A Broadcasting Scheme for Continuous Media Data with Restrictions in Data Division

Tomoki Yoshihisa†, Masahiko Tsukamoto‡, and Shojiro Nishio*

†Grad. Sch. of Inf. Sci. and Tech., Osaka University
1-5, Yamadaoka, Suita-Shi, Osaka, 565-0871, Japan, yosihisa@ist.osaka-u.ac.jp
‡Faculty of Eng., Kobe University
1-1, Rokkodai, Nada-Ku, Kobe-Shi, Hyogo, 657-8501, Japan, tuka@kobe-u.ac.jp
* Grad. Sch. of Inf. Sci. and Tech., Osaka University, nishio@ist.osaka-u.ac.jp

ABSTRACT

Due to the recent proliferation of digital broadcasting, various schemes for broadcasting continuous media data have been studied. In general broadcasting systems, since data is broadcast repetitively, clients have to wait until their desired data is broadcast. These schemes reduce this waiting time by dividing the data into several segments of equal size. However, data often have appropriate dividing boundaries. For example, data encoded by MPEG2 can be played every GOP (Group of Pictures). Hence, it is reasonable to divide the data into each GOP from the viewpoint of efficient implementation. In this paper, we propose a scheme to reduce the waiting time considering such restrictions. Our proposed scheme divides a continuous media data at every unit for playing portions of the data. By scheduling divided data so that clients finish receiving a unit before starting to play the unit, waiting time is reduced.

Keywords: Broadcasting, Continuous Media Data, Video on Demand, Group of Pictures, Scheduling

1 Introduction

Broadcasting systems have recently been digitized. Since the influence of noise on digital broadcasting is fainter than that on analog broadcasting, there is room for the system's bandwidth to expand, up to approximately 24 Mbps. Moreover, since it is easy for digital broadcasting to multiplex data, many channels are available. Accordingly, various schemes for delivering continuous media data such as music or movies using attractive digital broadcasting systems have been studied[1]–[8], [10]. In general broadcasting systems, a server broadcasts data repetitively. While the server can concurrently deliver the data to many clients, clients have to wait until their desired data are broadcast. In continuous media data broadcasting, it is usually postulated that clients can play the data without any interruptions until the end of the data. Hence, these schemes reduce clients' waiting time under the postulate.

Previous studies usually assume that a client can play data as soon as it starts receiving the data. However, in many cases, clients can play the data after finishing receiving units for playing portions of the data. For example:

- A unit for playing data encoded by MPEG is a GOP (Group of Pictures)[9]. Clients can play a GOP only after finishing receiving the GOP.
- Continuous media data are sometimes encoded by intra-frame compression. In this case, a unit for playing data must be a frame.
- In the case where data is encoded by inter-frame compression, a unit for playing data must be a group of these interconnected frames. MPEG is one of these cases.

These can be regarded as a restriction in that the data divided into given data sizes cannot be played as soon as clients start receiving the divided data. Accordingly, clients must wait until a unit is received. This problem cannot be efficiently solved only by artificially delaying the start of playback by the largest size of units. Because a comparison with conventional schemes, which are extended to wait the start of playback, reveals that our proposed scheme gives shorter waiting time than these extended conventional schemes.

In this paper, we propose a broadcasting scheme to reduce waiting time considering such a restriction. By dividing data into each unit to be played, and scheduling them so that clients finish receiving a unit before starting to play the unit, our proposed scheme reduces the waiting time.

The remainder of this paper is as follows: In Section 2, we discuss related researches. We propose a scheme for continuous media data with restrictions in data division in Section 3. We analyze it in Section 4 and evaluate it in Section 5. In Section 6, we discuss our proposed scheme and conclude the paper in Section 7.

2 Related Works

In continuous media data broadcasting, clients have to wait until their desired data is broadcast. Accordingly, various schemes have been proposed to reduce this waiting time. These studies assume that a client can play the data as soon as it starts receiving the data.

The Harmonic Broadcasting (HB) scheme divides a continuous media data into N segments $S_1, \cdots, S_N$ of equal size. In addition, a segment S_i ($i = 1, \cdots, N$) is divided into i sub-segments $S_{i,1}, \cdots, S_{i,i}$. Segments or sub-segments can be played as soon as clients start receiving them and cannot be

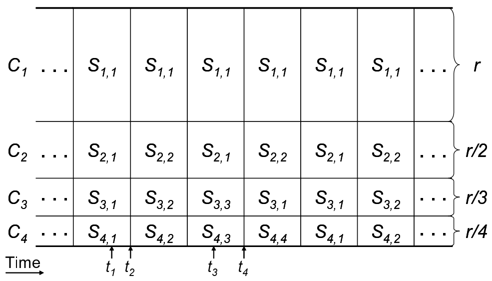

Figure 1: A broadcasting schedule under the HB scheme ($N = 4$).

received midstream. The server uses N channels $C_1, \cdots, C_N$. The bandwidth of C_i is r/i. Here, r is the consumption rate of the data. For example, when a server broadcasts a 5 Mbps data encoded by MPEG2, $r = 5$ Mbps. The server broadcasts $S_{i,1}, \cdots, S_{i,i}$ via C_i repetitively. Since S_1 is frequently broadcast in C_1, clients' waiting time can be reduced. Fig. 1 shows an example of the broadcasting schedule under the HB scheme when $N = 4$.

The Cautious Harmonic Broadcasting (CHB) scheme is an extended scheme of the HB scheme. The CHB scheme divides a segment S_j ($j = 4, \cdots, N$) into $j - 1$ sub-segments. The server uses $N - 1$ channels. The bandwidth of C_1 or C_2 is r and that of C_k ($k = 3, \cdots, N - 1$) is r/k. The server broadcasts S_1 in C_1, S_2 and S_3 in C_2, and $S_{k+1,1}, \cdots, S_{k+1,k}$ in C_k repetitively. Since clients finish receiving S_k before the time to play S_k, a client that plays S_1 as soon as it starts receiving S_1 can play the data all the way through without interruption. Though the average waiting time under the CHB scheme is longer than that under the HB scheme, this enables more simple receiving mechanism.

In this paper, we do not assume that a client can play the data as soon as it starts receiving the data.

3 Proposed Scheme

We propose a scheduling scheme, the *Asynchronous Harmonic Broadcasting* (AHB) scheme, to reduce clients' waiting time when units to be played are given. The AHB scheme adopts the HB scheme's idea that the bandwidth of each channel is adjusted so that a client finishes receiving a segment before it plays the segment. A difference between the AHB scheme and conventional schemes such as the HB or the CHB is that the AHB scheme divides data into separate units to be played. Due to the recent digitalization of broadcasting systems, waiting time reduction on digital broadcasting is still required. Hence, the AHB scheme is useful for current broadcasting systems.

3.1 Assumed Environment

- Continuous media data cannot be played as soon as a client starts receiving the data. The data can be played after the client finishes receiving a unit to be played.
- A server broadcasts divided data repetitively via multiple channels.

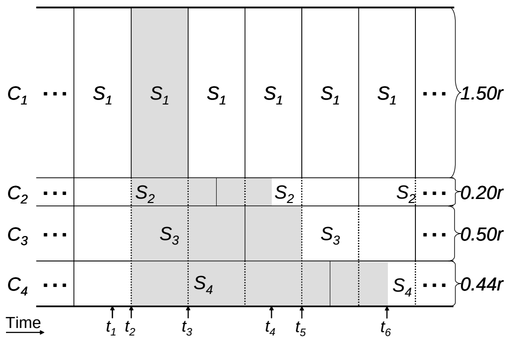

Figure 2: An example of a broadcasting schedule under the AHB scheme.

- A client can concurrently receive the data via multiple channels.
- A client has a buffer large enough to store the data.
- A client starts receiving the data after the client's user demands the data.

An example of our assumed environment is broadcasting movie data encoded by MPEG2 via a satellite digital broadcasting system.

3.2 The Scheduling Process

We denote D as the playing time of a continuous media data, and r as the data consumption rate. Note that r is constant while clients play the data (constant bit rate). Furthermore, $a_1, \cdots, a_N$ denote the data sizes for units to be played, i.e., the data size needed to play some portions of a continuous media data such as a GOP or a frame. According to $a_1, \cdots, a_N$, the data is divided into N segments $S_1, \cdots, S_N$. That is, the data size for S_i ($i = 1, \cdots, N$) is a_i. The playing time of S_i, which is denoted by p_i, becomes a_i/r and $D = p_1 + \cdots + p_N = (a_1 + \cdots + a_N)/r$. The scheduling process is shown below:

1. Divide the data into $S_1, \cdots, S_N$ so that the data size for S_i is a_i ($i = 1, \cdots, N$).
2. Use N channels $C_1, \cdots, C_N$ and set the bandwidth of C_j ($j = 2, \cdots, N$), which is denoted by b_j, by the following equation:

$$b_j = \frac{a_j}{\frac{a_1}{b_1} + p_1 + \cdots + p_{j-1}}. \quad (1)$$

3. Broadcast S_i via C_i repetitively. Note that, since segments are not received midstream, S_j is divided into several sub-segments.

Furthermore, b_1 is adjusted so that the total bandwidth $B = b_1 + \cdots + b_N$ is equal to the available bandwidth. By setting the bandwidth by (1), a client can finish receiving S_j before it starts playing S_j. For example, Fig. 2 shows a broadcasting schedule under the AHB scheme where $B = 2.64r$, $a_1 = 3r$, $a_2 = r$, $a_3 = 3r$, and $a_4 = 4r$. In this case, b_1 becomes

$1.5r$. A dashed line indicates a separator of sub-segments and the shaded area indicates the data received by a client that demands the data at t_1.

3.3 Implementation

A server divides a continuous media data into each unit to be played. The server adjusts b_1 according to the available bandwidth and broadcasts segments repetitively according to the schedule produced by the AHB scheme. When the server broadcasts a segment, it adds the data number and the segment number to the front of the segment, enabling clients to receive the data number and the segment number of the received segment. Since this additional information is often written in front of segments, clients cannot receive segments from midstream. Since the data size for additional information is small compared with that for the whole data, we ignore the time needed to broadcast it.

When a user demands to play a continuous media data, the user's client starts receiving the desired data from the broadcast data. Then, the client plays the data as soon as it finishes receiving S_1. While the client plays the data, it receives the broadcast data and stores the data into its buffer. After all subsegments of S_i ($i = 2, \cdots, N$) are received, the client quickly combines them to play S_i. The client plays the combined S_i as soon as the client finishes playing S_{i-1}. In this way, the client can play the data right through without any interruptions. For example, suppose that a user demands the data at t_1 (Fig. 2). The user's client starts receiving S_1 from t_2 and starts playing S_1 at t_3 as soon as the client finishes receiving S_1. Since $b_1 = 1.5r$, the playing time of S_1 is longer than the time needed to broadcast S_1. Accordingly, the client finishes playing S_1 at t_4. Since the client finishes receiving S_2 at t_4, the client can play S_2 without interruption. The client finishes receiving S_3 before the client finishes playing S_2 (t_5), and finishes receiving S_4 before it finishes playing S_3 (t_6). Accordingly, the client can play the data right through without any interruptions.

4 Analysis

The necessary bandwidth for broadcasting a continuous media data with the AHB scheme B is:

$$\begin{aligned} B &= \sum_{i=1}^{N} b_i = b_1 + \sum_{i=2}^{N} \frac{a_i}{\frac{a_1}{b_1} + p_1 + \cdots + p_{i-1}} \\ &= b_1 + \sum_{i=2}^{N} \frac{a_i b_1 r}{a_1 r + (a_1 + \cdots + a_{i-1})\, b_1}. \end{aligned} \quad (2)$$

The maximum waiting time is $2a_1/b_1$, and the minimum waiting time is a_1/b_1. Since the waiting time is supposed to be uniformly distributed, the average waiting time W_{ave} is:

$$W_{ave} = \frac{3a_1}{2b_1}. \quad (3)$$

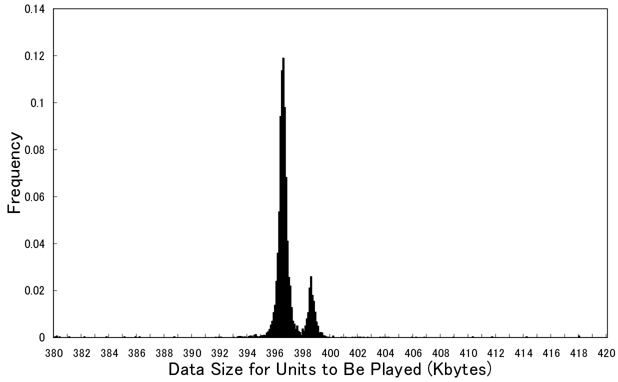

Figure 3: The histogram of data size needed to play the data.

Table 1: Statistics of the data for evaluation.

Minimum (Kbytes)	70
Maximum (Kbytes)	458
Average (Kbytes)	392
Variance $(\text{Kbytes})^2$	1,163
Total (Mbytes)	2,349
Segments	5,994

4.1 Playing without Any Interruptions

When a server broadcasts a continuous media data according to the broadcasting schedule produced by the AHB scheme, a client that starts playing S_1 as soon as it finishes receiving S_1 can play the data without any interruptions until its end. The reason is given below.

A client finishes receiving S_1 after a_1/b_1 from the time that it starts receiving S_1. Shortly after that, the client starts playing S_1. The client starts playing S_i ($i = 2, \cdots, N$) after $p_1 + \cdots + p_{i-1}$ from the time that it starts playing S_1. Therefore, it takes $a_1/b_1 + p_1 + \cdots + p_{i-1}$ for the client to start playing S_i after it starts receiving S_1. On the other hand, since the bandwidth of C_i is given by (1), the broadcasting interval of S_i is $a_1/b_1 + p_1 + \cdots + p_{i-1}$. Hence, it takes $a_1/b_1 + p_1 + \cdots + p_{i-1}$ for the client to finish receiving S_i after it starts receiving S_1. In this way, the client can finish receiving S_i before it starts playing S_i.

The name AHB is derived from the fact that the broadcasting intervals of S_i are not synchronized with that of S_1.

5 Performance Evaluation

In this section, we evaluate the necessary bandwidth and the average waiting time under the AHB scheme. Moreover, we compare the average waiting time with that under the HB scheme and the CHB scheme in the case where a client waits so that it can play the data without any interruption. In this section, we use data for 60 minutes encoded by MPEG2 ($r =$ 5 Mbps). Since units for playing a data encoded by MPEG2 are GOP, $a_1, \cdots, a_N$ are given by the data sizes for the GOP. Fig. 3 shows the histogram of units to be played, i.e., the GOP. Tab. 1 shows the relevant statistics. Although the distribution of units to be played does not always resemble the histogram, it is clear that data sizes for units to played are not constant.

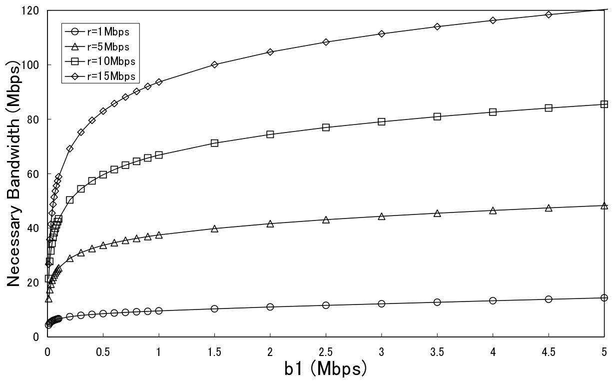

Figure 4: Necessary bandwidth under the AHB scheme.

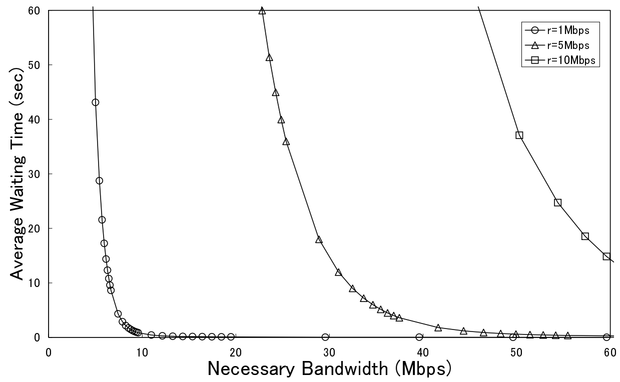

Figure 5: The necessary between and the average waiting time under the AHB scheme.

5.1 The Necessary Bandwidth under the AHB scheme

The necessary bandwidth for broadcasting continuous media data under the AHB scheme is shown in Fig. 4. The horizontal axis represents b_1 and the vertical axis the necessary bandwidth. We can see that a larger b_1 requires a larger bandwidth and the increasing rate becomes lower as b_1 increases. For example, in the case where $b_1 = 0.1$ Mbps, the necessary bandwidth is 25 Mbps, and where $b_1 = 0.2$ Mbps, the necessary bandwidth is 28 Mbps. Where $b_1 = 0.9$ Mbps, the necessary bandwidth is 36 Mbps, and where $b_1 = 1.0$ Mbps, the necessary bandwidth is 37 Mbps. Although b_1 is increased by the equivalent 0.1Mbps, the necessary bandwidths increase by $28 - 25 = 3$ Mbps or $37 - 36 = 1$ Mbps. This is because the interval of S_1 $(= a_1/b_1)$ decreases in inverse proportion to b_1. Hence, a larger b_1 further decreases the interval of S_1. Since the interval of S_1 is included in the denominator of (1), a larger b_1 increases the necessary bandwidth at the lower rate.

For example, in the case where $b_1 = 80$ Kbps, the necessary bandwidth is 24 Mbps. Since the bandwidth of a general satellite digital broadcasting system is approximately 24 Mbps, this is practical.

5.2 The Average Waiting Time

The average waiting time under the AHB scheme is shown in Fig. 5. The horizontal axis represents the bandwidth and the vertical axis the average waiting time. It is clear that a larger bandwidth reduces the average waiting time more. This is because clients can receive S_1 more frequently since a larger bandwidth further decreases the interval of S_1. In the case of broadcasting MPEG2 (5 Mbps) data for 60 minutes via a 24 Mbps satellite digital broadcasting system, the average waiting time is 47.3 seconds.

5.3 Comparison with the HB scheme and the CHB scheme

By extending conventional schemes to delay the start of playback by the largest size of units, clients can play the data without interruption. Therefore, we compare the AHB scheme

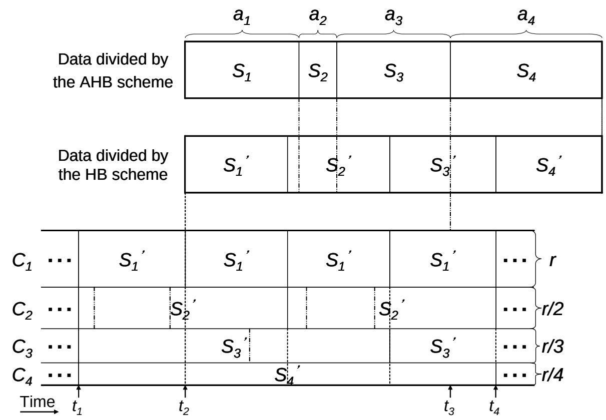

Figure 6: An example of data division under the AHB scheme and the HB scheme.

with these extended conventional schemes such as the HB scheme and the CHB scheme.

In the HB scheme and the CHB scheme, the data is assumed to be played as soon as clients start receiving it. Hence, in cases where the data is played after clients finish receiving a unit to be played, clients cannot always finish receiving the unit before the time to play it. Accordingly, clients have to wait to finish receiving the unit before playing it. Therefore, the waiting time is different from that in the case where the data is played as soon as the client starts receiving it. An example follows: In the case where $a_1 = 3r$, $a_2 = r$, $a_3 = 3r$, and $a_4 = 4r$, the AHB scheme divides the data into $S_1, \cdots, S_4$ according to the data sizes for the units to be played. However, the HB scheme divides the data into $S'_1, \cdots, S'_4$ without concerning $a_1, \cdots, a_4$, and broadcasts them as shown in Fig. 6 (a single segment includes some GOPs). Suppose that a client that demands the data at t_1 starts playing the data at t_2 as soon as it finishes receiving S_1 (some portion of S_1 included in S'_2 is received in C_2). To play the data without interruption, the client has to finish receiving S_4 before t_3, i.e., the time to play S_4. However, since the client finishes receiving S'_4, which is included in S_4, at t_4, the client cannot finish receiving S_4 until t_3. In this case, by starting to play S_1 after $t_4 - t_3$ from the time it finishes receiving

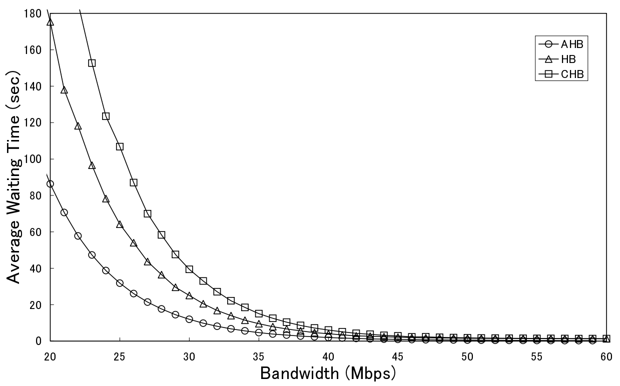

Figure 7: Average waiting time under the AHB scheme, the HB scheme, and the CHB scheme.

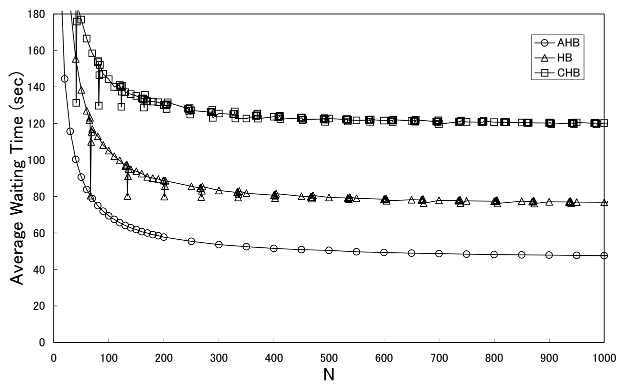

Figure 8: The number of segments and the average waiting time.

S_1', the client can play the data without interruption until the end of the data. In this case, the waiting time increases by $t_4 - t_3$. In this way, we calculated the average waiting time under the HB scheme and the CHB scheme. The result is shown in Fig. 7. The horizontal axis represents the bandwidth and the vertical axis the average waiting time. The value of b_1 under the AHB scheme and the number of segments under the HB scheme and the CHB scheme are adjusted according to the available bandwidth. The average waiting time under the HB scheme and the CHB scheme are longer than that under the AHB scheme. The average waiting time under the HB scheme is shorter than that under the CHB scheme. This is because the number of segments under the HB scheme is larger than that under the CHB scheme. Since the interval of S_1 under the HB scheme is shorter than that under the CHB scheme, the average waiting time is shorter. However, since the average waiting time depends on the data sizes for units to be played, the average waiting time under the HB scheme is not always shorter than that under the CHB scheme. For example, in the case where $a_1 = a_2 = a_3 = a_4 = r$, and the bandwidth is $2.34r$, the average waiting time under the CHB scheme is shorter than that under the HB scheme.

When broadcasting MPEG2 data (5 Mbps) for 60 minutes via a 24 Mbps satellite digital broadcasting system, the average waiting time under the AHB scheme is 47.3 seconds, whereas that under the HB scheme is 78.3 seconds and that under the CHB scheme stretches out to 123 seconds.

6 Discussion

The AHB scheme reduces the average waiting time where units to be played are given beforehand. By changing b_1, the necessary bandwidth is adjusted to the available bandwidth.

6.1 Comparison with Previous Studies

As mentioned above, the average waiting time under the AHB scheme is shorter than that under the HB scheme and the CHB scheme. In this way, the average waiting time under the AHB scheme is shorter than that under schemes that do not consider units to be played.

6.2 The Number of Segments and the Average Waiting Time

Since the average waiting time under the AHB scheme depends on the number of segments N, we discuss the effect of N on the average waiting time.

We assume that the data sizes for units to be played are equal ($a_1 = \cdots = a_N = a$). We adjust a so that the total amount of units, $a_1 + \cdots + a_N$, is $5\ \mathrm{M}{\times}60 \times 60/8 = 2250$ Mbytes. Fig. 8 shows the average waiting time when broadcasting such continuous media data ($r = 5$ Mbps) for 60 minutes via a 24 Mbps bandwidth. The horizontal axis represents the number of segments N and the vertical axis the average waiting time. It is clear that a larger N has a greater effect at reducing the average waiting time. However, a larger N further decreases the reduction rate because a becomes smaller in inverse proportion to N. For example, in the case where $N = 50$, a is 45 Mbytes. To achieve the necessary bandwidth of 24 Mbps, b_1 becomes 5.96 Mbps. In this case, the average waiting time is $(3 \times 45\ \mathrm{M}{\times}8)/(2 \times 5.96\ \mathrm{M}) = 90.6$ seconds. Where $N = 100$, a is 22.5 Mbytes and the average waiting time is $(3 \times 22.5\ \mathrm{M}{\times}8)/(2 \times 3.89\ \mathrm{M}) = 69.4$ seconds. Whereas, in the case where $N = 950$, a is 2.37 Mbytes and the average waiting time is $(3 \times 2.37\ \mathrm{M}{\times}8)/(2 \times 596\ \mathrm{K}) = 47.7$ seconds. Where $N = 1000$, a is 2.25 Mbytes and the average waiting time is $(3 \times 2.25\ \mathrm{M}{\times}8)/(2 \times 568\ \mathrm{K}) = 47.5$ seconds. In this way, although N is equivalently increased by 50, since the decreasing rate of a is different, the rate of the average waiting time reduction decreases.

Under the HB scheme and the CHB scheme, which divide data into segments in equal size, the average waiting time is sharply reduced repetitively. This occurs because the boundaries that divide the data match the boundaries of units to be played. Hence, a client that starts playing the data as soon as it finishes receiving the first unit to be played (S_1 in Fig. 6) can play the data without any interruptions. When using 24 Mbps, the HB scheme divides the data into 67 segments, and the CHB scheme divides the data into 41 segments (These numbers of segments are calculated by the given equations in [2] and [5]). Hence, if N is some integral times of these values, the average waiting time is reduced sharply. For ex-

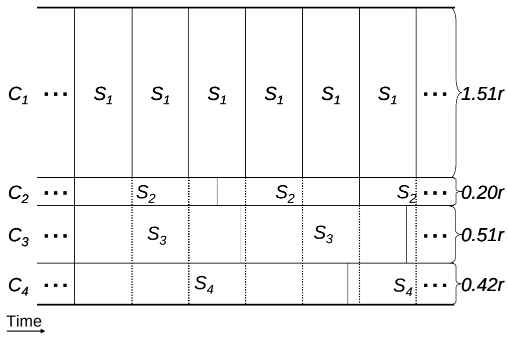

Figure 9: A broadcasting schedule under the AHB scheme for VBR.

ample, when $N = 67, 134, 201, \cdots$, the average waiting time under the HB scheme is reduced sharply. Particularly, when $N = 67$, the boundaries that divide the data under the AHB scheme are the same as those under the HB scheme. Hence, the average waiting time under the HB scheme is equal to that under the AHB scheme.

6.3 A Modification for VBR

MPEG has two types of bit rate. One is CBR, in which the consumption rate is constant while the data is played. The other is VBR (Variable Bit Rate), in which the consumption rate varies while the data is played. In this paper, although the data was assumed to be CBR, here, we introduce an application of the AHB scheme to VBR. In VBR, since the consumption rate varies with every unit to be played, let r_i ($i = 1, \cdots, N$) indicate the consumption rate of S_i. In the case of VBR, since the same arguments in Section 4.1 are established ($p_i = a_i/r_i$), a client that starts playing the data as soon as it finishes receiving S_1 can play the data completely without any interruptions by giving the bandwidth of C_j ($j = 2, \cdots, N$) by (1). Hence, the AHB scheme can be easily implemented also in the case of VBR. The necessary bandwidth is not given by (2), but:

$$B = b_1 + \sum_{i=2}^{N} \frac{a_i b_1}{a_1 + \left(\frac{a_1}{r_1} + \cdots + \frac{a_{i-1}}{r_{i-1}}\right) b_1}. \quad (4)$$

Fig. 9 shows the broadcasting schedule in the case where B, $a_1, \cdots, a_4$ are the same as in Fig. 2, and the consumption rates of segments are $r_1 = r$, $r_2 = 1.2r$, $r_3 = 0.8r$, and $r_4 = 1.1r$.

7 Conclusion

In many cases, continuous media data has a restriction in that the data divided into given data sizes are not played as soon as clients start receiving it. Accordingly, clients have to wait until a unit is received to play portions of the data. In this paper, we proposed a scheme to reduce the waiting time considering such restrictions in data division.

Our proposed scheme, the AHB scheme, reduces clients' waiting time where units to be played are given beforehand. We revealed that the average waiting time under the AHB scheme is shorter than that under the HB scheme and the CHB scheme in cases where the data is assumed to be played after clients finish receiving a unit to be played. Our future work includes development of a scheme to reduce the number of channels in order to simplify implementation.

Acknowledgement

This research was supported in part by The 21st Century Center of Excellence Program "New Information Technologies for Building a Networked Symbiotic Environment" and in part by Grant-in-Aid for Scientific Research (B) numbered 15300033 from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

REFERENCES

[1] J.-F. Paris, A Simple Low-Bandwidth Broadcasting Protocol for Video-on-Demand, Proc. of the Int. Conf. on Computer Communications and Networks (IC3N '99), pp. 118–123 (1999).

[2] J.-F. Paris, S. W. Carter, and D. D. E. Long, Efficient Broadcasting Protocols for Video on Demand, Proc. of the Int. Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS '98), pp. 127–132 (1998).

[3] J.-F. Paris, S. W.Carter, and D. D. E. Long, A hybrid broadcasting protocol for video on demand, Proc. of the Multimedia Computing and Networking Conf. (MMCN '99), pp. 317–326 (1999).

[4] K. A. Hua and S. Sheu, Skyscraper broadcasting: a new broadcasting scheme for metropolitan video-on-demand systems, Proc. of the ACM SIGCOMM, pp. 89–100 (1997).

[5] L.-S. Juhn and L. M. Tseng, Harmonic broadcasting for video-on-demand service, IEEE Trans. on Broadcasting, Vol. 43, No. 3, pp. 268–271 (1997).

[6] L.-S. Juhn and L. M. Tseng, Fast data broadcasting and receiving scheme for popular video service, IEEE Trans. on Broadcasting, Vol. 44, No. 1, pp. 100–105 (1998).

[7] R. Janakiraman and M. Waldvogel, Fuzzycast: Efficient Video-on-Demand over Multicast, Proc. of the IEEE Infocom (2002).

[8] S. Viswanathan and T. Imielinski, Pyramid broadcasting for video on demand service, Proc. of the SPIE Multimedia Computing and Networking Conf. (MMCN '95), pp. 66–77 (1995).

[9] T. Sikora, MPEG-1 and MPEG-2 Digital Video Coding Standards, http://wwwam.hhi.de/mpeg-video/papers/sikora/mpeg1_2/mpeg1_2.htm.

[10] T. Yoshihisa, M. Tsukamoto, and S. Nishio, A Scheduling Scheme to Enable Fast-Forward for Continuous Media Data Broadcasting, Proc. of the IEEE Int. Conf. on Multimedia Expo (ICME'04), in CD-ROM (2004).