Inter–Destination Synchronization Quality in an Integrated Wired and Wireless Network with Handover

Shuji Tasaka, Yutaka Ishibashi and Masahiro Hayashi
Department of Electrical and Computer Engineering,
Nagoya Institute of Technology, Nagoya 466–8555, Japan

Abstract—**This paper presents experimental results of inter–destination synchronization control in an integrated wired and wireless network. The experimental system consists of interconnected wired and wireless networks; the former is a router–based IP network, and the latter is a wireless LAN with two base stations. A server in the wired network multicasts a pair of audio and video streams to a terminal in the same wired network and a wireless terminal, which performs handover from one base station to the other. We exert inter–destination synchronization control between the two terminals using the synchronization maestro scheme, which the authors previously proposed. In the experiment, we measured the synchronization quality of inter–destination as well as intra–stream and inter–stream when handover occurred, and we confirmed the effectiveness of the control.**

I. Introduction

Mobile access is a rapidly increasing demand for the Internet. This leads to the integrated wired and wireless network configuration for the Internet. A variety of application services are being available over such networks.

Multicasting of audio–video streams is one of the most promising applications of this type. Multimedia conferencing and distance learning are typical examples. In order to offer this type of services with good quality, we need various *QoS* (*Quality–of–Service*) control schemes, among which *media synchronization* control [1] plays an important role in the streaming services.

Media synchronization for continuous media means the preservation of the temporal relations between *media units* (*MU*s) such as video frames and voice packets. It can be classified into three categories: *intra–stream synchronization*, *inter–stream synchronization*, and *inter–destination* (or *group*) *synchronization*. The first and second ones are required in all applications of continuous media. The first one keeps the continuity of a single stream; it outputs each MU at the destination at the same intervals as the generation ones at the source. The second is synchronization among plural media streams; a typical example is synchronization between spoken voice and the movement of the speaker's lips (i.e., video), which is called *lip sync*. The third one is required in multicast communications: It controls the output timing of each MU multicast to two or more destinations so that the MU can be output simultaneously at all the destinations.

Inter–destination synchronization is an indispensable function to support some types of applications; also, it is necessary to realize the fairness among destinations in many applications. In multimedia conferencing, for instance, if the output timing of speech by a participant largely varies from destination to destination, the conference itself cannot hold. This problem arises more notably in integrated wired and wireless networks than in purely wired networks, since wireless subnetworks have many delay components different from those of wired subnetworks. They include multiple access delays, delays due to retransmission for recovery of transmission error and changes of packet routes associated with *handover* (or *handoff*). In particular, handover produces a large amount of end–to–end delay variation in a short period of time. Thus, if a media stream is multicast to both wireless destinations and wired ones at the same time, then the output timing can be quite different at the two kinds of destinations. This degrades the inter–destination synchronization quality.

We can find several researches on inter–destination synchronization in the literature [2]–[7]. In [2], Escobar *et al.* propose a flow synchronization protocol with centralized control. Akyildiz and Yen present a group synchronization mechanism for networks with known delay bounds in [3]. Also, in [4]–[7], the authors propose and evaluate group synchronization mechanisms which are merged into the *Virtual–Time Rendering* (*VTR*) media synchronization algorithm [8]. The VTR algorithm is applicable to networks with unknown delay bounds and dynamically adjusts the MU rendering–time according to the network condition. References [4] and [5] deal with centralized control. The scheme in the former is referred to as the *master–slave destination* scheme, where a single destination serves as the master to which the others adjust their output timing. The latter studies the *synchronization maestro* (or *synchronization manager*) scheme. In this scheme, the synchronization maestro collects output timing information from the destinations and distributes control information to them in order to arbitrate the output timing at each destination. Also, a distributed control scheme is described in [6]. Reference [7] proposes a group synchronization mechanism based on the synchronization maestro scheme which can be used jointly with the adaptive causality control.

As mentioned above, there have already been several approaches to the inter–destination synchronization problem; however, none of them dealt with the integrated wired and wireless network configuration, where the problem can be more serious than in purely wired networks. Thus, we do not see whether the previous methods of inter–destination synchronization control are effectively applicable to this type of networks or not.

On the other hand, many handover papers have already been published, since handover is a key technology that realizes continuity of communications for mobile users. With respect to previous studies on mobility management including handover, the reader is referred to [9], for instance. In spite of a large number of papers on this subject, to the best of the authors' knowledge, we cannot find any paper assessing media synchronization quality when handover occurs.

The purpose of this paper is twofold. The first one is to observe how the media synchronization issue, especially the inter–destination synchronization one, arises in an integrated wired and wireless network when handover occurs and then identify important technical problems. The second one is to examine whether the media synchronization schemes proposed before can provide some solutions to the problems, that is, whether they are successfully applicable to this environment or not and to gain some insight into new design methodology for media synchronization schemes in this environment. For these purposes, we carry out an experiment on media synchronization control with a scheme previously proposed by the authors; in the experiment we assess the synchronization quality of inter–destination as well as intra–stream and inter–stream when handover occurs and then confirm the effectiveness of the control.

The rest of the paper is organized as follows. Section II outlines the media synchronization control scheme adopted in our experiment for intra–stream, inter–stream and inter–destination synchronization. Section III illustrates a methodology for the quality assessment, including the experimental system configuration, experimental methods, QoS parameters and its assessment method. Section IV presents experimental results. Section V concludes the paper.

II. Media Synchronization Control Scheme

In this study, we employ a control scheme proposed in [7] for intra–stream, inter–stream and inter–destination synchronization. Its intra–stream and inter–stream synchronization control is performed according to the VTR algorithm, and inter–destination synchronization control is carried out by the synchronization maestro. In what follows, we outline the basic idea of the scheme. For details, see [7].

The VTR algorithm selects a media stream as the *master stream* and the others as *slave streams*, which are synchronized to the master. The algorithm exerts intra–stream synchronization control over both master and slave streams, while it performs inter–stream synchronization control only on slave streams after the intra-stream control. In this paper, we consider the transmission of an audio stream and the corresponding video stream. Audio is selected as the master stream and video as the slave stream since audio is more sensitive to intra–stream synchronization error than video.

We first consider intra–stream synchronization control. The disturbance of media synchronization appears in some form of delay jitters; therefore, we can achieve media synchronization by absorbing the jitters at the destination. This is carried out by buffering MUs for an appropriate period of time. It is clear that the period of time should be the maximum delay jitter. However, we cannot necessarily set the buffering time to this value, because getting the exact value in the Internet is very hard, and even if we can know it, setting the value may destroy the real–time property.

The VTR algorithm assumes no exact knowledge of the network delay jitter; by utilizing the timestamp provided to each MU at the source, it adaptively changes the buffering time according to the amount of delay jitters of MUs received at the destination. Initially, the buffering time is set to a rough estimate of the maximum delay jitter, which is denoted by $J_{\rm max}$; this value may be different from destination to destination. When inter–destination synchronization control is applied, however, a constant delay value δ instead of individual buffering times $J_{\rm max}$'s is used commonly to all the destinations; this is referred to as the *target delay time*, which is defined as the time from the moment an MU is generated until the instant the MU should be output. After the first MU is received, the buffering time or the target delay time can be changed by the modification of the *target output time*[†] of each received MU. The application form of the modification depends on the kind of media treated, i.e., stored or live. In the case of stored media, the target output time is put backward only; this is the virtual–time expansion, which corresponds to expansion of the buffering time or the target delay time. On the other hand, live media need forward movement (virtual–time contraction) as well as the backward movement, since the real–time property must be preserved. For live media, we can set the *maximum allowable delay* $\Delta_{\rm al}$ so that the modification of the target output time does not make MU delay exceed this limit. Note that δ is set to a value not larger than $\Delta_{\rm al}$. Only the master stream can modify the target output time for itself, and accordingly the slave stream modifies it by the same amount at the same time.

[†] The target output time is the time at which the MU should be output [8].

Inter–stream synchronization control is exerted over the slave stream; the output timing of each slave MU is controlled so that the difference in output time between the slave MU and the corresponding master MU can agree with the difference in timestamp between the two MUs. In this paper, we suppose *loosely–coupled media streams*, where each slave MU is not provided with the sequence number of the corresponding master MU.

Inter–destination synchronization is achieved by adjusting the MU buffering time at each destination so that its output timing can be the same at all the destinations. In order to do so, each destination reports the output timing information to the synchronization maestro at the beginning of the output of the first MU, when the target output time is modified, or when a constant number of consecutive MUs each have arrived earlier (or later) than their target output times. On the basis of this information, the maestro determines the *reference output timing* and then multicasts it to all the destinations. Each destination adjusts its own output timing to the reference output timing by modifying the target output time of the master stream.

III. Methodology for Quality Assessment

To assess the media synchronization quality and demonstrate the effectiveness of the media synchronization scheme in integrated wired and wireless network environments, we developed an experimental system. In this section, we describe the system configuration, a method of experiment, a quality assessment method and QoS parameters adopted.

A. Configuration of the Experimental System

In this study, as the wireless subnetwork, we employ a WaveLAN network [10], which offers compatibility with the IEEE 802.11 standard for operation in the 2.4–GHz band: Its transmission rate is 2 Mbps with a fallback to 1 Mbps, and the MAC protocol is CSMA/CA with Acknowledgment.

Figure 1 illustrates the configuration of the experimental system. It comprises five personal computers (PC1 through PC5), two base stations (BS1 and BS2), four 10Base–T Ethernet–hubs, two routers (Router 1 and Router 2), and a data link simulator.

CPU's and the operating systems of the PC's are as follows: PC1 (PentiumII/266 MHz, FreeBSD 3.2R), PC2 (PentiumIII/1 GHz, FreeBSD 4.4R), PC3 (PentiumII/300 MHz, FreeBSD 3.4R), PC4 (PentiumII/400 MHz, Windows NT), and PC5 (PentiumII/300 MHz, Windows 2000). Among them, PC3 and PC5 are mobile terminals each equipped with the WaveLAN/IEEE card to communicate with BS1 or BS2. In the experiment, PC3 moves so as to perform handover from BS1 to BS2, while PC5 is located at a fixed position from which it can communicate only with BS2.

BS1 and BS2 are the WavePOINT–II base stations, which are laid down on a single floor of a building as shown in Fig. 2. Each base station periodically broadcasts beacon messages to the mobile terminals for decision on handover. A beacon message contains information about the base station itself as well as link measurement data. Using received beacon messages, each mobile terminal measures the signal–to–noise ratio (SNR) from which it judges whether handover is necessary or not. When necessary, the mobile terminal sends its current base station a handover request identifying the new base station. When the current base station receives the request, it notifies the new base station of the reception via the wired subnetwork. On receiving the notification, the new base station responds with a handover acceptance message, which is sent to the mobile terminal. The reception of the acceptance message by the mobile terminal completes the handover.

Routers 1 and 2 are Cisco System's 2514 and 4700–M, respectively. They are connected to each other by a V.35 serial

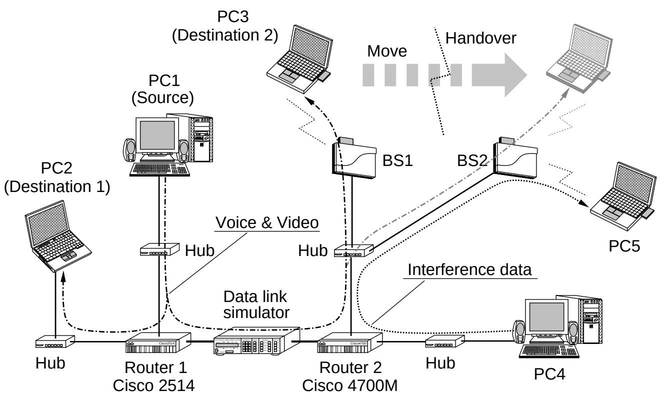

Fig. 1. Configuration of the experimental system.

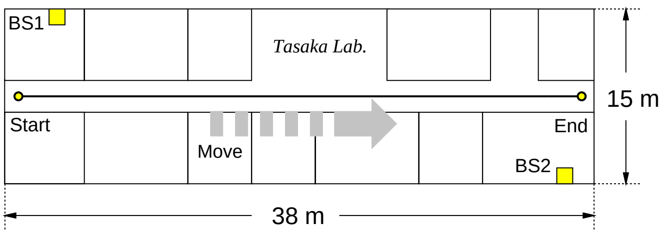

Fig. 2. Floor layout and movement method of PC3.

line through the data link simulator (ADTECH SX/12). The transmission rate of the serial line is set to 2 Mbps in our experiment.

The data link simulator can simulate various transmission error patterns and propagation delay values. In this study, we utilize the capability of producing a constant propagation delay which can take any value in the range from 0 to 2 seconds by ms; in the experiment, we set it to 50 ms. By this delay, we can simulate a simple environment of the wide area network.

B. Method of Experiment

In this study, we suppose the transmission of live audio–video streams. In the experiment, however, we used stored ones in order to generate the same amount of the stream traffic in each experimental run.

Our experiment focused on lip sync, and we employed a girl's voice and her head view video as the audio stream and video one, respectively. Table 1 shows the specifications of voice and video. The voice is encoded by the ITU–T G.711 μ–law, and a set of 400 voice samples is treated as an audio MU. The video stream is MPEG1 with only I pictures, each of which is defined as a video MU.

TABLE 1

SPECIFICATIONS OF VOICE AND VIDEO.

item	voice	video
coding scheme	ITU–T G.711 μ-law	MPEG1 GOP I
image size [pixels]	—	256 × 192
average MU size [bytes]	400	4000
original average MU rate [MU/s]	20	20
original average bit rate [kbps]	64	640
original recording time [s]	90	90

In the experiment, PC1 (the source) multicasts the voice and video as two separate transport streams to PC2 (*destination 1*) and PC3 (*destination 2*) by IP multicast [11]. UDP is used as the transport protocol. As illustrated in Fig. 2, PC3 moves from the start point to the end point at a walking speed of approximately 1.2 m/s. It starts to move about 25 seconds after the beginning of the reception of the streams, in order for the stream quality at the PC to become stable. Handover occurs during the movement.

In order to examine whether the inter–destination synchronization control is effective or not, we impose different amounts of traffic on BS1 and BS2, which makes difference in the end–to–end delay distribution between before and after the handover. For that purpose, while PC3 is moving, PC4 sends fixed–size data messages of 1472 bytes each to PC5 under the UDP protocol at exponentially distributed intervals. The amount of the data traffic is adjusted by changing the average of the interval. Let us define the *data load* as the average number of interference data bits transmitted in a second by PC4. In the measurement, the data load is set to 0.6 Mbps, 0.8 Mbps and 0.9 Mbps, which correspond to light, medium, and heavy traffic conditions, respectively, in our wireless LAN.

In this paper, we compare the media synchronization quality of three schemes: (1) joint application of the VTR algorithm and inter–destination synchronization control, which is referred to as *VTR/IDS* here, (2) application of only the VTR algorithm, which we denote simply by *VTR*, and (3) no media synchronization control, which is represented by *NC*. In the experiment, we measured the media synchronization quality of each scheme ten times and took the average.

The source of the streams (i.e., PC1) also served as the synchronization maestro. In the experiment, as the reference output timing, the maestro selected the maximum one from among the set of output timing information (i.e., the *total slide time* or the *recommended total slide time*) latest received from each destination. The maestro usually multicast the reference output timing every five seconds. However, if the current reference output timing was different from the previous one, it multicast one second later; this was done in order to cope with abrupt changes of the output timing due to handover.

Various threshold and parameter values for the media synchronization control were set to the same as those in [7] and [12] except δ=200 ms in VTR/IDS, $J_{\max}$=100 ms at destination 1 and $J_{\max}$=200 ms at destination 2 in VTR. We also set $\Delta_{\rm al}$=400 ms in both VTR/IDS and VTR except Fig. 12, where the experimental result for $\Delta_{\rm al}$=1000 ms is also shown in order to examine the effect of $\Delta_{\rm al}$. The choice of $\Delta_{\rm al}$=400 ms was made on the basis of ITU–T recommendation G.114 [13].

C. Quality Assessment Method and QoS Parameters

In order to examine the influence of handover on the media synchronization quality, especially the inter–destination synchronization quality, we measured the quality every three seconds; the measurement was made for 30 seconds before the handover and also for 30 seconds after it. We assessed the quality in each time interval of three seconds for MUs which were output in the interval.

Media synchronization quality is regarded as a kind of application–level QoS [14]. As a QoS parameter for the inter–destination synchronization quality, we adopt the *mean square error of inter–destination synchronization*, which is defined as the average square of the difference between the output time of each MU at destination 1 and that of the MU at destination 2. We also employ the *average MU delay*, the *average MU rate*, the *mean square error of intra–stream synchronization*, and the *mean square error of inter–stream synchronization*. The average MU delay denotes the average time in milliseconds from the moment an MU is generated until the instance the MU is

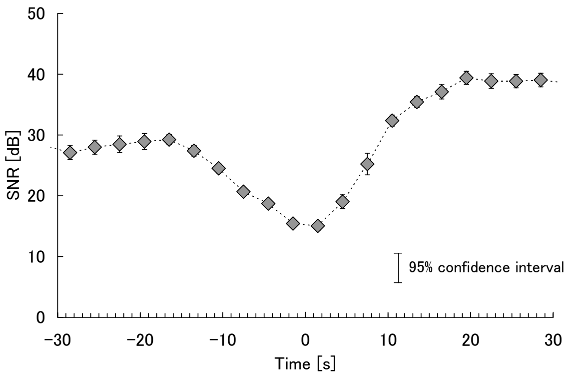

Fig. 3. SNR at destination 2 (data load: 0.8 Mbps).

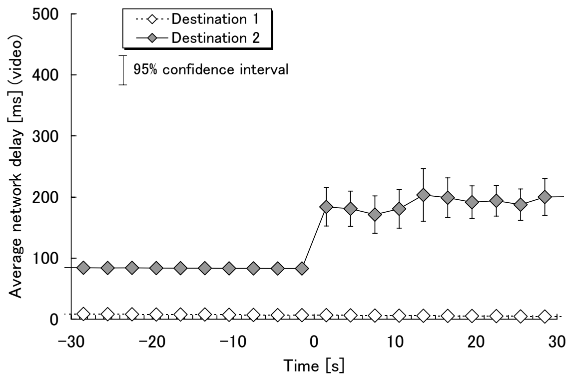

Fig. 4. Average network delay of video (data load: 0.8 Mbps).

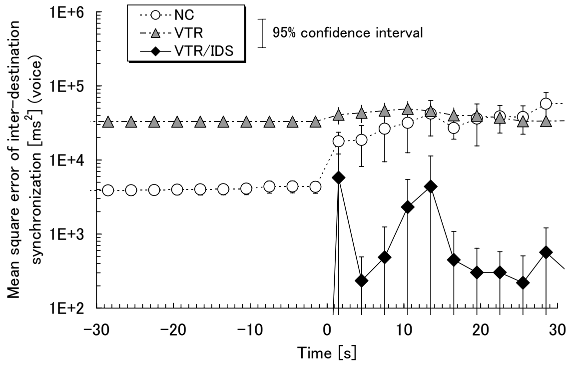

Fig. 5. Mean square error of inter–destination synchronization for voice (data load: 0.8 Mbps).

output. The average MU rate is defined as the average number of MUs output in a second. The mean square error of intra–stream synchronization is the average square of the difference between the output time of each MU and the target output time of the MU. The error represents how accurately the temporal structure of each stream is preserved. The mean square error of inter–stream synchronization, which is used as a QoS parameter for the inter–stream synchronization quality, denotes the average square of the difference between the output time of each slave MU and that of the corresponding master MU plus the relative generation time of the slave MU to the master MU. According to the subjective assessment results in [15], errors less than 6400 ($= 80^2$) ms^2 lead to inter–stream synchronization of high quality; errors larger than 25600 ($= 160^2$) ms^2 correspond to asynchrony.

IV. Experimental Results

Here we make a quality comparison among VTR/IDS, VTR, and NC as described in Subsection III-B.

A. Network State around the Handover at the Data Load of 0.8 Mbps

Before we examine the quality in terms of the QoS parameters defined in the previous section, we clarify the network state around the handover when the data load is 0.8 Mbps. This gives us useful information for getting better understanding of measured QoS parameters. We show SNR at destination 2 and the *average network delay* of video, which is the average time from the moment a video MU is sent until the instance the MU is received at the application layer, in Figs. 3 and 4, respectively. It should be noted that the two measures do not depend on which scheme is employed. In the figures, we set the instant at which the handover occurred to time 0 and plot the average of measured values since we carried out each experiment ten times as mentioned in Subsection III-B. We also display the 95 % confidence intervals of the measures; however, when the interval is smaller than the size of the corresponding symbol representing the experimental result, we do not show it in the figures.

From Fig. 3, we see that SNR gradually decreases from around -18 to 0 second, then it increases from around 0 to 18 second; therefore, destination 2 had almost the smallest SNR when the handover occurred. In Fig. 4, we observe that the average network delay at destination 2 jumps up at about time 0. This is because at the time destination 2 started to communicate with BS2, whose load was heavier than the load of BS1. At destination 1, the average network delay is small and constant independently of the time. The reason is that destination 1 is connected to the wired subnetwork. We also find in Fig. 4 that the average network delay at destination 2 is larger than 50 ms. This is because the data link simulator produces the additional propagation delay of 50 ms and destination 2 is connected to the wireless subnetwork.

B. Measurement Results at the Data Load of 0.8 Mbps

We first examine the case where the data load is 0.8 Mbps. We show the mean square error of inter–destination synchronization for voice as a function of time in Fig. 5. Figures 6 through 9 display the average MU delay of video, the average MU rates of voice and video, and the mean square error of intra–stream synchronization for voice versus the time at the data load. Although we measured the mean square error of inter–destination synchronization for video, the average MU delay of voice, and the mean square error of intra–stream synchronization for video, we do not show the measurement results here since the results for voice (or video) had almost the same tendency as the corresponding results for video (or voice). We also illustrate the mean square error of inter–stream synchronization versus the time at the data load in Fig. 10.

In Fig. 5, we can confirm that VTR/IDS has the smallest mean square error of inter–destination synchronization among the three schemes. How large mean square error is allowable depends on the type of applications and has not been clarified yet; this is for further study. We also observe in the figure that the mean square errors of all the schemes after time 0 (i.e., after the handover) are larger than those before time 0. This is because the average network delay jumps up at destination 2 at about time 0 (see Fig. 4).

From Fig. 6, we find that the average MU delays of VTR and NC at destination 2 are largely different from those at destination 1, which are independent of the time. We also observe that

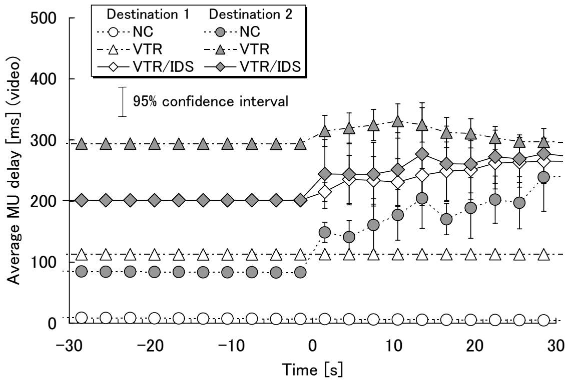

Fig. 6. Average MU delay of video (data load: 0.8 Mbps).

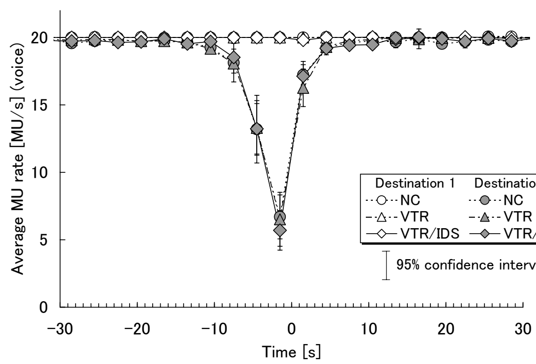

Fig. 7. Average MU rate of voice (data load: 0.8 Mbps).

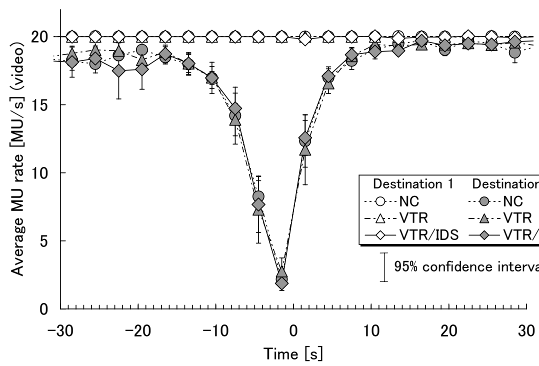

Fig. 8. Average MU rate of video (data load: 0.8 Mbps).

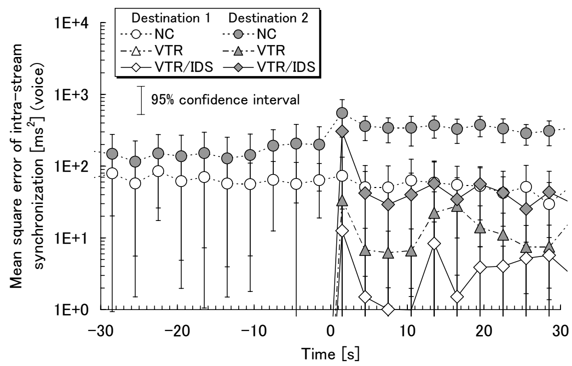

Fig. 9. Mean square error of intra–stream synchronization for voice (data load: 0.8 Mbps).

the average MU delays of NC are close to the average network delays (see Fig. 4). This is because NC outputs each MU on receiving it. VTR has the average MU delays larger than 100 ms at destination 1, which are approximately equal to the value of $J_{\max}$ (i.e., 100 ms) plus the average network delay (see Fig. 4), and those larger than 250 ms at destination 2. The reason why the average MU delay at destination 2 is larger than 250 ms is that $J_{\max}$ is set to 200 ms and the additional propagation delay is 50 ms. It should be noted that since the average network delay is around 83 ms at destination 2 before time 0 (i.e., before the handover) in Fig. 4, the average MU delay is somewhat larger than the value of $J_{\max}$ (200 ms) plus 83 ms. We further see that the difference in average MU delay between destinations 1 and 2 becomes larger after time 0. In the case of VTR/IDS, destination 1 has almost the same average MU delay as destination 2 (the average MU delays of VTR/IDS are 200 ms before time 0 since δ =200 ms); therefore, the mean square error of inter–destination synchronization is the smallest among the three schemes (see Fig. 5). This is the effect of the inter–destination synchronization control.

Figures 7 and 8 reveal that the three schemes have almost the same average MU rates. We also notice that the average MU rates at destination 2 have "gorges" (i.e., sudden fall and the following sharp rise) at about time 0. This is because the number of lost MUs increases owing to deterioration in SNR (see Fig. 3) and interruption of communication associated with handover. On the other hand, the average MU rates at destination 1 are 20 MU/s (i.e., the original average MU rate) independently of the time.

We observe in Fig. 9 that NC has the largest mean square errors of intra–stream synchronization among the three. VTR/IDS has somewhat larger mean square errors than VTR after time 0. The reason is that the output timing of MUs is delayed under the inter–destination synchronization control. However, we hardly perceived the difference in quality between VTR/IDS and VTR.

In Fig. 10, we see that the mean square errors of inter–stream synchronization of NC are the largest at destinations 1 and 2. However, since the errors are less than 6400 ms^2, the quality of inter–stream synchronization is high in all the schemes.

C. Measurement Results at the Data Loads of 0.6 Mbps and 0.9 Mbps

Next, let us assess the quality in terms of the mean square errors of inter–destination synchronization for voice at the data loads of 0.6 Mbps and 0.9 Mbps. We here show it as a function of the time in Figs. 11 and 12. Since the mean square error of inter–destination synchronization of VTR/IDS was less than 100 ms^2 at the data load of 0.6 Mbps, we do not plot the error in Fig. 11. In Fig. 12, we also plot the errors at $\Delta_{\rm al}$ of 1000 ms as well as 400 ms for VTR/IDS and VTR.

We see in Fig. 11 that the mean square error of VTR hardly depends on the time. The error of VTR is larger than 22500 $(= 150^2)$ ms^2 since the difference in $J_{\max}$ between destinations 1 and 2 is 100 ms and the additional propagation delay is 50 ms. In Fig. 12, we find that the errors of all the schemes suddenly increase up to around 10^6 ms^2 at about time 0. Then the error of VTR/IDS decreases down to approximately 5×10^5 and 30000 ms^2 at $\Delta_{\rm al}$ of 400 ms and 1000 ms, respectively. From this, we can say that as the value of $\Delta_{\rm al}$ increases in VTR/IDS, the inter-destination synchronization quality is improved more largely under heavy traffic conditions at the expense of the interactive property. On the other hand, in the figure, the errors of NC and VTR keep constant after about time 0; the error of NC is somewhat larger than that of VTR,

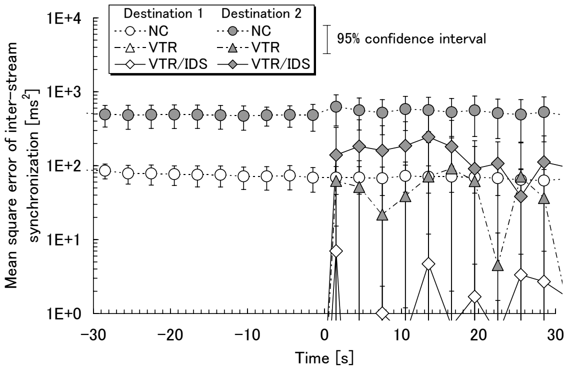

Fig. 10. Mean square error of inter–stream synchronization (data load: 0.8 Mbps).

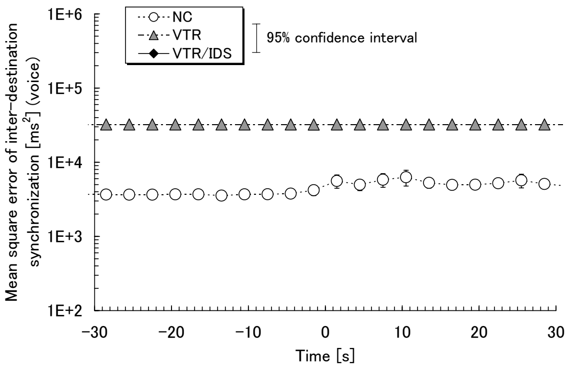

Fig. 11. Mean square error of inter–destination synchronization for voice (data load: 0.6 Mbps).

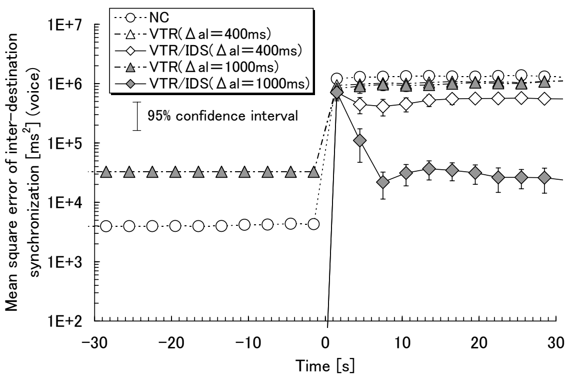

Fig. 12. Mean square error of inter–destination synchronization for voice (data load: 0.9 Mbps).

which hardly depends on the value of $\Delta_{\rm al}$.

From the above observations, we can conclude that VTR/IDS achieves higher quality of inter–destination synchronization than VTR and NC without large degradation of the other kinds of quality such as the intra–stream and inter–stream synchronization quality.

V. Conclusions

This paper examined the influence of handover on the inter–destination synchronization quality in an integrated wired and wireless network. By making a quality comparison among VTR/IDS, VTR, and NC, we confirmed the effectiveness of inter–destination synchronization control. That is, we found that VTR/IDS achieves higher quality of inter–destination synchronization than VTR and NC without largely degrading the other kinds of quality such as the intra–stream and inter–stream synchronization quality.

In this paper, we adopted the synchronization maestro scheme for the inter–destination synchronization control. In our experiment, the synchronization maestro multicast the reference output timing one second later when the current reference output timing was different from the previous one. The optimum transmission interval of reference output timing is dependent on how the network delay changes around the handover. This is one of our future research subjects. We also plan to carry out the same experiment by using other schemes such as the distributed control scheme [6] instead. In addition, we need further experimental studies for more than two destinations and in a variety of network environments.

Acknowledgment

The authors thank Taro Yokoyama for his contribution to the experiment at an early stage of this study. This work was supported by the Grant–In–Aid for Scientific Research of Japan Society for the Promotion of Science under Grant 12555109.

References

[1] G. Blakowski and R. Steinmetz, "A media synchronization survey: Reference model, specification, and case studies," *IEEE J. Sel. Areas in Commun.*, vol. 14, no. 1, pp. 5–35, Jan. 1996.

[2] J. Escobar, C. Partridge, and D. Deutsch, "Flow synchronization protocol," *IEEE/ACM Trans. Networking*, vol. 2, no. 2, pp. 111–121, Apr. 1994.

[3] I. F. Akyildiz and W. Yen, "Multimedia group synchronization protocols for integrated services networks," *IEEE J. Sel. Areas in Commun.*, vol. 14, no. 1, pp. 162–173, Jan. 1996.

[4] Y. Ishibashi, A. Tsuji, and S. Tasaka, "A group synchronization mechanism for stored media in multicast communications," in *Proc. IEEE INFOCOM'97*, pp. 693–701, Apr. 1997.

[5] Y. Ishibashi and S. Tasaka, "A group synchronization mechanism for live media in multicast communications," in *Conf. Rec. IEEE GLOBECOM'97*, pp. 746–752, Nov. 1997.

[6] Y. Ishibashi and S. Tasaka, "A distributed control scheme for group synchronization in multicast communications," in *Proc. ISCOM'99*, pp. 317–323, Nov. 1999.

[7] Y. Ishibashi, S. Tasaka, and Y. Tachibana, "Adaptive causality and media synchronization control for networked multimedia applications," in *Conf. Rec. IEEE ICC2001*, pp. 952–958, June 2001.

[8] Y. Ishibashi and S. Tasaka, "A synchronization mechanism for continuous media in multimedia communications," in *Proc. IEEE INFOCOM'95*, pp. 1010–1019, Apr. 1995.

[9] I. F. Akyildiz, J. McNair, J. S. M. Ho, H. Uzunalioglu, and W. Wang, "Mobility management in next–generation wireless systems," *Proc. IEEE*, vol. 87, no. 8, pp. 1347–1384, Aug. 1999.

[10] A. Kamerman and L. Monteban, "WaveLAN-II: A high-Performance wireless LAN for the unlicensed band," *Bell Labs Technical Journal*, pp. 118–133, Summer 1997.

[11] S. Deering, "Host extension for IP multicasting," RFC 1112, Aug. 1989.

[12] S. Tasaka, T. Nunome, and Y. Ishibashi, "Live media synchronization quality of a retransmission–based error recovery scheme," in *Conf. Rec. IEEE ICC2000*, pp. 1535–1541, June 2000.

[13] ITU–T Recommendation G.114: "Transmission systems and media, general characteristics of international telephone connections and international telephone circuits: One–way transmission time," Feb. 1996.

[14] S. Tasaka and Y. Ishibashi, "Mutually compensatory property of multimedia QoS," in *Conf. Rec. IEEE ICC2002*, pp. 1105–1111, Apr./May 2002.

[15] R. Steinmetz, "Human perception of jitter and media synchronization," *IEEE J. Sel. Areas in Commun.*, vol. 14, no. 1, pp. 61–72, Jan. 1996.