Psychometric Analysis of the Effect of End-to-End Delay on User-Level QoS in Live Audio-Video Transmission

Yoshihiro Ito, Shuji Tasaka and Yoshihiko Fukuta
Department of Computer Science and Engineering
Graduate School of Engineering
Nagoya Institute of Technology, Nagoya 466–8555, Japan

Abstract—This paper investigates the effect of the mean and jitter of end-to-end delay on user-level QoS of live audio–video transmission. The temporal structure of continuous media can be disturbed by delay jitter of packets. The temporal disturbance causes subjective degradation of the media. On the other hand, long end-to-end delay also degrades quality of live media subjectively. Therefore, it is necessary to study the effect of the mean delay and the delay jitter on user–level QoS degradation. In this paper, we assess the effect by experiment and a psychometric method. In our experiment, we actually delay packets of live audio and video according to Pareto–normal distribution with a network emulator and assess user–level QoS of the live transmission with the method of successive categories, which is a psychometric method. As a result, we show that the standard deviation of delay is more dominant than the mean delay in the effect on the user–level QoS. We also perform QoS mapping from application–level to user–level with multiple regression analysis. From the QoS mapping, we clarify the effect of degradation of the application–level QoS caused by the mean and standard deviation of delay on the user–level QoS.

I. INTRODUCTION

Owing to diffusion of broadband access networks of the Internet and high performance terminals, many applications which treat continuous media, such as audio and video, over the Internet are becoming popular. Among them, interactive applications, such as TV conference and Internet TV chat, which transmit live continuous media, are widely used.

The interactive applications can often be degraded by the following two factors. First, the continuous media have a temporal structure. The structure can be disturbed by delay jitter of packets. The disturbance causes subjective degradation of the continuous media. Second, in live media transmission, long delay also causes subjective quality degradation. Therefore, it is necessary to investigate the effect of the mean delay and delay jitter on human subjectivity of live continuous media transmission over the Internet.

Many researchers have reported the effect of delay on subjectivity of multimedia. For example, in the psychological field, Dixon and Spitz studied the subjective tolerance of asynchrony between hearing and vision [1]. Reference [2] shows the subjective degradation of voice caused by one-way delay over the telephone network. Steinmetz investigated the tolerance of skew, which is the difference between audio delay and video one [3]. These researches, however, treated fixed delay. In the literature, we can find no quantitative study that reports the effect of delay jitter on the user–level QoS of live audio–video simultaneous transmission over packet networks.

On the other hand, by using a playout buffer, we can remedy the disturbance of the temporal structure caused by delay jitter. For example, the VTR (Virtual Time Rendering) algorithm, which adjusts the buffering time dynamically, were proposed [4], [5]. Reference [6] investigates lip synchronization with playout buffer. However, the utilization of the playout buffer produces additional end-to-end delay. To eliminate the effect of more amount of jitter, longer buffering time is required. The longer buffering time leads to longer end-to-end delay. Therefore, in order to perform buffering control for live continuous media, we must find a subjective compromise between long delay and temporal structure disturbance. As the first step of study on the subjective compromise, it is useful to investigate the effect of the mean delay and delay jitter on the human subjectivity of live audio–video transmission over packet networks.

In this paper, we treat human subjectivity of live audio–video transmission over the Internet. We refer to the subjectivity as user–level QoS. Because of a layered structure of functions of the Internet, QoS also has a layered structure. For example, reference [7] considers six levels of QoS: physical–level, network–level, network–level, end-to-end–level, application–level and user–level. The user–level QoS is perceptual one.

This paper quantitatively assesses the effect of the mean and standard deviation (i.e., jitter) of end-to-end delay on user–level QoS of live audio–video transmission over an IP network. We can regard the mean and standard deviation of delay as end-to-end–level QoS parameters in our experiment. First, in order to perform user–level QoS assessment, we utilize the method of successive categories, which is one of the psychometric methods [10]. Then, we clarify the relationship between the end-to-end–level QoS and application–level one. Moreover, by QoS mapping from application–level to user–level, we investigate the effect of application–level QoS degradation caused by the delay on user–level QoS.

The rest of the paper is organized as follows. Section II discusses the assessment of user–level QoS with psychometric methods. Section III describes a QoS mapping method with multiple regression analysis. Sections IV and V show our experiment and its results, respectively.
II. USER-LEVEL QoS ASSESSMENT

A. Assessment with psychometric methods

In order to assess user-level QoS of multimedia transmission quantitatively, the authors proposed utilization of psychometric methods [8], [9]. In the psychological field, many schemes were proposed to measure human subjectivity quantitatively. These schemes are referred to as psychometric methods [10]. In [8], the authors use the method of paired comparisons and Thurstone’s law of comparative judgment [10]. Moreover, in [9], the method of successive categories [10] is utilized.

With the psychometric methods, the human subjectivity can be represented by a measurement scale. 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 [10]. 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. With the psychometric methods used in [8], [9], we can represent user-level QoS by an interval scale. 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 to assess user-level QoS quantitatively. We will show an outline of this method in the following subsection.

B. Method of successive categories

In the method of successive categories, a subjective score is measured by the rating-scale method [10]. In the method, subjects (or observers) classify each stimulus into one of a certain number of categories. Here, a stimulus means an object for evaluation, such as audio and video. Each category has a predefined number. For example, “excellent” is assigned 5, “good” 4, “fair” 3, “poor” 2 and “bad” 1. However, since the assigned number is an ordinal scale, we cannot 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 operations to the scale.

C. 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 \( j (j = 1, \ldots, n) \) is presented to a subject, a psychological value designated by \( s_j \) occurs on 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 \( g (g = 1, \ldots, m + 1) \) by \( e_g \) and define \( e_0 = -\infty \) and \( e_{m+1} = +\infty \). The subject sorts \( n \) stimuli into the \( m + 1 \) categories \((n > m + 1)\) by comparing \( s_j \) with \( e_g \). If \( e_{g-1} \leq s_j < e_g \), 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 \( e_g \) 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 \( \sigma_j \). Then, we can consider \( R_j \) as an interval scale.

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 proposed a method of testing the goodness of fit for a scale calculated with Thurstone’s law of comparative judgment [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.

III. QoS MAPPING

A. QoS mapping with multiple regression analysis

Since we cannot control user-level QoS directly, it is desirable to control QoS at lower levels so as to keep user-level QoS high. To do this, we need to clarify the relationship between user-level QoS and QoS at lower levels. We call the relationship QoS mapping. QoS mapping is also useful for investigating lower-level QoS parameters which affect user-level QoS. As a method of QoS mapping, we proposed the utilization of multiple regression analysis [8], [9]. In this method, we consider the user-level QoS parameter as a criterion variable, and QoS parameters at lower levels as predictor variables.

In this paper, we adopt multiple regression analysis as a QoS mapping method and consider application-level QoS parameters as predictor variables since the application-level is beneath the user-level. By performing QoS mapping, we study the effect of application-level QoS degradation caused by mean delay and delay jitter (i.e., standard deviation) of packet on user-level QoS of live audio-video transmission.

B. Application-level QoS parameters

To perform multiple regression analysis, we must select some application-level QoS parameters as predictor variables. Preservation of the temporal structure of audio-video streams concerns media synchronization quality. Therefore, measures of media synchronization quality are appropriate for candidates of the predictor variables.

In general, media synchronization is classified into intra-stream synchronization and inter-stream synchronization. The former keeps the continuity of a single stream (audio or video), while the latter is synchronization between an audio stream and the corresponding video stream. In this paper, we select audio as the master stream and video as the slave stream, which is synchronized to the master.

In order to represent media synchronization quality, reference [8] uses nine application-level QoS parameters. We use
seven application-level QoS parameters out of the nine in [8]. 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 $C_v$ for audio and by $C_v$ for video. Second, we use the average MU rate for audio $R_{av}$ or that for video $R_v$, which is defined as the average number of (either audio or video) MUs output in a second at the destination. 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 $E_{av}$ for audio and by $E_{sv}$ for video. These six parameters indicate the intra-stream synchronization quality.

Note that, in [8], the MU loss rate for audio and that for video, which are the ratio of the number of lost MUs to the total number of generated MUs, are adopted. Since the MU loss rate highly correlates with the average MU rate, we do not treat the MU loss rates in this paper.

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 master and corresponding slave MUs and their timestamp difference.

In order to examine the application-level QoS degradation caused by the mean delay, in addition to the seven parameters, we evaluate the average MU delay, which is the average time in seconds from the moment an MU is generated until the instant the MU is output, and we denote it by $D_{av}$ for audio and by $D_{sv}$ for video.

IV. Experiment

A. Experimental configuration

Figure 1 shows our experimental configuration. In this configuration, two terminals each with a subject are connected with each other via a network emulator with Ethernet interfaces. The network emulator is NistNet [13], which can delay packets according to a specified probability distribution. The two terminals send audio-video streams of the subject to each other in real time. The audio-video streams are transmitted with UDP packets. Table I shows the media specification of the audio-video streams.

In this experiment, we do not exert any media synchronization control. That is, the terminals output MUs on receiving them; thus, no playout buffer is used. In this experiment, we suppose multimedia conversation, such as TV conference, as an application of live media transmission. During the experiment, it is desirable for the subjects to talk to each other continuously. Then, in order to prevent the subjects from becoming silent, we asked them to perform a simple task. In this task, the two subjects alternately count a large number as possible aloud. After the task, the subjects assessed the quality of the media by the rating-scale method.

In this experiment, we delayed packets according to Pareto–normal distribution to emulate packet delay of the Internet. Reference [14] 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 0, 50, 100, 150 and 200 msec. Also we chose 0, 5, 10, 15 and 20 msec 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 less than 0, NistNet sets the actual delay to 0. If the frequency with which the value of the random variable becomes less than 0 increases, the distribution of delay is extremely distorted compared with the expected distribution. Therefore, we set the standard deviation of delay so that it does not exceed one third of the mean. That is, when the mean of delay is 0 msec, we set the standard deviation of delay to only 0 msec. Moreover, when the mean of delay is 50 msec, we do not set the standard deviation of delay to 20 msec. Consequently, we utilized $5 \times 5 - 4 - 1 = 20$ combinations of the mean and the standard deviation.

In this experiment, we connected the two terminals via only the network emulator. Therefore, we approximately regard the mean and the standard deviation of delay set by the network emulator as end-to-end-level QoS parameters.

B. Subjective assessment

In the rating-scale method, we used 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. The subjects were non-experts in the sense that they were not directly concerned with voice and video quality as a part of their normal work. They
are men and women, and their ages were between 20 and 25. The number of subjects is 47. It took about thirty minutes per subject to finish all assessment.

V. EXPERIMENTAL RESULTS

A. User-level QoS assessment

Table II shows the number of subjects who classified the stimulus into each category by the rating-scale method.

<table>
<thead>
<tr>
<th>Mean age</th>
<th>Std. dev.</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>35</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>40</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>45</td>
<td>3</td>
<td>5</td>
</tr>
</tbody>
</table>

We translate the data measured by the rating-scale method into an interval scale with the law of categorical judgment. In the law, we can consider four conditions, conditions A, B, C and D, which differ in assumptions, approximations, and degree of simplification [11]. In this paper, we try condition D.

Let the probability that $s_j$ is less than $c_g$ be $p_{jg}$. We regard the proportion of times that $s_j$ is less than $c_g$ to the number of subjects as the observed value of the probability. Under condition D, the law of categorical judgment can be represented by

$$t_g - R_j = Z_{jg}$$

where $Z_{jg}$ is the normal deviate, which is defined as the distance from the mean of a normal distribution with a standard deviation of unity for $p_{jg}$.

In the law of categorical judgment, we assume that there exist the true values for the quantities introduced in the previous subsection and represent the corresponding observed and estimated values from observed data by attaching the prime (′) and the double prime (″), respectively, to the true values.

First, from Table II, we can get the probability $p_{jg}′$, which provides the estimated value $Z_{jg}′$. Using $Z_{jg}′$, we calculate the estimated values $t_{jg}′$ and $R_{jg}′$ for $t_g$ and $R_j$, respectively. If an observed probability $p_{jg}′$ is 0 or 1, $Z_{jg}′$ becomes negative or positive infinity. In this case, $Z_{jg}′$ is considered as a missing entry. If some missing entries exist, we have the alternative of finding $t_{jg}′$ first or $R_{jg}′$ first. We select the former here; that is, we calculate $t_{jg}′$ first and then use the obtained values to determine $R_{jg}′$.

In order to determine $t_{jg}′$, we first estimate the average width of each category. Then, we regard one boundary as the origin and calculate boundaries of the others. An estimated value $t_{jg}′ - t_{jg}″$ can be calculated by

$$t_{g+1}″ - t_{g}″ = \frac{1}{q_g} \sum_{j} q_g (Z_{jg+1}″ - Z_{jg}″)$$

where $\sum_{j} q_g$ means the summation for $j$ for which both $Z_{jg}″$ and $Z_{jg+1}″$ are available, and $q_g$ is the number of the available data for a given $g$ [9]. Table III shows $Z_{jg+1}″ - Z_{jg}″$ for $g = 1, 2$ and 3.

<table>
<thead>
<tr>
<th>Mean age</th>
<th>Std. dev.</th>
<th>$Z_{jg}″ - Z_{jg+1}″$</th>
<th>$Z_{jg}″ - z_j$</th>
<th>$Z_{jg+1}″ - z_j$</th>
<th>$Z_{jg}″ - z_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>3</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
</tr>
<tr>
<td>30</td>
<td>3</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
</tr>
<tr>
<td>35</td>
<td>3</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
</tr>
<tr>
<td>40</td>
<td>3</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
</tr>
<tr>
<td>45</td>
<td>3</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
</tr>
</tbody>
</table>

From Table III, we can calculate $t_{g+1}″ - t_{g}″$. By regarding the mean $t_1″$ of the upper boundary of category 1 as the origin of the obtained interval scale, we can obtain the mean of the upper boundary of each category. Thus, we have $t_1″ = 0.000$, $t_2″ = 1.281$, $t_3″ = 2.378$ and $t_4″ = 3.588$.

Next, $R_j″$ can be obtained by

$$R_j″ = \frac{1}{q_g} \sum_{j} t_{jg}″ - Z_{jg}″$$

where $\sum_{j} q_g$ means the summation for $g$ for which $Z_{jg}″$ is available, and $q_g$ is the number of the available data for a given $g$ [9].

Table IV shows $R_j″$ in the rightmost column. The obtained $R_j″$ is an interval scale, that is, the user-level QoS parameter.

<table>
<thead>
<tr>
<th>Mean age</th>
<th>Std. dev.</th>
<th>$Z_{jg}″ - Z_{jg+1}″$</th>
<th>$Z_{jg}″ - z_j$</th>
<th>$Z_{jg+1}″ - z_j$</th>
<th>$Z_{jg}″ - z_j$</th>
<th>$R_j″$</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>3</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
</tr>
<tr>
<td>30</td>
<td>3</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
</tr>
<tr>
<td>35</td>
<td>3</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
</tr>
<tr>
<td>40</td>
<td>3</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
</tr>
<tr>
<td>45</td>
<td>3</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
<td>1.23</td>
</tr>
</tbody>
</table>
In order to test the obtained interval scale, we perform Mosteller’s test. As a result of Mosteller’s test, the null hypothesis that the obtained internal 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.

Let us now investigate the relationship between the end-to-end QoS parameters and the obtained user-level QoS parameter. Figure 2 shows the user-level QoS parameter versus the standard deviation of delay for the five values of delay. Figure 3 shows the user-level QoS parameter versus the mean delay for the five values of the standard deviation of delay. In these figures, for convenience, we have reset the minimum value of the obtained interval scale to the origin.

![Graph](image)

**Fig. 2.** User-level QoS parameter versus standard deviation of added delay.

![Graph](image)

**Fig. 3.** User-level QoS parameter versus mean of delay.

From Figs. 2 and 3, we find the following. First, by comparing Fig. 2 with Fig. 3, we see that the increase of the standard deviation of delay causes more user-level QoS degradation than the one of the mean of delay. This implies that the standard deviation of network delay is a more dominant end-to-end QoS parameter for user-level QoS than the mean. Second, from Fig. 2, we see that the value of user-level QoS parameter largely decreases as the standard deviation increases when the standard deviation of delay exceeds 5 m sec. Third, in Fig. 3, the line of 0 m sec of the standard deviation is close to the one of 5 m sec of the standard deviation. This suggests that the subjects did not notice the increase of standard deviation of added delay from 0 m sec to 5 m sec in our task.

In this subsection, we have studied the relationship between the end-to-end QoS and the user-level one. However, the application-level QoS is considered to have more relevance to user-level QoS than the end-to-end QoS since the application-level is beneath the user-level. Therefore, in the rest of this section, we first investigate the relationship between the end-to-end QoS and the application-level one; then, we perform QoS mapping from application-level to user-level.

### B. Classification of application-level QoS parameters

In order to investigate the relationship between the end-to-end QoS and the application-level one, we first classify the application-level QoS parameters according to correlation coefficients between them. Table V shows correlation coefficients between the nine application-level QoS parameters described in Subsection III-B. From Table V, we find that the nine parameters can be classified into two groups:

**group a)** $D_a$ and $D_v$

**group b)** $R_{av}$, $R_{cv}$, $C_a$, $C_v$, $E_{av}$, $E_v$ and $E_{int}$

The parameters in the same group highly correlate with each other.

Next, we study the relationship between the mean of delay and the application-level QoS parameter and the application-level QoS parameter and one between the standard deviation of delay and the application-level QoS parameter. From Table VI, we find that the application-level QoS parameters in group a) and those in group b) highly correlate with the mean and the standard deviation of added delay, respectively.

<table>
<thead>
<tr>
<th>Table V</th>
<th>CORRELATION COEFFICIENTS BETWEEN APPLICATION-LEVEL QoS parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_{ea}$</td>
<td>$R_{ea}$</td>
</tr>
<tr>
<td>0.75</td>
<td>0.75</td>
</tr>
<tr>
<td>0.75</td>
<td>0.75</td>
</tr>
<tr>
<td>0.75</td>
<td>0.75</td>
</tr>
<tr>
<td>0.75</td>
<td>0.75</td>
</tr>
<tr>
<td>0.75</td>
<td>0.75</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table VI</th>
<th>CORRELATION COEFFICIENTS BETWEEN MEAN DELAY AND APPLICATION-LEVEL QoS parameters</th>
<th>CORRELATION COEFFICIENTS BETWEEN DELAY JITTER AND APPLICATION-LEVEL QoS parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_{ea}$</td>
<td>$R_{ea}$</td>
<td>$C_a$</td>
</tr>
<tr>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
</tr>
</tbody>
</table>
C. QoS mapping

By multiple regression analysis, we perform QoS mapping from the application-level QoS parameters to the user-level QoS parameter. That is, we consider the application-level QoS parameters and the user-level QoS parameter as predictor variables and the criterion variable, respectively. Then, we calculate a multiple regression line.

Before multiple regression analysis, we select some out of the nine application-level QoS parameters as predictor variables. In order to avoid the effect of multi-collinearity, we select one application-level QoS parameter from each group described in the previous subsection. Consequently, the number of the combination of the application-level QoS parameters becomes $2 \times 7 = 14$. In this paper, we first perform multiple regression analysis with all combinations of application-level QoS parameters as predictor variables. Then, we select a combination which indicates the highest contribution rate adjusted for degrees of freedom. Table VII shows contribution rates adjusted for degrees of freedom for the 14 combinations of the application-level QoS parameters.

<table>
<thead>
<tr>
<th>TABLE VII</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTRIBUTION RATES ADJUSTED FOR DEGREES OF FREEDOM</td>
</tr>
<tr>
<td>$D_a$</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>$R^2$</td>
</tr>
</tbody>
</table>

From Table VII, we see that two combinations, $(D_v, C_v)$ and $(D_a, C_v)$ show the highest contribution rate adjusted for degrees of freedom (0.986). Here, we select a parameter regarding audio and one concerning video. Since $C_v$ is common to the two combinations and concerns video, we choose $D_a$ as one of the two predictor variables. Figures 4 and 5 show the user-level QoS parameter versus $D_a$ and $C_v$, respectively.

![Fig. 4. User-level QoS parameter versus $D_a$.](image)

Figs. 4 and 5 indicate the relationship between the application-level QoS and the user-level one. However, Fig. 4 is similar to Fig. 3 since $D_a$ highly correlates with the mean of delay. On the other hand, Fig. 5 differs from Fig. 2 in some ways. In order to indicate the relevance of the standard deviation of delay and $C_v$ to the user-level QoS quantitatively, we calculate correlation coefficients. The correlation coefficient between the standard deviation of delay and the user-level QoS parameter is $-0.960$, while the one between $C_v$ and the user-level QoS parameter becomes $-0.983$. Therefore, the application-level QoS parameter, $C_v$, has more relevance to the user-level QoS than the end-to-end QoS parameter, the standard deviation of delay. This is because the application-level is located beneath the user-level and concerns it more than the lower levels.

Multiple regression analysis provides

$$S = 3.94 - 2.750 \times 10^{-3} D_a - 2.175 C_v$$  \hfill (4)

where $S$ is an estimate of the user-level QoS parameter. As shown in Table VII, the contribution rate adjusted for degrees of freedom of the obtained regression line is 0.986.

In order to investigate the tradeoff between the degradation caused by the mean delay and the one caused by the delay jitter, we have used the two predictor variables. However, as shown in this subsection, $C_v$ highly correlates with the user-level QoS parameter by itself; thus only $C_v$ seems to be necessary as a predictor variable. This requires us to statistically test whether $D_a$ makes a significant contribution to the multiple regression line. The result of the statistical test [15] shows that the partial regression coefficient of $C_v$ and that of $D_a$ are statistically significant. That is, both $C_v$ and $D_a$ affect the user-level QoS.

Since the scale thus obtained is an interval scale, we can select an arbitrary origin and any scale unit. Therefore, we cannot compare absolute values of the user-level QoS parameter directly. Then, we rewrite Eq. (4) into

$$S = 3.94 - 2.750 \times 10^{-3} (D_a + 7.909 \times 10^2 C_v)$$  \hfill (5)

From Eq. (5), we find that $D_a$ and $790.9C_v$ cause the same
subjective degradation. Thus the increase of $D_a$ by 79.09 msec subjectively equals the one of $C_v$ by 0.1, for example.

Moreover, using Eq. (4), we can calculate a pair of the application-level QoS parameter values which give the psychological scale value $P$ as follow:

$$C_v = 1.811 - 4.598 \times 10^{-2} P - 1.264 \times 10^{-3} D_a$$  \hspace{1cm} (6)

Figure 6 plots $D_a$ versus $C_v$ subject to a fixed $P$ when $P$ is 0, 1.0, 2.0 and 3.0. From Fig. 6, we see the tradeoff between $C_v$ and $D_a$. Considering a value of the user-level QoS parameter, we are tolerant of large $D_a$ when $C_v$ is small. For example, the user-level QoS parameters when $(D_a, C_v) = (0.1, 0.765)$ and $(D_a, C_v) = (0.5, 0.265)$ take the same value of 2.0.

It is desirable to utilize the tradeoff relation between $C_v$ and $D_a$ in buffering control for media synchronization. Media synchronization control decreases $C_v$ while $D_a$ increases; this can keep the user-level QoS high if the control is appropriate. The tradeoff relation obtained in this experiment is approximately applicable to the buffering control, when the buffering time is small. However, if it becomes large, the tradeoff relation is not applicable since the end-to-end delay distribution will change owing to the buffering control. The method of buffering control in this case is our future work.

VI. CONCLUSIONS

We quantitatively assessed the user-level QoS of live audio-video transmission with the method of successive categories by experiment. We delayed packets of the audio-video stream according to Pareto-normal distribution by using the network emulator while we measured the user-level QoS of the live audio-video transmission in the simple task with the psychometric method. As a result, we found that the effect of the delay jitter on the user-level QoS was more dominant than the one of the mean delay. Moreover, we performed QoS mapping between user-level and application-level by multiple regression analysis. From the QoS mapping, we clarified the subjective tradeoff between the application-level QoS degradation caused by the mean delay and the one caused by the delay jitter.

Some important issues are left as future work. First, in our experiment, we used one kind of task. However, the effect of the application-level QoS degradation caused by the mean delay and delay jitter on user-level QoS may depend on the kind of task. Therefore, we will investigate the subjective effect for many kinds of the tasks.

Second, it is necessary to find the subjective compromise between the long end-to-end delay and the temporal disturbance when we utilize buffering control. Although we investigated the relationship between the application-level QoS parameters in this paper, the relationship is considered to change when we use buffering control. If the relationship is clarified, we can actually perform buffering control according to the relationship so as to keep the user-level QoS high.

ACKNOWLEDGMENTS

This work was supported by the Grant-In-Aid for Scientific Research of Japan Society for the Promotion of Science under Grant 14350200.

REFERENCES