Journal of Mobile Multimedia, Vol. 12, No.3&4 (2017) 181-196

A SCHEDULING METHOD FOR SWITCHING PLAYBACK SPEED IN SELECTIVE CONTENTS BROADCASTING

DAICHI FUKUI

Graduate School of Natural Science and Technology, Okayama University, Okayama, JAPAN

YUSUKE GOTOH

Graduate School of Natural Science and Technology, Okayama University, Okayama, JAPAN
gotoh@cs.okayama-u.ac.jp

In selective contents broadcasting, i.e. watching contents users selected themselves, the server can deliver several contents to many users. However, when users watch the data continuously, waiting time occurs by decreasing the available bandwidth and increasing the number of contents. Therefore, many researchers have proposed scheduling methods to reduce the waiting time. Although the conventional method reduces waiting time by producing the broadcast schedule in fast-forwarding, that for playing contents in normal playback becomes lengthened. In this paper, we propose a scheduling method for switching the playback speed in selective contents broadcasting. Our proposed method can make the broadcast schedule based on the configuration of the program and the available bandwidth. In addition, waiting time can be reduced by dividing each content into two types of data for fast-forwarding and normal playback and scheduling them.

Key words: fast-forwarding, play-sequence graph, scheduling, selective contents broadcasting, waiting time

1 Introduction

Selective contents, i.e. watching contents selected by users themselves, have attracted great attention. In selective contents broadcasting, the server can deliver several contents to many users. However, waiting time occurs based on the system environment. Therefore, many researchers have proposed scheduling methods to reduce the waiting time. In the conventional method, waiting time can be reduced in fast-forwarding by dividing each content into two types of data for fast-forwarding and normal playback and scheduling them. On the other hand, because the server schedules the data for normal playback after that for fast-forwarding, waiting time for playing the program in normal playback becomes lengthened.

In this paper, we propose a scheduling method for selective contents broadcasting with switching the playback speed while playing the content. Our proposed method sets the number of channels based on the configuration of the program and the available bandwidth. In addition, waiting time can be reduced by dividing each content into two types of data for fast-forwarding and normal playback and scheduling them.

The remainder of the paper is organized as follows. Selective contents are explained in Section 2. The details of delivery systems are explained in Section 3. We explain conventional scheduling methods in Section 4. Related works are introduced in Section 5. Our proposed method is explained in Section 6, evaluated in Section 7 and discussed in Section 8. Finally, we conclude the paper in Section 9.

2 Selective Contents

Since selective contents have sequences in which to play them, and their structures can be described by state-transition graphs. Therefore, in [1], we have proposed a play-sequence graph. Here, we explain the play-sequence graph to make the paper self-contained. In a play-sequence graph, each node represents a state in which a user plays a content. Contents are scenes of a few minutes in duration. When finished playing the content, the user state transits to the next node. For example, a play-sequence graph for a quiz program is shown in Figure 1-(A). Node S_1 is a state where the user plays a video that presents the quiz. The playing time of S_1 is 1 min. After 1 min. from the start of S_1, the state transits to the next node S_2, which is a state where the user plays a video that explains answers X and Y. The user selects his/her answer from X or Y while playing the video. The playing time of S_2 is 1 min. If the user selects X, the state transits to S_3, and if Y is selected, the state transits to S_4. When the user does not select an answer, the state transits to S_2 again, and the user repetitively plays the same video from the beginning. The playing time of S_3 or S_4 is 1 min. In this way, the state transits to the next node based on the choices.

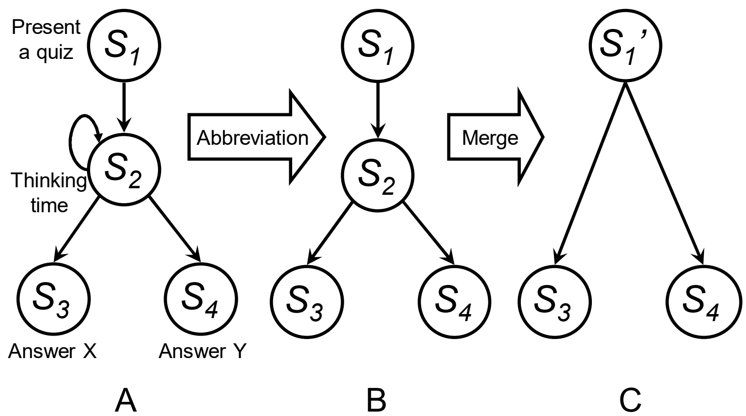

Figure 1 Play-sequence graph.

Play-sequence graphs can be simplified by applying the following three operations: abbreviation, merge, and split.

(1) Abbreviation: Users can play received contents whenever they want by storing them. Accordingly, since users are ready to transit to previous states, state transitions that transit backward along the time can be abbreviated.

(2) Merge: Nodes without multiple branches can be merged with the next node. For example, in Figure 1-(B), S_1 and S_2 can be merged. By merging nodes, we can simplify the play-sequence graph.

(3) Split: A node can split into two nodes without branches. By splitting a node, we can synchronize the playing time for the branches.

By applying the above operations, we can simplify the play-sequence graph for a quiz program to Figure 1-(C). In this paper, by producing a broadcast schedule using play-sequence graphs, we reduce the waiting time for continuously playing the data.

3 Delivery System

In webcasts, there are mainly two types of delivery systems: broadcasting and video on demand (VoD). The situations that cause waiting times in VoD and broadcasting are shown in Figure 2. In a broadcasting system for delivering selective contents, the server delivers the same contents data to many clients using a constant bandwidth. The server can reduce the increase of network load. Therefore, systems using broadcast are effective in the case of many clients. However, clients have to wait until their desired data are broadcast.

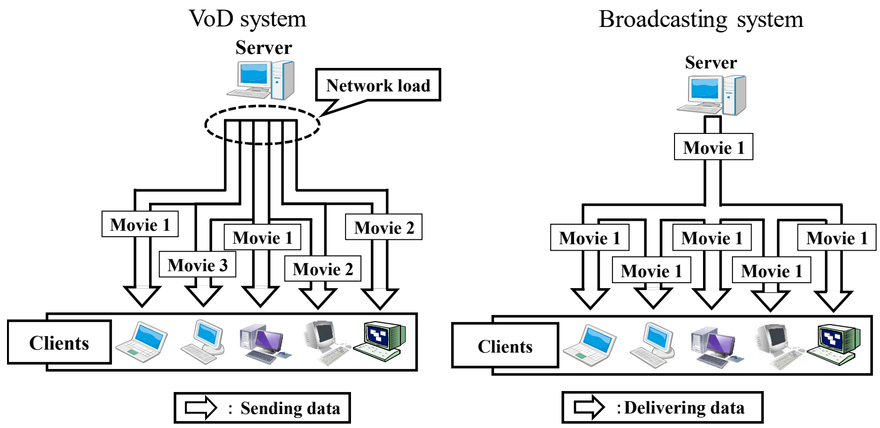

Figure 2 VoD and broadcasting environments.

In a VoD system, the server can set bandwidth for each clients, and users can watch the data immediately. However, the server increases network load with an increase of clients. In this paper, we assume a broadcasting system for delivering selective contents.

4 Scheduling Method

4.1 Basic Idea

In selective contents broadcasting by producing a broadcast schedule using play-sequence graphs, we reduce the available bandwidth. However, since the server delivers many contents concurrently, the necessary bandwidth increases. When the available bandwidth is less than the necessary bandwidth, users have to wait to watch the data. Therefore, many researchers have proposed scheduling methods to reduce the waiting time.

4.2 Simple Method

In the simple method, the available bandwidth is equally divided into the same number of channels as the number of nodes that are maximum at each depth.

When the server broadcasts a program whose play-sequence graph is shown in Figure 3, the broadcast schedule under the simple method is shown in Figure 4. The playing time of each content is 60 sec. Suppose the case where the consumption rate is 3.0 Mbps and the number of branches is a maximum of 2, the necessary bandwidth is $3.0 \times 2 = 6.0$ Mbps. In the simple method, when the number of nodes at a depth is less than the number of nodes that are maximum at each depth, the server does not broadcast contents. For example, in Figure 4, when the server broadcasts S_1 in C_1, the server does not broadcast the content in C_2.

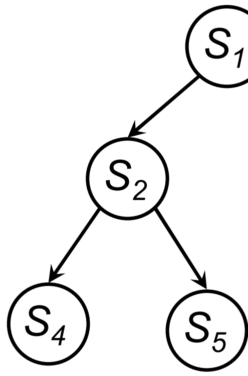

Figure 3 Play-sequence graph for news program.

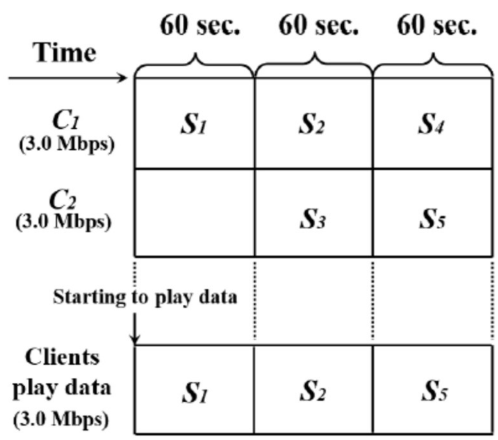

Figure 4 Example of broadcast schedule under simple method (Available bandwidth: 6.0 Mbps).

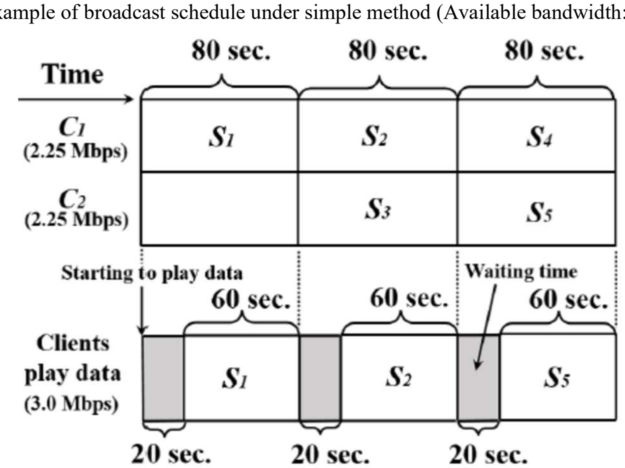

Figure 5 Example of broadcast schedule under simple method (Available bandwidth: 4.5 Mbps).

When an upper limit exists in the bandwidth, clients have to wait to watch contents. For example, when the available bandwidth is limited to 4.5 Mbps, the broadcast schedule is shown in Figure 5. The necessary bandwidth is 10 Mbps and the available bandwidth is 4.5 Mbps. Since the bandwidth for each channel is as large as simply $4.5 \div 6.0 = 0.75$ times, the bandwidth for C_1 and C_2 is $3.0 \times 0.75 = 2.25$ Mbps and the data size for selective contents is $60 \times 3.0 \div 8 = 22.5$ Mbytes. Since the bandwidth for each channel is less than the consumption rate, it takes $22.5 \times 8 \div 2.25 = 80$ sec. Since the playing time is 60 sec., the user waits $80 - 60 = 20$ sec. to play S_1. Much the same is true for $S_2, \ldots, S_5$: the user waits 20 sec. For example, when clients watch S_1, S_2, and S_5 continuously, the waiting time is 60 sec.

4.3 CCB Method

The Contents Cumulated Broadcasting (CCB) method [1] reduces the necessary bandwidth. The broadcast schedule under the CCB method is shown in Figure 6. In the CCB method, waiting time does not occur by scheduling contents considering the configuration of all programs and the available bandwidth. In Figure 6, when the bandwidths for C_1 and C_2 are each 3.0 Mbps, which is the same as the consumption rate, the server schedules S_1 and S_3 to C_1, and S_2 and S_4 to C_2. Since the broadcasting time of S_5 is 120 sec., the bandwidth for C_3 is $3.0 \times 60 \div 120 = 1.5$ Mbps. Therefore, the server broadcasts all contents for 120 sec. by using the necessary bandwidth, which is $3.0 \times 2 + 1.5 = 7.5$ Mbps.

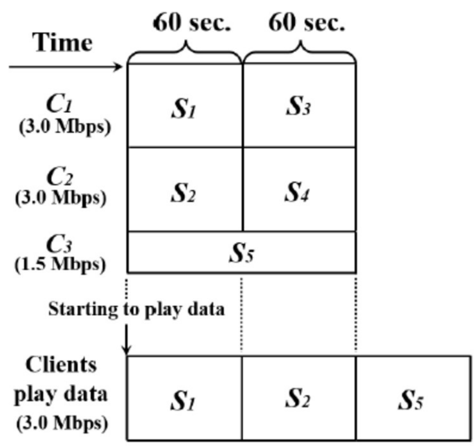

Figure 6 Example of broadcast schedule under CCB method.

Next, when the available bandwidth is limited to 4.5 Mbps, since the necessary bandwidth is 7.5 Mbps, the bandwidth for each channel is as large as simply $4.5 \div 7.5 = 0.6$ times. The bandwidths for C_1 and C_2 are $3.0 \times 0.6 = 1.8$ Mbps, and that for C_3 is $1.5 \times 0.6 = 0.9$ Mbps. It takes $22.5 \times 8 \div 1.8 = 100$ sec. to broadcast the content for 60 sec. When clients watch S_1, S_2, and S_5 continuously, the waiting time is 80 sec.

4.4 CCB-CP Method

We explain the mechanism for waiting time generation in fast-forwarding. By applying the abbreviation, the user can watch the content again.

The Contents Cumulated Broadcasting Considering Prefetching (CCB-CP) method [2] reduces the waiting time in fast-forwarding. The broadcast schedule under the CCB-CP method is shown in Figure

7. The CCB-CP method acquires channel bandwidth that is the same as the data consumption rate. In addition, waiting time can be reduced by dividing each content into two types of data for fast-forwarding and normal playback and scheduling them. When the available bandwidth is 7.5 Mbps and the consumption rate is 3.0 Mbps, the bandwidths for C_1 and C_2 are each 3.0 Mbps, and that for C_3 is 1.5 Mbps. In Figure 7, when users watch the data in fast-forwarding, the server divides n contents into two types of segments, which are D_{if} ($i = 1, \ldots, n$) and D_{ig}. In this case, waiting time does not occur.

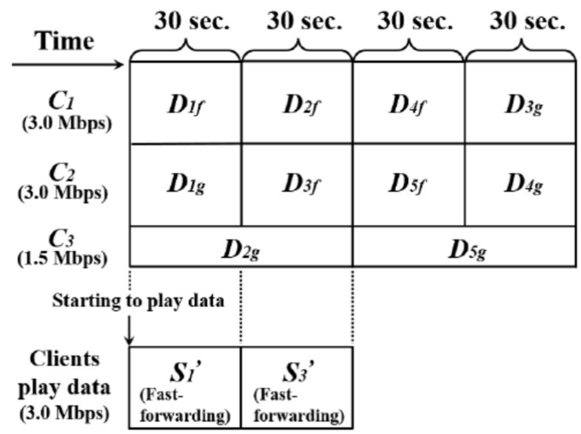

Figure 7 Example of broadcast schedule under CCB-CP method.

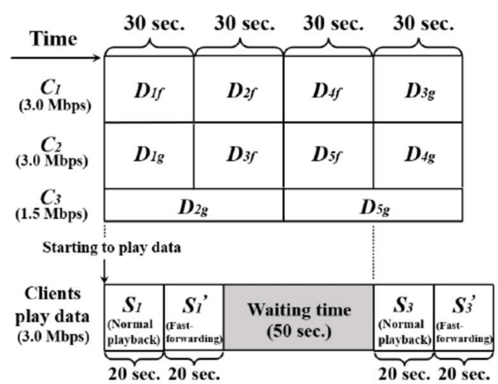

Figure 8 Broadcast schedule under CCB-CP method with fast-forwarding.

However, in the CCB-CP method, waiting time occurs in normal playback by reducing the waiting time in fast-forwarding. When users watch the data in normal playback for the first 20 sec. and that in fast-forwarding for the next 20 sec. sequentially, the broadcast schedule under the CCB-CP method is shown in Figure 8. Suppose the playing time of each content is 60 sec., and the user watches S_1 and S_3. S_i is the content for normal playback by combining D_{if} and D_{ig}, and S_i' is the content for fast-forwarding by D_{if}. By using D_{3f} and D_{3g}, the user can watch S_3 in normal playback. Although the server starts delivering D_{3f} after 30 sec., since the server starts delivering D_{3g} after 90 sec., the user has to wait $90 - 40 = 50$ sec. to play S_3 in normal playback.

5 Related Works

5.1 Selective Contents Broadcasting

In selective contents broadcasting, several methods have been proposed to reduce waiting time [3, 4]. The Contents Cumulated Broadcasting - Considering Bandwidth (CCB-CB) method [5] reduces waiting time by acquiring channel bandwidth that is the same as the consumption rate. Several scheduling methods to reduce waiting time for continuous media data have been proposed [6, 7, 8, 9, 10]. The Cautious Harmonic Broadcasting (CHB) method [11] divides the data into several segments and frequently broadcasts the first segments. The Discontinuous Interactive Cautious Harmonic Broadcasting (DICHB) method [12] reduces the waiting time in fast-forwarding. The DICHB method divides the data into two types of data for fast-forwarding and normal playback. Users can watch the data continuously when they select the data for playback speed. The Harmonic Division (HD) method [13] divides the data into several segments so that clients can receive the next segment immediately in Near-VoD (NVoD) system.

In on-demand delivery, the Restricted waiting time for Selective Contents (RSC) method [14] reduces waiting time by setting the upper limit in waiting time after selecting the content.

5.2 Division-based Broadcasting

In division-based broadcasting systems, scheduling methods have been proposed to reduce waiting times. The Fast Broadcasting (FB) method [15] sets the ratio of the data size for each segment and schedules it without interruption. When the ratio of the data size in the first segment is 1, the FB method sets it in the *n*th segment as $2^n - 1$. The server schedules segments using channels where all the available bandwidths are equal. When the available bandwidth of each channel equals the consumption rate, the waiting time under the FB method can be reduced more than that under the BE-AHB method [16]. However, when the available bandwidth of each channel is less than the consumption rate, interruption occurs while playing the data.

In Optimized Periodic Broadcast (OPB) [17], each bit of data is separated into two parts. The server uses several broadcast channels and distributes each segment on each channel. When clients completely receive the precedent parts of the content, they start receiving the remaining portions of the data. Since clients can get the sub-segment data in advance, waiting times can be reduced. However, the bandwidth increases as the amount of contents increases.

In Heterogeneous Receiver-Oriented Broadcasting (HeRO) [18], the server divides the data into *K* segments. Let J be the data size for the first segment. The data sizes for the segments are $\mathrm{J}, 2\mathrm{J}, 2^2 J, \ldots, 2^{K-1} J$. The server broadcasts these segments using *K* channels. However, since the data size of the *K*th channel becomes half of the data, clients may have delays and interruptions.

6 Proposed Method

6.1 Basic Idea

In selective contents broadcasting, we propose a scheduling method to enable fast-forwarding for reducing waiting time called the ``Contents Cumulated Broadcasting for Switching Speed (CCB-SS)" method. The CCB-SS method sets the number of channels based on the configuration of the program and the available bandwidth. In addition, waiting time can be reduced by dividing each content into two types of data for fast-forwarding and normal playback and scheduling them. In the case of where the server divides each content into two types of data for fast-forwarding and normal playback, if each bandwidth of channels is more than half of the consumption rate, users can play the program without interruption.

6.2 Assumed Environment

Our assumed system environment is summarized below.

(1) The broadcast data is selective contents.

(2) The playing time of each content is the same.

(3) The server can concurrently broadcast data with multiple channels.

(4) Clients have adequate buffer to store the received data.

(5) Once clients start playing the program, they can play it without interruption.

(6) Fast-forwarding is defined as double speed.

(7) Clients can play the content in fast-forwarding.

6.3 Scheduling Process

n is the number of states, the maximum depth is d, B is the available bandwidth, and r is the consumption rate. D_{if} (i = 1, …, n) is the state in which the client plays the content at double speed. D_{ig} (i = 1, …, n) is the state in which the client plays the content at normal speed. The scheduling process continues as follows.

(1) When m is the number of channels, the server calculates m according to formula (A):

$$m = \begin{cases} \left\lceil \frac{2n}{d} \right\rceil & (B \geq \frac{r}{2} \times \left\lceil \frac{2n}{d} \right\rceil) \\ \left\lfloor \frac{2n}{d} \right\rfloor & (B < \frac{r}{2} \times \left\lceil \frac{2n}{d} \right\rceil) \end{cases} \qquad (A)$$

(2) The available bandwidths of C_1, …, C_m are set to $\frac{B}{m}$.

(3) From the root to leaves, the server selects all contents as the candidate in the depth, which is not scheduled.

(4) The server schedules D_{if} sequentially using C_1, …, C_m, where the finishing time when delivering it is the fastest.

(5) Next, the server schedules D_{ig} using C_1, …, C_m, where the finishing time when delivering it is the fastest.

(6) Steps (3) and (4) continue repetitively until scheduling of all contents is finished.

6.4 Implementation

An example of a broadcast schedule under the CCB-SS method is shown in Figure 9. We use the play-sequence graph shown in Figure 10. In Figure 10, e is 2 and d is 4. The playing time of each content is 60 sec., the available bandwidth is 7.5 Mbps, and the consumption rate is 3.0 Mbps. In step 1, the number of channels is set to 3. In step 2, each bandwidth of C_1, C_2, and C_3 is $7.5 \div 3 = 2.5$ Mbps. In step 3, the server selects S_1 as the candidate. In step 4, the server schedules D_{1f} by C_1 for 36 sec. Next, the server schedules D_{1g} by C_2 for 36 sec. In step 3, the server selects S_2 and S_3 as the candidate. In step 4, the server schedules D_{2f} by C_3 for 36 sec., and D_{3f} by C_1 for 36 sec. Next, the server schedules D_{2g} by C_2 for 36 sec., and D_{3g} by C_3 for 36 sec. Finally, D_{4f}, D_{5f}, D_{6f}, D_{4g}, D_{5g}, and D_{6g} are scheduled sequentially. In Figure 9, when users play each content in normal playback for the first 20 sec. and that in fast-forwarding for the next 20 sec. sequentially, the waiting time for playing S_1, S_3, S_4, and S_5 sequentially occurs.

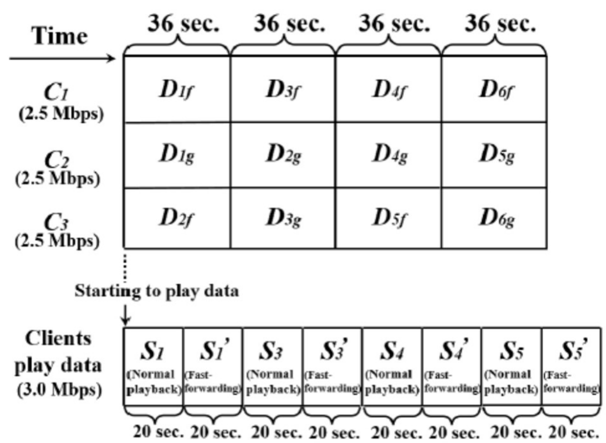

Figure 9 Broadcast schedule under CCB-SS method.

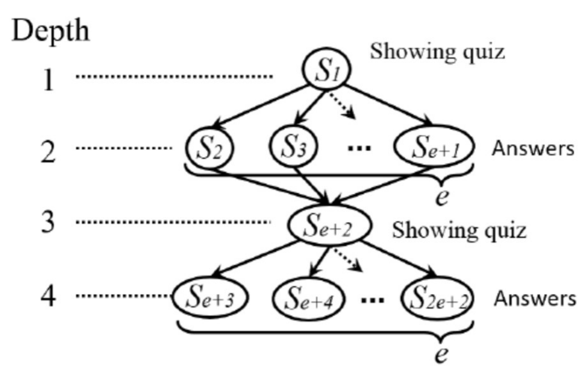

Figure 10 Play-sequence graph for quiz program.

7 Evaluation

7.1 Waiting Time

Here, we evaluate the performance of the CCB-SS method. Depending on the broadcast program, a play-sequence graph can have many patterns. However, evaluating the performance of our proposed method for all of these patterns is not realistic. Therefore, in this paper, we use the play-sequence graph shown in Figure 10. In Figure 10, the play-sequence graph corresponds to broadcasting several quizzes. Each quiz has e potential answers, and the depth is d $(\mathrm{d} \geq 2)$. We compare the average waiting time with the CCB-SS and the CCB-CP methods.

7.2 Effect of Available Bandwidth

Since the available bandwidth influences the average waiting time, the server needs to set the bandwidth based on the waiting time. Hence, we calculate the average waiting time under different available bandwidths. The number of states is 30, the playing time is 60 sec., and the consumption rate is 5.0 Mbps. We evaluate three types of watching styles below.

(1) Clients watch the content with only fast-forwarding.

(2) Clients watch the content with only normal playback.

(3) Clients watch the first half of content with normal playback and the latter half with fast-forwarding.

7.2.1 Case of Fast-forwarding

When clients play data in fast-forwarding for 30 sec., the result is shown in Figure 11. The horizontal axis is the available bandwidth, and the vertical axis is the average waiting time.

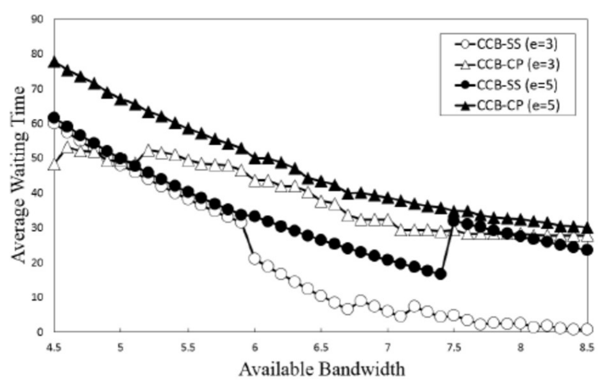

Figure 11 Average waiting time and available bandwidth with fast-forwarding.

In Figure 11, the average waiting time with the CCB-SS method is longer than the CCB-CP method. In the CCB-SS method, each bandwidth of channels is the available bandwidth divided by the number of channels and is more than half of the consumption rate. In this case, since users watch contents at double speed with interruption, waiting time becomes longer. However, we assume that users watch each content at double speed after finishing playing it at normal speed. In addition, we assume that users do not watch contents at double speed from the beginning in general.

7.2.2 Case of Normal Playback

When clients play data in normal playback for 60 sec., the result is shown in Figure 12. The horizontal axis is the available bandwidth, and the vertical axis is the average waiting time.

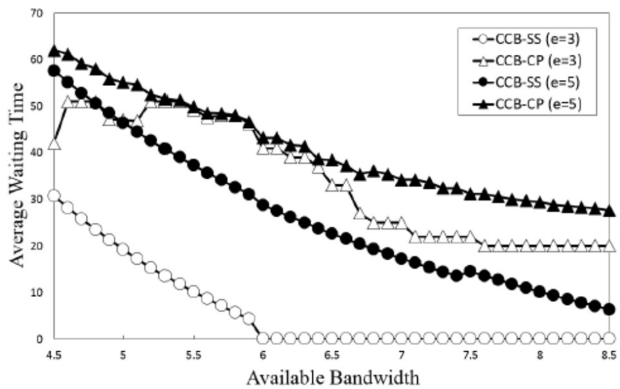

Figure 12 Average waiting time and available bandwidth with fast-forwarding.

In Figure 12, the average waiting time under the CCB-SS method is shorter than the CCB-CP method. For example, when the available bandwidth is more than 6.0 Mbps and *e* is 3, waiting time is not occurred in the CCB-SS method. In the CCB-SS method, since each bandwidth of channels is more than half of the consumption rate, users can play the program without interruption when the server schedules them for different channels at the same starting time. Therefore, when users play contents in normal playback, waiting time is not occurred.

7.2.3 Case of Fast-forwarding after Normal Playback

When clients play data in fast-forwarding for 20 sec. after playing it in normal playback for 20 sec., the result is shown in Figure 13. The horizontal axis is the available bandwidth, and the vertical axis is the average waiting time.

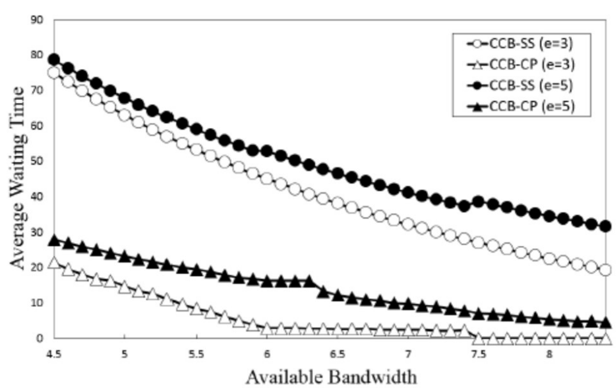

Figure 13 Average waiting time and available bandwidth with both fast-forwarding and normal playback.

In Figure 13, when e is 3, the average waiting time under the CCB-SS method is shorter than the CCB-CP method. In the CCB-SS method, each bandwidth of channels is more than half of the consumption rate. Since the server divides each content into two types of data for fast-forwarding and normal playback, users can play the program without interruption when the server schedules them for different channels at the same starting time. Therefore, when users play contents at normal speed, the average waiting time under the CCB-SS method is shorter than the CCB-CP method. For example, suppose the case where the available bandwidth is 7.0 Mbps and e is 3, waiting time under the CCB-SS method is 5.9 sec., and that under the CCB-CP method is 32.3 sec. Therefore, the average waiting time under the CCB-SS method is reduced $(32.3 - 5.9)\times 100 \div 32.3 \cong 81.7\%$ compared to the CCB-CP method.

7.3 Effect of Number of States

To evaluate the influence of the number of states, we calculated the waiting time under different numbers of states. The available bandwidth is 6.5 Mbps, the playing time is 60 sec., and the consumption rate is 5.0 Mbps. As well as the case in Subsection 7.2, we evaluate the average waiting time for three types of watching styles.

7.3.1 Case of Fast-forwarding

When clients play data in fast-forwarding for 30 sec., the result is shown in Figure 14. The horizontal axis is the number of states, and the vertical axis is the average waiting time.

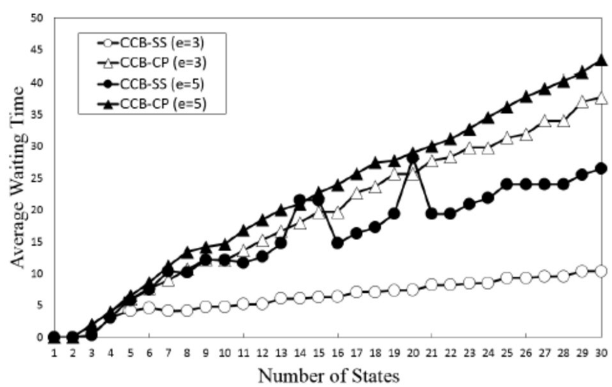

Figure 14 Average waiting time and number of states with fast-forwarding.

In Figure 14, when the number of states is 7 and e is 5, the average waiting time under the CCB-SS method is longer than the CCB-CP method. In the case of where the number of states is 7, since the depth number is 3, users can select the quiz content in all routes. However, since the number of routes is not increased if the number of quiz contents increases, the waiting time for selecting all routes becomes longer.

7.3.2 Case of Normal Playback

When clients play data in normal playback for 60 sec., the result is shown in Figure 15. The horizontal axis is the number of states, and the vertical axis is the average waiting time.

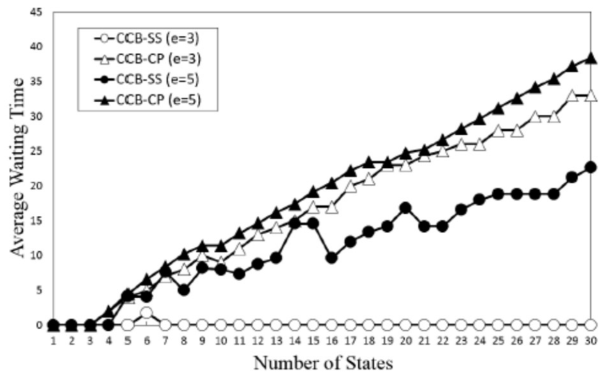

Figure 15 Average waiting time and number of states with normal playback.

In Figure 15, the average waiting time under the CCB-SS method is shorter than the CCB-CP method. Users can watch contents at normal speed with the data for fast-forwarding and that for normal playback. When the two types of data for fast-forwarding and normal playback are scheduled, users have to wait for watching contents at normal speed until delivering the data for normal playback from the server. Since the two types of data for fast-forwarding and normal playback are closely scheduled, waiting time is not lengthened. Therefore, the average waiting time under the CCB-SS method is shorter than the CCB-CP method.

7.3.3 Case of Fast-forwarding after Normal Playback

When clients play data in fast-forwarding for 20 sec. after playing it in normal playback for 20 sec., the result is shown in Figure 16. The horizontal axis is the number of states, and the vertical axis is the average waiting time.

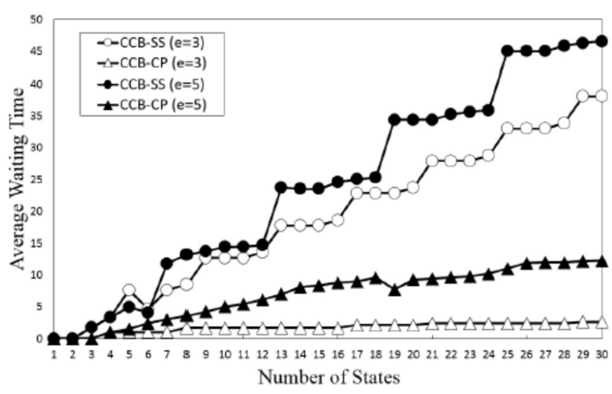

Figure 16 Average waiting time and number of states with both fast-forwarding and normal playback.

In Figure 16, the average waiting time under the CCB-SS method is shorter than the CCB-CP method. When the number of states is 20 and *e* is 5, the average waiting time under the CCB-SS method becomes longer. In the CCB-SS method, when the number of states increases, since the number of channels increases, each bandwidth of channels is less than the consumption rate. Since the content data for fast-forwarding is scheduled, the average waiting time under the case of playing data in fast-forwarding after playing data in normal playback becomes longer.

8 Discussion

For three types of watching styles as shown in Subsection 7.2, we calculated the waiting time under several available bandwidths. The result is shown in Figure 17. The horizontal axis is the available bandwidth, and the vertical axis is the average waiting time. We use the CCB-SS method, and *e* is 5.

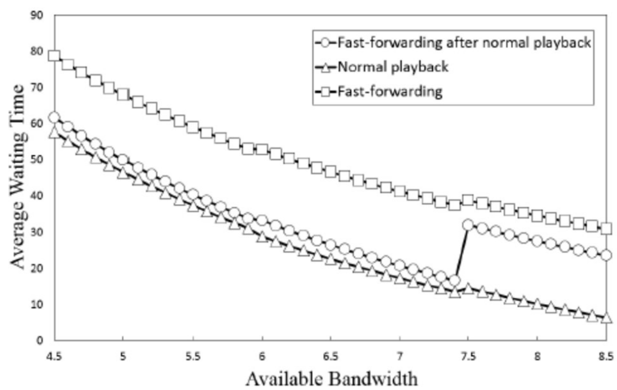

Figure 17 Average waiting time and available bandwidth with three types of watching styles.

In Figure 17, when the available bandwidth is 7.5 Mbps, the average waiting time for fast-forwarding after normal playback is longer than that when the available bandwidth is 7.4 Mbps. In the CCB-SS method, the server acquires the bandwidth that is half of the consumption rate and makes effective schedules. When the available bandwidth of each channel is less than the consumption rate, since the broadcasting time of the content is longer than the playing time, waiting time becomes lengthened. For example, when the available bandwidth is 7.5 Mbps, the number of channels is 5 and the available bandwidth of each channel is 1.5 Mbps, which is the same as the consumption rate. On the other hand, when the available bandwidth is 7.4 Mbps, the number of channels is 4 and the available bandwidth of each channel is 1.875 Mbps, which is more than the consumption rate. Our proposed method makes the broadcast schedule considering the case where clients can select the three types of watching styles.

9 Conclusion

In this paper, we proposed a scheduling method for reducing the waiting time to enable fast-forwarding selective contents broadcasting called the CCB-SS method. While clients watch a program, if they are interested in the next content, they want to finish playing the current content as soon as possible. The

CCB-SS method sets the number of channels based on the configuration of the program and the available bandwidth. In addition, the CCB-SS method divides each content into the two types of data for fast-forwarding and normal playback and schedules them. Since each bandwidth of channels is set to more than half of the consumption rate, users can watch contents at normal speed without interruption. In our evaluation, we confirmed that the average waiting time under the CCB-SS method is reduced more than that under the CCB-CP method. For example, when the available bandwidth is 7.0 Mbps and a user plays the data in fast-forwarding for 20 sec. after playing it in normal playback for 20 sec., the average waiting time under the CCB-SS method is reduced 81.7% compared to the CCB-CP method.

A further direction of this study will be to investigate watching styles, including real watching style and the same play-sequence graph as the broadcast program. In addition, we will make a scheduling method considering the transition probability in a play-sequence graph in which clients play contents with fast-forwarding.

Acknowledgements

This work was partially supported by JSPS KAKENHI Grant Number 26730059, 15H02702, and 16K01065. In addition, this work was supported in part by Sanyo Broadcasting Foundation.

References

1. Yoshihisa T., A Scheduling Method for Bandwidth Reduction in Selective Contents Broadcasting. Proc. of 3rd International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2006), 2006, 60-67.
2. Gotoh Y., Yoshihisa T., Taniguchi H., Kanazawa M., Rahayu W. and Chen Y.P.P., A Scheduling Method for Selective Contents Broadcasting with Fast-forwarding. Proc. of 2nd International Workshop on Streaming Media Delivery and Management Systems (SMDMS 2011), 2011, 344-349.
3. Gotoh Y., Yoshihisa T., Kanazawa M. and Takahashi Y., A Scheduling Method to Reduce Waiting Time Considering Transition Probability for Selective Contents Broadcasting, Proc. of 2008 IEEE International Symposium on Wireless Communication Systems (ISWCS '08), 2008, 149-153.
4. Gotoh Y., Yoshihisa T., Taniguchi H. and Kanazawa M., A Scheduling Method on Selective Contents Broadcasting with Node Relay Based Webcast Considering Available Bandwidth, Proc. of 5th International Conference on Networked Computing and Advanced Information Management (NCM 2009), 2009, 1368-1373.
5. Gotoh Y., Yoshihisa T., Kanazawa M. and Takahashi Y., A Broadcasting Scheme for Selective Contents Considering Available Bandwidth, IEEE Trans. on Broadcasting, Vol.55, Issue 2, 2009, 460-467.
6. Viswanathen S. and Imilelinski T., Pyramid Broadcasting for Video on Demand Service, Proc. of Multimedia Computing and Networking Conference (MMCN '95), Vol.2417, 1995, 66-77.
7. Yoshihisa T., Tsukamoto M. and Nishio S., A Scheduling Scheme for Continuous Media Data Broadcasting with a Single Channel, IEEE Trans. on Broadcasting, Vol.52, Issue 1, 2006, 1-10.
8. Mahanti A., Eager D., Vernon M. and Stukel D., Scalable On-demand Media Streaming with Packet Loss Recovery, IEEE/ACM Trans. on Networking, Vol.11, 2003, 195-209.
9. Zhao Y., Eager D. and Vernon M., Scalable On-demand Streaming of Nonlinear Media, IEEE/ACM Trans. on Networking, Vol.15, 2007, 1149-1162.

10. Yi S. and Shin H., A Hybrid Scheduling Scheme for Data Broadcast over a Single Channel in Mobile Environments, Journal of Intelligent Manufacturing, Vol.23, Issue 4, 2012, 1259-1269.
11. Paris J-F., Carter S.W. and Long D.D.E., efficient Broadcasting Protocols for Video on Demand, Proc. of ACM International Multimedia Conference (Multimedia '99), 1999, 189-197.
12. Yoshihisa T., Tsukamoto M. and Nishio S., A Scheduling Scheme to Enable Fast-Forward for Continuous Media Data Broadcasting, Proc. of IEEE International Conference on Multimedia Expo (ICME '04), 2004, in CD-ROM.
13. Yoshihisa T. and Nishio S., A Division-Based Broadcasting Method Considering Channel Bandwidths for NVoD Services, IEEE Trans. on Broadcasting, Vol.59, Issue 1, 2013, 62-71.
14. Gotoh Y., Yoshihisa T. and Kanazawa M., A Scheduling Method for On-demand Delivery of Selective Contents, Proc. of 4th International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2008), 2008, 17-24.
15. Juhn L.-S. and Tseng L.M., Fast Data Broadcasting and Receiving Scheme for Popular Video Service, IEEE Trans. on Broadcasting, Vol.44, Issue 1, 1998, 100-105.
16. Yoshihisa T., Tsukamoto M. and Nishio S., A Broadcasting Scheme for Continuous Media Data with Restrictions in Data Division, Proc. of 2nd International Conference on Mobile Computing and Ubiquitous Networking (ICMU 2005), 2005, 90-95.
17. Juhn L.-S. and Tseng L.M., Harmonic broadcasting for video-on-demand service, IEEE Trans. on Broadcasting, Vol.43, Issue 3, 1997, 268-271.
18. Hua K.A., Bagouet O. and Oger D., Periodic Broadcast Protocol for Heterogeneous Receivers, Proc. of Multimedia Computing and Networking (MMCN '03), Vol.5019, No.1, 2003, 220-231.