Vol. 41 No. 2 Transactions of Information Processing Society of Japan Feb. 2000

Regular Paper

Performance Evaluation of Media Synchronization for Multimedia Presentation

Jun Sato,† Koji Hashimoto,†† Michiaki Katsumoto††† and Yoshitaka Shibata††

In order to realize a general multimedia applications such as video-on-demand, multimedia conference and electronic museum, it is required to support a variety of presentations which take account of media synchronization. In this paper, we suggested a unified multimedia presentation protocol architecture considering a synchronization reference model to provide various synchronization functions depending on presentations in various applications. We designed and implemented a multimedia conference system which concurrently provides both live and stored presentations based on our proposed architecture. We also evaluated performance of a lip synchronization method in the prototype system. As a result, our synchronization method could provide effective and correct media presentations.

1. Introduction

The realization of multimedia information network applications such as video-on-demand (VoD), multimedia conference and electronic museum, has been expected as the development of the high speed network and high speed computer technologies. However, it is required to support a variety of presentation types with different media synchronization on these multimedia applications. For example, in the multimedia conference system, there are live presentations in which user's images and voices are exchanged each other by taking lip synchronization, and also stored presentation in which the electronic materials related with the conference are multicasted by taking scene synchronization. These live and stored presentations may be concurrently provided in the conference. Furthermore, in the electronic museum system, the system must provide more sophisticated presentation which is organized by several different types of media data such as audio, video, text and image based on the presentation scenario, and the system should also perform hypermedia linking functions to navigate from the current presentation to another presentations by taking context switching. Historically, these synchronization technologies have been separately developed for each level in a multimedia system. For example, in the papers of Refs. 1)–3), intra media synchronization methods in single media streams at the operating system and lower communication layers were introduced. In the papers of Refs. 4)–6), inter media synchronization methods in multiple media streams at run-time supported level were provided. In the papers of Refs. 7)–9), the synchronization between time-dependent and time independent media at presentation level were presented. On the other hand, the paper of Ref. 10) introduced a unified multimedia presentation model which integrates various synchronization methods into a layered architecture. However, specification and implementation issues involving this model were not discussed in the paper. Furthermore, those methods did not consider the influence of the dynamic load change during the actual service.

So far we have already investigated Packet Audio/Video System (PAVS)[11] the layered protocol architecture, and designed and implemented audio/video synchronization[11], dynamic rate control[12], and QoS guarantee control function[13] for Video-on-Demand based on our prototyped protocol architecture to reduce the influence of dynamic load changes in computing and network resources.

However, our PAVS architecture supported just only the stored audio/video presentation with lip synchronization, but did not support a variety of presentation types such as both stored and live presentations.

In this paper, we reorganize the previous PAVS architecture into a unified multimedia

† Faculty of Engineering, Toyo University
†† Faculty of Software and Information Science, Iwate Prefectural University
††† Communication Research Laboratory, Ministry of Posts and Telecommunications

presentation protocol architecture considering the synchronization reference model[10]. In this protocol architecture, presentation control functions including three new controllers: service controller (SC), presentation controller (PC) and media controller (MC) are introduced to provide different types of presentations such as live presentation and stored presentation[14]. We also introduce the several media synchronization methods to integrate different types of media streams in a presentation. As an example of application based on our proposed protocol architecture, we implemented the multimedia conference system which provides both stored and live presentations. In addition, we evaluated performance of lip synchronization which is realized based on our suggested protocol architecture for different environment by changing system parameters. Through the performance evaluation, our synchronization method could provide effective and correct media presentation.

In the followings, the unified multimedia presentation protocol architecture for media synchronization is proposed in Section 2. Presentation control functions organized by three new controllers based on the protocol architecture are explained in Section 3, and media synchronization methods are explained in Section 4. Furthermore, implementation of multimedia conference system is explained in Section 5 and evaluation of lip synchronization is explained in Section 6.

2. Unified Multimedia Presentation Protocol Architecture

Figure 1 illustrates a unified multimedia presentation protocol architecture which provides multimedia services uniformly for various applications.

In this architecture, three functional layers including: synchronization, media transform and media flow control layers to provide multimedia services are introduced. The synchronization layer takes various media synchronization[11] including the inter/intra media synchronizations between the continuous as well as discrete media. The media transform layer performs media format conversion, media compression and decompression. The media flow control layer performs packet flow control and packet loss control according to the load condition of the computers and networks[12]. Besides of those functions above, each layer has the QoS guarantee functions considering resource management, QoS mapping and admission control to maintain the End-to-End QoS[13].

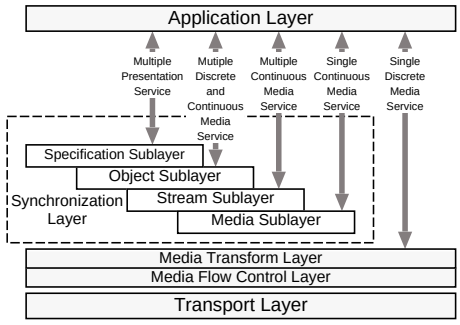


Fig. 1 Protocol architecture.

Furthermore, the synchronization layer is divided into four sub-layers to provide various multimedia services to applications as follows[10]:

Media Sub-layer Intra-media synchronization within a single continuous media stream such as audio or video is performed.

Stream Sub-layer Lip synchronization between the related audio and video is performed.

Object Sub-layer Scene synchronization among the different type of media stream such as audio, video, image and text is performed based on the presentation scenario.

Specification Sub-layer Multiple presentations are controlled to realize sophisticated multimedia application. For example, both the stored presentation and the live presentation are integrated into a presentation window for real time communication while hypermedia functions in another presentation window is used interactively.

For example, a service which is consisted of signle discrete media such as a image or a text is provided on the media transform layer, a service consisted of single continuous media service such as video or audio is provided on the media sub-layer in the synchronization layer, a service of multiple continuous media such as combination of video and audio is provided on the stream sub-layer, a service consisted of multiple discrete and continuous media such as combination of video, audio, image and text is provided on the object sub-layer, and a service consisted of multiple presentations which are consisted of multiple or single continuous/discrete media streams is provided on the specification sub-layer.

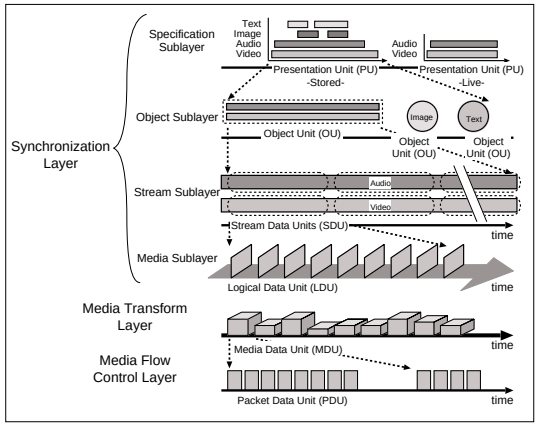


Fig. 2 Protocol data unit.

In addition, we define the protocol data units which are delivered between each layer independently to perform functions required in each layer as shown in **Fig. 2**:

Packet Data Unit (PDU) The data unit which is delivered between media flow control layer and transport layer. It is equivalent to one packet transmitted on the network.

Media Data Unit (MDU) The data unit which composes the media data, and is handled under the media transform layer. It is equivalent to a compressed audio/video frame.

Logical Data Unit (LDU) The data unit which also composes the media data, and is handled over the media transform layer. It is equivalent to a decompressed audio/video frame.

Stream Data Unit (SDU) All of LDUs in a single media, between the synchronization point at which lip synchronization between audio and video is taken.

Object Unit (OU) The data unit which is organized by the lip-synchronized audio and rideo, image or text.

Presentation Unit (PU) It is equivalent to one presentation, which is organized by multiple OUs.

3. Presentation Control Functions

It is expected that more than two presentations may be provided concurrently in one multimedia application. For example, not only live presentations with faces and voices of users are exchanged among them but also stored presentations may be also provided as the reference information in the multimedia conference system. We designed presentation control functions which handle multiple presentations as shown in **Fig. 3**.

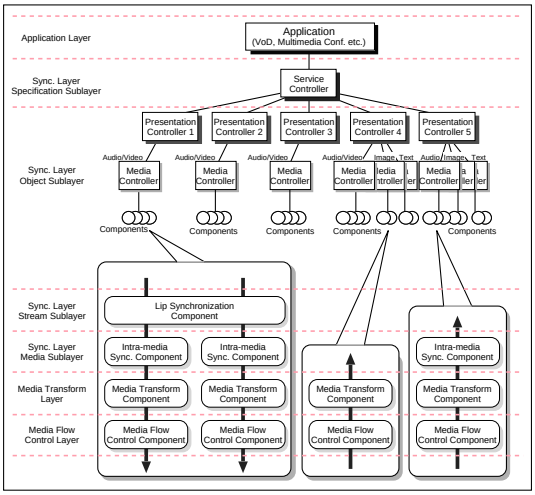


Fig. 3 Presentation control functions.

The application is located at the application layer in the protocol architecture and the functions needed for application are performed. For example, in the multimedia conference system, floor control among users and QoS negotiation are performed in the application. Here, we newly introduce three controllers: service controller (SC), presentation controller (PC) and media controller (MC) to handle different types of presentations independently. The SC controls multiple presentations in a multimedia service. The PC controls multiple media streams in one presentation. The MC performs the media transmission which is realized by several components such as a lip-synchronization component, a media transform component and a media flow control component. These components are automatically selected by the MC according to user's QoS requirements and the characteristics of media streams and presentations.

4. Media Synchronization

Multimedia presentations are realized by integrating multiple continuous and discrete media streams. In this section, several synchronization methods including intra-media synchronization and inter-media synchronization are introduced.

4.1 Intra-media Synchronization

In the media sub-layer at the synchronization layer, intra-media synchronization which refers the time interval of single continuous media

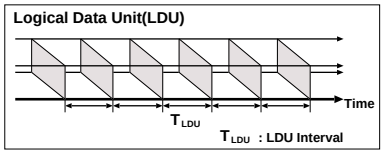


Fig. 4 Logical data units (LDUs).

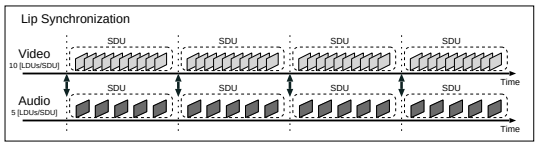


Fig. 5 Method of lip synchronization by stream data units (SDUs).

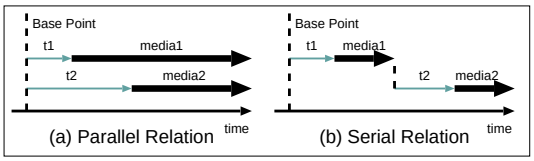


Fig. 6 Scene synchronization.

stream, such as audio or video is performed. Logical data units (LDUs)[10], which are a sequence of information units consisting continuous media stream, such as video frames, are adjusted at suitable point on the time line as shown in **Fig. 4**.

On the receiver side, the media sub-layer adjusts LDU time interval in the intra-media synchronization. For example, in order to display video frames with 30 [fps], each video frame must be displayed on every 33.3 [msec]. In order to adjust LDU interval on the stable points, we consider to use time stamp and sequence number to each LDU header. A time stamp should be used in the live presentation to represent LDUs when they are captured, and a sequence number should be used for a stored presentation to represent LDUs at a constant. However, these methods can be combined together because they can easily detect an LDU loss and end-to-end delay.

On the sender side, LDUs must be constantly transmitted so as to keep in time at the receiver side.

4.2 Inter-media Synchronization

Two inter-media synchronizations, namely the lip synchronization and the scene synchronization have been proposed. The lip synchronization takes synchronization between more than two continuous media streams such as audio and video.

The stream data units (SDUs) which are a collection of LDUs of each continuous media stream are synchronized each time at the sender and the receiver. SDU is a common unit to take lip synchronization for each media stream. In the stream sub-layer in synchronization layer, lip synchronization is realized by adjusting the position of audio SDU and video SDU as shown in **Fig. 5**.

The number of LDU in a SDU, N_{SDU} can be calculated from the lip synchronization interval T_{SDU} [sec], and the number of LDU in a second N_{LDU} [LDUs/sec] by the following equation:

$$N_{SDU} = N_{LDU}\,[\text{LDUs/sec}] \times T_{SDU}\,[\text{sec}]$$

For example, in the case where the synchronization interval is 0.5 [sec] and the video frame rate is 30 [fps], the composition unit of SDU for video is calculated as follows:

$$30\,[\text{LDUs/sec}] \times 0.5\,[\text{sec}] = 15$$

Thus, in this example, lip synchronization must be taken for each 15 video frames.

In order to provide a multimedia presentation which integrates not only continuous media but also discrete media, the synchronization function to integrate these different types of media streams based on the customized presentation scenario is required. Scene synchronization adjusts the start of navigation time of several different types of media streams.

As example, in **Fig. 6** (a), when media 1 and media 2 have a parallel relationship each other, then both are started after $t1$ [sec] and $t2$ [sec] from the base point of the original time, respectively. On the other hand, as shown in Fig. 6 (b), when media 1 and media 2 have a serial relationship each other, then the media 1 is started after $t1$ [sec] from the base point of the original time, and media 2 is started after $t2$ [sec] from the end of the media 1.

5. Implementation of Multimedia Conference System

In this section, we describe the implementation of a multimedia conference system as an example of applications based on our suggested protocol architecture.

In this multimedia conference system, while the live audio/video presentations are provided between participants, stored media multimedia presentations are also concurrently provided as reference information for the conference. Each media stream used for the stored media presentation is distributed in the databases on the network. In order to realize the multimedia confer-

ence system, we introduce the following structure model consisted of user stations (USs), service agents (SA) and multimedia data bases (MDBs) as shown in **Fig. 7**.

The USs provide presentations to users and capture the live audio/video and transmit them to other users. The MDBs store several media streams as reference information used for the conference, and each media stream is managed by the SA. The SA also manages the user information participated in the conference and provides QoS negotiation and floor control functionalities.

In the multimedia conference system, we implemented the USs, SA and MDBs on several SGI Workstations by C-language, and a number of controllers and components are realized concurrently using multiple processes and POSIX thread technologies. Through this system, the stored presentation which is organized by audio, video, image and text could be provided while an audio/video live presentation is also concurrently provided[14].

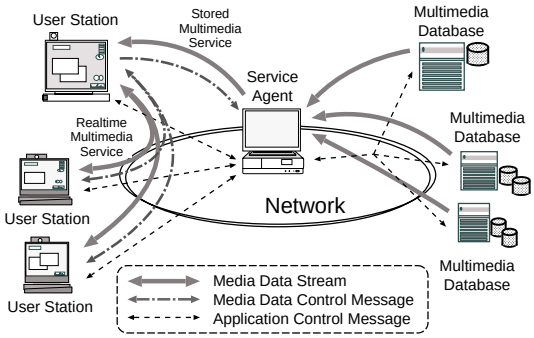


Fig. 7 A multimedia conference system.

5.1 Flow of a Multimedia Conference

Figure 8 illustrates the actual flow on the multimedia conference as an application. In this figure, it is assumed that the users A, B and C participate the conference and user A is a chair parson of the conference.

At the beginning of the conference, all of the users who want to participate the conference have to entry the conference and send request by **Entry_Conf** message to a SA which manages the conference before the conference is started **(Entry Phase)**. After the entry phase has been completed, the chair parson sends a **Open_Conf** message to the SA to open the conference.This message is multicasted to all of the USs through the SA. Furthermore, the SA and all of the USs execute their own SCs to open service after receiving **Open_Conf** messages **(Open Phase)**. During the **Service Phase**, live audio/video presentations among the users are started at first. Then each US's application sends a **Open_Presentation** message to their own SCs to open presentation. In addition, when one user requires the reference material used for the conference, the application user's US sends

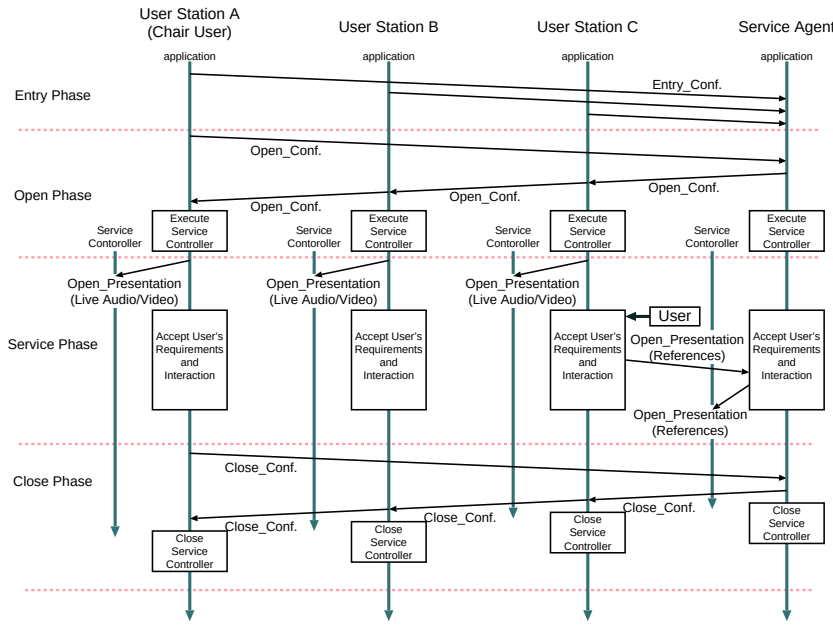


Fig. 8 Flow of a multimedia conference.

a **Open_Presentation** message to the SA. When the SA receives this message, the SA also sends a **Open_Presentation** message to his own SC to provide the new presentation to the user as the reference material. The user's presentation requirements are repeated until a **Close_Conf** message is issued from the chair parson to SA **(Service Phase)**. Finally, when the conference is terminated, the chair parson sends a **Close_Conf** message to the SA. In addition, when the all of the USs receive the **Close_Conf** message through the SA, the conference is actually closed **(Close Phase)**. During Service Phase, if a user who wants to join to the conference, he can obtain an agreement from the chair parson, the user can join to the conference during the conference. In the same way, if a user who wants to leave from the conference, he can obtain an agreement from the chair parson.

6. Performance Evaluation

We evaluated performance of lip synchronization method which takes synchronization between audio SDU and video SDU on the prototyped multimedia conference system. Each SDU is realized by one thread and synchronized by exchanging condition signals between these threads. **Figure 9** illustrates the prototyped system for our evaluation.

On the prototyped system, full color 320 × 240 [pixels] Motion-JPEG video stream with 30 [fps] and 44.1 [KHz] stereo audio stream are transmitted on the 100 [Mbps] Fast-Ethernet (**Table 1**). The lip synchronization was carried out on every 1.0 [sec] and 2.0 [sec], and the time difference between audio output and video output was measured in the three cases: 1) synchronization is carried at both sender and receiver, 2) only at sender, and 3) no synchronization.

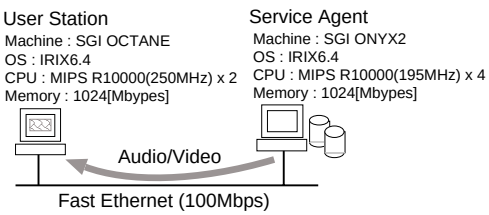


Fig. 9 Prototype system.

Table 1 Used audio and video parameters.

SDU Interval	1.0 [sec], 2.0 [sec]
Video Format	Software Motion JPEG
Video Size	320 × 240 [pixels/frame]
Video Frame Rate	30 [fps]
Audio Format	μ-law
Audio Sampling Rate	44100 [Hz]
Audio Channel	Stereo

Figures 10, **11** and **12** show in the cases where lip synchronization was taken on every 1.0 [sec]. In the case where no synchronization is executed as shown in Fig. 10, the time difference between the audio and video graduately increased and the actual audio/video presentation became unnatural. In the case where lip synchronization is only executed at the sender side as shown in Fig. 11, the video immediately delayed at the beginning of the video transmission then approached to the constant value while audio is almost constant without delay.

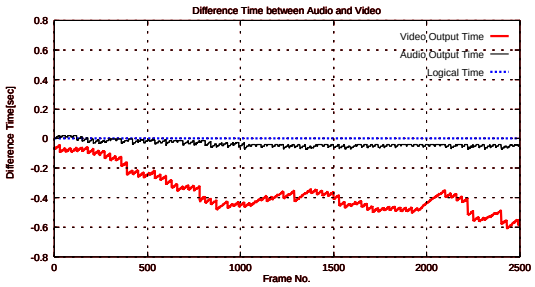


Fig. 10 Without synchronization (SDU interval = 1.0 [sec]).

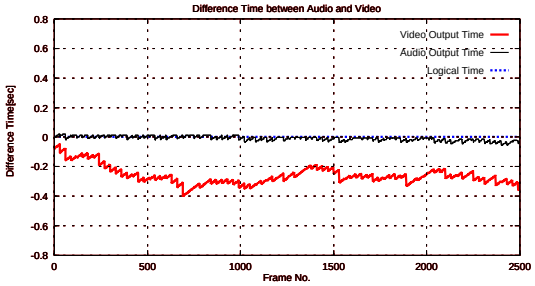


Fig. 11 With synchronization at only sender (SDU interval = 1.0 [sec]).

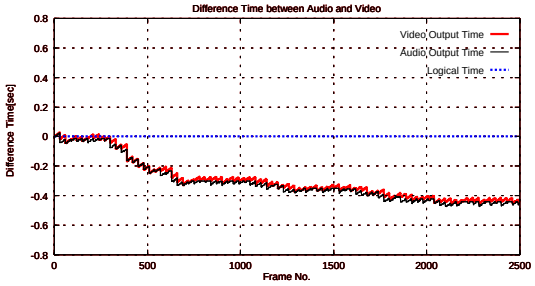


Fig. 12 With synchronization at both sender and receiver (SDU interval = 1.0 [sec]).

As a result, the time difference between the audio and video was about 0.3 [sec] in average. The reason of the delay of the video against the audio was due to the processing load by the software JPEG decoding at the receiver is so large depending on the size of the video frame. As a result, the actual video output could not be maintained within the set frame rate. In the case where lip synchronization is executed at both sender and receiver, as shown in Fig. 12, the time difference between the video and audio is almost the same although both audio and video initially are delayed but approach to the constant value. This was because the audio SDU is buffered to synchronize with the delayed video to be output at the same time.

Figures 13, **14**, and **15** show in the cases where lip synchronization was taken on every 2.0 [sec]. By comparing the cases where lip synchronization interval was 1.0 [sec] as shown in Figs. 10 through 12, the curves of all of the Figs. 13 through 15 were similar. However, the granularity of the lip synchronization was relatively coarse. This was due to the difference of relatively coarse-grained lip synchronization. On the other hand, since the number of lip synchronization in a second degreased at one-half, the processing load could be reduced, eventually the delay of the video frame could be reduced, as can been seen by comparing Fig. 11 with Fig. 14. Through this performance evaluation, our suggested synchronization method could be effective and useful.

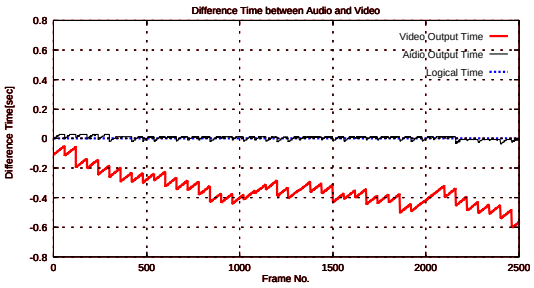


Fig. 13 Without synchronization (SDU interval = 2.0 [sec]).

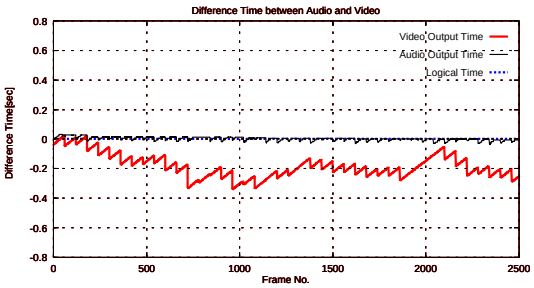


Fig. 14 With synchronization at only sender (SDU interval = 2.0 [sec]).

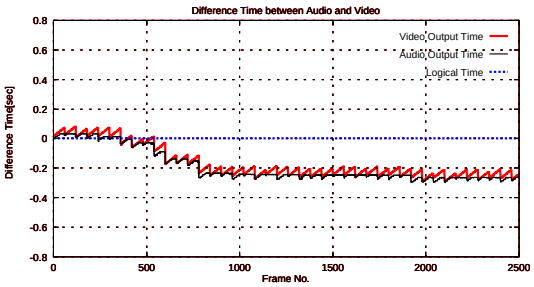


Fig. 15 With synchronization at both sender and receiver (SDU interval = 2.0 [sec]).

7. Conclusions

In this paper, we proposed the unified multimedia presentation protocol architecture and the presentation control functions which include three controllers, SC, PC and MC to support variety of presentation types such as both stored and live presentation. Furthermore, we introduced media synchronization methods including intra-media synchronization and inter-media synchronization to integrate different types of media streams in a presentation. We implemented the multimedia conference system based on our proposed protocol architecture and evaluated performance of lip synchronization. As a result, audio and video streams could be played concurrently without time difference between the audio and video when lip synchronization was applied at both sender and receiver stations.

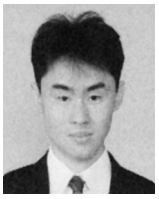
Currently, we are evaluating media synchronization methods combined with rate control under more various environments and apply the QoS guarantee functions in the synchronization methods in future research.

References

1) Nahrstedt, K. and Steinmetz, R.: Resource management in multimedia networked systems, Distributed Systems LAB79, University of Pennsylvania, Technical Report MS-CIS-94-29 (1994).
2) Oikawa, S. and Tokuda, H.: User-level real-time threads: An approach toward high performance multimedia threads, *Proc. 4th Int. Workshop on Network and Operating System Support for Digital Audio and Video*, pp.61–71 (1993).

3) Parris, C., Zhang, H. and Ferrari, D.: Dynamic management of guaranteed-performance multimedia connections, *Multimedia Syst.*, Vol.1, No.6 (1994).
4) Anderson, D.P. and Chan, P.: Toolkit support for multiuser audio/video applications, Herrtwich, R.G. (Ed.): *Network and Operating System Support for Digital Audio/Video*, (Proc. Second Int. Workshop, Heidelberg Nov. 1991), pp.230–241, Springer, Berlin (1992).
5) Anderson, D.P. and Homsy, G.: A continuous media I/O server and its synchronization mechanisms, *IEEE Computer*, Vol.24, pp.51–57 (1991).
6) Coulson, G., Garcia, F., Campbell, A. and Hutchison, D.: Orchestration services for distributed multimedia synchronisation, *Proc. 4th IFIP Int. Conf. High Performance Networking* (*HPN*), pp.14–18 (1992).
7) Blakowski, G.: High level services for distributed multimedia applications based on application media and environment descriptions, *Proc. ACSC-15* (15th Australian Computer Science Conf., 1992), also *Australian Computer Science Communications*, Vol.14, pp.93–109 (1992).
8) Li, L., Karmouch, A. and Georganas, N.: Multimedia teleorchestra with independent sources: Part 1 – temporal modeling of collaborative multimedia scenarios, *Multimedia Syst.*, Vol.1, No.4, pp.143–153 (1994).
9) ISO Multimedia and Hypermedia Information Coding Expert Group, ISO/IEC JTC1/SC29/WG12, Information technology–coded representation of multimedia and hypermedia information (MHEG), Part 1: Base notation (ASN.1), Committee Draft ISO/IEC CD 13522-1 (1993).
10) Blakowski, G. and Steinmetz, R.: A Media Synchronization Survey, Reference Model, Specification, and Case Studies, *IEEE J. Select. Areas Commun.*, Vol.14, No.1, pp.5–35 (1996).
11) Shibata, Y., Seta, N. and Katsumoto, M.: Media Synchronization Protocol for Packet Audio/Video System on Multimedia Information networks, *Proc. of HICSS-28*, pp.594–601 (1995).
12) Sato, J., Hashimoto, K., Kohsaka, Y., Shibata, Y. and Shiratori, N.: Compressed Video Transmission Protocol Considering Dynamic QoS Control, *Proc. ICPP'98 Workshop on Flexible Communication Systems*, pp.95–104 (1998).
13) Hashimoto, K. and Shibata, Y.: Performance Evaluation of End-to-End QoS Using Prototyped VOD System, *Proc. ICOIN-12*, pp.175–178 (1998).
14) Sato, J., Hashimoto, K., Katsumoto, M., Mori, H. and Shibata, Y.: Implementation of Multimedia Conference System Based on Unified Multimedia Transmission Protocol, *Proc. ICOIN-13*, Vol.2, pp.11C-2.1–11C-2.6 (1999).

(Received May 11, 1999)
(Accepted December 2, 1999)



Jun Sato was born in 1974. He received the B.S. and M.S. degrees from Toyo University in 1997 and 1999. Since 1999, he is working now at AT&T Janes. He is a member of IPSJ.

Koji Hashimoto received the B.S. and M.S. degrees from Toyo University in 1994 and 1996. From 1996 to 1998, he was working at CSK Research Institute Corp. (CRI). Since 1998, he is working as a assistant in Faculty of Software and Information Science, Iwate Prefectural University. He is a member of IPSJ.

Michiaki Katsumoto received the B.S., M.S. and Ph.D. degrees from Toyo University in 1991, 1993 and 1996 respectively. He is working now Communications Research Laboratory of Ministy of Posts and Telecommunications. His research interests include hypermedia system and multimedia database. He is a member of IPSJ, IEICE, IEEE Computer Society and ACM.

Yoshitaka Shibata received his Ph.D. in computer science from the University of California, Los Angels (UCLA) in 1985. From 1981 to 1985, he was a doctoral research associate in the Computer Science Department, where he engaged in software development of an array processor for high-speed simulation. From 1985 to 1989, he was a research member in Bell Communication Research (former Bell Laboratory), where he was working in the area of higher-layer protocol design and end-to-end performance analysis of multimedia information services. From 1989 to 1998, he conducts an intelligent multimedia network laboratory of Information and Computer Science Department in Toyo University as professor. Since 1998, he is a director of Media Center and a professor of Faculty of Software and Information Science in Iwate Prefectual University. He is a member of IPSJ, IEICE, IEEE, and ACM.