PAPER

Tradeoff Relationship between Fidelity and Latency in Interactive Audio-Video Applications over IP Networks

Yoshihiro ITO[†a)], ***Member and*** **Shuji TASAKA**[†b)], ***Fellow***

SUMMARY Interactive audio-video applications over IP networks have subjective tradeoffs between fidelity and latency owing to packet buffering at the receiver. Increasing the buffering time improves the fidelity, whereas it degrades the latency. This paper makes the subjective tradeoff between fidelity and latency clear in a quantitative way. In addition, we examine the effect of tasks on the subjective tradeoff. In evaluating the effect of tasks, we use two tasks according to ITU-T Recommendation P.920. An experiment was conducted to measure user-level QoS of an interactive application with the psychometric methods. We then investigate the subjective tradeoff quantitatively by QoS mapping. The experimental results confirm that there exists the buffering time which makes user-level QoS the highest. The results also show that the optimum buffering time depends on the kind of task.

key words: *user-level QoS assessment, interactive audio-video application, subjective tradeoff*

1. Introduction

In audio-video transmission over IP networks, its temporal structure can be easily disturbed by delay jitter of packets. The disturbance decreases *fidelity* of the audio-video stream.

The impairment of the fidelity can be remedied by *a playout buffer* in the receiver; packets which arrive at the receiver are stored in its buffer so that the delay jitter can be absorbed. We refer to the packet delay caused by the playout buffer as the *buffering time*. To absorb larger delay jitter, more buffer space is required.

On the other hand, the utilization of the playout buffer increases *latency* because of the buffering time. The latency is the difference between the time when media are generated at the sender and the time when the media are output at the receiver. The increase of the latency causes degradation of interactivity between users in interactive audio-visual applications, such as TV conferences. Thus, larger buffering time does not always contribute toward improving quality in those applications.

In interactive audio-visual applications, users' subjective quality is important. It is affected by the fidelity and the latency. Increase of the buffering time improves the fidelity, while the latency rises. That is, in the interactive audio-visual applications, there exists *a subjective tradeoff between the fidelity and the latency* by the buffering control. Therefore, regardless of a buffering control scheme, it is important to find appropriate buffering time which makes subjective quality high in the interactive audio-visual applications. The subjective quality corresponds to *user-level QoS* in the context of the network architecture.

In general, QoS has a layered structure. For example, Tasaka and Ishibashi [1] identified six levels of QoS: *physical-level*, *node-level*, *network-level*, *end-to-end-level*, *application-level* and user-level. The user-level QoS is subjective one. On the other hand, the fidelity and the latency are application-level QoS.

We can find many studies on application-level QoS for audio only or video only in interactive applications. For example, see [2] through [7]. In the great majority of papers, however, buffering control is the main subject; they do not assess user-level QoS.

On the other hand, few researches report the effect of buffering control on user-level QoS of both audio and video transmission. Even, these researches treat only the fidelity. For example, Kouvelas et al. [8] showed that a reconstruction buffer must be added to a video system for lip synchronization. In [8], the effectiveness of lip synchronization is confirmed by subjective assessment. Steinmetz [9] assumed that audio and video are individually buffered to absorb delay jitter; he investigated the tolerance of skew, which is difference between audio delay and video one caused by buffering control.

Thus, in the literature, we had found no study that treats the subjective tradeoff between the fidelity and the latency caused by buffering control in interactive audio-visual applications.

Then, the authors addressed themselves to this problem and showed the subjective tradeoff between fidelity and latency in [10]. However, in [10], the tradeoff is not clarified enough. Moreover, the study treats only one specific task.

The purposes of the paper are two-hold: One is to show the subjective tradeoff between fidelity and latency clearly. The other is to investigate the effect of tasks on the tradeoff. In order to assess the user-level QoS, we utilize the *psychometric methods* [11]. The psychometric methods have been proposed in the psychological field and are effective in assessment of human subjectivity. Moreover, to investigate the tradeoff quantitatively, we utilizes QoS mapping from application-level to user-level. As a QoS mapping method, we use the multiple regression analysis.

The rest of the paper is organized as follows. Section 2 introduces a method of assessing user-level QoS with psy-

Manuscript received June 9, 2006.
Manuscript revised November 22, 2006.

†The authors are with the Department of Computer Science and Engineering, Graduate School of Engineering, Nagoya Institute of Technology, Nagoya-shi, 466-8555 Japan.
a) E-mail: yoshi@nitech.ac.jp
b) E-mail: tasaka@nitech.ac.jp

DOI: 10.1093/ietcom/e90–b.5.1112

chometric methods. Section 3 describes application-level QoS parameters we use in this paper. Section 4 explains our experiment. We show our results and consideration in Sect. 5.

2. Psychometric Methods for User-Level QoS Assessment

2.1 Four General Classes of Measurement Scales

Before we explain the psychometric methods, let us consider four general classes of measurement scales which represent human subjectivity. In general, we can define four basic types of the measurement scales according to the mathematical operations that can be performed legitimately on the numbers obtained by the measurement; from lower to higher levels, we have *nominal, ordinal, interval* and *ratio* scales [11]. In the nominal scale, we use a number only as a label for a class or a category. The numbers assigned in the ordinal scale have the property of rank order. In the interval scale, numerically equal distances stand for empirically equal distances in some psychological aspect of objects. However, the origin and the unit of the interval scale are meaningless. In the ratio scale, the unique origin can be determined in addition to the property of the interval scale. Since almost all the statistical procedures can be applied to the interval scale and the ratio scale, it is desirable to represent the user-level QoS by an interval scale or a ratio scale. In this paper we utilize an interval scale since it is generally easier to calculate an interval scale than a ratio scale.

2.2 Psychometric Methods

In [10], [12] and [13], to obtain an interval scale as user-level QoS parameter, we adopted two psychometric methods: the *method of paired comparisons and Thurstone's law of comparative judgment* [11] and the *method of successive categories* [11]. Reference [12] assesses the user-level QoS of audio-video transmission by the method of paired comparisons and Thurstone's law of comparative judgment. References [10] and [13] utilize the method of successive categories. The method of paired comparisons and Thurstone's law of comparative judgment can give more accurate values of the interval scale but takes longer experimental time than the method of successive categories. In this paper, we utilize the method of successive categories. The method of successive categories is introduced in the next subsection.

2.3 Method of Successive Categories

In the method of successive categories, a subjective score is measured by the *rating-scale method* [11]. In the method, experimental subjects (or observers) classify each stimulus into one of a certain number of categories. Note that a stimulus means an object, such as audio and video, for evaluation. Each category has a predefined number. For example, "excellent" is assigned 5, "good" 4, "fair" 3, "poor" 2 and "bad" 1. However, the numbers assigned to the categories only have a greater-than-less-than relation between them, that is, the assigned number is nothing but an ordinal scale. Therefore, it is not desirable to use the assigned number for obtaining the user-level QoS parameter.

In order to obtain an interval scale as the user-level QoS parameter, we first measure the frequency of each category with which the stimulus was placed in the category by the rating-scale method. With the *law of categorical judgment* [11], we can translate the frequency obtained by the rating-scale method into an interval scale. We can apply almost all the statistical operations to the scale.

2.4 The Law of Categorical Judgment

The law of categorical judgment makes the following assumptions. Let the number of the categories be $m+1$. When stimulus j ($j = 1, \cdots, n$) is presented to a subject, a psychological value designated by s_j occurs on a *psychological continuum*, which is an interval scale, in him/her. For the $m + 1$ categories, their *boundaries* have values on the interval scale. We denote the upper boundary of category g ($g = 1, \cdots, m+1$) by c_g and define $c_0 \stackrel{\Delta}{=} -\infty$ and $c_{m+1} \stackrel{\Delta}{=} +\infty$. The subject classifies n stimuli into the $m + 1$ categories ($n > m + 1$) by comparing s_j with c_g. If $c_{g-1} < s_j \leq c_g$, then stimulus j is classified into category g. The categories can be arranged in a rank order, in the sense that each stimulus in category g is judged to have a psychological value which is "less than" the one for any stimulus in category $g + 1$. This statement holds for all values of g from 1 to m. The variable c_g is normally distributed with mean t_g and standard deviation d_g. Also, the variable s_j is normally distributed with mean R_j and standard deviation σ_j. Then, we can consider R_j as an interval scale. In this paper, we refer to the obtained interval scale as *psychological scale* and treat it as a user-level QoS parameter.

Since the law of categorical judgment is a suite of assumptions, we must test goodness of fit between the obtained interval scale and the measurement result. Mosteller [14] proposed a method of testing the goodness of fit for a scale calculated with Thurstone's law of comparative judgment [11], which we use in [12]. The method can be applied to a scale obtained by the law of categorical judgment. In this paper, we use Mosteller's method to test the goodness of fit.

3. Application-Level QoS Parameters

In this paper, we consider the fidelity and the latency as application-level QoS. Thus, in order to treat them quantitatively, we need to express them in terms of some application-level QoS parameters.

In this paper, we assume that the fidelity indicates how exactly the temporal structure of media is preserved. Therefore, the fidelity relates to media synchronization quality [1]. The media synchronization is defined for multimedia in general. In this paper, however, we treat only two types of media: audio and video.

The media synchronization for audio and video can be classified into *intra-stream synchronization* and *inter-stream synchronization* [15]. The former indicates the continuity of a single stream (audio or video), while the latter is synchronization between an audio stream and the corresponding video one. We consider measures of media synchronization quality as application-level QoS parameters about the fidelity.

In order to represent media synchronization quality, Ref. [12] uses nine application-level QoS parameters. We use seven application-level QoS parameters out of the nine in [12]. First, we adopt the *coefficient of variation of output interval*, which is defined as the ratio of the standard deviation of the MU output interval of a stream to its average. MU stands for "media unit," which indicates an information unit for media synchronization. This parameter is denoted by C_a for audio and by C_v for video. Second, we use the *MU loss ratio* for audio L_a and that for video L_v, which are the ratio of the number of lost MUs to the total number of generated MUs. Third, we treat the *mean square error of intra-stream synchronization*, which is defined as the average square of the difference between the output interval of MU at the destination and the generation one at the source. We denote it by E_a for audio and by E_v for video. These six parameters indicate the intra-stream synchronization quality.

The QoS parameter for the inter-stream synchronization is the *mean square error* E_{int}, which is defined as the average square of the difference between the output-time difference of the audio and corresponding video MUs and their timestamp difference. We use the seven application-level QoS parameters introduced so far as the fidelity measure.

Note that, in [12], the *average MU rate* for audio or that for video, which is defined as the average number of (either audio or video) MUs output in a second at the destination, is adopted. Since the MU loss ratio highly correlates with the average MU rate, we do not treat the average MU rates in this paper.

For the purpose of examination of the application-level QoS from a latency point of view, we evaluate two application-level QoS parameters which are used in [10]: the *average MU delay* of audio and that of video. The average MU delay is the average time in seconds from the moment an MU is generated until the instant the MU is output. We denote it by D_a for audio and by D_v for video.

4. Experiments

4.1 Experimental Environment

This paper sets up an experimental environment shown in Fig. 1. In the experimental environment, a pair of subjects sit in front of their terminals, and each terminal transmits a pair of audio-video streams of the subject to each other over a network emulator which produces delay jitter. The subjects assess the output audio-video stream subjectively.

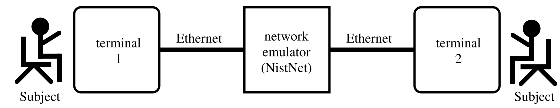

Fig. 1 Experimental environment.

Table 1 Specifications of audio-video streams.

	audio	video
coding scheme	unsigned 8 bit PCM	MPEG1
image size [pixels]	-	240×180
picture pattern	-	IPPP
average MU size [byte]	400	3138
average MU rate [MU/s]	20	20
average MU interval [ms]	50.0	50.0
average bit rate [kb/s]	64	502
playout time [s]	20	20

Each terminal can perform the buffering control. By adjusting the buffering time at the terminals, we can realize the tradeoff between the fidelity and the latency. If we increase the buffering time, the fidelity improves, but the latency rises. Conversely, if we reduce the buffering time, the latency decreases, but the fidelity deteriorates.

The audio-video streams are encapsulated in UDP datagrams. The media specifications of the audio-video streams are shown in Table 1.

To investigate the effect of tasks on the subjective tradeoff between the fidelity and the latency, we gave orders for the subjects to perform two tasks in the subjective assessment of the audio-video streams. Considering ITU-T Recommendation P.920 [16], we selected two tasks: *Task 1* and *Task 2*. In Task 1, each pair of subjects take turns in counting. In Task 2, one subject imitates the motion of the other. They change their roles alternately. Task 1 is performed to evaluate the effects of speech delay on communication quality. Task 2 is to evaluate the effects of audiovisual delay and/or transmission errors on communication quality.

The subjects assess their subjectivity of the audio-video stream with the rating-scale method. In this method, we use five categories (i.e., $m = 4$) of impairment:"imperceptible" assigned integer 5, "perceptible, but not annoying" 4, "slightly annoying" 3, "annoying" 2, and "very annoying" 1.

Two terminals are connected to each other via a network emulator (NistNet) [17]. The network emulator can delay packets according to a specified probability distribution.

In this experiment, we delayed packets according to Pareto-normal distribution to emulate packet delay of the Internet. Reference [18] shows that Pareto distribution is the most appropriate model of tail-parts of packet delay distributions in the Internet. It also indicates that the normal or the log-normal distribution is an appropriate model of the entire packet delay distribution in the Internet. Therefore, we have chosen Pareto-normal distribution, which is the normal distribution with Pareto tail, as the distribution of delay.

We set the mean of delay to 50, 100 and 150 ms. Also we chose 20, 40 and 60 ms as the standard deviations of delay. It should be noted that NistNet delays packets according to a specified distribution. When the value of the random variable for the distribution becomes negative, NistNet sets the actual delay to 0. As a result, if the frequency with which the value of the random variable becomes negative increases, the distribution of delay is extremely distorted compared with the expected distribution. Therefore, we should utilize the distribution whose probability of being negative is sufficiently small. Thus, we set the standard deviation of delay so that it does not exceed half of the mean. For example, when the mean of delay is 50 ms, we set the standard deviation of delay to only 20 ms. If the mean of delay is 100 ms, we set the standard deviation of delay to 20 and 40 ms. Consequently, we utilized 6 combinations of the mean and the standard deviation.

4.2 Scheme for Buffering Control

In order to change the buffering time at the terminals in our experimental environment, we utilize a simple buffering control scheme, which is introduced below.

For a description of the scheme, we define the following notations for stream j ($j = 1$ for audio, and $j = 2$ for video). Firstly, we let $T_n^{(j)}$ ($n = 1, 2, \cdots$) denote the timestamp of the n-th MU in stream j, which is attached when it generates, and define $\sigma_{n,m}^{(j)} \triangleq T_m^{(j)} - T_n^{(j)}$ ($n \leq m$; $m = 1, 2, \cdots$). Secondly, let $J_{\max}$ be an estimate of the maximum network delay jitters. In order to absorb delay jitters at the receiver, we set the *initial buffering time* to $J_{\max}$. Thirdly, let $A_n^{(j)}$ and $D_n^{(j)}$ represent the arrival time and output time, respectively, of the n-th MU in stream j at the destination. Thirdly, let $A_n^{(j)}$ and $D_n^{(j)}$ represent the arrival time and output time, respectively, of the n-th MU in stream j at the destination.

First, we determine the output time of the first MU in each stream, which is also used to obtain the time-origin for output control at the destination. Defining $A_1 \triangleq \max\left(A_1^{(1)}, A_1^{(2)}\right)$ and $T_1 \triangleq \min\left(T_1^{(1)}, T_1^{(2)}\right)$, we set the output time of the first MU in stream j ($j = 1$ and 2) to

$$D_1^{(j)} = A_1 + T_1^{(j)} - T_1 + J_{\max} \tag{1}$$

Next, we define the *ideal target output time* $x_n^{(j)}$ of the n-th MU in stream j as

$$x_1^{(j)} = D_1^{(j)} \tag{2}$$

$$x_n^{(j)} = x_1^{(j)} + \sigma_{1,n}^{(j)} (n = 2, 3, \cdots) \tag{3}$$

We calculate the output time of each MU with the ideal target output time.

If $A_n^{(j)} \leq x_n^{(j)}$, $D_n^{(j)}$ is set to $x_n^{(j)}$. Otherwise, the n-th MU in stream j is dropped.

In our experiment, we vary the initial buffering time, which is set to $J_{\max}$, to change the buffering time. We chose ten values from 0 ms to 2000 ms as those of the initial buffering time.

4.3 Subjects

We used 50 subjects in the subjective assessment. The subjects were non-experts in the sense that they were not directly concerned with audio and video quality as a part of their normal work. They are male and female, and their ages were between 20 and 25. It took about thirty minutes for a subject to finish all assessment.

5. Experimental Results

5.1 Results of Application-Level QoS Assessment

Figures 2 through 10 plot measured application-level QoS parameters, which were introduced in Sect. 3. The figures

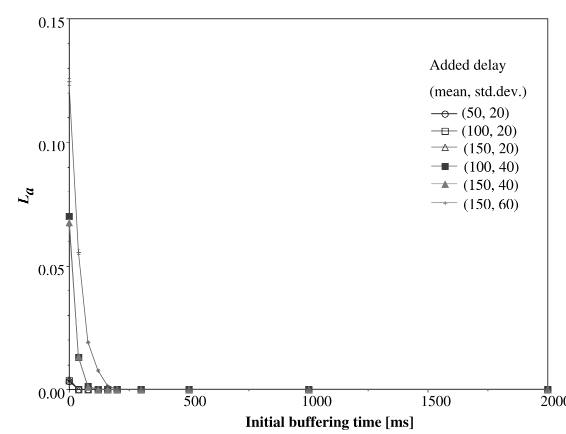

Fig. 2 MU loss ratio for audio L_a versus initial buffering time.

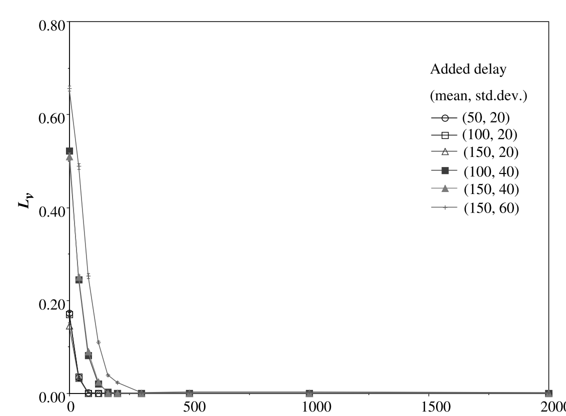

Fig. 3 MU loss ratio for video L_v versus initial buffering time.

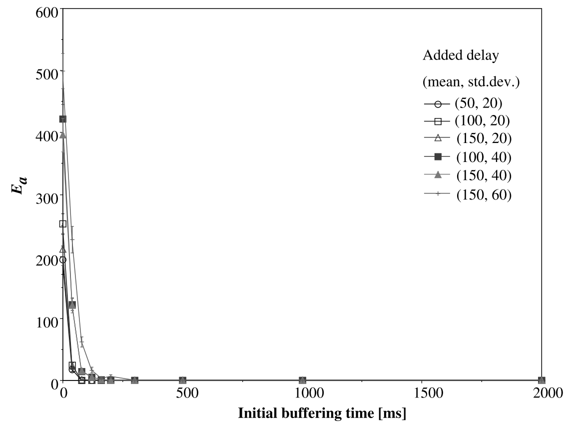

Fig. 4 Mean square error of intra-stream synchronization for audio E_a versus initial buffering time.

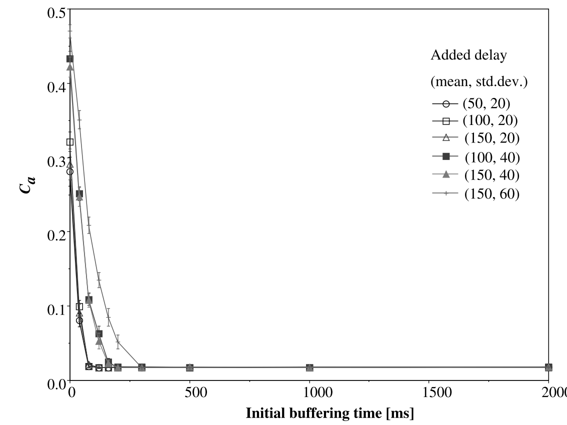

Fig. 6 Coefficient of variation of output interval for audio C_a versus initial buffering time.

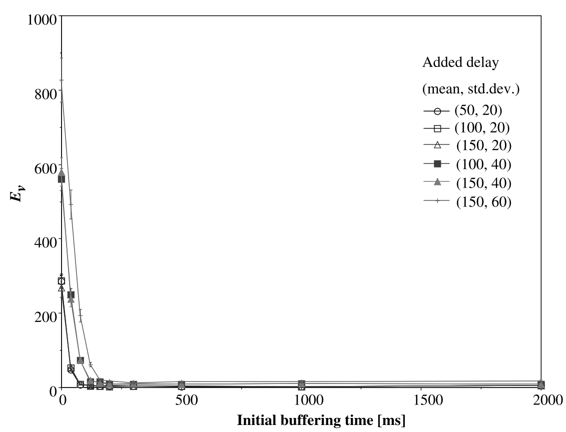

Fig. 5 Mean square error of intra-stream synchronization for video E_v versus initial buffering time.

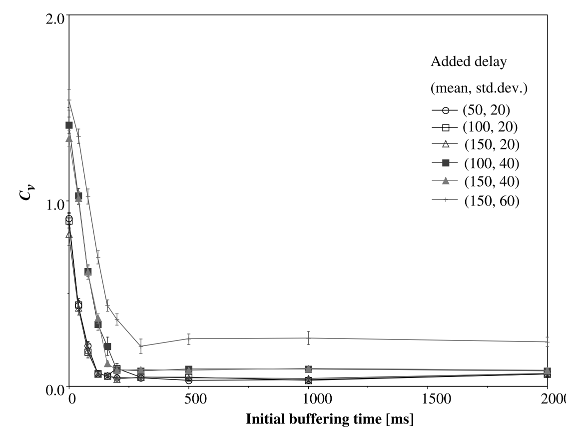

Fig. 7 Coefficient of variation of output interval for video C_v versus initial buffering time.

also show the 95% confidence interval on each measured value.

Figures 2 and 3 display the MU loss ratio for audio versus the initial buffering time and that for video, respectively. Note that the scales of the ordinate in the two figures are different. From Figs. 2 and 3, we find that increase of the initial buffering time decreases the MU loss ratio for audio and that for video. The MU loss ratio for video takes higher value than that for audio. This is because a video MU consists of more than one IP packet while one IP packet composes an audio MU. Moreover, when the initial buffering time exceeds 300 ms, both MU loss ratio for audio and that for video are 0.

Figures 4 and 5 denote the mean square error of intra-stream synchronization for audio versus the initial buffering time and that for video, respectively. They show that the mean square error of intra-stream synchronization of audio and that of video decrease as the initial buffering time increases. When the initial buffering time is 300 ms or more, both mean square errors of intra-stream synchronization are very small; they are less than 20.

The coefficients of variation of output interval of audio and that of video are indicated in Figs. 6 and 7, respectively. From these figures, we see that the value of the coefficient decreases as the initial buffering time increases. They also show that the coefficient of variation of output interval for video takes larger values than that for audio. This was caused by the difference in MU size between audio and video. An audio MU can be transmitted with a single IP packet while a video MU consists of a few IP packets. Note that even if only one of the packets which compose a video MU arrives late, the output of the MU must be delayed.

Figure 8 indicates the mean square error of inter-stream synchronization versus the initial buffering time. From Fig. 8, we find that the mean square error of inter-stream synchronization diminishes as the initial buffering time grows. This is because more buffering time can absorb larger delay jitter.

From Figs. 2 through 8, we can confirm that the fidelity of the audio-video streams improves with the increase of the initial buffering time. Moreover, a larger standard deviation of added delay degrades the fidelity.

Figures 9 and 10 plot the average MU delay of audio and that of video, respectively. These figures indicate that

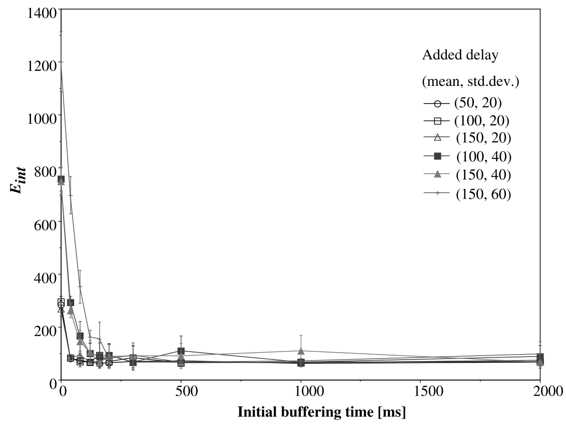

Fig. 8 Mean square error of inter-stream synchronization E_{int} versus initial buffering time.

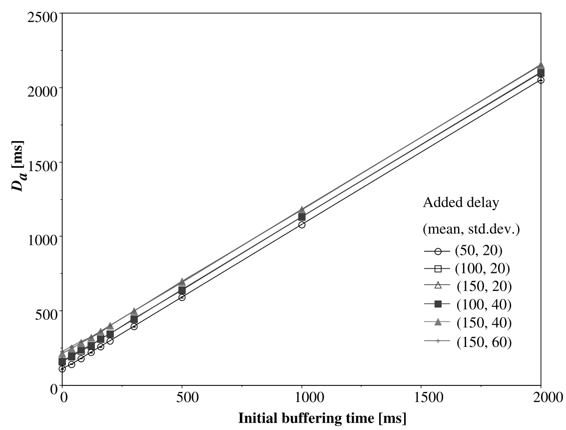

Fig. 9 Average MU delay for audio D_a versus initial buffering time.

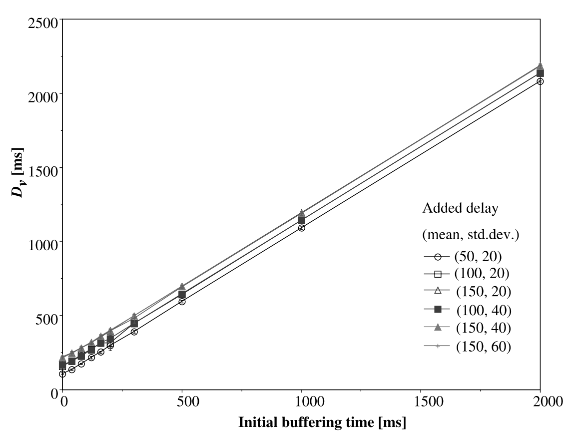

Fig. 10 Average MU delay for video D_v versus initial buffering time.

the average MU delay, that is, the latency, increases linearly as the initial buffering time rises. Therefore, we can confirm that the increment of the initial buffering time improves the fidelity but increases the latency.

5.2 Calculation of User-Level QoS Parameter

Tables 2 and 3 indicate the measurement result for Task 1 and that for Task 2, respectively, by the rating-scale method. Note that Category 5 corresponds to “imperceptible” impairment, Category 4 “perceptible, but not annoying,” Category 3 “slightly annoying,” Category 2 “annoying” and Category 1 “very annoying,” as already stated in Sect. 4.2.

Each entry in this table represents the number of subjects who classified the stimulus into the entry.

From Tables 2 and 3, we calculated psychological scales, that is, user-level QoS parameter, with the law of categorical judgment. To know the detail of the calculation of psychological scale with the law of categorical judgment, see [10] and [13].

Table 4 displays obtained the psychological scale, that is, the value of the user-level QoS parameter.

To verify the obtained psychological scale, we have performed Mosteller’s test. As a result of Mosteller’s test, the null hypothesis that the obtained interval scale fits the observed data cannot be rejected at significance level 0.05. That is, if the hypothesis is right, the probability that the hypothesis is rejected by mistake is less than 0.05. Therefore, we consider that the obtained scale is appropriate for the user-level QoS parameter.

We plot the calculated user-level QoS parameter for Task 1 and that for Task 2 in Figs. 11 and 12, respectively. Note that, in an interval scale, we can select an arbitrary origin and any unit of scale. For convenience, then, we set the smallest value of the user-level QoS parameter to the origin. From Figs. 11 and 12, we see that the value of the user-level QoS parameter first grows as the initial buffering time increases. Then, the value of the user-level QoS parameters begin to decrease. From Figs. 2 through 10, we have found that application-level QoS about the fidelity improves but that about the latency deteriorates with the increment of the initial buffering time. Therefore, we confirm the subjective tradeoff between the fidelity and the latency caused by the buffering control. Moreover, from these figures, we find that the buffering time which makes the user-level QoS parameter value the highest for Task 1 is smaller than that for Task 2 for each delay distribution. This means that Task 1 requires higher interactivity than Task 2. These figures also show that longer mean delay degrades user-level QoS more.

5.3 QoS Mapping

To investigate the subjective tradeoff between the fidelity and the latency by increasing the initial buffering time quantitatively, we perform QoS mapping between application-level QoS and user-level QoS. In [10], [12] and [13], the authors perform QoS mapping between application-level QoS and user-level one by *multiple regression analysis*. In this paper, we also utilize multiple regression analysis for the QoS mapping.

We consider the user-level QoS parameter and

Table 2 Measurement result by the rating-scale method for Task 1.

Added delay		Buffering	Category				
Ave. [ms]	Std. Dev. [ms]	time [ms]	1	2	3	4	5
50	20	0	0	1	7	13	29
50	20	40	0	0	7	23	20
50	20	80	0	1	3	28	18
50	20	120	0	2	6	21	21
50	20	160	0	0	11	21	18
50	20	200	0	6	14	20	10
50	20	300	0	2	25	18	5
50	20	500	4	18	17	10	1
50	20	1000	26	19	3	2	0
50	20	2000	37	9	2	2	0
100	20	0	1	4	16	15	14
100	20	40	0	0	8	25	17
100	20	80	0	1	11	21	17
100	20	120	0	1	8	30	11
100	20	160	0	7	9	19	15
100	20	200	1	6	15	20	8
100	20	300	3	12	13	19	3
100	20	500	7	21	11	8	3
100	20	1000	25	20	2	3	0
100	20	2000	41	6	2	1	0
150	20	0	0	3	18	19	10
150	20	40	0	1	16	21	12
150	20	80	0	2	17	12	19
150	20	120	1	7	17	18	7
150	20	160	1	8	11	24	6
150	20	200	0	6	24	15	5
150	20	300	1	10	22	14	3
150	20	500	7	19	18	4	2
150	20	1000	29	14	3	4	0
150	20	2000	40	6	3	1	0
100	40	0	2	24	15	7	2
100	40	40	1	7	23	13	6
100	40	80	0	2	15	24	9
100	40	120	0	3	9	24	14
100	40	160	0	9	4	23	14
100	40	200	3	6	6	22	13
100	40	300	3	14	16	11	6
100	40	500	8	20	15	7	0
100	40	1000	22	22	3	3	0
100	40	2000	42	5	2	1	0
150	40	0	7	22	16	5	0
150	40	40	1	12	21	13	3
150	40	80	0	3	20	22	5
150	40	120	0	7	15	22	6
150	40	160	0	2	19	21	8
150	40	200	1	8	18	19	4
150	40	300	0	14	23	11	2
150	40	500	7	24	10	9	0
150	40	1000	22	18	6	3	1
150	40	2000	42	6	1	1	0
150	60	0	14	24	9	1	2
150	60	40	2	24	15	6	3
150	60	80	1	10	17	18	4
150	60	120	1	10	18	12	9
150	60	160	0	8	22	17	3
150	60	200	0	6	19	19	6
150	60	300	2	8	19	14	7
150	60	500	13	21	11	3	2
150	60	1000	27	18	2	2	1
150	60	2000	43	5	1	1	0

Table 3 Measurement result by the rating-scale method for Task 2.

Added delay		Buffering	Category				
Ave. [ms]	Std. Dev. [ms]	time [ms]	1	2	3	4	5
50	20	0	0	3	15	23	9
50	20	40	0	3	9	21	17
50	20	80	0	0	4	21	25
50	20	120	0	0	8	20	22
50	20	160	0	0	5	24	21
50	20	200	0	1	16	19	14
50	20	300	0	3	9	19	19
50	20	500	1	7	17	18	7
50	20	1000	8	24	16	2	0
50	20	2000	35	9	4	1	1
100	20	0	1	7	13	21	8
100	20	40	0	0	4	23	23
100	20	80	0	3	6	22	19
100	20	120	0	1	9	23	17
100	20	160	0	1	11	25	13
100	20	200	0	2	8	24	16
100	20	300	1	4	15	22	8
100	20	500	4	9	18	15	4
100	20	1000	9	25	12	4	0
100	20	2000	32	13	4	1	0
150	20	0	1	7	18	19	5
150	20	40	0	1	6	24	19
150	20	80	0	2	9	27	12
150	20	120	0	1	15	26	8
150	20	160	0	4	7	31	8
150	20	200	0	7	16	19	8
150	20	300	0	5	15	26	4
150	20	500	0	10	19	16	5
150	20	1000	14	23	9	4	0
150	20	2000	29	13	4	3	1
100	40	0	7	19	14	10	0
100	40	40	0	7	18	15	10
100	40	80	0	2	12	20	16
100	40	120	0	0	6	30	14
100	40	160	0	0	17	19	14
100	40	200	0	1	9	32	8
100	40	300	1	4	15	24	6
100	40	500	1	13	19	13	4
100	40	1000	9	25	11	3	2
100	40	2000	33	12	4	1	0
150	40	0	9	18	15	6	2
150	40	40	1	7	22	15	5
150	40	80	0	3	20	18	9
150	40	120	1	2	12	25	10
150	40	160	0	3	13	19	15
150	40	200	0	2	15	29	4
150	40	300	0	5	21	20	4
150	40	500	1	10	28	7	4
150	40	1000	7	24	13	5	1
150	40	2000	33	10	6	0	1
150	60	0	11	21	16	1	1
150	60	40	4	13	23	7	3
150	60	80	1	7	20	16	6
150	60	120	0	4	19	20	7
150	60	160	0	4	10	30	6
150	60	200	0	5	11	18	16
150	60	300	1	7	15	21	6
150	60	500	3	20	15	11	1
150	60	1000	11	28	9	2	0
150	60	2000	37	8	4	1	0

Table 4 User-level QoS parameter.

Added delay Ave. [ms]	Added delay Std. Dev. [ms]	Buffering time [ms]	Task 1	Task 2
50	20	0	3.757	3.006
50	20	40	3.590	3.290
50	20	80	3.707	3.879
50	20	120	3.521	3.599
50	20	160	3.384	3.717
50	20	200	2.869	3.301
50	20	300	2.797	3.326
50	20	500	1.836	2.645
50	20	1000	0.601	1.257
50	20	2000	0.362	0.777
100	20	0	2.892	2.717
100	20	40	3.468	3.829
100	20	80	3.456	3.395
100	20	120	3.406	3.501
100	20	160	3.015	3.379
100	20	200	2.726	3.382
100	20	300	2.247	2.802
100	20	500	1.804	2.238
100	20	1000	0.683	1.309
100	20	2000	0.120	0.397
150	20	0	2.979	2.583
150	20	40	3.260	3.617
150	20	80	3.257	3.279
150	20	120	2.645	3.183
150	20	160	2.665	3.068
150	20	200	2.554	2.736
150	20	300	2.368	2.717
150	20	500	1.680	2.460
150	20	1000	0.732	1.140
150	20	2000	0.195	0.990
100	40	0	1.912	1.691
100	40	40	2.545	2.753
100	40	80	3.090	3.296
100	40	120	3.233	3.473
100	40	160	2.999	3.092
100	40	200	2.741	3.307
100	40	300	2.248	2.757
100	40	500	1.549	2.358
100	40	1000	0.769	1.507
100	40	2000	0.094	0.379
150	40	0	1.494	1.671
150	40	40	2.322	2.532
150	40	80	2.798	2.920
150	40	120	2.692	2.976
150	40	160	2.993	3.173
150	40	200	2.519	2.926
150	40	300	2.070	2.616
150	40	500	1.581	2.316
150	40	1000	1.116	1.540
150	40	2000	0.029	0.753
150	60	0	1.296	1.306
150	60	40	1.961	2.009
150	60	80	2.471	2.584
150	60	120	2.581	2.815
150	60	160	2.402	2.945
150	60	200	2.674	3.101
150	60	300	2.481	2.647
150	60	500	1.439	1.877
150	60	1000	0.906	1.045
150	60	2000	0.000	0.302

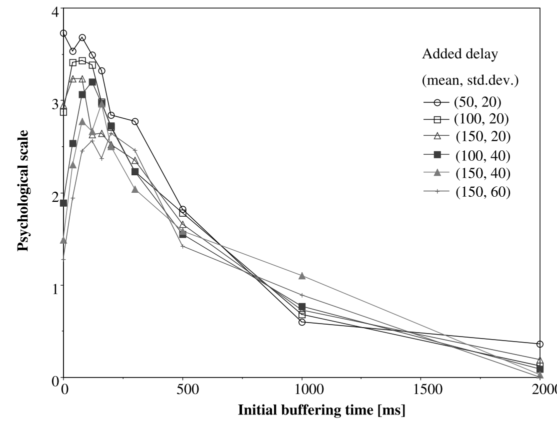

Fig. 11 User-level QoS parameter for Task 1 versus initial buffering time.

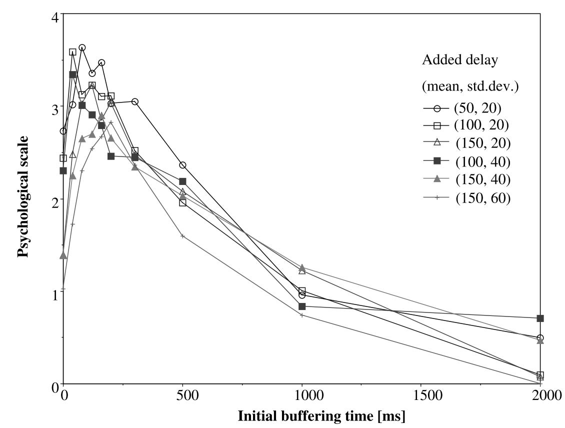

Fig. 12 User-level QoS parameter for Task 2 versus initial buffering time.

application-level QoS parameters as the criterion variable and predictor variables, respectively. We select two application-level QoS parameters as predictor variables; one is an application-level QoS parameter about the fidelity and the other is one about the latency. Therefore, we first select one parameter from among L_a, L_v, E_a, E_v, C_a, C_v and E_{int}. Then, we adopt either D_a or D_v. In this case, we must consider 7×2 combinations of application-level QoS parameters as predictor variables. We try to select one combination of the application-level QoS parameters whose contribution rate adjusted for degrees of freedom is the highest. The contribution rate adjusted for degrees of freedom indicates goodness of fit of the obtained multiple regression line. However, in order to compare two QoS mapping lines for Task 1 and Task 2 directly, we use the same parameters as predictor variables, though the combination does not provide the highest contribution rate.

Tables 5 and 6 show contribution rates adjusted for degrees of freedom for the 14 combinations of the application-level QoS parameters for Task 1 and those for Task 2, respectively. From Table 6, we find that two combinations, (D_a, E_v) and (D_v, E_v) have the highest value of the contribution rate for Task 2. In this paper, we select a parameter

Table 5 Contribution rates adjusted for degrees of freedom for Task 1.

	L_a	E_a	C_a	L_v	E_v	C_v	E_{int}
D_a	0.906	0.898	0.894	0.900	0.901	0.889	0.905
D_v	0.905	0.897	0.893	0.900	0.901	0.888	0.904

Table 6 Contribution rates adjusted for degrees of freedom for Task 2.

	L_a	E_a	C_a	L_v	E_v	C_v	E_{int}
D_a	0.905	0.918	0.914	0.913	0.921	0.905	0.916
D_v	0.906	0.918	0.914	0.913	0.921	0.905	0.916

regarding audio and one concerning video. Therefore, we choose (D_a, E_v) as predictor variables for Task 2. On the other hand, Table 5 shows that the combination (D_a, E_v) also have high value of the contribution rate for Task 1, though it is not the highest. Then, we regard (D_a, C_v) as predictor variables for Task 1 and Task 2.

As a result of multiple regression analysis, we have the multiple regression lines as follows:

$$S_1 = 3.310 - 1.701 \times 10^{-3} D_a - 1.510 \times 10^{-3} E_v \quad (4)$$

$$S_2 = 3.361 - 1.588 \times 10^{-3} D_a - 2.346 \times 10^{-3} E_v \quad (5)$$

where S_1 and S_2 are estimate of the user-level QoS parameter for Task 1 and that for Task 2, respectively.

From Eqs. (4) and (5), we find the following. When we increase the initial buffering time, for example, we suppose that the values of D_a and E_v increase by ΔD_a and ΔE_v, respectively. The negative value of ΔE_v means that the fidelity improves by the increment of the buffering time, while the positive value indicates that the fidelity deteriorates. In the same way, the negative value of ΔD_a indicates the decrease of the latency, while the positive one stands for the increase of the latency. For Task 1, if $\left(1.701 \times 10^{-3} \Delta D_a + 1.510 \times 10^{-3} \Delta E_v\right)$ takes a negative value, the user-level QoS parameter of Task 1 will gain by increasing the initial buffering time. Otherwise, we should not increase the initial buffering time. Comparing Eqs. (4) and (5), we see that the absolute value of the coefficient of D_a in Eq. (4) is larger than that in Eq. (5). This means that large latency causes subjective degradation for Task 1 more than Task 2. This has been shown in Sect. 5.2.

By using Eqs. (4) and (5), we can estimate user-level QoS from application-level one. Figures 13 and 14 depict estimates of user-level QoS parameter for Task 1 and that for Task 2, respectively. To investigate user-level QoS parameter values around the peak values of them, in Figs. 13 and 14, we show the result when the initial buffering time are less than 1000 ms. From these figures, we can confirm the subjective tradeoff between fidelity and latency more clearly.

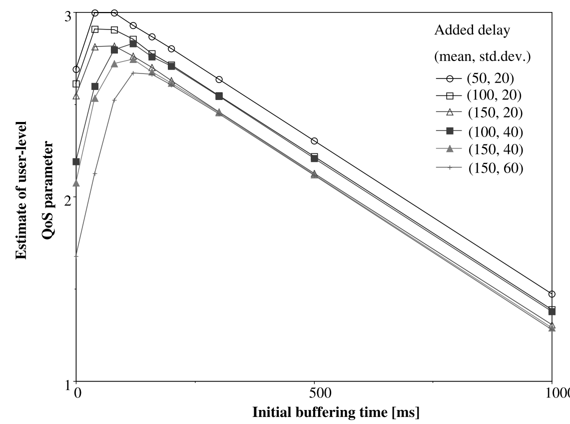

Fig. 13 Estimate of user-level QoS parameter for Task 1 versus initial buffering time.

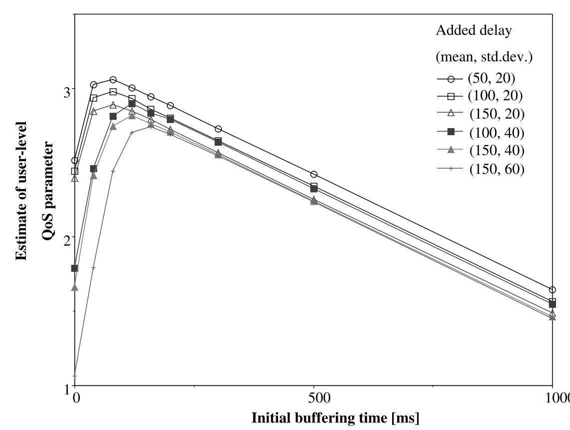

Fig. 14 Estimate of user-level QoS parameter for Task 2 versus initial buffering time.

6. Conclusions

In this paper, we investigated the subjective tradeoff between the fidelity and the latency in an interactive audio-video application. By experiment, we changed the initial buffering time while assessing the user-level QoS with the method of successive categories for two tasks. As a result, we clarified the subjective tradeoff between the fidelity and the latency. We also showed that the tradeoff is affected by the task.

We have some issues to be investigated as our future work. First, we will treat other tasks than those in this paper. Second, we will try other environments than that in this paper. Especially, we will perform experiments in actual networks, such as the Internet. Finally, it seems feasible to utilize the subjective tradeoff between the fidelity and the latency for QoS control. We will study schemes for it and implement them.

Acknowledgment

The authors thank Kouji Sonohara for his assistance in the experiment. This work was supported by the Telecommunications Advancement Foundation and by the Grant-In-Aid for Scientific Research of Japan Society for the Promotion of Science under Grant 17360179.

References

[1] S. Tasaka and Y. Ishibashi, “Mutually compensatory property of multimedia QoS,” Conf. Rec. IEEE ICC2002, pp.1105–1111, April/May 2002.

[2] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne, “Adaptive playout mechanisms for packetized audio applications in wide-area networks,” Proc. IEEE INFOCOM1994, pp.680–688, June 1994.
[3] C.J. Sreenan, J.-C. Chen, P. Agrawal, and B. Narendran, “Delay reduction techniques for playout buffering,” IEEE Trans. Multimed., vol.2, no.2, pp.100–112, June 2000.
[4] Y.J. Liang, E.G. Steinbach, and B. Girod, “Real-time voice communication over the Internet using packet path diversity,” Proc. ACM Multimedia2001, pp.431–440, Sept./Oct. 2001.
[5] N. Laoutaris and I. Stavrakakis, “Intrastream synchronization for continuous media streams: A survey of play out schedulers,” IEEE Network Mag., vol.16, no.3, pp.30–40, May 2002.
[6] T.-C. Wang, H.-C. Fang, and L.-G. Chen, “Low-delay and error-robust wireless video transmission for video communications,” IEEE Trans. Circuits Syst. Video Technol., vol.12, no.12, pp.1049–1058, Dec. 2002.
[7] M. Kalman, E. Steinbach, and B. Girod, “Adaptive media playout for low-delay video streaming over error-prone channels,” IEEE Trans. Circuits Syst. Video Technol., vol.14, no.6, pp.841–851, June 2004.
[8] I. Kouvelas and A. Watson, “Lip synchronisation for use over the Internet: Analysis and implementation,” Conf. Rec. IEEE GLOBECOM96, vol.2, pp.893–898, Nov. 1996.
[9] R. Steinmetz, “Human perception of jitter and media synchronization,” IEEE J. Sel. Areas Commun., vol.14, no.1, pp.61–72, Jan. 1996.
[10] Y. Ito, S. Tasaka, and Y. Fukuta, “Psychometric analysis of the effect of buffering control on user-level QoS in an interactive audio-visual application,” Proc. ACM Workshop on Next-Generation Residential Broadband Challenges, pp.2–10, Oct. 2004.
[11] J.P. Guilford, Psychometric Methods, McGraw-Hill, New York, 1954.
[12] Y. Ito and S. Tasaka, “Quantitative assessment of user-level QoS and its mapping,” IEEE Trans. Multimed., vol.7, no.3, pp.572–584, June 2005.
[13] S. Tasaka and Y. Ito, “Psychometric analysis of the mutually compensatory property of multimedia QoS,” Conf. Rec. IEEE ICC2003, pp.1880–1886, May 2003.
[14] F. Mosteller, “A media synchronization survey: Reference model, specification and case studies,” Psychometrika, vol.16, no.2, pp.207–218, June 1951.
[15] 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.
[16] ITU-T Rec. P.920, “Interactive test methods for audiovisual communications,” May 2000.
[17] “URL http://www.itl.nist.gov/div892/itg/carson/nistnet/”
[18] K. Fujimoto, S. Ata, and M. Murata, “Statistical analysis of packet delays in the internet and its application to playout control for streaming applications,” IEICE Trans. Commun., vol.E84-B, no.6, pp.1504–1512, June 2001.

Yoshihiro Ito received the B.S., M.S., and Ph.D. degrees from Nagoya Institute of Technology, Nagoya, Japan, in 1991, 1993, and 2002, respectively. From 1993 to 2001, he was with KDDI. In 2001, he joined Nagoya Institute of Technology, in which he is now an Associate Professor in the Department of Computer Science and Engineering, Graduate School of Engineering. His research interests include multimedia communication over the Internet and user-level QoS assessment. Dr. Ito is a member of IEEE and Information Processing Society of Japan.

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, Graduate School of 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* (MIT Press, Cambridge, MA, 1986). Dr. Tasaka is a member of IEEE, ACM and Information Processing Society of Japan.