Abstract—This paper proposes a method for testing feasibility of QoS control based on QoS mapping. We focus on the issue in wireless IP networks with IEEE 802.11 MAC. In wireless multimedia applications, it is important to investigate how we enhance user-level QoS of audio–video transmission. As one of the methods, we can control QoS at lower levels so that user–level QoS keeps high by means of QoS mapping. However, there is no guarantee that a given scheme of QoS control at lower–levels can give desirable user–level QoS. Therefore, we must examine the feasibility of the QoS control. In order to test it, we introduce concepts of the feasibility space and the decision boundary, which are obtained by the QoS mapping with multiple regression analysis. To show the effectiveness of our method, we utilize the simulation results in a previous work by the authors’ research group. The simulation treats application–level QoS control and MAC–level one; we adopt EDCA of IEEE 802.11e as MAC–level control.

I. INTRODUCTION

Recently, IEEE 802.11 wireless LANs have been widely used as access networks of the Internet. In this environment, audio–video transmission services are becoming popular. However, it is not easy to keep the quality of service (QoS) high. Therefore, we need some QoS guarantee mechanisms. For that purpose, IEEE 802.11 Task Group E has standardized the enhancement of IEEE 802.11 MAC protocols.

QoS of IP networks has a layered structure. For example, reference [1] identifies six levels of QoS: physical–level, node–level, network–level, end–to–end–level, application–level and user–level. Since users are the ultimate recipient of audio–video transmission services, the user–level QoS is the most important. In other words, the final goal of QoS control is to obtain high user–level QoS.

We cannot control user–level QoS directly because it is perceptual. However, it is feasible to control QoS at lower levels so that user–level QoS can keep high. In order to do so, we must find the correspondence of user–level QoS to QoS at lower levels. The correspondence is referred to as QoS mapping. In this paper, we adopt application–level QoS as the lower–level QoS since the application–level is beneath the user–level.

In [2] and [3], the authors utilize multiple regression analysis as a QoS mapping method. They regarded application–level QoS parameters and user–level one as predictor variables and the criterion variable, respectively. As a result of multiple regression analysis, we have obtained multiple regression lines for QoS mapping. By using the obtained lines, we can estimate values of the user–level QoS parameter from the application–level ones.

The multiple regression line is useful for QoS control. We can utilize it in order to adjust application–level QoS parameters so that user–level QoS can keep high.

QoS control can be exerted at many levels: application–level, end–to–end–level, and so on. In addition, the QoS control at each level interacts with each other. Therefore, it is not easy to find a feasible combination of application–level QoS parameters which gives a desirable user–level QoS parameter value. In some cases, there may exist no combination of application–level QoS parameters which can achieve a target value of the user–level QoS parameter. Therefore, it is necessary to test feasibility of QoS control based on QoS mapping. The feasibility in this paper means that we can obtain a set of application–level QoS parameters which give a desirable value of the user–level QoS parameter. In the literature, we can find no paper which treats the feasibility of the QoS control.

In this paper, we propose a method of feasibility test for QoS control based on the mapping to user–level QoS in IEEE 802.11 wireless LANs. In this method, we utilize a feasibility space and a decision boundary. In the feasibility space, we can decide the feasibility of QoS control intuitively. By simulation, we show the effectiveness of the proposed method. The simulation treats QoS control at two levels in an IP network: application–level and MAC–level. As MAC–level QoS control, we utilize EDCA (Enhanced Distributed Channel Access) of IEEE 802.11e [4].

The rest of the paper is organized as follows. Section II presents a method of testing the feasibility. Section III describes application–level QoS parameters. Section IV explains the assessment of user–level QoS. Sections V and VI show our assessment method and numerical results, respectively. Finally, we conclude our paper in Section VII.

II. TEST OF FEASIBILITY

A. QoS mapping with multiple regression analysis

We explain the method of QoS mapping with multiple regression analysis, which is proposed in [3]. In this method,
we consider the user–level QoS parameter as the criterion variable, and application–level QoS parameters as predictor variables. An estimate $\hat{S}$ of the user–level QoS parameter can be expressed with $n$ application–level QoS parameters $A_1, A_2, \cdots, A_n$ by

$$\hat{S} = \beta_0 + \beta_1 A_1 + \cdots + \beta_n A_n$$

(1)

where $\beta_i$ (1 ≤ $i$ ≤ $n$) is the partial regression coefficient of the $i$-th application–level QoS parameter, and $\beta_0$ is the intercept.

B. Selection of predictor variables

Before multiple regression analysis, we must select a certain number of application–level QoS parameters as predictor variables of multiple regression analysis. To do this, at the beginning, we present some candidates for application–level QoS parameters and classify them by the principal component analysis.

We show the procedure for selection of predictor variables in the following. First, as a result of the principal component analysis, we obtain the cumulative contribution rate for each principal component. According to it, we can decide the number of the principal components which we should consider. Second, we calculate the principal component loading of each principal component for all the application–level QoS parameters. With the value of the principal component loading for each application–level QoS parameter, we can classify the application–level QoS parameters into some groups. Third, to avoid the effect of multi–colinearity, we select one application–level QoS parameter from each group as predictor variables. As a result, we can obtain some combinations of application–level QoS parameters. Then, we treat all the obtained combinations of the application–level QoS parameters as predictor variables and perform multiple regression analysis; we select the combination which indicates the highest contribution rate adjusted for degrees of freedom. Finally, we carry out multiple regression analysis with selected predictor variables again. By the statistical test [5], we remove predictor variables which give little contribution to the multiple regression line.

C. Method for feasibility test

As described in the previous subsection, we can select a certain number of application–level QoS parameters as predictor variables. From the selected application–level QoS parameters, we can estimate the user–level QoS parameter value.

Let us imagine an $n$–dimensional space whose axes represent predictor variables; $n$ is the number of predictor variables. We refer to the $n$–dimensional space as the feasibility space. In this space, the combinations of application–level QoS parameter values which make the user–level QoS parameter value constant are expressed as a set of plots. In general, they become an $(n–1)$–dimensional space. For example, if we adopt three predictor variables, the combinations can be expressed as a 2–dimensional plane in the 3–dimensional space. In this paper, the $(n–1)$–dimensional space which consists of the set of plots is referred to as the decision boundary.

We can test the feasibility of a given scheme of QoS control in the feasibility space. If we adopt a certain number of decision boundaries, the feasibility space can be divided into regions whose number equals to the one of decision boundaries. Then, we measure values of the predictor variables with the QoS control scheme and plot them in the feasibility space. Then, by inspecting the feasibility space, we can test the feasibility of the QoS control scheme.

III. APPLICATION–LEVEL QOS MEASUREMENT

In this paper, we regard measures of media synchronization quality of audio–video transmission as candidates of the predictor variables. In order to represent media synchronization quality, reference [6] defines nine application–level QoS parameters. We also employ them in this paper. First, we use 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”; it indicates the information unit for media synchronization. This parameter is denoted by $C_a$ for audio and by $C_v$ for video. Second, we utilize the average MU rate for audio $R_a$ and that for video $R_v$; this is defined as the average number of (either audio or video) MUs output in a second at the destination. Third, we treat the MU loss rate for audio $L_a$ and that for video $L_v$; this is the ratio of the number of lost MUs to the total number of generated MUs. Fourth, we adopt 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 $E_{\text{in}}$ for audio and by $E_{\text{in}}$ for video. Finally, we treat mean square error of inter–stream synchronization $E_{\text{int}}$, 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.

IV. USER–LEVEL QOS ASSESSMENT

To carry out QoS mapping, we must express user–level QoS quantitatively. The authors proposed the utilization of the method of successive categories for quantitative assessment of user–level QoS in audio–video transmission [2]. The method is one of the psychometric methods, which were proposed to measure human subjectivity quantitatively in the psychological field [7].

Before we introduce the method of successive categories, we must know four types of measurement scales. 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 [7]. 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 general, however, it is difficult to obtain a ratio scale.

The method of successive categories can obtain an interval scale. In the method, a subjective score is measured by the rating–scale method [7], in which subjects 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 often has a predefined number to calculate MOS (Mean Opinion Score). For example, “excellent” is assigned 5, “good” 4, “fair” 3, “poor” 2 and “bad” 1. However, in the strict sense, we cannot use the assigned number for assessing the user–level QoS since the assigned number is an ordinal scale.

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. The utilization of the law of categorical judgment [8] can translate the frequency obtained by the rating–scale method into an interval scale.

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 \( c_g \) and define \( c_0 = -\infty \) and \( c_{m+1} = +\infty \). The subject sorts \( n \) 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 \( \sigma_j \). 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 [9]. The method can be applied to a scale obtained by the law of categorical judgment [7]. In this paper, we use Mosteller’s method to test the goodness of fit.

By utilizing the method of successive categories, we can also obtain the boundary values of all categories. These values are useful in deciding the criterion of the user–level QoS parameter value.

V. ASSESSMENT METHOD

A. Simulation model

In this paper, we utilize the simulation results in [10], where QoS of audio–video transmission in a wireless IP network is evaluated by a computer simulation. Reference [10] utilizes EDCA of IEEE802.11e as MAC–level QoS control; it also exerts VTR (Virtual Time Rendering) media synchronization control [11][12][13] and dynamic video resolution control at the application–level. To simplify our discussion, this paper adopts EDCA and VTR as QoS control methods.

Figure 1 illustrates the network model in [10]. It is simulated by network simulator 2 [14]. This network consists of an access point (AP), four multimedia stations and five data stations. The channel rate is fixed at 11 Mb/s. The AP transmits a pair of audio and video streams to each of the four multimedia stations. The audio and video are transmitted as separate transport streams by using UDP/IP. Table I presents the specifications of the audio and video in the simulation. The content of audio and video is a clip of a music video. While the AP transmits a pair of audio and video streams, the five data stations generate fixed–size UDP datagrams of 1472 bytes in its payload at exponentially distributed intervals and send them to the AP as load traffic. The average amount of load traffic is set to 500, 600, 700 and 800 kb/s. We show the parameter values of the EDCA used in the simulation in Table II.

<table>
<thead>
<tr>
<th>Sample data</th>
<th>Audio</th>
<th>Video</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coding scheme</td>
<td>G.711</td>
<td>MPEG1</td>
</tr>
<tr>
<td>Image size [pixels]</td>
<td>320×240</td>
<td></td>
</tr>
<tr>
<td>Picture pattern</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Average MU rate [MU/s]</td>
<td>8</td>
<td>20</td>
</tr>
<tr>
<td>Average MU interval [msec]</td>
<td>125</td>
<td>50</td>
</tr>
<tr>
<td>Average bit rate [kb/s]</td>
<td>64</td>
<td>800</td>
</tr>
<tr>
<td>Simulation time [sec]</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>
Reference [10] compares the performance of EDCA with that of DCF (Distributed Coordination Function) [15], which is designed for asynchronous data transmission with CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). In the case of the DCF, CW_{min} and CW_{max} are set to 31 and 1023, respectively. The duration of a slot, the value of DIFS (DCF InterFrame Space) and that of SIFS (Short InterFrame Space) equal 20\mu s, 50\mu s and 10\mu s, respectively. The maximum allowable number of retransmissions of an MPDU (MAC Protocol Data Unit) is set to 7. Each source buffer at the MAC layer in a station or the AP can have a maximum of 50 MPDUs. If the station or the AP generates a new MPDU and if its buffer does not have space to accommodate the MPDU, the MPDU is discarded. In the simulation, the RTS(Request To Send)/CTS(Clear To Send) frames are not exchanged before transmission of an MPDU, and only one MPDU is transmitted per TXOP (Transmission Opportunity) in the EDCA.

Each multimedia station plays out received audio and video with/without VTR media synchronization control as an application-level QoS control method. If a multimedia station utilizes VTR, its initial buffering time for the media synchronization \( J_{\text{max}} \) is set to 200 ms.

The reader is referred to [10] for further details of the simulation.

### B. Subjective assessment

In [10], user-level QoS is assessed with the method of successive categories. In the rating–scale method, reference [10] uses 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 in the user-level QoS assessment were non-experts in the sense that they were not directly concerned with audio and video quality as a part of their normal work. Their ages were 20s. The number of subjects is 15.

### VI. Numerical results

#### A. User-level QoS assessment

In this paper, we utilize the simulation result in [10] except for that of the dynamic video resolution control. Therefore, we assess user-level QoS, using the result in [10] again.

Table III shows the number of subjects who classified the stimulus into each category by the rating–scale method. In Table III, NC (No Control) means that VTR is not carried out. 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 [8]. In this paper, we tried condition D. As a result, we obtained the upper boundary of each category. For convenience, we set the minimum scale value obtained to the origin of the interval scale. Thus, we have \( t_1' = 0.842 \), \( t_2' = 1.964 \), \( t_3' = 3.016 \) and \( t_4' = 4.727 \), where \( t_g' \) means the value of \( t_g \) estimated from the experimental result.

In order to test the obtained interval scale, we 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 regarded the obtained scale as the user-level QoS parameter. Table IV shows the obtained user-level QoS parameter values.

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

Table V shows correlation coefficients between the nine application-level QoS parameters. By using the correlation coefficients in Table V, we carried out the principal component analysis. As a result, we see that the cumulative contribution rate for the first three principal components becomes 90.8\% . This means that the first three principal components
can present 90.8% of information involved by the nine application-level QoS parameters. Therefore, we adopt the first three principal components. Table VI displays the principal component loading of the each principal component.

![Image of Table VI](image)

From Table VI, we find that the nine parameters can be classified into four groups:

- **group a** $C_a$, $E_a$, $L_a$, $R_a$, $C_v$, $E_v$, $L_v$ and $R_v$
- **group b** $E_{int}$
- **group c** $D_a$ and $D_v$
- **group d** $E_a$

The parameters in group a) and those in group c) highly correlate with the first principal component and the third one, respectively. The parameter in group b) correlates with the second principal component. On the other hand, the parameter in group d) slightly correlates with the first and third principal components.

### C. QoS mapping

According to the result of the principal component analysis, we select some out of the nine application-level QoS parameters as predictor variables. We selected 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 $7 \times 1 \times 2 \times 1 = 14$.

We performed multiple regression analysis of all the combinations as predictor variables. Table VII presents contribution rates adjusted for degrees of freedom. Table VIII and Table IX show the combination $(C_v, D_a, E_a, E_{int})$ and $(C_v, D_v, E_a, E_{int})$, show the highest contribution rate adjusted for degrees of freedom. According to the result of the statistical test, we find that $D_a$, $D_v$ and $E_{int}$ do not make any significant contribution to the multiple regression lines. Therefore, we adopt the combination $(C_v, E_a)$ as predictor variables.

The obtained multiple regression line becomes

$$
\hat{S} = 4.971 - 3.513C_v - 1.467 \times 10^{-3} E_a
$$

(2)

where $\hat{S}$ is an estimate of the user-level QoS parameter. The contribution rate adjusted for degrees of freedom of the obtained regression line is 0.87. As a result of removing $D_a$, $D_v$ and $E_{int}$, the contribution rate adjusted for degrees of freedom has improved by only 0.01.

We now investigate measured values of $C_v$ and those of $E_a$ in detail. As a result, we see that the values of $E_a$ distribute very widely. Then, we used $\sqrt{E_a}$ as a predictor variable instead of $E_a$, it produces

$$
\hat{S} = 5.214 - 3.501C_v - 7.012 \times 10^{-2}\sqrt{E_a}
$$

(3)

The contribution rates adjusted for degree of freedom of Eq. (3) becomes 0.93. This values is higher than that of Eq. (2). Therefore, we adopt Eq. (3) for QoS mapping.

From Eq. (3), we can estimate the user-level QoS parameter from two application-level QoS parameters: $C_v$ and $\sqrt{E_a}$. Inversely, we must find a pair of $(C_v, \sqrt{E_a})$ whose values give a desirable value of the user-level QoS parameter, using Eq. (3).

### D. Result of feasibility test

Since we obtained two predictor variables as described in the previous subsection, we treat a two-dimensional feasibility space. Consequently, the decision boundary becomes a single-dimensional space, that is, a line. In the feasibility space, one axis is $C_v$, and the other is $\sqrt{E_a}$.

From Eq. (3), we can obtain some decision boundaries which give target user-level QoS parameter values. Eq. (3) can be rewritten as

$$
\sqrt{E_a} = 7.436 \times 10 - 1.426 \times 10 P - 4.993 \times 10 C_v
$$

(4)

where $P$ is a desirable user-level QoS parameter value. If we plot $C_v$ versus $\sqrt{E_a}$ over the feasibility space subject to a fixed value $P$, we can show the decision boundary for $P$ in the feasibility space. We set the values of $P$ to 0.842, 1.964, 3.016 and 4.727, which are the average values of the upper boundary of categories 1, 2, 3 and 4 obtained in subsection VI-A. Each region separated by the lines (the decision boundaries) means that the plot in the region gives one of the user’s subjective decisions: “imperceptible”, “perceptible, but not annoying”, “slightly annoying”, “annoying” and “very annoying”.

![Image of Table VII](image)
IP networks, such as a DiffServ network. We will try other network environments. Second, we must consider implementation of the proposed method. For example, a pair of terminals sends the result of the feasibility test at the start of the communication to each other. According to the feasibility, they can control the QoS effectively. SIP can be utilized as a method of notification of the feasibility.

ACKNOWLEDGMENTS

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