multimedia Computing and Systems (ICMCS '96), pp. 118–126, June 1996.

[6] Bluetooth SIG, "The Official Bluetooth Wireless Info Site," http://www.bluetooth.com.