IEICE TRANS. COMMUN., VOL.E85–B, NO.1 JANUARY 2002

PAPER *Special Issue on Internet Technology II*

Media Synchronization Quality of Packet Scheduling Algorithms**

Kenji ITO†*, **Shuji TASAKA**†, ***and*** **Yutaka ISHIBASHI**†, ***Regular Members***

SUMMARY This paper studies effect of packet scheduling algorithms at routers on media synchronization quality in live audio and video transmission by experiment. In the experiment, we deal with four packet scheduling algorithms: First-In First-Out, Priority Queueing, Class-Based Queueing and Weighted Fair Queueing. We assess the synchronization quality of both intra-stream and inter-stream with and without media synchronization control. The paper clarifies the features of each algorithm from a media synchronization point of view. A comparison of the experimental results shows that Weighted Fair Queueing is the most efficient packet scheduling algorithm for continuous media among the four.

key words: *packet scheduling algorithm, continuous media, media synchronization, experiment*

1. Introduction

One of the most rapidly increasing demands in the Internet is the transmission of continuous media such as audio and video with high quality. Currently, however, the main service provided by the Internet is the best-effort one, which does not always guarantee the high quality.

In research communities of the Internet, a variety of efforts have been made to give solutions to the problem. Among them, the incorporation of *Quality-of-Service* (*QoS*) guarantee mechanisms into the Internet is the approach that attracts most attention of the communities. It includes the *integrated services* (*IntServ*) [1], [2] and the *differentiated services* (*DiffServ*) [3], both of which are being investigated by the Internet Engineering Task Force (IETF) in recent years.

IntServ is based on end-to-end resource reservation; it adopts the *Resource Reservation Protocol* (*RSVP*) signaling [4] to provide *per-flow* QoS guarantee. Therefore, it requires every router to have the reservation function and keep all flow-related state information. As a result, the amount of the state information increases as the number of flows increases; this leads to the scalability problem in IntServ.

DiffServ, on the other hand, does not take the per-flow approach but utilizes a concept of *per-hop behavior* (*PHB*) [3]. Each router treats packets with the same *DiffServ codepoint* [5] in the same way independently of the other routers, although those packets may belong to different flows. Thus, the amount of information each router has to keep in DiffServ becomes small, and all routers do not need to have the same functions; this can realize scalability in DiffServ. For this reason, DiffServ is considered more promising than IntServ as the future Internet service architecture, though the former cannot necessarily guarantee end-to-end QoS.

In both IntServ and DiffServ, the packet scheduler at each router is an essential ingredient; it stores incoming packets into specific queues according to some classification criteria and then schedules the queues with an algorithm. Although the scheduler works at individual routers, it eventually affects the end-to-end QoS. The algorithms proposed for this purpose includes *Weighted Fair Queueing* [6], [7] and *Deficit Round-Robin* [8], for instance.

Many papers on performance evaluation of the packet scheduling algorithms have been published [9]–[13]. However, the great majority of them deal with packet-level performance such as packet throughput and packet delay; we find no performance studies on continuous media in higher layers, including audio/video quality. The main difference between the packet-level performance and the continuous-media-level one exists in the latter's necessity for the consideration on the temporal structure of the media.

The preservation of the temporal structure of continuous media is referred to as *media synchronization* [14]. It can be classified into *intra-stream synchronization* and *inter-stream synchronization*. The former refers to the temporal constraints within a single stream, e.g., the preservation of time intervals between two successive video frames of a video stream. The latter is synchronization among plural media streams, e.g., between a voice stream and a video stream. In multimedia communications, both kinds of synchronization quality are important QoS measures in higher layers. However, to the best of the authors' knowledge, no publications whose subject is QoS of the packet scheduling algorithms are related to media synchronization quality; in particular, we cannot find any publication handling quantitative assessment of the media synchronization quality.

Manuscript received May 10, 2001.
Manuscript revised August 15, 2001.

†The authors are with the Department of Electrical and Computer Engineering, Nagoya Institute of Technology, Nagoya-shi, 466-8555 Japan.
*Presently, with Toyota Central Research and Development Laboratories, Inc., Aichi-ken, 480–1192 Japan.

**This paper was presented in part at IEEE ICC2001, Helsinki, Finland, June 2001.

This paper considers the transmission of a live audio stream and the corresponding video stream over the Internet and investigates effect of packet scheduling algorithms on both intra-stream and inter-stream synchronization quality. We treat four algorithms: *First-In First-Out* (*FIFO*), *Priority Queueing* (*PQ*), *Class-Based Queueing* (*CBQ*) [15] and *Weighted Fair Queueing* (*WFQ*). FIFO is widely used in the current Internet because of its simplicity. PQ is priority-based scheduling, which is useful for transfer of real-time-traffic but produces an unfairness problem among traffic classes. CBQ guarantees allocated bandwidth of classes with the same priority under a static condition. WFQ is one of the flow-based scheduling algorithms that dynamically allocate bandwidth to each flow on a fair basis. By experiment, we assess the media synchronization quality achieved by the four algorithms with and without media synchronization control. The *Virtual-Time Rendering* (*VTR*) [16], [17] media synchronization algorithm is utilized in the experiment. Referring to the experimental results, we try to clarify the features of each algorithm from a media synchronization quality point of view.

The rest of the paper is organized as follows. Section 2 describes principles of the four packet scheduling algorithms. Section 3 gives the basic concept of the VTR media synchronization algorithm. In Sect. 4, we illustrate the configuration of the experimental system and a method of the experiment, which includes implementation details of the packet scheduling algorithms and performance measures. Section 5 presents measurement results and discusses the features of the packet scheduling algorithms.

2. Packet Scheduling Algorithms

In this section, we introduce four scheduling algorithms (FIFO, PQ, CBQ and WFQ) used in our experiment.

2.1 First-In First-Out (FIFO)

In the current Internet, FIFO is the most common scheduling algorithm. It stores packets arriving at a router into a buffer and forwards them in the order of arrival without isolation between flows. Consequently, if the router has no buffer space to store newly arriving packets, then they are discarded. Thus, packets belonging to a specific flow can be lost successively even if its buffer occupancy and arrival rate are low at that time; this produces an unfairness problem among flows, and the requested bandwidth cannot be guaranteed.

2.2 Priority Queueing (PQ)

PQ sets a certain number of priority classes according to some parameter of a packet, e.g., its IP address, protocol type or incoming interface, and each router prepares a buffer per priority class. The algorithm puts each incoming packet into a buffer corresponding to its priority class. It first serves packets in the highest priority buffer until the buffer becomes empty and then moves to the next highest priority buffer. This process goes down to lower priority classes in the same way. The priority mechanism among the classes can be used in order to give preferential treatment to a special class, e.g., a class for real-time traffic. However, this algorithm also has an unfairness problem.

2.3 Class-Based Queueing (CBQ)

Class-Based Queueing [15] classifies each arriving packet according to its attribute such as the application and protocol type and stores it into the buffer corresponding to the class. For each class, the allocated bandwidth and priority are predetermined. Within classes of the same priority, this scheduling policy serves the buffers in a fashion of byte-by-byte Weighted Round-Robin (WRR) [18], with weights proportional to the bandwidth allocations of the classes. The weight determines the number of bytes that the class is allowed to send in a round. CBQ is statically configured and does not automatically adapt to changing network conditions.

2.4 Weighted Fair Queueing (WFQ)

Weighted Fair Queueing [6], [7] is a flow-based algorithm, and it dynamically allocates available bandwidth to each flow according to its weight. The weight of a certain flow is determined on the basis of the request rate of the flow. If the sum of all the request rates is smaller than the output link capacity, then each request rate is assured; otherwise, the allocated rate becomes smaller than the request one. In this case, the fraction of the output link capacity available to a flow is the ratio of the request rate of the flow to the sum of the request rates of the *backlogged* flows at the instant. Thus, WFQ automatically adapts to changing network traffic conditions.

3. Basic Concept of the VTR Algorithm

The temporal structure of continuous media can be disturbed by various causes; in QoS non-guaranteed networks like the Internet, network delay jitter is a dominant one. In this case, we can achieve media synchronization by absorbing the jitters at the destination. This is carried out by buffering the information unit such as a video frame or voice packet, which is referred to as an *MU* (*Media Unit*), 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 is a media synchronization algorithm proposed by the authors [16], [17], [19]. It assumes no exact knowledge of the network delay jitter and adaptively changes the buffering time according to the amount of delay jitters of MUs received at the destination and MU loss. Initially, the buffering time is set to a rough estimate of the maximum delay jitter, which is denoted by $J_{\max}$; after the first MU is received, it 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. On the other hand, live media need both forward and 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.

In this paper, we transfer live audio and video over a QoS non-guaranteed network, where MU loss occurs. A video frame is defined as a video MU, and an audio packet consisting of a constant number of audio samples as an audio MU. 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. 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. We utilize the VTR algorithm enhanced for MU loss; the reader is referred to [19] for its details.

4. Experimental System

In our experiment, we transfer live audio and JPEG video streams to investigate effects of the packet scheduling algorithms at a router on media synchronization quality. We measure the performance of two schemes: a scheme with the enhanced VTR algorithm [19] and without media synchronization control. The former is denoted here simply by *VTR*, and the latter is referred to as *NC*. We compare their media synchronization quality.

In this section, we first show the configuration of the experimental system for quality assessment. We then explain implementation details of the packet scheduling algorithms at the router and also illustrate a method for experiment. Finally, we define measures for quality assessment.

4.1 System Configuration

Figure 1 illustrates the configuration of the experimental system. It comprises six workstations (WS1 through WS6), five 10Base-T Ethernet-hubs and three routers (Routers 1, 2 and 3).

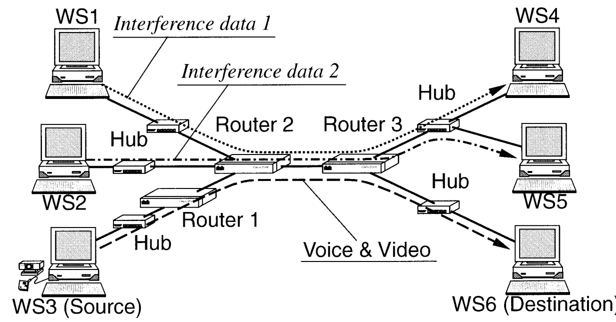

Fig. 1 Configuration of the experimental system.

Table 1 Specifications of voice and video.

item	voice	video
coding scheme	ITU-T G.711 μ-law	JPEG
image size [pixels]	—	320×240
average MU size [bytes]	400	3150
original average MU rate [MU/s]	20.0	
original average inter-MU time [ms]	50.0	
original average bit rate [kbps]	64.0	504.0
measurement time [s]	120.0	

WS3 is a SUN Ultra 2 workstation (200 MHz) which has a main memory of 128 Mbytes. The workstations except for WS3 are SUN Ultra 1 workstations (143 MHz) each with a 64 Mbyte main memory. WS3 and WS6 each have JPEG video boards (PowerVideo from Parallax Graphics, Inc.). All the workstations run Solaris 2.5.1 with OpenWindow 3.5.1.

Routers 1 through 3 each are Cisco System's 2514. Router 2 runs Cisco IOS 11.3, and the other two routers run Cisco IOS 11.2. Router 1 and Router 2, as well as Router 2 and Router 3, are connected to each other by a V.35 serial line. The transmission rate of the serial lines is set to 2 Mbps in our experiment.

4.2 Method of the Experiment

Our experiment focuses on lip synchronization, and we use a lady's voice and her head view video as the audio stream and video stream, respectively. Table 1 shows the specifications of the voice and video.

In the experiment, WS3 and WS6 are used as the source of the voice and video streams and its destination, respectively. WS3 inputs the voice and video streams from a video cassette recorder in order to generate the media traffic of the same amount in each experimental run. Using RTP/UDP as the transport protocol, WS3 transfers the voice and video as two separate transport streams to WS6.

WS1, WS2, WS4 and WS5 are used to generate traffic flows of interference. WS1 sends fixed-size data messages of 1472 bytes each to WS4 under the UDP protocol at exponentially distributed intervals; these

Table 2 The scheduling algorithm types for performance measurement of PQ.

type	priority of incoming interface		
	voice & video	interference data 1	interference data 2
PQ1	medium	normal	normal
PQ2	normal	medium	normal
FIFO	—	—	—

Table 3 The scheduling algorithm types for performance measurement of CQ.

type	the byte count value [bytes]		
	voice & video	interference data 1	interference data 2
CQ1	3000	4500	1500
CQ2	1500	1500	1500
CQ3	1500	4500	1500
FIFO	—	—	—

data messages are referred to as *interference data 1*, and WS2 also transfers data messages to WS5 in the same way as WS1; we refer to them as *interference data 2*. The amount of the interference traffic is adjusted by changing the average of the interval. In the experiment, we change the amount of interference data 1 from 0.5 Mbps to 1.0 Mbps and keep that of interference data 2 at 0.5 Mbps. In addition, we employ the same values of the thresholds and the parameters, which include $J_{\max}$ and Δ_{al}, for the VTR algorithm as those in [19]; for example, $J_{\max} = 100$ ms and $\Delta_{\mathrm{al}} = 300$ ms.

In the experiment, Router 2 adopts one of the four scheduling algorithms, i.e., either FIFO, PQ, CBQ or WFQ, and we investigate effects of those algorithms on media synchronization quality. Routers 1 and 3 use only FIFO. We next explain how the packet scheduling algorithms except for FIFO are implemented.

4.2.1 Implementation of PQ

In the router of our experimental system, four priorities are defined in the following descending order: *high*, *medium*, *normal* and *low* [20]. In PQ, each arriving packet is placed in one of four queues based on its assigned priority. The packets that are not assigned priority fall into the *normal* queue. We set one of the four priorities to each incoming interface in Router 2.

Table 2 shows scheduling algorithm types to be compared for performance measurement of PQ. PQ1 gives the voice and video streams preferential treatment over interference data 1 and 2. Oppositely, in PQ2, interference data 1 is given priority over the voice and video.

4.2.2 Implementation of CBQ

In Cisco System's routers, CBQ is implemented with a name of *Custom Queueing* (*CQ*) [20]. The difference between CBQ and CQ is that the latter has no priority among the classes. Therefore, CQ schedules all classes in a fashion of byte-by-byte WRR.

In the experiment with CQ, we associate three classes with three incoming interfaces of Router 2, which correspond to the voice-video flow, interference data 1 and interference data 2. So, CQ stores packets arriving at the same incoming interface into the same buffer. We determine the number of bytes served from each buffer in a cycle, which is referred to as the *byte count value*.

Table 3 shows scheduling algorithm types to be compared for performance measurement of CQ. We set the byte count value of each type to multiples of 1500 bytes. The reason is as follows. A Cisco's router sends packets from a buffer until the byte count value is exceeded. Once it occurs, the packet that is being sent at that time will be completely sent. We send the interference data with 1500 byte-size including IP and UDP headers. Therefore, for simplicity of implementation, we set the byte count values for interference data 1 and 2 to multiples of 1500 bytes. For the voice and video transmission, we transfer voice packets of 440 bytes and video packets of at most 1052 bytes†, including IP, UDP and RTP headers. Since the sum of the voice packet size and the maximum video packet size is about 1500 bytes, we also set the byte count value to multiples of 1500 bytes.

In our experimental system, the output link capacity of Router 2 is set to 2 Mbps. Therefore, in CQ1, a half of the total bandwidth (about 1 Mbps) can be allocated to interference data 1, one third (about 667 kbps) to voice and video, one sixth (about 334 kbps) to interference data 2. The allocated bandwidth to the voice and video transmission in our experiment is enough because the total bit rate of the voice and video is about 568.0 kbps. Note that the allocated bandwidth to the voice and video can change in each cycle because the number of bytes of the voice and video actually served in a cycle is not always equal to the byte count value. In CQ2, the allocated bandwidth of every interface is one third (about 667 kbps). This type also corresponds to the case in which the bandwidth for the voice and video transmission is sufficient. In CQ3, three fifths of the output link capacity (about 1.2 Mbps) is allocated to interference data 1, and the allocated bandwidth for the voice and video transmission is one fifth (about 400 kbps), which is equal to that for interference data 2. In this type, the allocated bandwidth to the voice and video transmission becomes insufficient as the amount of interference data 1 increases.

†Note that the size of a video MU varies from MU to MU owing to the variable bit rate coding of JPEG, while that of a voice MU is constant.

4.2.3 Implementation of WFQ

WFQ stores packets into a buffer per flow (i.e., per stream) in a Cisco's router [20]. The bandwidth allocated to a flow in WFQ, which is referred to as the *service rate*, is determined by the weight of the flow in the router. WFQ sets the weight of a flow using the IP precedence field value (three bits) included in the *type of service* (*ToS*) field of the IPv4 packet header. The IP precedence field takes a value between 0 and 7, where 0 is the lowest priority and 7 is the highest. As the precedence value of a flow increases, this algorithm increases the service rate of the flow and allocates more bandwidth to that flow.

We now explain the decision policy of the service rate using the IP precedence values [20]. The fraction of the output link capacity available to a flow is given by the ratio of *precedence value + 1* of the flow to the sum of all flows' ones.

Table 4 shows scheduling algorithm types to be compared for performance measurement of WFQ. Let us calculate the service rate in each type. WFQ1 has one stream with precedence 0, two streams with precedence 1 and one stream with precedence 2. Therefore, the total of the "*precedence value + 1*" values is eight. The voice and video each can use two eighths of the output link capacity (about 500 kbps), and three eighths (about 750 kbps) and one eighth (about 250 kbps) are allocated to interference data 1 and 2, respectively. Note that the actual service rate of each flow is time-varying.

Similarly, we can calculate the service rate of each flow in WFQ2 and WFQ3. WFQ2 allocates the available bandwidth equally to the voice, video, interference data 1 and 2; that is, the service rate of each flow is one fourth of the output bandwidth (about 500 kbps). On the other hand, in WFQ3, the service rates of the voice, video and interference data 2 each are one sixth (about 334 kbps), and that of interference data 1 is three sixths (about 1 Mbps). It is clear that the service rate of the video becomes insufficient as the amount of interference data 1 increases in this type.

In the experiment, we set the IP precedence field value of each flow at the source workstations, i.e., WS1, WS2 and WS3.

Table 4 The scheduling algorithm types for performance measurement of WFQ.

type	IP precedence field value			
	voice	video	interference data 1	interference data 2
WFQ1	1	1	2	0
WFQ2	0	0	0	0
WFQ3	0	0	2	0
FIFO	—	—	—	—

4.3 Performance Measures

The measures for quantitative assessment of media synchronization quality are necessary. We employ measures the authors have introduced and used in their previous studies on media synchronization [17], [19].

For the quality assessment of intra-stream synchronization for voice or video, we first evaluate the *coefficient of variation of output interval*, which represents the smoothness of output of a stream. In addition, we use the *average MU rate*, which is defined as the average number of MUs output in a second at the destination.

For the inter-stream synchronization quality, we calculate the *mean square error*, which is defined as the average square of the difference between the output time of each slave MU and its *derived output time*. The derived output time of each slave MU is defined as the output time of the corresponding master MU plus the difference between the timestamps of the two MUs [16].

Also, the *average MU delay* is a key measure for live media; it is the average time in seconds from the moment an MU is generated until the instant the MU is output; that is, it represents the real-time property.

Finally, in order to measure the performance of interference data, we use the *throughput* of interference data. The throughput of interference data 1 and that of interference data 2 are measured at WS4 and WS5, respectively.

5. Experimental Results

We examine detrimental effects of the interference data load on the quality of VTR and NC using the four scheduling algorithms.

5.1 Quality Measurement of PQ

We first assess the intra-stream synchronization quality. Figures 2 and 3 show the coefficient of variation of output interval for voice and that for video, respectively, as a function of the data load (interference data 1). In these figures, in VTR, we observe that for both

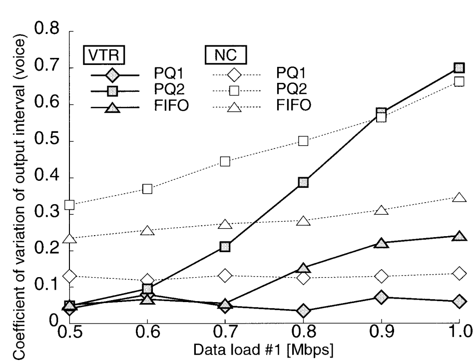

Fig. 2 Coefficient of variation of output interval for voice (PQ).

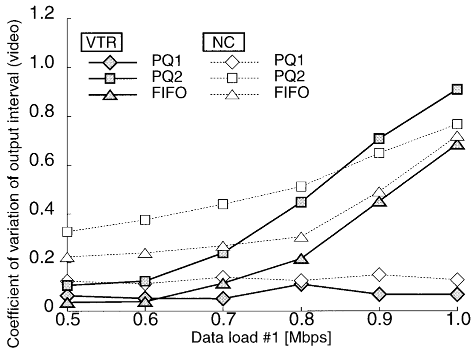

Fig. 3 Coefficient of variation of output interval for video (PQ).

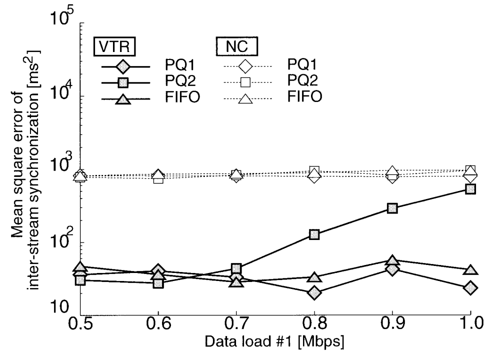

Fig. 5 Mean square error of inter-stream synchronization (PQ).

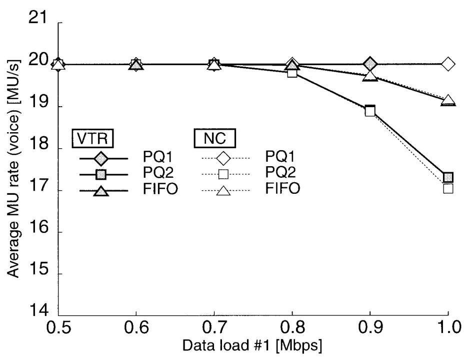

Fig. 4 Average MU rate for voice (PQ).

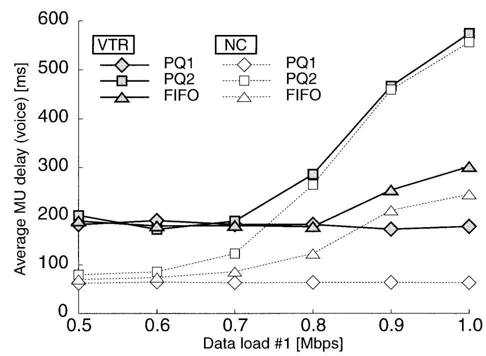

Fig. 6 Average MU delay for voice (PQ).

voice and video PQ1 provides the minimum coefficient of variation for almost all the data loads here, while PQ2 gives the largest. This is because voice and video packets are in the lower priority queue and therefore they are served after interference data 1. The relationship among the three types of NC also shows the same tendency as that of VTR.

In Fig. 4, we display the average MU rate for voice. We notice that PQ1 achieves the best performance for voice. On the other hand, we find that PQ2 provides the worst performance for voice, when the data load exceeds around 0.8 Mbps. The reason is that voice and video of PQ2 are scheduled after interference data 1 and then packet loss increases. The result of the average MU rate for video revealed the same tendency as that for voice; so, we did not show it here.

In subjective assessment of the smoothness, we hardly perceived discontinuity of voice and jerkiness of video in VTR and NC with PQ1, for all the data loads here. However, the media qualities in the other types are damaged when the data load exceeds about 0.8 Mbps.

Figure 5 plots the mean square error of inter-stream synchronization versus the data load. We can confirm by this figure that VTR provides smaller mean square error of inter-stream synchronization than NC for all the types. However, even NC has the values of at most about 1000 ms^2, which is much smaller than a threshold of high inter-stream synchronization quality, that is, 6400 ms^2 ($= 80^2$ ms^2) [21]. This is a characteristic of live media [22]. In fact, all the types attain high quality of inter-stream synchronization, and we hardly noticed the difference among the types in the subjective assessment.

We next examine the average MU delay of voice, and Fig. 6 displays it. The average MU delay of voice in PQ1 hardly change even if the data load increases. However, when the priorities of voice and video are lower than that of interference data 1, i.e., in PQ2, the average MU delay increases. We also notice that the average MU delays of all the types except PQ2 are smaller than the maximum allowable delay Δ_{al}, i.e., 300 ms. In addition, we observed that the average MU delay for video was almost the same as that for voice.

5.2 Quality Measurement of CQ

Figures 7 and 8 show the coefficient of variation of output interval for voice and that for video, respectively, as a function of the data load (interference data 1). Figure 9 depicts the average MU rate of video versus the data load. We measured the average MU rate of voice. As a result, we saw that the average MU rate of voice was very similar to that of video. Figures 10, 11, and 12 plot the mean square error of inter-stream synchronization, the average MU delay of voice and that of video,

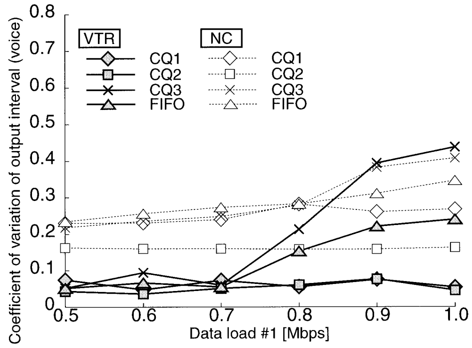

Fig. 7 Coefficient of variation of output interval for voice (CQ).

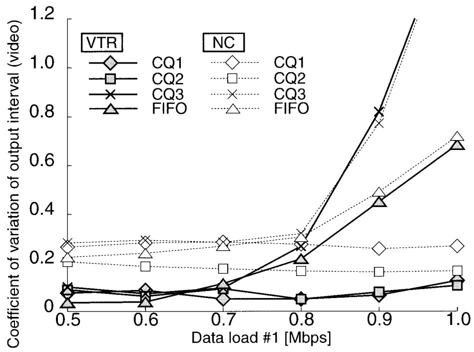

Fig. 8 Coefficient of variation of output interval for video (CQ).

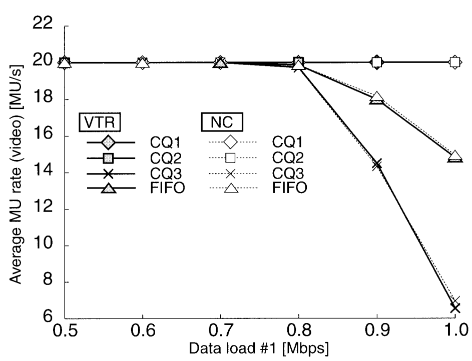

Fig. 9 Average MU rate for video (CQ).

respectively, versus the data load. Furthermore, Fig. 13 displays the throughput of the interference data.

In Figs. 7 and 8, we see that for all the data loads here, the coefficient of variation of output interval for voice and that for video in both CQ1 and CQ2 remain almost constant in VTR and NC. The reason is that the bandwidth for the voice and video transmission is guaranteed in both CQ1 and CQ2 as mentioned earlier. Also, in the case of NC, the coefficient of variation of output interval for voice and that for video in CQ1 are larger than those in CQ2, while in VTR CQ1 is comparable to CQ2. The reason is as follows. The network delay jitters of voice and video in CQ1 are larger than those in CQ2 because the allocated bandwidth of in-

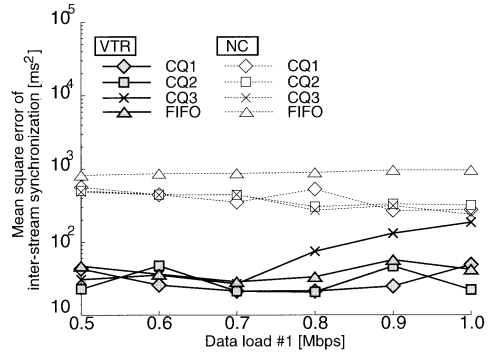

Fig. 10 Mean square error of inter-stream synchronization (CQ).

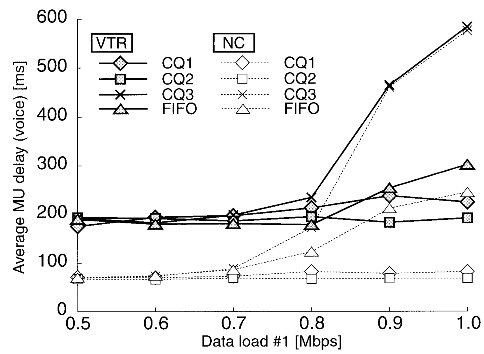

Fig. 11 Average MU delay for voice (CQ).

terference data 1 in CQ1 is larger than that in CQ2. However, VTR can absorb the network delay jitters.

In CQ3, when the data load becomes heavier than 0.8 Mbps, the coefficient of variation of output interval for voice and that for video increase, because the allocated bandwidth for voice and video becomes insufficient. Therefore, the number of lost packets increases, and the average MU rate decreases. We can also find that in Fig. 9.

We made the subjective assessment of the smoothness. The output quality of voice and that for video in CQ2 and in CQ1 with VTR were excellent. However, when the data load exceeds around 0.8 Mbps, the voice and video qualities in CQ3 were damaged.

Also, we find from Fig. 10 that the result for the mean square error of inter-stream synchronization is similar to that of PQ (see Fig. 5). Therefore, we can say that all the types achieve high quality of inter-stream synchronization.

We can confirm in Figs. 11 and 12 that the average MU delays of the types in which the allocated bandwidth to the voice and video transmission is guaranteed, i.e., CQ1 and CQ2, and that of FIFO for all the data loads here are smaller than 300 ms.

In Fig. 13, we observe that when the data loads are heavier than 0.8 Mbps, the throughput of interference data 1 in CQ2 keeps about the same value and that value is larger than the allocated bandwidth (about

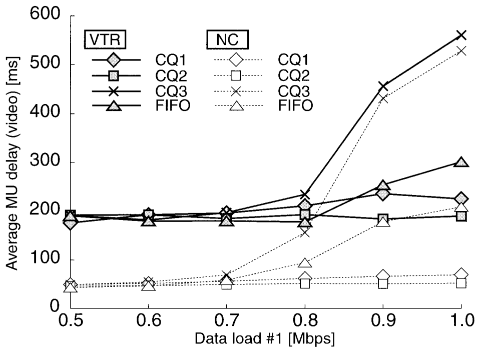

Fig. 12 Average MU delay for video (CQ).

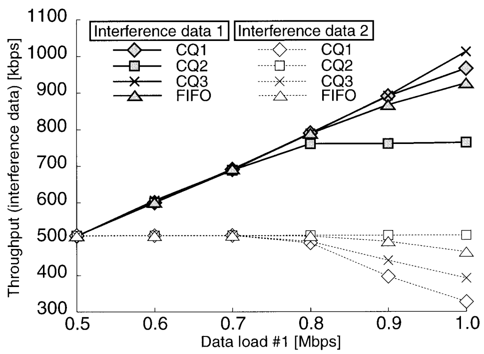

Fig. 13 Throughput of interference data (CQ).

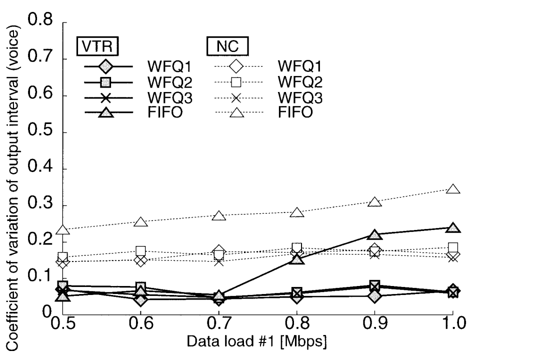

Fig. 14 Coefficient of variation of output interval for voice (WFQ).

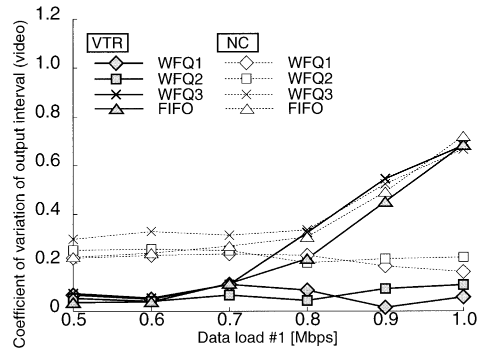

Fig. 15 Coefficient of variation of output interval for video (WFQ).

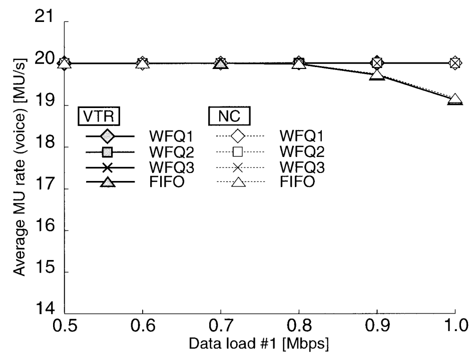

Fig. 16 Average MU rate for voice (WFQ).

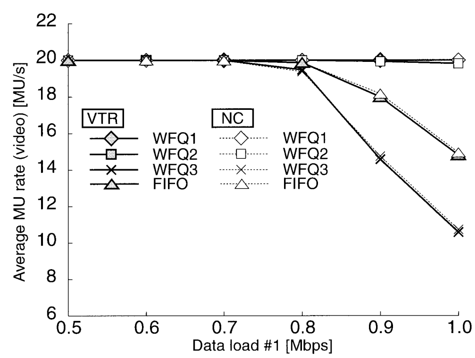

Fig. 17 Average MU rate for video (WFQ).

667 kbps). The reason is as follows. The actually utilized bandwidth for voice and video (about 568 kbps) is smaller than the allocated bandwidth (about 667 kbps). Then the unused bandwidth is utilized by interference data 1. In addition, we can confirm the throughput of interference data 1 becomes equal to its original transmission rate, as its allocated bandwidth increases.

5.3 Quality Measurement of WFQ

We first assess the intra-stream synchronization quality. Figures 14 and 15 show the coefficient of variation of output interval for voice and that for video, respectively. We observe in Fig. 14 that the coefficients of variation of output interval of the types except FIFO hardly change, even if the data load increases. In our measurement environment, the bandwidth for voice is guaranteed in each of the types because its bit rate (i.e., 64 kbps) is smaller than the allocated bandwidth.

On the other hand, in Fig. 15, the coefficient of variation of output interval for video in WFQ3 increases as well as that in FIFO, when the data load exceeds around 0.8 Mbps. This is because the precedence of interference data 1 is the highest, and the available bandwidth for the video stream is smaller than its original average bit rate, i.e., 504 kbps.

Also, we plot the average MU rate of voice and that of video in Figs. 16 and 17, respectively. The average

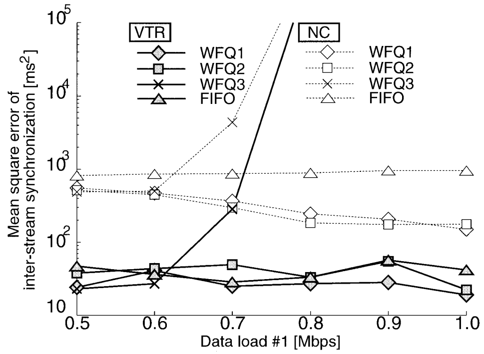

Fig. 18 Mean square error of inter-stream synchronization (WFQ).

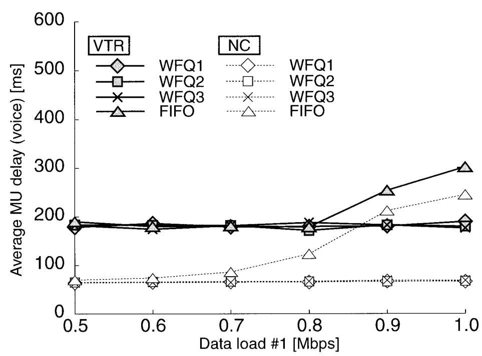

Fig. 19 Average MU delay for voice (WFQ).

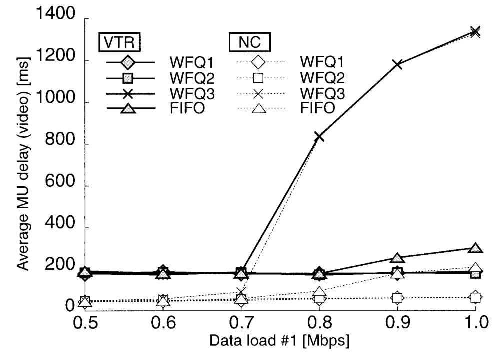

Fig. 20 Average MU delay for video (WFQ).

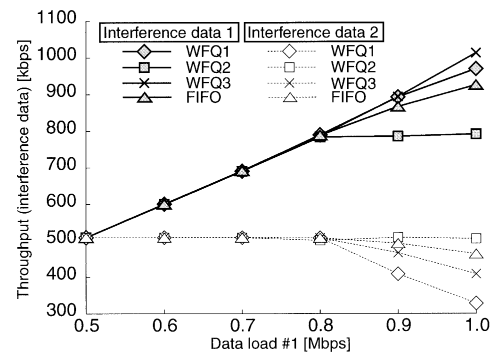

Fig. 21 Throughput of interference data (WFQ).

MU rates of voice for the types except FIFO are approximately the same as the original MU rate. In video, the average MU rate of FIFO and that of WFQ3 decrease when the data load becomes heavier than 0.8 Mbps. This is due to the same reason that we explained for the coefficient of variation of output interval.

Figure 18 shows the mean square error of inter-stream synchronization. The values of the types except WFQ3 are smaller than 6400 ms^2 for all the data loads, but that of WFQ3 exceeds the value, when the data load becomes larger than about 0.7 Mbps. In WFQ, the router stores each incoming packet into the buffer corresponding to the kind of its stream. Therefore, the voice and video packets are treated distinctively. Then, the voice packets in WFQ3 usually arrive earlier than the corresponding video packets because the available bandwidth for the video stream is insufficient.

We can also confirm from Figs. 19 and 20 that the average MU delays of voice in three types with WFQ and those of video except WFQ3 and FIFO keep about the same value. On the other hand, for video, WFQ3 produces larger MU delays than the others at heavy data loads.

Finally, we present the throughput of the interference data in Fig. 21. In this figure, we find that the throughput of the interference data in each type can be more than its allocated bandwidth. This is due to the same reason as before.

5.4 Features of the Four Scheduling Algorithms and Comparison

On the basis of the observations in the previous subsections, we clarify features of the four scheduling algorithms.

In FIFO, the output quality of both audio and video deteriorates as the network load increases, since FIFO has no QoS guarantee mechanisms.

In PQ, we first found that high quality of audio and video is provided if its priority is higher than the priorities of the other traffic. Otherwise, both audio and video qualities are degraded as the amount of the other traffic increases. In this case, the output quality with PQ deteriorates more rapidly than that with FIFO.

In CQ and WFQ, if the total transmission rate of audio and video is less than the allocated bandwidth, high quality can be achieved since the allocated bandwidth is guaranteed. Otherwise, the quality is degraded. We first discuss the features of these algorithms in the former case, i.e., CQ1, CQ2, WFQ1 and WFQ2 in our experiment with VTR. Figure 22 shows the probability distribution of network delay for audio MUs when the data load is 0.8 Mbps. From this figure, we find that CQ1 has a more gently-sloping distribution than the other types. This is because CQ is statically configured and does not automatically adapt to chang-

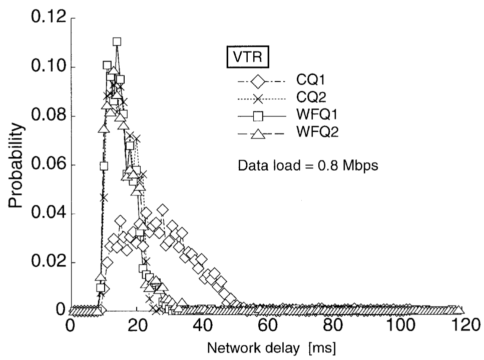

Fig. 22 Probability distribution of network delay for voice in CQ1, CQ2, WFQ1, and WFQ2.

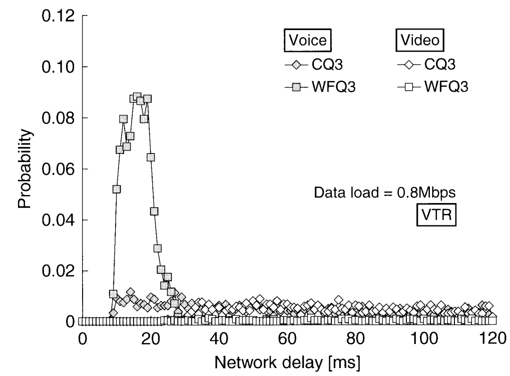

Fig. 23 Probability distribution of network delay for voice and video in CQ3 and WFQ3.

ing network conditions.

Regarding the latter case, from Figs. 7 and 14, we find that the coefficient of variation of output interval for the voice in WFQ3 is smaller than that in CQ3 when the data load is heavier than around 0.8 Mbps. That is, WFQ3 has higher intra-stream synchronization quality of the voice than CQ3. As for the video, we see from Figs. 8 and 15 that in the same area CQ3 and WFQ3 have large coefficients of variation of output interval (the average MU rates of video in CQ3 and WFQ3 deteriorate seriously in Figs. 9 and 17, respectively). Therefore, the intra-stream synchronization quality of the video is not good. Also, from Figs. 10 and 18, we notice that the mean square error of inter-stream synchronization in WFQ3 is larger than that in CQ3 (that is, WFQ3 has lower inter-stream synchronization quality than CQ3); from Figs. 12 and 20, WFQ3 has larger average MU delays of video than CQ3. It should be noted that the allocated bandwidth to the video in CQ3 (about 336 kbps) is approximately equal to that to the video in WFQ3 (about 334 kbps). This is because in WFQ3 the service rates of voice and video each are one sixth of the output link capacity, and in CQ3 one fifth of the output link capacity is allocated to both voice and video together. Since WFQ dynamically allocates bandwidth to each flow, the assigned bandwidth to the video is smaller than the transmission rate of the video and that to the voice is large enough in WFQ3. Therefore, in WFQ3, a number of video MUs arrive at the destination later than the corresponding voice MUs; on the other hand, in CQ3, voice MUs tend to arrive late in the same way as video MUs. We can confirm this in Fig. 23, which illustrates the probability distribution of network delay for voice and video MUs when the data load is 0.8 Mbps. Thus, WFQ3 has higher intra-stream synchronization quality of the voice (i.e., the master stream) than CQ3, in which the quality of the voice deteriorates in the same way as that of the video. Since generally the quality of the master stream (i.e., the voice in this paper) is more important than that of the slave stream (i.e., the video), it is effective to allocate bandwidth to each flow dynamically.

From the above observations, we can say that WFQ is an efficient packet scheduling algorithm for continuous media.

6. Conclusions

We investigated effect of the four packet scheduling algorithms (FIFO, PQ, CBQ and WFQ) on media synchronization quality in live audio and video transmission. By experiment, we assessed the media synchronization quality achieved by the four scheduling algorithms with the VTR media synchronization control and without the control.

In our experimental results, we saw that the quality of both audio and video in FIFO, PQ and CQ can be disturbed severely. On the other hand, we found that WFQ provides high quality of audio because it dynamically schedules the packets on a per-flow basis. Thus, we confirmed that WFQ is the most efficient packet scheduling algorithm for continuous media among the four.

As the next step of our research, we need to investigate the effects of packet scheduling algorithms on media synchronization quality under more complicated network conditions as in [23], e.g., when the number of continuous media streams input to a router increases and when all routers adopt the same scheduling algorithms, by simulation as well as experiment. In addition, we plan to measure the quality of continuous media in IntServ and in DiffServ.

Acknowledgement

This work was supported by the Grant-In-Aid for Scientific Research of Japan Society for the Promotion of Science under Grant 12650378. The authors thank Jun Takeo for his assistance in the experiment.

References

[1] J. Wroclawski, "The use of RSVP with IETF integrated services," RFC2210, Sept. 1997.

[2] P.P. White, "RSVP and integrated services in the Internet: A tutorial," IEEE Commun. Mag., vol.35, no.5, pp.100–106, May 1997.
[3] S. Blake, D. Blake, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," RFC2475, Dec. 1998.
[4] R. Braden, ed., L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource reservation protocol (RSVP)," RFC2205, Sept. 1997.
[5] K. Nichols, S. Blake, F. Baker, and D. Black, "Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers," RFC2474, Dec. 1998.
[6] A. Demers and S. Shenker, "Analysis and simulation of a fair queueing algorithm," Proc. SIGCOMM'89, vol.19, no.4, pp.1–12, Sept. 1989.
[7] A.K. Parekh and R.G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single-node case," IEEE/ACM Trans. Networking, vol.1, no.3, pp.344–357, June 1993.
[8] M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round-robin," IEEE/ACM Trans. Networking, vol.4, no.3, pp.375–385, June 1996.
[9] A. Varma and D. Stiliadis, "Hardware implementation of fair queueing algorithms for asynchronous transfer mode networks," IEEE Commun. Mag., vol.35, no.12, pp.54–68, Dec. 1997.
[10] D.A. Hayes, M. Rumsewicz, and L.L.H. Andrew, "Quality of service driven packet scheduling disciplines for real-time applications: Looking beyond fairness," Proc. IEEE INFOCOM'99, pp.405–412, March 1999.
[11] D. Stiliadis and A. Varma, "Latency-rate servers: A general model for analysis of traffic scheduling algorithms," IEEE/ACM Trans. Networking, vol.6, no.5, pp.611–624, Oct. 1998.
[12] R. Guerin, S. Kamat, V. Peris, and R. Rajan, "Scalable QoS provision through buffer management," Proc. SIGCOMM'98, vol.28, no.4, pp.29–40, Oct. 1998.
[13] G. Hasegawa, T. Matsuo, M. Murata, and H. Miyahara, "Comparisons of packet scheduling algorithms for fair service among connections on the Internet," Proc. IEEE INFOCOM2000, pp.1253–1262, March 2000.
[14] 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.
[15] S. Floyd and V. Jacobson, "Link-sharing and resource management models for packet networks," IEEE/ACM Trans. Networking, vol.3, no.4, pp.365–386, Aug. 1995.
[16] Y. Ishibashi and S. Tasaka, "A synchronization mechanism for continuous media in multimedia communications," Proc. IEEE INFOCOM'95, pp.1010–1019, April 1995.
[17] Y. Ishibashi, S. Tasaka, and A. Tsuji, "Measured performance of a live media synchronization mechanism in an ATM network," Conf. Rec. IEEE ICC'96, pp.1348–1354, June 1996.
[18] M. Katevenis and C. Courcoubetis, "Weighted round-robin cell multiplexing in a general-purpose ATM switch chip," IEEE J. Sel. Areas Commun., vol.9, no.8, pp.1265–1279, Oct. 1991.
[19] S. Tasaka, T. Nunome, and Y. Ishibashi, "Live media synchronization quality of a retransmission-based error recovery scheme," Conf. Rec. IEEE ICC2000, pp.1535–1541, June 2000.
[20] "Cisco IOS(TM) software quality of service solutions," White Paper, Cisco Systems, 1998.
[21] R. Steinmetz, "Human perception of jitter and media synchronization," IEEE J. Sel. Areas Commun., vol.14, no.1. pp.61–72, Jan. 1996.
[22] S. Tasaka and Y. Ishibashi, "A performance comparison of single-stream and multi-stream approaches to live media synchronization," IEICE Trans. Commun., vol.E81-B, no.11, pp.1988–1997, Nov. 1998.
[23] K. Ito, S. Tasaka, and Y. Ishibashi, "Effect of packet scheduling algorithms at multiple nodes on media synchronization quality," The 2001 IEICE General Conference, B-11-18, March 2001.

Kenji Ito received the B.S. and M.S. degrees from Nagoya Institute of Technology, Nagoya, Japan, in 1999 and 2001, respectively. In April 2001, he joined Toyota Central Research and Development Laboratories, Inc. His research interests include media synchronization mechanisms, QoS control mechanisms and multimedia communication protocols.

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 Electrical and Computer 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, high-speed networks and 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.

Yutaka Ishibashi received the B.S., M.S. and Ph.D. degrees from Nagoya Institute of Technology, Nagoya, Japan, in 1981, 1983 and 1990, respectively. From 1983 to 1993, he was with NTT Laboratories. In 1993, he joined Nagoya Institute of Technology as an Associate Professor in the Department of Electrical and Computer Engineering. In August 2001, he was promoted to a Professor. From June 2000 to March 2001, he was a Visiting Professor in the Department of Computer Science and Engineering at the University of South Florida. His research interests include distributed multimedia applications, media synchronization algorithms, and multimedia communication protocols. Dr. Ishibashi is a member of the IEEE, ACM, Information Processing Society of Japan, and the Institute of Image Information and Television Engineers.