PAPER

Inter-Destination Synchronization Schemes for Continuous Media Multicasting: An Application-Level QoS Comparison in Hierarchical Networks

Toshiro NUNOME[†], *Member and* Shuji TASAKA[†], *Fellow*

SUMMARY This paper presents an application-level QoS comparison of three inter-destination synchronization schemes: the master-slave destination scheme, the synchronization maestro scheme, and the distributed control scheme. The inter-destination synchronization adjusts the output timing among destinations in a multicast group for live audio and video streaming over the Internet/intranets. We compare the application-level QoS of these schemes by simulation with the Tiers model, which is a sophisticated network topology model and reflects hierarchical structure of the Internet. The comparison clarifies their features and finds the best scheme in the environment. The simulation result shows that the distributed control scheme provides the highest quality of inter-destination synchronization among the three schemes in heavily loaded networks, while in lightly loaded networks the other schemes can have almost the same quality as that of the distributed control scheme.

key words: *multicast, live media, media synchronization, inter-destination synchronization, QoS*

1. Introduction

Live audio and video streaming is one of the most promising applications in the Internet, and multicasting is an important technique for the application. Usually, multicasting in the networks employs IP multicast [1], which provides the best-effort service and no *QoS* (*Quality of Service*) control mechanism. Thus, the temporal relations of continuous media may be disturbed by delay and its jitter during the transmission.

The disturbance can be a serious problem for interactive and collaborative applications such as quiz show and multimedia conferencing. In a quiz show, for example, contestants may feel unfairness because the contestant at the shortest delay destination gets an advantage over the others. In order to preserve the temporal relations, we can exert *media synchronization* control [2], which is one of the *application-level QoS* control [3].

We identify three types of media synchronization: *intra-stream synchronization*, *inter-stream synchronization* and *inter-destination* (or *group*) *synchronization*. The intra-stream synchronization control is necessary for the preservation of the timing relation between *media units* (*MUs*) such as video frames in a single media stream; an MU is the information unit for media synchronization. The inter-stream synchronization is required for keeping the temporal relations among MUs in multiple media streams. The inter-destination synchronization control is needed in multicast communications. The purpose of the control is to output each MU simultaneously at different destinations. 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. In hierarchical networks such as the Internet, the output timing at a destination can be quite different from that at another destination. Therefore, inter-destination synchronization is an indispensable function to support these applications.

We can find several researches on inter-destination synchronization in the literature [4]–[10]. A flow synchronization protocol is proposed in [4], where an initiator manages the distribution of control information among destinations. It is also assumed that *globally synchronized clocks* are employed; that is, clock ticks at the sources and destinations have the same advancement, and the current local times are also the same. However, the validity of the protocol has not been demonstrated sufficiently. Akyildiz and Yen present group synchronization protocols in [5], where clocks run at different rates (i.e., locally available clocks). By simulation, they evaluate the maximum, minimum and average amount of asynchrony in seconds. However, the simulation assumes only a single media stream and employs a dummy stream as the media. Furthermore, they also assume that the network delay bounds are known; however, the bounds cannot be known exactly in the Internet. In [6], Benslimane proposes an inter-destination synchronization scheme which does not need globally synchronized clocks. In the scheme, a source terminal manages control information and then announces it to all the destinations. He also assumes that the network delay bounds are known. The effectiveness of the proposed scheme has been shown by simulation; however, he does not consider the temporal structures of media streams.

The papers mentioned above have some limitations when we apply their schemes to continuous media transfer. For example, they assume that the network delay bounds are known and do not consider the temporal structures of continuous media. That is, they do not discuss media synchronization quality; we regard the quality as the major part of application-level QoS.

On the other hand, in [7]–[9], inter-destination synchronization schemes based on the *virtual-time rendering* (*VTR*) media synchronization algorithm [11] are proposed; they are the *master-slave destination* scheme, the *synchro-*

Manuscript received June 4, 2003.

Manuscript revised February 2, 2004.

[†]The authors are with the Graduate School of Engineering, Nagoya Institute of Technology, Nagoya-shi, 466–8555 Japan.

nization maestro (or *synchronization manager*) scheme, and the *distributed control* scheme. The VTR algorithm is applicable to networks with unknown delay bounds by dynamically adjusting the MU buffering time according to the network condition. It employs globally synchronized clocks.

The master-slave destination scheme is proposed in [7]. In this scheme, destinations are grouped into a *master destination* and *slave destinations*. Each slave destination adjusts the output timing of MUs to that of the master destination. This is suitable for applications in which a single destination has priority over the others; in multimedia conferencing, for instance, we can select the chairperson's terminal as the master destination. However, the scheme cannot treat all the destinations fairly.

The synchronization maestro scheme is proposed in [8]. This scheme can handle all destinations fairly. 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.

The master-slave destination and synchronization maestro schemes are centralized control ones. In these schemes, if the master or maestro cannot communicate with the other terminals owing to some trouble, no destination can take over the inter-destination synchronization control. In order to solve the problem, the distributed control scheme is proposed in [9].

The effectiveness of the master-slave destination scheme, the synchronization maestro scheme, and the distributed control scheme in terms of the application-level QoS was examined by simple experiment with a source and two destinations. For example, in [10], Tasaka et al. examine the influence of handover on the application-level QoS including inter-destination synchronization quality in an integrated wired and wireless network, where the synchronization maestro scheme is adopted.

As multicast communications become popular, its scale will grow. Furthermore, multicasting over the Internet will become popular. However, the effectiveness of the inter-destination synchronization control schemes in multicast environments with many destinations has not been clarified. Furthermore, no quantitative comparison among these schemes has been made, although the qualitative comparison is performed in [9].

In this paper, we compare the three schemes for inter-destination synchronization in terms of application-level QoS by simulation. In the simulation, we employ the *Tiers model* [12], [13], which is a sophisticated network topology model and reflects hierarchical structure of the Internet. Owing to this property of the Tiers model, we can clarify their features and find the best scheme in practical environments.

The rest of this paper is organized as follows. Section 2 describes principles of the inter-destination synchronization schemes. Section 3 illustrates a methodology for the application-level QoS assessment, including the network configuration, simulation method and QoS parameters. The simulation results are presented and discussed in Sect. 4.

2. Inter-Destination Synchronization Schemes

As the basis of inter-destination synchronization control in this paper, we employ the scheme in [14], which is based on the VTR media synchronization algorithm and proposed as a synchronization maestro scheme. We also modify the control scheme so that it can realize the master-slave destination control or the distributed control. We further enhance the distributed control scheme so as to smooth traffic due to the control packets.

In this section, we first describe an outline of the VTR media synchronization algorithm. We then explain the three inter-destination synchronization schemes: the synchronization maestro scheme, the master-slave destination scheme, and the distributed control scheme. Furthermore, we present the enhancement of the distributed control scheme.

2.1 VTR Media Synchronization Algorithm

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 as shown in Fig. 1. 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 jitter; therefore, we can achieve media synchronization by absorbing the jitter 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

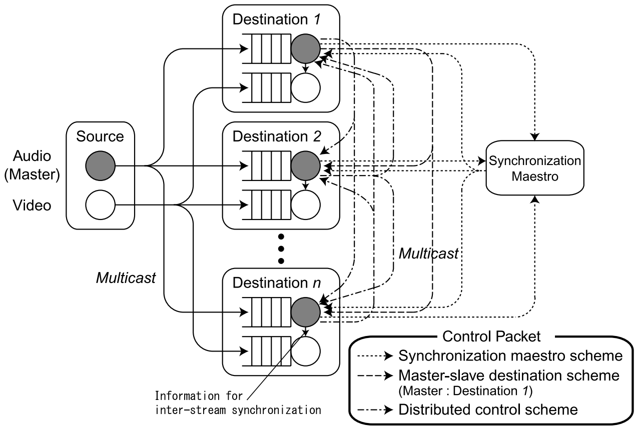
Destination 1
Destination 2
Destination n
Source
Audio (Master)
Video
Synchronization Maestro
Multicast
Multicast
Control Packet
Synchronization maestro scheme
Master-slave destination scheme (Master : Destination 1)
Distributed control scheme
Information for inter-stream synchronization

Fig. 1 A system model.

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 jitter 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 $J_{\max}$; 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 $J_{\max}$'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 target output time is the time when an MU should be output. When the MU arrives at the destination too late after the target output time, the target output time itself is expanded to absorb the jitter. In order to preserve the real-time property of live media, we can set the *maximum allowable delay* Δ_{al} [14] so that the modification of the target output time does not make MU delay exceed this limit. Furthermore, the target output time can be contracted when the amount of delay jitter decreases; this means that the buffering time decreases. 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.

Figure 2 shows an example of the timing diagram of intra-stream and inter-destination synchronization control. It depicts time lines of a source and two destinations. Destinations A and B output the n-th MU at the same time. Here, the $(n + 1)$-st MU arrives late at destination B. Then, destination B modifies the target output time.

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.

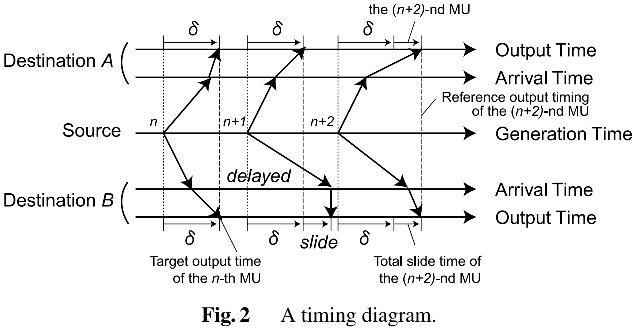
Total slide time of the (n+2)-nd MU
δ
δ
δ
Destination A
Output Time
Arrival Time
Reference output timing of the (n+2)-nd MU
Source
n
n+1
n+2
Generation Time
delayed
Destination B
Arrival Time
Output Time
δ
δ
slide
δ
Target output time of the n-th MU
Total slide time of the (n+2)-nd MU

Fig. 2 A timing diagram.

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; the timing is referred to as the *reference output timing*. For example, in Fig. 2, we assume that the latest output timing among all the destinations is selected as the reference one; it is shown by the dashed line. Each destination tries to output MUs at their reference output timings.

We describe the three inter-destination synchronization schemes below.

2.2 Synchronization Maestro Scheme

The synchronization maestro scheme employs a synchronization maestro, which gathers the information on the output timing from all destinations and adjusts the output timing among the destinations by distributing control packets as represented by the dotted arrows in Fig. 1. The maestro can be chosen from among the sources and destinations.

At the beginning of the output of the first MU in the master stream, every destination inquires of the synchronization maestro whether the target output time should be modified or not, by sending the information on the output timing to the maestro. The purpose is to adjust the output timing of the succeeding MUs among all the destinations. In this paper, we represent the output timing in terms of the *total slide time*; it denotes the total amount of modification of the target output time. Therefore, the destination sends a recommended value of the total slide time to the maestro; it is referred to as the *recommended total slide time* in this paper.

After the beginning of the output, when the destination receives a constant number of consecutive MUs each of which has arrived later or earlier than its target output time (N_c consecutive MUs have arrived later or N_d successive MUs have arrived earlier [14]), it notifies the maestro of the recommended total slide time. The recommended total slide time is different from the total slide time in that the latter is the accumulation of the slide times, while the former is employed for inquiry about the modification of the target output time in advance. The destination also notifies the synchronization maestro of the total slide time whenever the target output time is changed by the intra-stream synchronization control. In order to decide the total slide time or recommended one, the scheme specifies three parameters: r_1, r_2 and r_3 [14]. Here, r_1 is the maximum value of the slide time in the case where the total slide time is increased under the intra-stream synchronization control. The parameter r_2 is the maximum value of the recommended slide time for increment of the recommended total slide time, when N_c consecutive MUs each have arrived later than their target output times. On the other hand, r_3 is the maximum absolute value of the recommended slide time for decrement of the recommended total slide time, when N_d successive MUs each have arrived earlier than their target output times.

When the synchronization maestro receives the total slide time or recommended one from each destination, it determines the reference value of the total slide time as the reference output timing. This is performed by comparing the output timings received from the destinations. Then, the maestro multicasts the information of the reference total slide time to all the destinations when the time is changed. It also multicasts the information at regular intervals (say, 5 seconds).

Each destination gradually adjusts its own total slide time to the reference one when it receives the information on the reference output timing. Parameters r_4 and r_5 [14] are defined in order to adding and subtracting the total slide time, respectively.

2.3 Master-Slave Destination Scheme

In the master-slave destination scheme, destinations are grouped into a master destination and slave destinations. Each slave destination does not send any information on the output timing. It adjusts the target output time of MUs to that of the master destination.

This scheme uses the total slide time or the recommended total slide time at the master destination as the reference total slide time of all the destinations. When the reference output timing is changed, the master destination sends it to all the slave destinations. For example, in Fig. 1, Destination 1 is selected as the master and then it multicasts the information as indicated by the dashed arrows. In addition, the master destination periodically sends the total slide time for recovering loss of control packets (every 5 seconds in our simulation).

Each slave destination gradually adjusts its own total slide time to the reference one received from the master destination. It is performed in the same way as that for a destination under the synchronization maestro scheme. However, it should be noted that no slave destination sends any control packet including the information on the output timing.

2.4 Distributed Control Scheme

The distributed control scheme can perform inter-destination synchronization without the centralized control terminal such as the synchronization maestro or master destination. In the distributed control scheme, each destination decides the reference output timing from among the output timing of itself and that of the other destinations.

Each destination multicasts the total slide time or recommended one, which is decided in the same way as that of the synchronization maestro scheme, to all the other destinations as shown by the dot-dashed arrows in Fig. 1. In addition, each destination periodically sends the total slide time as the master destination under the master-slave destination scheme does.

In this scheme, each destination also has the same function as the synchronization maestro. When the destination outputs an MU, it decides the reference total slide time from among the output timing of itself and that of the other destinations. Then, the destination gradually adjusts its own total slide time to the reference one in the same way as that in the two centralized control schemes.

2.5 Enhancement of Distributed Control Scheme

In the original version of the distributed control scheme, a control packet is generated just after the output of an audio MU. Thus, multiple destinations may send control packets at the same time because of inter-destination synchronization. As a result, bursty traffic due to the control packets degrades the output quality of media streams.

A variety of studies on reliable multicast protocols with retransmission-based error recovery have been reported [15]. These studies have solved the *feedback-implosion* problem [16]. That is, if every receiver reports the success or failure of the data transfer, the sender will be overwhelmed with feedback packets. As an example of the solution, each destination sets a random backoff timer before sending a feedback packet [17].

This paper employs the above approach for the distributed control scheme. That is, each destination sets a random backoff timer before sending a control packet. This timer generates a value uniformly distributed between 0 and 50 ms in units of 1 ms[†]. The destination sends the control packet after the waiting time generated by the timer.

3. Methodology for Quality Assessment

We compare the application-level QoS of the inter-destination synchronization schemes by computer simulation with *ns-2* (*network simulator version 2*) [18]. In the simulation, we employ the Tiers model, which reflects hierarchical structure of the Internet.

3.1 Network Configuration

Figure 3 illustrates the network configuration in the sim-

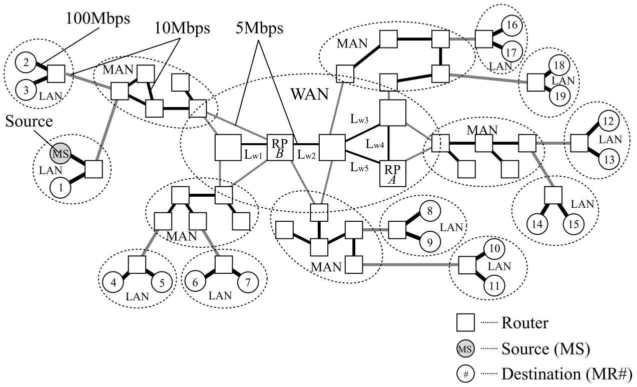

Fig. 3 Network configuration.

[†] Each control packet is generated just after the output of an audio MU. Thus, the minimum generation interval between two control packets equals the output interval between two MUs. Furthermore, the media synchronization algorithm in this paper works in the millisecond unit.

ulation. This network consists of three levels of hierarchy, which are referred to as *Wide Area Networks* (*WANs*), *Metropolitan Area Networks* (*MANs*), and *Local Area Networks* (*LANs*).

In the figure, each square denotes a router node. This topology includes a single WAN with five routers, five MANs with five routers each, and two LANs per MAN. Each LAN consists of a router and two terminals; each terminal is shown by a circle with a number which represents the destination number. We refer to the source terminal as MS. Furthermore, the i-th destination is called MRi ($i = 1, 2, \cdots, 19$).

Every link in the network is a duplex one. The transmission rate of each WAN link, which is denoted by L_{w1} through L_{w5}, is 5 Mbps. In addition, the rate of each MAN link is 10 Mbps. Furthermore, the line capacity of each LAN link is 100 Mbps. The transmission rates of WAN-MAN and MAN-LAN links are 5 Mbps and 10 Mbps, respectively. The transmission delay of each WAN link is about 4 to 30 ms, and that of each MAN link is between around 1 and 5 ms[†]. In LANs, the delay is under 1 ms. Each link has a first-in first-out (FIFO) queue for each direction. The maximum allowable queue length is specified in terms of the number of packets; we set the length to 50[††].

In this paper, we employ the *Centralized Multicast* protocol [18] for multicast routing. This is a sparse mode implementation of multicast similar to *Protocol Independent Multicast-Sparse Mode* (*PIM-SM*) [19]. In the protocol, the root of a spanning tree for a multicast group is called the *rendezvous point* (*RP*). Data packets from the senders to a group are unicast to the rendezvous point, and then the rendezvous point multicasts the data packets by using the shortest path tree routing. The route computation is run exactly once prior to the start of the simulation.

In Centralized Multicast, the location of the rendezvous point may affect the characteristics of the multicast group. Then, we investigate two locations of the rendezvous point as shown in Fig. 3; that is, RP*A* and RP*B*. Here, RP*A* is the furthest WAN router from MS (i.e., many WAN routers exist between RP*A* and MS). On the other hand, there is no WAN router in the shortest path between RP*B* and MS; that is, RP*B* is the nearest WAN router to MS.

3.2 Method of Simulation

We assume MS as the video and voice sources. MS multicasts the media streams to all the destinations by using the RTP/UDP. We use a voice stream of ITU-T G.711 μ-law and an MPEG1 video stream. Table 1 shows the specifications of the voice and video. Furthermore, we take the media capturing and encoding delay time into consideration in the simulation. The capture duration of a voice MU equals the inter-MU time, which is 50 ms in this paper, and the encoding time is negligible; therefore, we set the capturing and encoding delay time of each voice MU to 50 ms. On the other hand, the capture duration of a video MU is just a moment. However, it spends much time to encode a video frame. In this paper, we set the capturing and encoding delay time of each video MU to 8 ms, which is the same time as that in the experimental system in [20][†††]. Each MU leaves the source the capturing and encoding delay time after its timestamp.

Table 1 Specifications of the voice and video.

item	voice	video
coding scheme	ITU-T G.711 μ-law	MPEG1 GOP I
image size [pixels]	—	256 × 192
original average MU size [bytes]	400	4000
original average MU rate [MU/s]	20.0	
original average inter-MU time [ms]	50.0	
original average bit rate [kbps]	64.0	640.0
measurement time [s]	120.0	

In the simulation, interference data traffic for the voice and video streams is generated. We assume that many terminals, which are not shown in Fig. 3, are connected to the network and they transmit data to other terminals by way of the WAN. In this case, the WAN has many data flows and may be a bottleneck. In order to simulate the situation simply, we generate the traffic at each WAN router. The routers unicast fixed-size IP datagrams of 1500 bytes each to another WAN router at exponentially distributed intervals. The amount of the interference traffic is adjusted by changing the average of the interval. We set the same average of the interval for all the pairs of WAN routers. Then, we refer to the average amount of the interference traffic for each pair as the *average load*.

We compare the application-level QoS of five schemes: *NC* (*No Control*), *VTR*, *Maestro*, *Master-Slave*, and *Distributed*. NC means that no media synchronization control is carried out. VTR exerts only intra-stream and inter-stream synchronization control based on the VTR algorithm. That is, it does not employ any inter-destination synchronization mechanism. Maestro and Master-Slave denote the synchronization maestro scheme and the master-slave destination scheme, respectively. Distributed means the distributed control scheme with the random backoff timer. In [21], we find that the distributed control scheme with the random backoff timer has an advantage over that without the timer from a media synchronization quality point of view. Thus, we do not employ the scheme without the timer.

In the simulation, as the average load increases, the WAN becomes congested, even when MANs and LANs are not congested. Thus, destinations far from the rendezvous point will be affected by the interference traffic more largely than those near the rendezvous point. This is because there are many WAN routers between the rendezvous point and

[†] The original ns-2 does not take the processing delay in routers into account. Thus, we set the transmission delay so as to include it.

[††] This is a default configuration of the FIFO queue in ns-2.

[†††] In [20], JPEG is employed for video codec. On the other hand, this paper handles MPEG video. However, because of the GOP pattern in this paper, we have assumed that the capturing and encoding delay time of each MU is approximately the same as that of JPEG video in [20].

the destination which is located far from the rendezvous point. Thus, in the centralized control schemes, the location of the centralized control terminal may affect the application-level QoS of the media streams. Therefore, this paper also examines the influence of the location of the synchronization maestro and that of the master destination.

We assume either MR1 or MR19 as the centralized control terminal. *Maestro* (*MR1*) and *Maestro* (*MR19*) mean that we choose MR1 and MR19 as the synchronization maestro terminal, respectively. In Master-Slave, *Master-Slave* (*MR1*) and *Master-Slave* (*MR19*) imply that we select MR1 and MR19 as the master destination, respectively.

In the VTR algorithm, we set the target delay time δ and the maximum allowable delay Δ_{al} to 100 ms and 300 ms, respectively. In addition, we set $N_c = 10$, $N_d = 20$, $r_1 = \infty$ and $r_2 = r_3 = r_4 = r_5 = 20$ ms. The other thresholds and parameters in the VTR algorithm have the same values as those in [20]. Furthermore, the synchronization maestro scheme and the distributed control scheme select the latest output timing from among the collected output timings as the reference one[†].

3.3 QoS Parameters

In order to assess the application-level QoS of the inter-destination synchronization schemes, we need to examine the inter-destination synchronization quality as well as the intra-stream and inter-stream synchronization quality.

For the inter-destination synchronization quality, we evaluate the *mean square error of inter-destination synchronization*[††]. In two destinations X and Y, it denotes the mean square of the difference between the output time of an MU (excluding skipped MUs) at destination X and that of the MU at destination Y. In this paper, we suppose many destinations. Thus, there are many combinations of two destinations. However, some combinations have the same tendency as other combinations or very small inter-destination synchronization error. Therefore, in this paper, we select a reference destination from among all the destinations and then calculate the average of mean square errors of inter-destination synchronization between the reference destination and another one; we use it for quality assessment. We have also measured the standard deviation of the mean square errors. However, we found that the standard deviation has the same tendency as that of the average; thus, we do not show the results of the standard deviation.

On the other hand, for the quality assessment of intra-stream synchronization for voice or video, we evaluate the *coefficient of variation of output interval*, which represents the smoothness of output of a media stream.

We have also assessed the inter-stream synchronization quality in the simulation. As a result, we noticed that all the schemes have high quality of inter-stream synchronization. Thus, we do not show the result.

4. Simulation Results

In this section, we first compare the application-level QoS of the inter-destination synchronization schemes when we employ RP*A* as the rendezvous point. We then show the results when RP*B* is selected as the rendezvous point. In the comparison, we focus on inter-destination synchronization quality, intra-stream synchronization quality at MR1, and that at MR19.

In this paper, each symbol in the figures to be shown represents the average of 10 measured values which were obtained by changing the random seed for generating the interference traffic. We also show 95% confidence intervals of the QoS parameters in the figures. However, when the interval is smaller than the size of the corresponding symbol representing the simulation result, we do not show it in the figures.

4.1 Case of RP*A*

4.1.1 Inter-Destination Synchronization Quality

Figures 4 and 5 show the average of mean square errors of inter-destination synchronization between MR1 and another destination for voice and that for video, respectively.

In Figs. 4 and 5, we see that the average of mean square errors of inter-destination synchronization with Master-Slave (MR19) is larger than that with Master-Slave (MR1). This is because the number of hops between MR19 and RP*A* is smaller than that between MR1 and RP*A*. Master-Slave (MR19) selects MR19 as the master destination. In the simulation, MUs received by MR19 are not affected by the interference traffic so largely because of the small number of hops. If the master destination is lightly loaded like

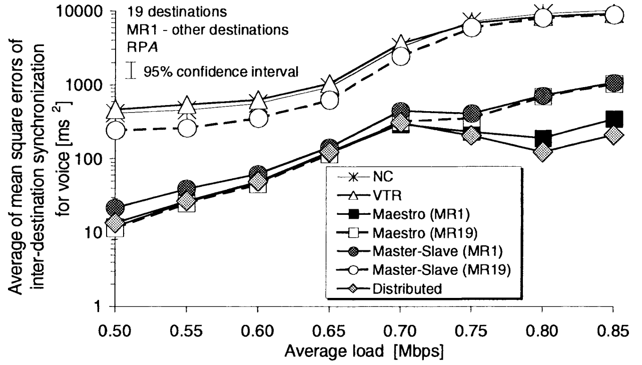

Fig. 4 Average of mean square errors of inter-destination synchronization for voice between MR1 and another destination (RP*A*).

[†] These configurations are set in order to improve the media synchronization quality rather than the real-time property. The optimization of the thresholds and parameters is a future study.

[††] It is not clear how large mean square error of inter-destination synchronization is allowable for applications. In order to answer this question, we need to carry out subjective assessment in a systematic way.

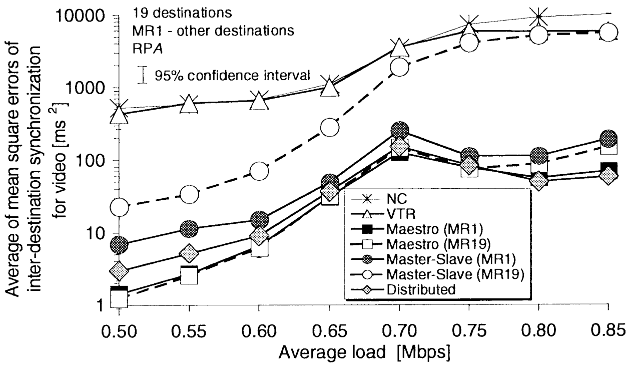

Fig. 5 Average of mean square errors of inter-destination synchronization for video between MR1 and another destination (RP*A*).

MR19, all the destinations set the buffering time to a small value. Therefore, the destinations with heavy loads cannot adjust their output timings to the reference output timing. Thus, the inter-destination synchronization quality with Master-Slave (MR19) degrades.

In these figures, we notice that the inter-destination synchronization error with Maestro is equal to or smaller than that with Master-Slave. Thus, we can say that Maestro provides higher quality of inter-destination synchronization than Master-Slave in the simulation.

We find in Fig. 5 that the averages of mean square errors of inter-destination synchronization with Maestro, Distributed, and Master-Slave (MR1) have peaks around 700 kbps. This is due to two opposing effects. That is, as the average load increases, network delay jitter also increases; this degrades the media synchronization quality. On the other hand, in these schemes, the reference total slide time expands largely when the average load becomes heavy; accordingly, most destinations can absorb network delay jitter effectively. Then, the difference in the output timings among destinations becomes small. Thus, the interaction between the two effects produces the peaks of the inter-destination synchronization errors.

In Fig. 4, we observe that when the average load is heavier than about 750 kbps, the average of mean square errors for voice with Distributed is the smallest among all the schemes, and that with Maestro (MR1) is the second smallest. Furthermore, in Fig. 5, we can confirm that for the average loads heavier than around 800 kbps, Distributed and Maestro (MR1) provide smaller inter-destination synchronization errors for video than the other schemes. This is because the waiting time in the queue of the bottleneck WAN router with these schemes becomes smaller than that with the other schemes. In order to explain the reason, we show the state of the router queue placed at the right edge of the link L_{w2} below; it is the transmit-queue of the output interface corresponding to L_{w2}[†]. It should be noted that L_{w2} is the most heavily loaded link among all the WAN links. In addition, the voice and video MUs which are sent by the rendezvous point to MR1 traverse L_{w2} from the right to the left.

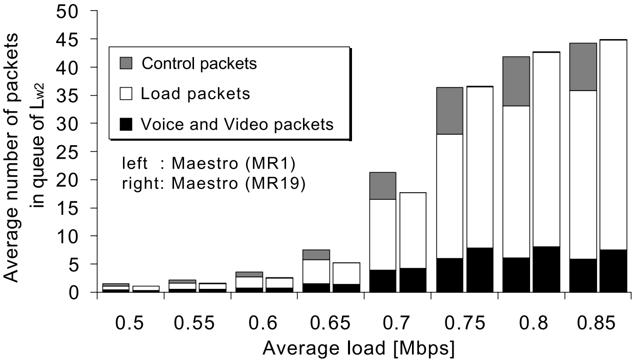

Fig. 6 Average number of packets in the queue of L_{w2} (RP*A*).

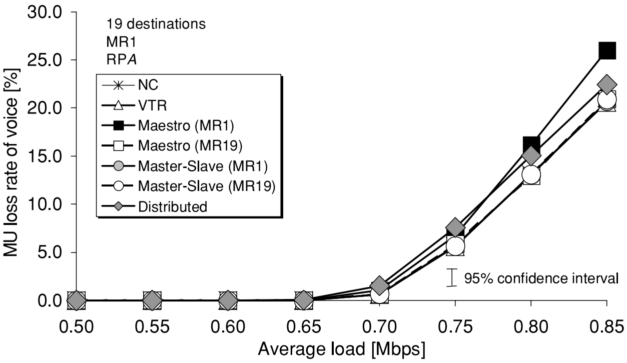

Fig. 7 MU loss rate of voice at MR1 (RP*A*).

4.1.2 State of Router Queue

Figure 6 depicts the average number of packets in the router queue placed at the right edge of L_{w2} for Maestro (MR1) and that for Maestro (MR19). This figure depicts the summation of the average numbers of three packet types: voice and video (i.e., media), load, and control packets. Furthermore, Fig. 7 shows the MU loss rate of voice at MR1 versus the average load. The MU loss rate is the ratio of the number of MUs lost to the total number of MUs generated.

We find in Fig. 6 that for the average loads heavier than 750 kbps, the average queue length for Maestro (MR1) is almost the same as that for Maestro (MR19). This is because many packets drop owing to the limitation of the allowable queue length. In the simulation, the maximum allowable queue length in a router is specified in terms of the number of packets; it does not depend on the total size of packets in the router[††]. Therefore, the average queue length for both schemes is almost the same.

In Fig. 6, we can also confirm that for the average loads heavier than 750 kbps, the control packets occupy about 20% of the average number of packets waiting in the queue

[†] The queue object in ns-2 models the output interface queue in the real router.

[††] This is the specification of the FIFO queue in ns-2. As a matter of fact, many router products employ the number of packets for the queue limitation.

in Maestro (MR1), whereas the queue in Maestro (MR19) has almost no control packet. The reason is as follows. In the configuration of the simulation, each destination has a different load condition. The reference total slide time is set to the largest value among the values from heavily loaded destinations as specified in Sect. 3.2. Meanwhile, lightly loaded destinations can perform the intra-stream and inter-stream synchronization control with much smaller total slide time than the reference one. Hence, the destinations repeatedly send the recommended total slide time in order to decrease the reference total slide time. In Maestro (MR1), MR1 has been selected as the synchronization maestro. Then, MR12 through MR19, which are located on the right edge of the network in Fig. 3, send the control packets to MR1 by way of L_{w2}. Thus, many control packets traverse L_{w2} in Maestro (MR1). On the other hand, in Maestro (MR19), no control packet sent by the destinations to the synchronization maestro traverse L_{w2} from the right to the left. Hence, the control packets are not input into the queue so much in Maestro (MR19).

Note that the maximum allowable queue length is specified in terms of the number of packets. That is, as the number of control packets in a router queue increases, the number of media and load packets decreases in the queue. Thus, the average number of media and load packets in the queue with Maestro (MR1) is smaller than that with Maestro (MR19). This is also confirmed in Fig. 7; it shows that the MU loss rate for Maestro (MR1) is larger than that for Maestro (MR19).

Every control packet is smaller than the other kinds of packets and then stays on a link for shorter duration. Therefore, as the percentage of the control packets in a router queue increases, the waiting time in the queue becomes small. We observed that when the average load is 750 kbps, the average waiting time of media packets in the queue placed at the right edge of L_{w2} for Maestro (MR1) is about 59 ms. In contrast, the waiting time in Maestro (MR19) is around 78 ms.

Additionally, we have investigated the queue state of Distributed. As a result, we found that Distributed has the same tendency as Maestro (MR1). That is, the control packets also affect the voice, video and load transfer in Distributed.

From the above discussion, we can say that Maestro (MR1) and Distributed have smaller average waiting time of packets in the queue than the other schemes. Thereby, these schemes have smaller difference in the MU delay times among destinations and then have better quality of inter-destination synchronization. However, it should be noted that these schemes lose more voice, video and load packets than the other schemes.

4.1.3 Real-Time Property and Intra-Stream Synchronization Quality

In Fig. 8, we present the average MU delay of voice at MR1 versus the average load. The coefficient of variation of output interval for voice at MR1 is shown as a function of the average load in Fig. 9. Furthermore, Figs. 10 and 11 are the results at MR19; they depict the average MU delay of voice and the coefficient of variation of output interval for voice.

We observe in Fig. 8 that when the average load is heavier than about 750 kbps, Distributed and Maestro (MR1) have smaller average MU delays of voice at MR1 than the other schemes. This is because the waiting time in the queue of the bottleneck WAN router with these schemes becomes smaller than that with the other schemes as mentioned earlier.

We find in Fig. 9 that for the average loads heavier than around 800 kbps, the coefficient of variation of output in-

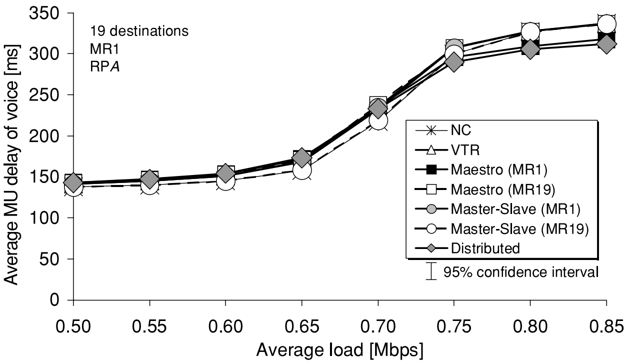

Fig. 8 Average MU delay of voice at MR1 (RP*A*).

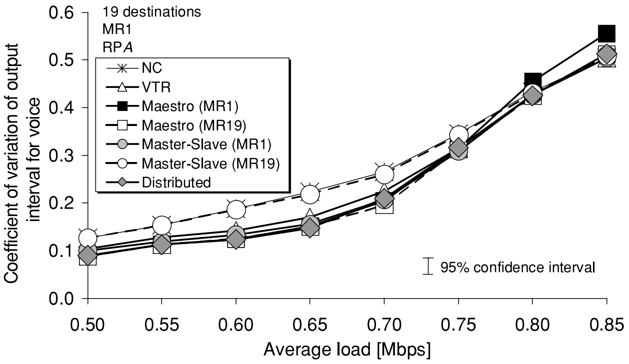

Fig. 9 Coefficient of variation of output interval for voice at MR1 (RP*A*).

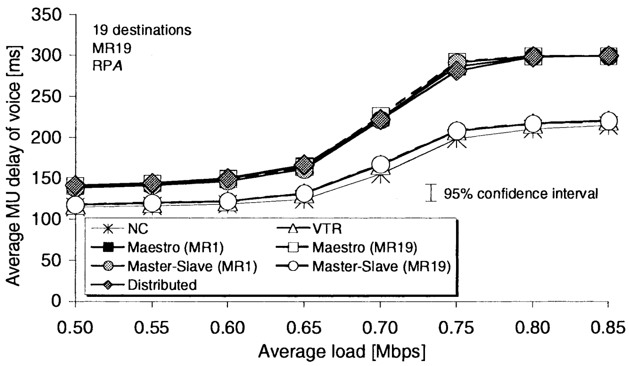

Fig. 10 Average MU delay of voice at MR19 (RP*A*).

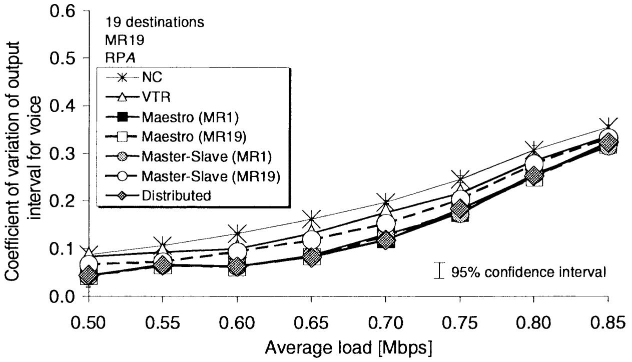

Fig. 11 Coefficient of variation of output interval for voice at MR19 (RP*A*).

terval for voice at MR1 with Maestro (MR1) is the largest among all the schemes. This is because many control packets cause the loss of voice MUs as seen from Figs. 6 and 7. The synchronization maestro scheme does not have the random backoff mechanism for smoothing the traffic due to the control packets[†]. Then, the many control packets degrade the application-level QoS of the media stream.

Figure 10 shows that the average MU delay of voice at MR19 with VTR is almost the same as that with Master-Slave (MR19); these schemes have smaller average MU delays than the other schemes except for NC. This is because MR19 is a lightly loaded destination. In these schemes, MR19 can set the buffering time independently of the other destinations. Thus, the average MU delays of these schemes are small.

In Fig. 11, we see that Maestro and Distributed provide smaller coefficients of variation for voice at MR19 than VTR. This is because the reference total slide time (i.e., the buffering time) in these schemes is set to the largest value among the values from heavily loaded destinations as specified in Sect. 3.2. When the buffering time is set to a large value, destinations can absorb network delay jitter effectively; however, the MU delay at every destination becomes large.

We have investigated the coefficient of variation for video, although we do not show the result. Then, we found that under heavily loaded conditions, all the schemes have approximately the same coefficients of variation for video.

From the above discussion, we see that when RP*A* is selected as the rendezvous point, Distributed provides the highest quality of inter-destination synchronization in heavily loaded networks. On that condition, Maestro has the second highest quality of inter-destination synchronization. On the other hand, in lightly loaded networks, the inter-destination synchronization schemes except for Master-Slave (MR19) can achieve high and almost the same quality of inter-destination synchronization. In Master-Slave, we should select one of the heavily loaded destinations as the master destination.

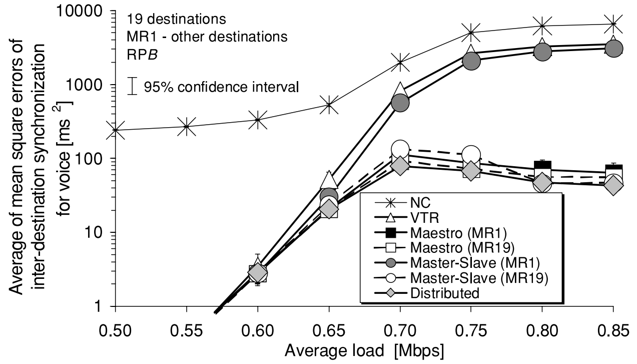

Fig. 12 Average of mean square errors of inter-destination synchronization for voice between MR1 and another destination (RP*B*).

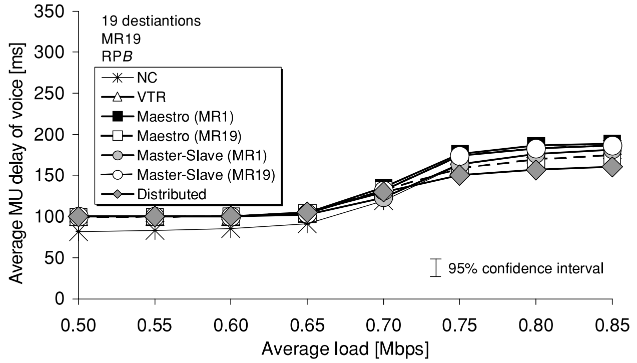

Fig. 13 Average MU delay of voice at MR19 (RP*B*).

4.2 Case of RP*B*

Figure 12 shows the average of mean square errors of inter-destination synchronization between MR1 and another destination for voice. In Fig. 13, we present the average MU delay of voice at MR19 versus the average load. The coefficient of variation of output interval for voice at MR19 is shown as a function of the average load in Fig. 14.

In Fig. 12, we notice that for all the average loads here, Maestro, Distributed and Master-Slave (MR19) have approximately the same averages of mean square errors of inter-destination synchronization for voice. Furthermore, the inter-destination synchronization errors with these schemes are smaller than those with VTR and NC. Thus, we can say that these schemes can improve the inter-destination synchronization quality.

Comparing Figs. 4 and 12, we find that the inter-destination synchronization errors with Maestro and Distributed when we select RP*B* are better than those when we select RP*A*. This is due to the difference in the number of WAN routers between the source and the rendezvous point. When the number of WAN routers is large, the voice and

[†] We assumed that the mechanism is not necessary for the synchronization maestro scheme, because the number of control packets with the scheme do not increase so largely as that with the distributed control scheme.

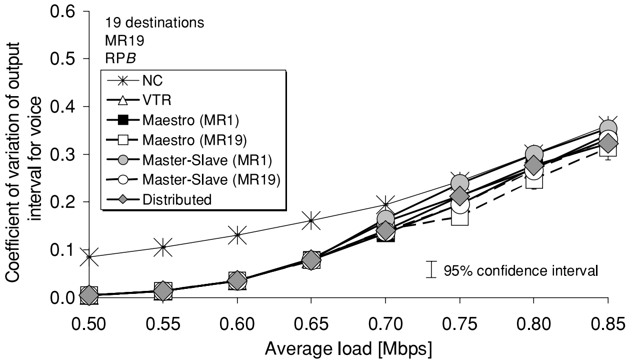

Fig. 14 Coefficient of variation of output interval for voice at MR19 (RP*B*).

video MUs sent by the source to the rendezvous point are affected by the interference traffic largely in the simulation.

Furthermore, we observe in Fig. 12 that when the average load is heavier than about 700 kbps, the inter-destination synchronization error with Master-Slave (MR1) is much larger than that with Master-Slave (MR19); the relationship between the two schemes is the opposite to that in Fig. 4. This is due to the difference in the number of hops between the rendezvous point and the master destination. That is, when we select a far destination from the rendezvous point as the master destination, Master-Slave achieves high quality of inter-destination synchronization.

In Fig. 13, we find that when the average load is heavier than around 750 kbps, Distributed has the smallest average MU delay of voice at MR19 among all the schemes. This is due to a huge number of control packets; the reason is the same as that in Fig. 8.

On the other hand, in Fig. 13, we observe that when the average load is lighter than about 650 kbps, NC has the smallest average MU delay of voice at MR19. This is because NC carries out no media synchronization control.

We can confirm in Fig. 14 that for the average loads heavier than around 750 kbps, the coefficient of variation of output interval for voice at MR19 with Maestro (MR19) is the smallest. This is due to the transmission timing of control packets. By tracing the packets in the simulation, we found that the control packets did not affect the voice MU transmission so largely, although they caused the loss of many interference data packets.

We have also investigated the coefficient of variation for voice at MR1. However, we saw that the coefficients with all the schemes are small enough. This is because MUs transmitted by RP*B* to MR1 went through no WAN router under the routing algorithm in the simulation. Then, MUs received by MR1 are not affected by the interference traffic. Thus, we do not show the results.

From these results, we can say that Distributed and Maestro provides high quality of inter-destination synchronization when we select RP*B* as the rendezvous point. Furthermore, the quality with these schemes when we select RP*B* is better than that when we select RP*A*.

On the other hand, in Master-Slave, we obtain the same conclusion as that in the previous subsection.

5. Conclusions

In this paper, we compared the application-level QoS of the three inter-destination synchronization schemes: the master-slave destination scheme, the synchronization maestro scheme, and the distributed control scheme. The comparison was performed in a Tiers model network. We then noticed that the inter-destination synchronization quality with the distributed control scheme is the highest among all the schemes in heavily loaded networks. However, in lightly loaded networks, the synchronization maestro scheme has high and almost the same quality of inter-destination synchronization as that with the distributed control scheme. Furthermore, the master-slave destination scheme achieves high quality of inter-destination synchronization as much as that with the other two schemes when we deploy the master destination far from the rendezvous point.

In addition, we investigated the effect of the location of the rendezvous point. Then, we found that as the number of WAN routers between the source and the rendezvous point decreases, the quality of media synchronization with the synchronization maestro and distributed control schemes becomes higher.

Furthermore, we saw that when we select a heavily loaded destination (i.e., a destination far from the rendezvous point) as the synchronization maestro in the synchronization maestro scheme, the quality of intra-stream synchronization at the destination is affected by the control packets largely.

On the other hand, we found that when we select a lightly loaded destination (i.e., a destination close to the rendezvous point) as the master destination, the master-slave destination scheme has lower quality of intra-stream synchronization than the synchronization maestro scheme and the distributed control scheme.

In this paper, we presented a method of the application-level QoS assessment of the inter-destination synchronization schemes by means of simulation with the Tiers model network. We can assess the QoS for a variety of network configurations by the same method.

As the next step of our research, we plan to assess the user-level QoS of the inter-destination synchronization schemes. We also need to investigate the relationship between the user-level QoS and the application-level QoS. In addition, it is important to examine the suitable locations of key nodes such as the master destination, maestro and rendezvous point in conjunction with network traffic monitoring schemes.

Acknowledgment

The authors are grateful to Prof. Yutaka Ishibashi for his valuable discussions. This work was supported by the Grant-In-Aid for Young Scientist (B) of Japan Society for the Promotion of Science under Grant 15760262 and the Grant-In-Aid for Scientific Research (B) under Grant 14350200.

References

[1] S. Deering, "Host extensions for IP multicasting," RFC1112, Aug. 1989.

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

[3] S. Tasaka and Y. Ishibashi, "Mutually compensatory property of multimedia QoS," Conf. Rec. IEEE ICC 2002, pp.1105–1111, April/May 2002.

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

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

[6] A. Benslimane, "A multicast synchronization protocol for multiple distributed multimedia streams," Conf. Rec. 7th International Conference on High Performance Computing (HiPC 2000), pp.469–476, Dec. 2000.

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

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

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

[10] S. Tasaka, Y. Ishibashi, and M. Hayashi, "Inter-destination synchronization quality in an integrated wired and wireless network with handover," Conf. Rec. IEEE GLOBECOM 2002, CQRS-05-1, Nov. 2002.

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

[12] M.B. Doar, "A better model for generating test networks," Conf. Rec. IEEE GLOBECOM'96, pp.86–93, Nov. 1996.

[13] K.L. Calvert, M.B. Doar, and E.W. Zegura, "Modeling Internet topology," IEEE Commun. Mag., vol.35, no.6, pp.160–163, June 1997.

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

[15] K. Obraczka, "Multicast transport protocols: A survey and taxonomy," IEEE Commun. Mag., vol.36, no.1, pp.94–102, Jan. 1998.

[16] G. Carle and E.W. Biersack, "Survey of error recovery techniques for IP-based audio-visual multicast applications," IEEE Netw., vol.11, no.6, pp.24–36, Nov./Dec. 1997.

[17] S. Floyd, V. Jacobson, C. Liu, S. McCanne, and L. Zhang, "A reliable multicast framework for light-weight sessions and application level framing," IEEE/ACM Trans. Netw., vol.5, no.6, pp.784–803, Dec. 1997.

[18] The network simulator –ns–2 –, (http://www.isi.edu/nsnam/ns/).

[19] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharm, and L. Wei, "Protocol independent multicast—Sparse mode (PIM-SM): Protocol specification," RFC 2236, June 1998.

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

[21] T. Nunome and S. Tasaka, "An application-level QoS comparison of inter-destination synchronization schemes in middle-scale multicast communications," IEICE Technical Report, CQ2002-86, MoMuc2002-34, Sept. 2002.

Toshiro Nunome received the B.S. and M.S. degrees from Nagoya Institute of Technology, Nagoya, Japan, in 1998 and 2000, respectively. From 2000 to 2001, he was with Pioneer Corp. In 2002, he joined Nagoya Institute of Technology as a Research Associate. His research interests include media synchronization and multicast communications.

Shuji Tasaka received the B.S. degree in electrical engineering from Nagoya Institute of Technology, Nagoya, Japan, in 1971, and the M.S. and Ph.D. degrees in electronic engineering from the University of Tokyo, Tokyo, Japan, in 1973 and 1976, respectively. Since April 1976, he has been with Nagoya Institute of Technology, where he is now a Professor in the Department of Computer Science and Engineering. In the 1984–1985 academic year, he was a Visiting Scholar in the Department of Electrical Engineering at the University of California, Los Angeles. His current research interests include wireless networks, multimedia QoS and multimedia communication protocols. He is the author of a book entitled *Performance Analysis of Multiple Access Protocols* (Cambridge, MA: The MIT Press, 1986). Dr. Tasaka is a member of the IEEE, ACM and Information Processing Society of Japan.