

QUVE: QoE Maximizing Framework for Video-Streaming

Takuto Kimura, Masahiro Yokota, Arifumi Matsumoto, Kei Takeshita, Taichi Kawano, Kazumichi Sato, Hiroshi Yamamoto, Takanori Hayashi, Kohei Shiimoto, *Senior Member, IEEE*, Kenichi Miyazaki,

Abstract— As video-streaming services for mobile terminals are becoming more popular, the volume of mobile traffic is growing rapidly. To deliver traffic-heavy network services without degrading users' overall experience, one must examine service characteristics and end-to-end network conditions. In this paper, we propose "QUVE," a framework for maximizing the user's quality of experience (QoE) of video streaming services. The QUVE framework consists of two key components: a QoE estimation model and QoE parameter estimation method. The QoE-estimation model is based on the rebuffering count and time, and content-encoding conditions. The QoE parameter-estimation method estimates forthcoming network quality and the corresponding rebuffering count and time that the user will experience. The effectiveness of this framework was demonstrated through a large-scale field trial for Niconico video service, one of the most popular video-streaming services in Japan. We gathered more than 1.4 billion pieces of feedback data from the in-service trial and found that our framework enhances user QoE by selecting the best encoding conditions suited for user network conditions.

Index Terms—Quality of Experience, video streaming service, QoE maximization, QoE estimation, network quality estimation, rebuffering condition estimation.

I. INTRODUCTION

VIDEO-streaming services have been becoming popular. In 2014, the network traffic of such services comprised 64% of all Internet traffic [1]. Some services provide the choice of very high bitrate content among other choices, such as tens of Mbps, that cannot be delivered over a relatively narrow-band network. In particular, in mobile environments, network bandwidth is relatively narrow, and its variation is higher than that of fixed-line environments due to the fluctuation in wireless conditions. Therefore, a serious problem for users and content providers is that viewing video content in a mobile network environment incurs bitrate degradation and a video playback interruption called rebuffering.

To address this problem, adaptive bitrate streaming (ABR) methods [2], [3] have been recently standardized [4] and implemented. With an ABR method, a content server has video data with several encoding conditions, such as bitrate, framerate, and resolution, and is divided into fragments of a couple of seconds called "chunks." An ABR client retrieves a lower quality and smaller chunk when the network throughput is low to prevent rebuffering, and it retrieves a higher quality and larger chunk when the network throughput is high.

However, the ABR method incurs initial and operational costs for preparing several ABR-compatible encoding conditions of content data and modifying the client player application. Thus, some video-streaming services still adopt a constant bitrate (CBR) based video delivery method. For example, Niconico video service [5], one of the most popular online video services in Japan, delivers videos with a CBR-based method.

Therefore, it is also important to enhance user's quality of experience (QoE) for CBR streaming services without replacing the existing equipment and content data. For this reason, we focused on QoE enhancement for CBR streaming services.

There have been several studies focused on techniques for enhancing video-streaming QoE. For example, several papers [6], [7] aim to enhance a single QoS or KQI, such as the rebuffering ratio and or video bitrate. However, video-streaming QoE does not depend on such a single parameter; it is known that multiple parameters contribute to QoE [8], [9]. Therefore, maximizing such a KQI parameter does not maximize QoE. Maximizing user engagement, which can be measured from the content-viewing time of each user, has been investigated [10], [11]. An engagement index depends on various factors, such as video quality, content quality, category, and service fee structure. Therefore, maximizing engagement does not necessarily maximize QoE.

In this paper, we propose "QUVE," a QoE maximizing framework for video-streaming. The QUVE framework recommends the encoding conditions to a streaming server or a user that maximize user QoE in individual content viewing. There are two challenges to implementing this framework:

- 1) establishing a QoE-estimation model for CBR video-streaming
- 2) estimating the parameters needed for the model

We studied and developed solutions to these challenges for QUVE and demonstrated it in a large-scale field trial for Niconico video service.

The remainder of this paper is organized as follows. We discuss the design space in Sec. II, and describe QUVE in Sec. III. In Sec. IV, we explain the QoE-estimation model, and in Sec. V, the QoE parameter-estimation method we developed for the framework. We explain the results of a field trial in Sec. VI and discuss related studies in Sec. VII. We conclude the paper in Sec. VIII.

II. DESIGN SPACE

In this study, we aimed to maximize QoE for CBR video-streaming services in the mobile environment. In a mobile environment, network bandwidth is relatively narrow, available

T. Kimura and M. Yokota and A. Matsumoto and K. Takeshita and T. Kawano and K. Sato and H. Yamamoto and T. Hayashi and K. Shiimoto are with NTT Network Technology Laboratories, NTT Corporation, 9-11, Midori-Cho 3-Chome Musashino-Shi, Tokyo, 180-8585 Japan

K. Miyazaki is with Dwango Co., Ltd, KABUKIZA TOWER., 15-12-4, Ginza Chuo-ku, Tokyo, 104-0061 Japan

network throughput varies in a short period, and is largely biased depending on the geographical location. It is also important that the solution is applicable to existing facilities and is effective. While taking these conditions into account, we first discuss the design space for this goal by exploring three natural dimensions: what, how, and where.

A. What

As mentioned in Sec. I, we adopted QoE as a measurement index, which is usually modeled using the above-mentioned indexes of QoS and KQI as parameters and is used as an objective function in a quality-maximizing method. The challenge is how to establish a QoE model that matches real user experience.

B. How

There are three possible and currently feasible methods for controlling video-streaming quality: streaming-server selection, encoding condition selection, and QoS control at network elements.

A server-selection based method was demonstrated [10]. This method assumed that content is available from multiple content delivery network (CDN) providers and that the end-to-end network paths between a client device and servers have different communication quality, that is, these paths are disjoint. However, for a star topology network, like the network in Japan, this assumption does not always hold true. Consequently, the applicability of this method is narrow.

The selection of encoding conditions seems to be an implementable method for common video streaming services. The servers are assumed to offer multiple encoding conditions of content data for user convenience. In particular, the bitrate value for receiving content is closely linked to the required network quality. Therefore, it would be a huge benefit to user QoE if we could select the encoding conditions suited for the network conditions.

Quality of service (QoS) control at network elements can be implemented in several ways. The ECN protocol [12] can notify a client or server of network congestion. However, the availability of this protocol is limited [13]. *Aquarema* [14], [15] is a framework of QoS control at network elements to enhance video service QoE. To introduce *Aquarema* to existing mobile equipment, a resource controlling mechanism has to be implemented at mobile base stations. There is no widely available mechanism that can be triggered from outside the mobile carrier.

Particularly, in a mobile network, the network condition changes dramatically depending on time and location. Therefore, it may be effective to change a streaming server or encoding condition in the midst of video streaming to meet the change in available network bandwidth. However, this involves non-trivial reconstruction of a client, server implementation, and possibly re-encoding of the prepared video data.

In summary, the above three methods for video-streaming quality control seem to be valid from the aspect of effectiveness. In contrast, from the aspect of applicability, only the server-selection and encoding-condition-selection methods

are feasible in the current Internet. In addition, the encoding-condition-selection method seems to have higher applicability than the streaming-server-selection method because it does not require multiple distributed servers or multiple CDNs. Therefore, we selected an approach to select appropriate encoding conditions from the perspectives of applicability and effectiveness.

C. Where

There are three possible locations to implement the encoding-condition-selection method mentioned above; in a server, client, or in-between network elements.

In the server approach, many client devices usually communicate to a server; thus, the server can obtain a large amount of network quality information from these devices. However, it is often the cases that a content provider deploys multiple servers, resulting in fragmentation of quality information.

In the client approach, a client device has very limited data on network quality, so it will exhibit at a particular location. In particular, a mobile device will exhibit differing network quality at different locations and at different times of day. Therefore, it is difficult to predict future network quality.

In the in-between network element approach, a network element selects an appropriate encoding condition and applies it by transcoding the video content in a product (see [16]). However, it is not realistic to deploy this kind of network element in every end-to-end path. Furthermore, if the video streaming packets are encrypted or transferred over HTTPS, transcoding at an in-between element is not feasible.

From these considerations, we adopt the server approach and avoid the fragmentation of quality information by splitting quality-related functions from content servers to a dedicated server. The dedicated server gathers network quality information from every client and estimates and recommends the best encoding conditions by analyzing the gathered quality information for each user request.

III. QUVE FRAMEWORK

On the basis of the design space discussion, we propose a framework called QUVE, which aimed to maximize each view's video streaming QoE by selecting the best encoding condition.

A. Overview

A video-streaming sequence using QUVE is illustrated in Fig. 1. The QUVE server is located in the Internet, and is split from streaming servers for ease of quality data aggregation. The QUVE server can also be managed by different providers than those of the streaming servers.

The details of the sequence are given below.

- 1) When a client user begins to view video content, the user's device sends a viewing request to a streaming server attached with the device's network information. The network information, consisting of a mobile carrier identifier and cell identifier (CID) of a mobile base station, is used to identify the user location in the network,

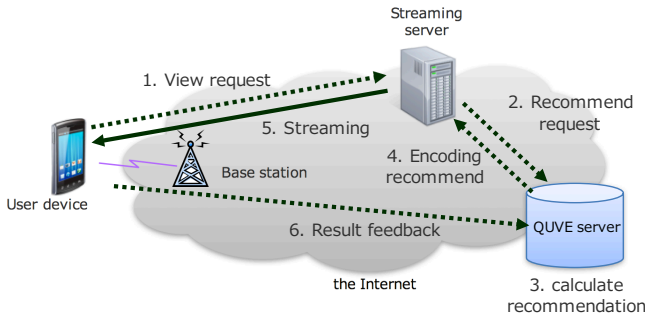


Fig. 1: Video streaming with QUVE

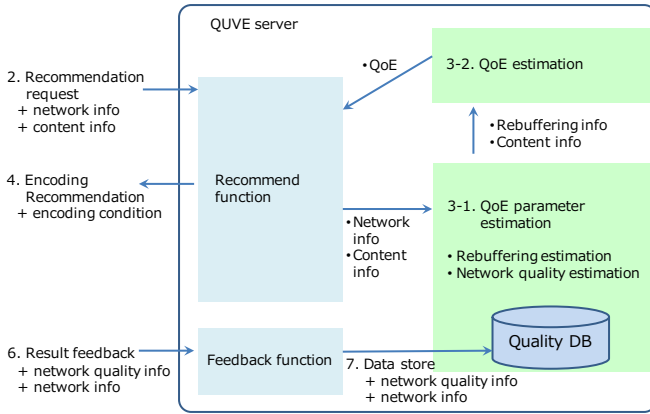


Fig. 2: QUVE server

- 2) The streaming server sends a recommendation request to the QUVE server that includes the network information received from the user device and content specific information. The content specific information means the encoding condition choices of the requested content that are prepared at the streaming server.
- 3) The best encoding condition is calculated in the QUVE server.
- 4) The QUVE server returns the best encoding condition to the streaming server.
- 5) Streaming starts with the content of the recommended encoding condition.
- 6) Upon viewing completion, the user device feeds back the experienced network quality and network location to the QUVE server.

To introduce QUVE to existing video-streaming client and server equipment, user devices and streaming servers have to implement the following additional functions. User devices have to retrieve and send their network information when starting to view content, measure network throughput while downloading the video data, and feed back the throughput results upon viewing completion. The streaming servers have to implement the function to query encoding condition recommendations to the QUVE server providing the user network information and choices of the encoding conditions of the requested content.

B. QUVE server

Upon receiving a recommendation request, the QUVE server calculates the QoE value for each encoding condition that the user will experience if the user selects the encoding condition. To estimate the QoE of the video, previous studies [8], [9] claim that the encoding and rebuffering conditions are the key influencing factors. The encoding condition is received from a streaming server, but the rebuffering condition cannot be known before watching the video because it depends on network quality.

Based on these insights, Fig. 2 illustrates the calculation sequence of the best encoding condition and feedback sequence inside the QUVE server.

- 3-1) We estimate the network quality for a requesting user using the network information provided by the user. Using the estimated network quality, we further estimate the corresponding rebuffering condition for each encoding condition of the requested video content.
- 3-2) By combining the estimated rebuffering and encoding conditions, the user QoE for each encoding condition is calculated. The encoding condition that maximizes the QoE is returned to the streaming server.
- 7) Upon viewing completion, the user device feeds back the pair of experienced network quality and network-location information to the QUVE server. The data are stored in the quality database, which helps improve accuracy for future network-quality estimation.

The above mentioned recommendation sequence has to be performed before video streaming begins. Thus, the overhead of processing and data transmission increases a user's waiting time for video playback. We also evaluated the overhead delay and discussed its impact on user QoE in the field trial section.

C. Challenges

The QUVE framework consists of two key components. One is the QoE-estimation model and the other is the QoE parameter-estimation method required for QoE calculation. The challenges of designing each component are described below.

1) *Designing QoE-estimation Model:* Studies on QoE estimation models for CBR video-streaming have been conducted [17]–[19]. However, these models are constructed for different resolution condition. Hence, we cannot compare the QoE values of a video that has multiple resolutions. In addition, these models do not support content longer than 10 seconds. Therefore, we constructed a new QoE estimation model, as discussed in Sec. IV.

2) *Designing QoE Parameter Estimation Method:* As mentioned above, the rebuffering and encoding conditions are necessary for video-streaming QoE calculation. The rebuffering conditions, such as the rebuffering count (RC) and rebuffering time (RT), are determined by the content bitrate, time series of available network throughput, and buffering behavior in

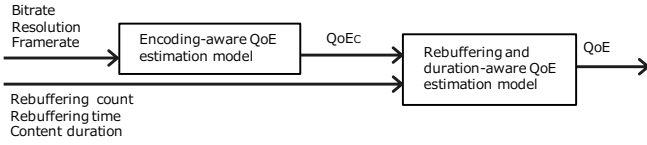


Fig. 3: Overview of our QoE-estimation model

a client device. The challenge here is to estimate the parameters, such as network quality and rebuffering conditions. Liu et al. [10] used past network-quality and rebuffering information to estimate future network quality and future rebuffering conditions. They used this information to select the best streaming server and proved the effectiveness of this approach. Hence, we follow the same approach of using past network-quality information obtained at a client device. The challenge is parameter estimation in a mobile environment. A mobile environment is negatively affected by a large amount of temporal variation and locational bias in network quality. Therefore, we needed to develop a method for estimating the network and rebuffering conditions for each user environment with high accuracy. This QoE parameter estimation method is described in Sec. V.

IV. QOE-ESTIMATION MODEL

In this section, we describe our QoE estimation model for CBR-based video streaming.

To estimate the QoE of video, previous studies [8], [9] claim that the encoding and rebuffering conditions are the key influencing factors. Based on this insight, ITU-T Rec. G.1071 [19] has been proposed. However, G.1071 has two problems when we adapt it to QUVE: we cannot compare the QoE values of a video that has different resolutions, and we cannot apply it to a video of less than 10 sec. Therefore, we conducted a subjective quality assessment test to construct a new QoE-estimation model that can handle a wider range of encoding parameters, rebuffering conditions, and content duration (*Dur*).

Figure 3 shows the overview of our QoE-estimation model. First, we constructed an encoding-aware QoE-estimation model based on content-encoding-related information, such as bitrate, framerate, and resolution. Second, we constructed a rebuffering and duration-aware QoE-estimation model. By combining these two QoE-estimation models, we constructed our QoE-estimation model for our framework, which solves the above problems.

A. Subjective Quality Assessment Conditions

The conditions of the subjective quality assessment test used to construct our QoE-estimation model are described below. These experimental conditions were determined based on ITU-T Rec. P.910 [20].

In the test, video quality was assessed using a five-point absolute category rating (ACR) method. Participants watched a video with a five-inch smartphone and assessed it on a five-point scale (1: Bad, 2: Poor, 3: Fair, 4: Good, 5: Excellent).

We prepared 8 different types of video content for the assessment such as “nature,” “sports,” “animation,” etc. Each

TABLE I: Test Conditions

Encoding condition	Values
Codec	H.264
Bitrate	100, 200, 370, 500, 700, 1000, 1600, 3200, 6400, 12800 kbps
Resolution	320x180, 480x270, 640x360, 960x540, 1280x720
Framerate	10, 15, 30 fps
Content duration	10, 30, 60, 120, 210 sec
Rebuffering time	0, 4, 8, 16 sec
Rebuffering count	0, 1, 2, 4, 6, 7 times

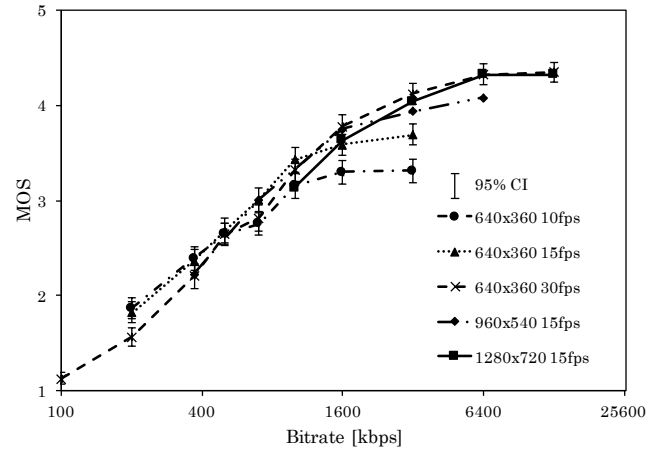


Fig. 4: MOS difference depending on encoding conditions

video was encoded with a variety of encoding conditions. Table I lists the encoding conditions used in the test. We used the experimental design to reduce the number of experiments and reduced these conditions to 138.

Twenty-four people aged 20–29, male and female, participated in this experiment. They were non-experts, not concerned with video quality assessment as part of their work, and not experienced assessors.

B. Encoding-aware QoE-estimation Model

In this subsection, we discuss the construction of the encoding-aware QoE-estimation model. Here, we focus on the MOS difference in encoding conditions but not that in rebuffering conditions or *Dur*. To construct this QoE model, we extracted 75 of the test conditions described in the above subsection. All the conditions had a fixed content length of 10 sec and had no rebuffering events.

To examine the characteristics of the model, we first analyzed the mean opinion score (MOS) difference depending on the encoding conditions. Figure 4 shows the results of the subjective quality assessment test. The x-axis is the content bitrate and the y-axis is the MOS, and lines are drawn for each condition of resolution and framerate.

Figure 4 shows the following four characteristics.

- 1) the MOS followed an S-curve based on the bitrate
- 2) the minimum MOS score was 1, and the MOS increased as the bitrate increased

- 3) the maximum MOS score is defined by resolution and framerate
- 4) the inflection point is defined by resolution and framerate

Considering characteristics 1, 2 and the existence of the inflection point, we used a logistic model to construct the encoding-aware QoE-estimation as Eq. (1).

$$QoE_C = I_1 + \frac{1 - I_1}{1 + \left(\frac{bitrate}{I_2}\right)^{i_1}} \quad (1)$$

Coefficient I_1 in Eq. (1) means the maximum value of QoE_C . This value depends on framerate and resolution, as mentioned in characteristic 3. In addition, QoE_C ranges from 1 to 5 according to the MOS definition [21]. Considering these characteristics, we constructed a model to calculate I_1 from the resolution and framerate. Figure 5a shows the relationship between the resolution and I_1 , and shows the following two characteristics.

- 1) I_1 increased as the resolution increased.
- 2) The increase rate decreased as the resolution increased.

Therefore, we constructed another I_1 model using a hyperbolic curve model, as Eq. (2).

$$I_1 = \frac{I_3 \cdot rs}{i_2 + rs} + 1 \quad (2)$$

Next, to take into account the framerate, we constructed a model to calculate I_3 in Eq. (2) using the framerate. Figure 5b shows the relationship between I_3 and framerate, and I_3 had the following characteristics.

- 1) I_3 became 0 when the framerate was 0
- 2) It increased as the framerate increased, and converged to 4.
- 3) The I_3 increasing rate reduced when the framerate became higher.

On the basis of these characteristics, we modeled I_3 as Eq. (3).

$$I_3 = 4 \cdot (1 - \exp(-i_3 \cdot fr)) \quad (3)$$

Next, we constructed a model for coefficient I_2 in Eq. (1). The I_2 indicates the x-axis value at the inflection point, and it varies depending on framerate and resolution, as detailed in characteristic 4. We followed the same manner as for I_1 to construct the model for I_2 . Figure 5c shows the relationship between resolution and I_2 and shows that

- 1) I_2 became 0 when the resolution was 0.
- 2) It increased as resolution increased.
- 3) The I_2 increasing rate reduced when the resolution reached a certain value.

On the basis of these characteristics, I_2 is modeled as Eq. (4).

$$I_2 = \frac{i_4 \cdot rs + I_4}{1 - \exp(-i_5 \cdot rs)} \quad (4)$$

To take into account the framerate in Eq. (4), we modeled I_4 using the framerate. Figure 5d shows the relationship of I_4 and framerate and shows that

- 1) I_4 became 0 when the framerate was 0.
- 2) it increased as the framerate increased.
- 3) the I_4 increasing rate reduced when the framerate became higher.

On the basis of these characteristics, we used the logarithmic model to model I_4 as Eq. (5).

$$I_4 = i_6 \cdot \log(i_7 \cdot fr + 1) \quad (5)$$

Parameters i_1 – i_7 can be determined and optimized by applying the results of the subjective quality assessment tests.

C. Rebuffering-aware QoE-estimation Model

We conducted a subjective quality assessment test to extend our encoding-aware QoE-estimation model to handle the difference in rebuffering conditions. In this test, under the test conditions described in Table I, we changed the encoding and rebuffering conditions with fixed Dur . In total, 15 conditions were tested, and the results were used to construct this model.

To clarify the effect of QoE degradation due to rebuffering, we examined the relationship between the bitrate and MOS with various rebuffering conditions. The result is shown in Fig. 6. In this figure, “4sec*2,” for example, means there were two 4 sec rebuffering events. This figure illustrates that

- 1) MOS decreased as RC and total RT increased
- 2) MOS decreased as RT increased even when the sum total of the rebuffering time was the same.

In this paper, RT is defined as the average rebuffering time of all rebuffering events. On the basis of these analyses, the rate of MOS decrease due to rebuffering (DR) is modeled as Eq. (7), and the rebuffering-aware QoE-estimation model as Eq. (6).

$$QoE = (QoE_C - 1)(1 - DR) + 1 \quad (6)$$

$$DR = c_1 \log\left(\frac{RC}{RT} + 1\right) \frac{1 - c_2^{RC}}{1 - c_2} \quad (7)$$

D. Duration-aware QoE-estimation Model

We conducted another subjective quality assessment test to extend the rebuffering-aware QoE-estimation model to handle the difference of Dur . In this test, we used the same 83 conditions mentioned in Subsec. IV-C. This data was used to construct the duration-aware QoE-estimation model.

The important factors related to a video-service QoE estimation model are degradation due to encoding and rebuffering. Therefore, we analyzed the effect of the difference in Dur on each type of degradation.

First, we analyzed the effect of the difference in Dur on encoding degradation, that is, we analyzed the test conditions with no rebuffering event. Figure 7 shows the relationship between MOS and Dur , and the MOS and encoding degradation did not change depending on the Dur . Therefore, we concluded that Dur does not impact encoding degradation.

Second, we analyzed the effect of Dur on QoE degradation due to rebuffering. Figure 8 shows the relationship between the total RT and $1 - DR$ with various Dur . and shows that $1 - DR$ increased with an increase in Dur . This means that the

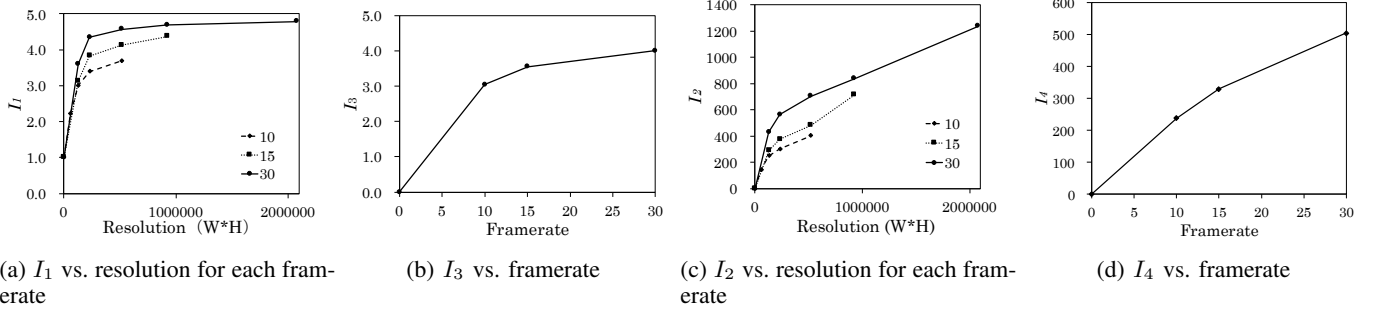
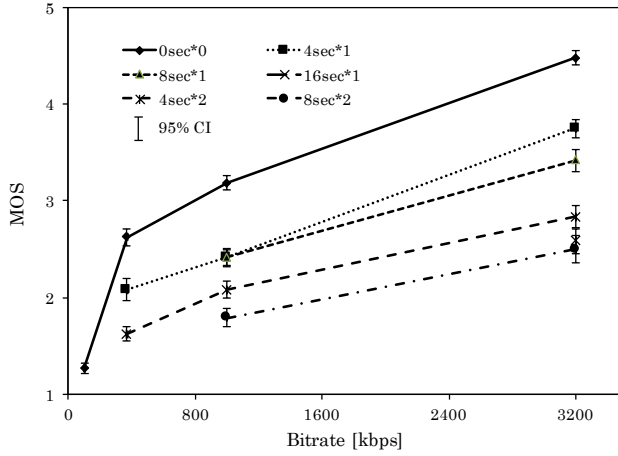
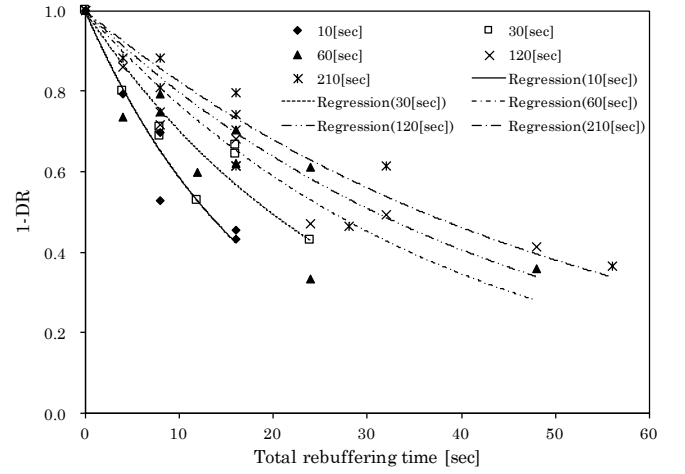
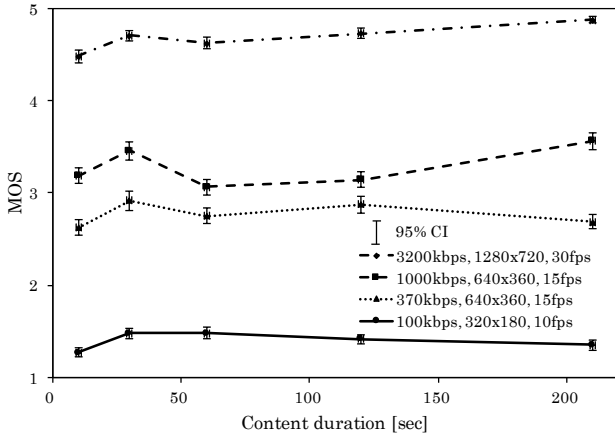
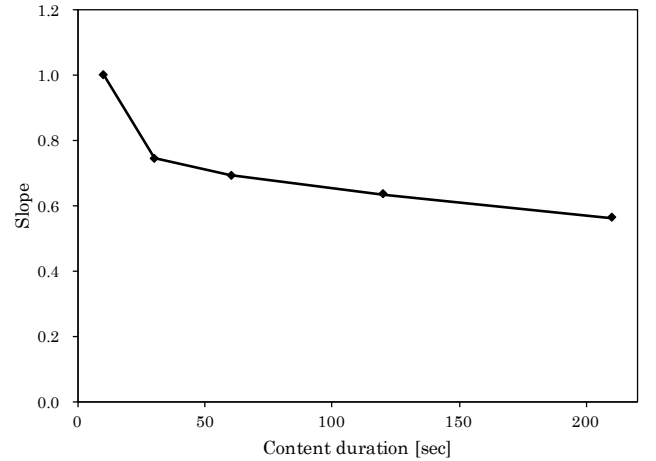

 Fig. 5: Characteristics of I_1 – I_4


Fig. 6: MOS vs. bitrate for various rebuffering conditions


 Fig. 8: $1 - DR$ vs. total RT for each content duration

 Fig. 7: MOS vs. Dur for each encoding condition

 Fig. 9: $Slope$ vs. Dur

longer the video duration becomes, the less effect rebuffering has. Furthermore, we confirmed that bitrate, resolution, and framerate do not affect the decrease in QoE.

We then introduced $Slope$ to adjust DR to correct it and handle the effects of Dur in Eq. (8).

$$DR' = Slope * DR \quad (8)$$

Figure 9 indicates the relationship between $Slope$ and Dur and shows that

- 1) $Slope$ became 1.0 when Dur was 10 sec.
- 2) $Slope$ decreased when Dur increased.
- 3) $Slope$ rate of decrease became lower as Dur became higher.

On the basis of these analyses, we modeled $Slope$ as Eq. (9).

$$Slope = \frac{c_3}{Dur + 10} + c_4 \quad (9)$$

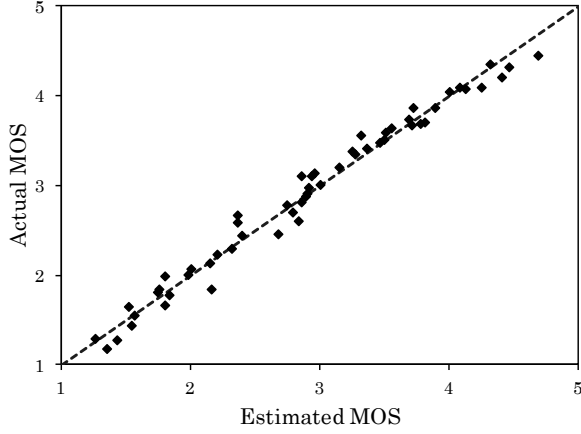


Fig. 10: Accuracy evaluation of encoding-aware QoE-estimation model

TABLE II: RMSE and PCC of estimated and measured MOSs in encoding-aware QoE estimation model

Evaluation index	Value
RMSE	0.131
PCC	0.990

Parameters c_1 – c_4 can be determined and optimized by applying the results of these subjective quality assessment tests.

E. Evaluation

This subsection contains the accuracy evaluation results for the QoE-estimation models discussed in Subsec. IV-B and IV-D. We used the Pearson product-moment correlation coefficient (PCC) and root mean square error (RMSE) to evaluate whether our complete QoE-estimation model is sufficient. The performance criteria for the model were set as described below.

- 1) $PCC \geq 0.94$
- 2) $RMSE \leq 0.27$

We followed the manner of ITU-T Rec. J.247 models [22] and adopted these as the criteria.

First, we evaluated the encoding-aware QoE-estimation model. We conducted a new subjective quality assessment test under 55 of the conditions that did not include rebuffering events, and had a 10 sec content length. Other conditions followed those in Table I.

The results are shown in Fig. 10. The x-axis is the MOS in the subjective quality assessment test, and the y-axis is the estimated MOS. The closer to the diagonal line the plot is placed, the more accurate the encoding-aware QoE-estimation model is analyzed to be. Most of the plots are placed near the diagonal line, which shows that the estimated MOS calculated with the encoding-aware QoE-estimation model was close to the actual measured MOS. Table II shows that RMSE was 0.131 and PCC was as high as 0.990. These results indicate that this model satisfies the performance criteria. Another advantage of this model is the support for high-definition video

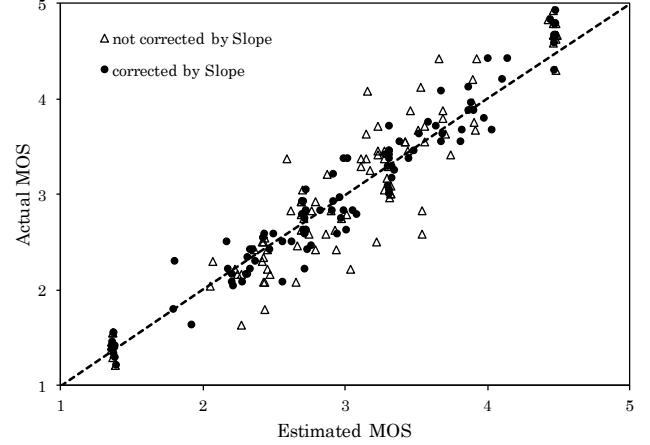


Fig. 11: Improvement in QoE-estimation accuracy with *Slope*

TABLE III: RMSE and PCC of estimated and measured MOSs in duration-aware QoE-estimation model

	RMSE	PCC
Corrected with Slope	0.214	0.940
Not corrected with Slope	0.324	0.847

content. We confirmed that the estimation accuracy for this kind of content had an RMSE of 0.172, which showed that this model is also effective for this kind of content.

Second, we evaluated the duration-aware QoE-estimation model that handles *RT* and *Dur*.

For the evaluation dataset, we used the same data mentioned in Subsec. IV-D. We conducted a 10-fold cross validation as the evaluation method; that is, we used randomly chosen nine tenths of data for learning parameters (i_1 – i_7 , c_1 – c_4), and the remaining one tenth for the evaluation. We conducted this evaluation ten times until all the data were used for the evaluation.

The evaluation results are shown in Fig. 11 and Table III. The x-axis is the measured MOS and y-axis is the estimated MOS. The RMSE, when not considering *Dur*, was 0.324 and the PCC was 0.847, which did not satisfy the performance criteria. However, the RMSE when considering *Dur* was 0.214 and the PCC was 0.940, which satisfy the performance criteria. Figure 11 shows that the estimated MOS of this model was close to the actual measured MOS. Therefore, the accuracy of this model was confirmed.

In the field trial mentioned in Sec. VI, we used a QoE-maximization framework that is based on our complete QoE-estimation model. The coefficient values in the field trial are listed in Table IV. These coefficients were calculated from all the data described in Subsec. IV-D.

V. QOE-PARAMETER-ESTIMATION METHOD

As shown in Fig. 3 in Subsec. IV-D, our complete QoE-estimation model requires bitrate, resolution, framerate, *RC*, *RT*, and *Dur* as the parameters. The bitrate, resolution,

TABLE IV: Coefficient values of QoE-estimation model used in field trial

Coefficient	Value
i_1	1.506228563
i_2	45539.70182
i_3	0.133277280
i_4	0.000361413
i_5	0.000262812
i_6	456.0187229
i_7	0.065806646
c_1	0.400750213
c_2	0.719806312
c_3	9.046965443
c_4	0.763117912

framerate and Dur are available at a streaming server that stores video content. However, the rebuffering conditions are not available until the actual video streaming is finished. Therefore, we designed a QoE parameter-estimation method for estimating the rebuffering conditions before streaming is started.

Our method has the following two features.

- 1) two estimation steps: network quality estimation and rebuffering conditions estimation
- 2) rebuffering conditions estimation is based on the emulation of the buffer transition

As described in Sec. III, a mobile environment is negatively affected by a large amount of temporal variation and locational bias in network quality, so the challenge is to estimate mobile network quality with high accuracy. We examined the characteristics of the mobile-network environments, and constructed an estimation method that predicts the average and standard deviation of network quality for each mobile base station. Then, based on these, we estimate RC and RT using the estimated transition of the amount of client device buffer.

A. Network-quality Estimation

In this subsection, we describe our QoE parameter-estimation method in terms of estimating network quality, i.e., throughput average and throughput standard deviation. To estimate future network quality, we use the selected network-quality-information data stored at the QUVE server that were measured under similar conditions. This is because network-quality values under similar network conditions are considered to have stationarity. The statistics calculated from these data is expected to estimate future network quality in a mobile-network environment.

To estimate the future network quality for users with high accuracy, we have to clarify which parameters of a user's network environment should be used. Therefore, we conducted a preliminarily experiment of throughput distribution by using Nikonico-video-service users' feedback data.

1) *Conditions used to estimate network quality:* We analyzed the actual video-streaming data to develop our QoE parameter-estimation method based on past network-quality information under similar conditions.

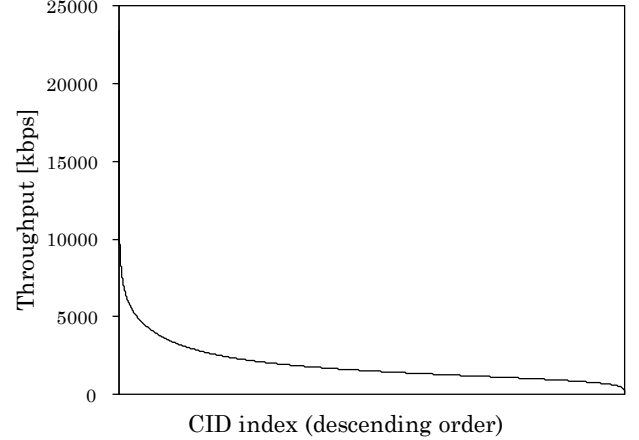


Fig. 12: Throughput distribution for each CID

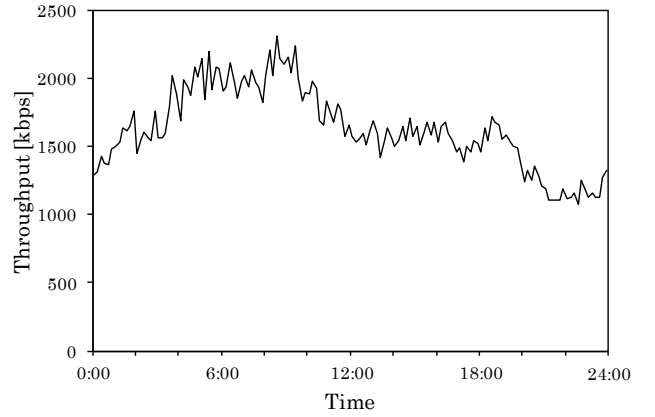


Fig. 13: Throughput transition in mobile site

In recent mobile network environments, the access network part is more likely to be the bottleneck point than the server, in-between network, or client device parts. Thus, we focused on the mobile-access network part and analyzed the network throughput from the perspective of the mobile CID. We did not consider cases in which there is a bottleneck point in the backbone-network or client-device part.

Figure 12 illustrates the distribution of the average throughput for each CID that had more than one hundred pieces of feedback data. It shows that the throughput largely varied depending on the CID and that we need to take into account the CID factor.

Figure 13 shows the transition during one day of average network throughput in an LTE environment. It shows that throughput also varied largely depending on the time slot of a day and that we need to take into account the time slot factor.

2) *Network-quality-estimation part of our method:* We describe the network-throughput-estimation part of our method based on the condition analyses above.

Feedback data having the same values for both factors, that is, the time slot in a day and CID, would be ideal for estimating future throughput. However, we cannot always have enough feedback data to achieve accurate throughput estimation for

TABLE V: Symbols used in this subsection

Symbol	Parameter
BR	encoding bitrate
Dur	content duration
T	average throughput
T_{std}	standard deviation of throughput
Th_{init}	threshold of start playing
Th_{resume}	threshold of resume playing
Th_{stop}	threshold of stop playing

every location. This is because there are more than hundreds of thousands of mobile cells at least in Japan.

Therefore, we introduced the following multiple-granularity grades of matching rules to select the feedback data for throughput estimation.

1) CID match rule

Both the time slot and CID match the estimating condition.

2) eNodeB match rule

Same as above except that the part of CID is used. Only the upper (left-most) 20 bits are used for eNodeB matching¹.

3) No location match rule

Only the time slot matches the estimating condition.

With our method, the first grade rule is tried first. If a sufficient amount of feedback data that match the condition is stored in the database, their average and standard deviation are used for estimation. If not, the coarser granularity grade rule is tried until a sufficient amount of data is found.

B. Rebuffering-time Estimation

In this and next subsection, we describe the rebuffering-condition-estimation part of our estimation method, that is, RC and RT , using the average throughput and throughput standard deviation of the network-throughput data described in Subsec. V-A. In this paper, the symbols used in this subsection are listed in Table V.

When the standard deviation of throughput T_{std} is 0, RT can be calculated in a closed-form expression. Therefore, we start with the rebuffering time RT_{std0} of such a case and extend it to handle T_{std} to calculate the RT .

We start with an equation when the T_{std} is 0. Figure 14 shows the client device buffering behavior and how rebuffering occurs, which is called leaky bucket model [24], [25]. A video starts playing when a playback buffer exceeds a certain threshold, called the “threshold of start playing (Th_{init})”. After starting the video, the buffer starts decreasing if the throughput is less than the content bitrate. Once the buffer drains below another threshold, called the “threshold of stop playing (Th_{stop})”, the playback stops while filling the buffer to a certain amount. After the buffer exceeds that certain threshold, playback resumes. We call this threshold the “threshold of resume playing (Th_{resume})”. In some player implementations, Th_{init} and Th_{resume} are the same.

¹In LTE, CID consists of 28 bits. The upper 20 bits represent an eNodeB identifier, and the lower 8 bits represent an antenna identifier attached to the eNodeB [23].

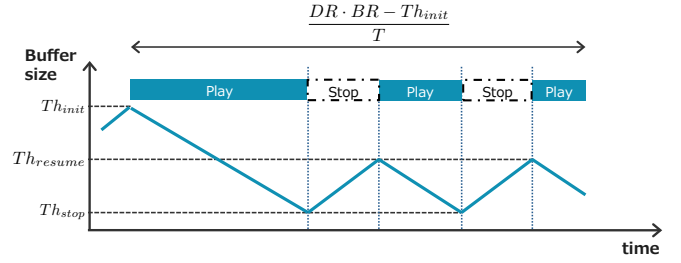


Fig. 14: Buffering behavior and rebuffering occurrence

TABLE VI: Parameters for generating throughput dataset

Parameter	Value
BR	500 kbps
Dur	500 sec
T	200–800 kbps
T_{std}/T	0–1
Th_{init}	12–37 Mbit
Th_{resume}	2–12 Mbit
Th_{stop}	0 Mbit

Considering this behavior of a video player, we can calculate RT_{std0} by using Eq. (10).

$$RT_{std0} = \frac{Dur \cdot BR - Th_{init}}{T} - Dur \quad (10)$$

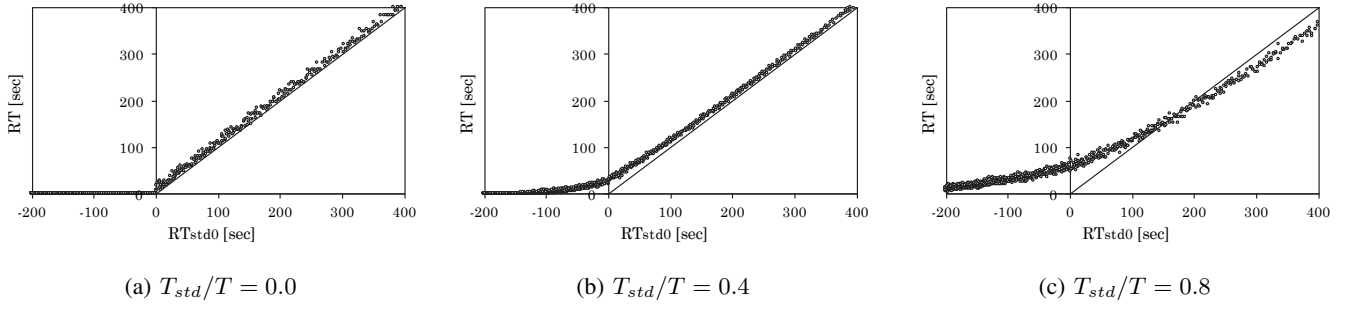
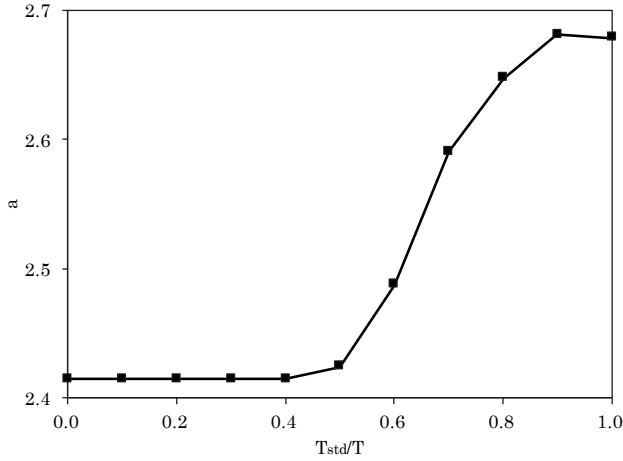
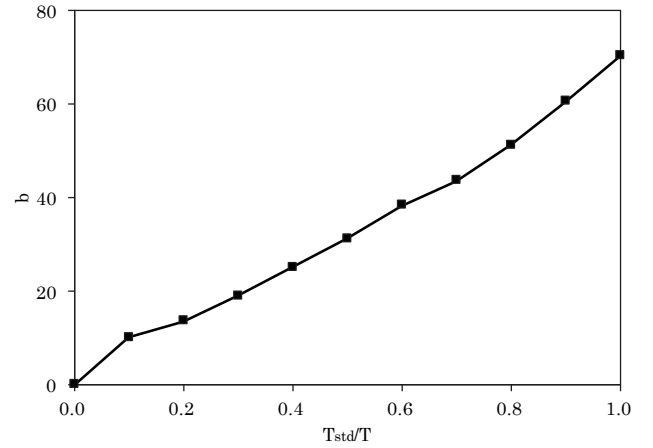
Next, we extend this equation to support the T_{std} and calculate RT . In this case, unlike RT_{std0} , we cannot construct a closed form expression. Therefore, we use the following process.

- 1) We generate an artificial and likely throughput transition based on T and T_{std} .
- 2) We estimate the transition of the playback buffer state using the throughput transition and selected encoding bitrate to calculate the estimated RT .
- 3) We apply steps 1 and 2 to various throughput conditions and analyze the relationship between RT_{std0} and RT .
- 4) On the basis of the relations in step 3, we design a best-fitting regression expression.

The generation of an artificial throughput transition is described below. There is no commonly used model for fitting the network throughput transition. Therefore, for ease of generation, we used a random walk. We calculate the average throughput and throughput standard deviation of the generated transitions and extracted the transitions that have the same average throughput and throughput standard deviation as T and T_{std} . Using these generated throughput-transition data, we follow the leaky bucket model to design the RT -estimation part.

We conducted a simulation test by applying the above process to the various throughput conditions listed in Table VI. For each condition (Th_{init} , Th_{stop} , T , T_{std}/T), we generated as many as 100 patterns of throughput transitions and calculated the average RT .

The results of the simulation test are in Fig. 15. When the T_{std}/T became large, the line form of RT gradually became rounder, and the increase rate of RT gradually decreased.

Fig. 15: RT vs. RT_{std0} for each T_{std}/T Fig. 16: Relationship between T_{std}/T and a Fig. 17: Relationship between T_{std}/T and b

Since this behavior is close to that of a hyperbolic curve, we model RT by using a hyperbolic function as Eq. (11). This equation is derived by rotating the hyperbolic function $\frac{x^2}{a^2b^2} - \frac{y^2}{b^2} = -1$ by the degree of $\theta = -\arcsin \frac{1}{\sqrt{1+a^2}}$. Here, a is a parameter to control the increase rate, and b is a parameter to control roundness.

$$RT = \frac{a * RT_{std0} + a\sqrt{RT_{std0}^2 + b^2(a^2 - 1)}}{a^2 - 1} \quad (11)$$

Next, we designed a model for estimating a and b in Eq. 11 using T_{std}/T . Figure 16 shows the transition of a depending on the transition of T_{std}/T . This figure shows the following characteristics of the relation between T_{std}/T and a .

- 1) a converges to a certain value when T_{std}/T is large or small enough.
- 2) The slope at the inflection point is not infinite.

Moreover, when $T_{std} = 0$, RT should be equal to RT_{std0} . Thus, the angle of the asymptotic line should be $3\pi/4$, and a needs to be equal to $\tan(3\pi/8)$. Considering the above characteristics, a is approximated as Eq. (12), using the Boltzmann equation.

TABLE VII: Coefficients in RT estimation equation

Coefficient	Value
a_1	2.675910
a_3	0.657144
a_4	0.053802
b_1	14.95978
b_2	54.04936

$$a = a_1 + \frac{a_2 - a_1}{1 + \exp\left(\frac{\frac{T_{std}}{T} - a_3}{a_4}\right)} \quad (12)$$

$$a_2 = \tan(3\pi/8) \quad (13)$$

Next, we designed an estimation equation of b . Figure 17 shows the relationship between b and T_{std}/T . The relationship between T_{std} and b follows a moderate quadratic curve. Therefore, we modeled b using the quadratic function in Eq. (14).

$$b = b_1 * \left(\frac{T_{std}}{T}\right)^2 + b_2 * \frac{T_{std}}{T} \quad (14)$$

The values of the coefficients a_1, a_3, a_4, b_1, b_2 are listed in Table VII. These values are calculated using the least-square method.

TABLE VIII: Coefficients in RC -estimation equation

Coefficient	Value
a_1	1.490317
a_3	0.621836
a_4	0.186416
b_1	30660.81
b_2	3193.760
b_3	0.949078
b_4	0.421037

C. Rebuffering-count Estimation

We used the same RC -estimation model as in a previous study [26]. Equation (15) is the RC -estimation part of the method we developed.

$$RC = \frac{a * SCC + a\sqrt{RCC^2 + b^2(a^2 - 1)}}{a^2 - 1} \quad (15)$$

$$RCC = RC_{std0} - C \quad (16)$$

$$C = \frac{b^2 a^2 - (a^2 - 2a - 1)RC_{max}^2}{2RC_{max}a} \quad (17)$$

$$RC_{max} = \frac{BR * Dur - (Th_{init} - Th_{resume})}{Th_{resume} - Th_{stop}} \quad (18)$$

$$a = a_1 + \frac{a_2 - a_1}{1 + \exp\left(\frac{T_{std} - a_3}{a_4}\right)} \quad (19)$$

$$a_2 = \tan(\tan 3\pi/8) \quad (20)$$

$$b = \frac{c_1}{Th_{resume} - Th_{stop}} + c_2 \quad (21)$$

$$c_1 = b_1 \frac{T_{std}}{T} + b_2 \quad (22)$$

$$c_2 = b_3 \frac{T_{std}}{T} - b_4 \quad (23)$$

The values for $a_1, a_3, a_4, b_1 \dots b_4$ are listed in Table.VIII. These values are calculated using the least-square method.

D. Evaluation

In this subsection, we evaluate our QoE parameter-estimation method for RC , and RT .

We begin with the evaluation of the RC estimation part of our method. We used the actual measured data of Niconico video streaming service. The evaluation data were collected from all users of the Niconico player application. The data contain the T , T_{std} , CID, and RC for each viewing. We used data collected during June 2015 for learning the network quality data for each measured CID and data collected during July 2015 for evaluation, where we compared the RC estimated with our QoE parameter-estimation method, which includes network-quality-estimation, with the actual RC .

Figure 18 shows the evaluation results. The x-axis is the estimated RC and y-axis is the average actual RC . The PCC and RMSE scores were 0.955 and 1.06, respectively. From this figure, the method estimated the RC with high accuracy, but it tended to output lower RC values. This seems to be because the video-seek operations were counted as a rebuffering event

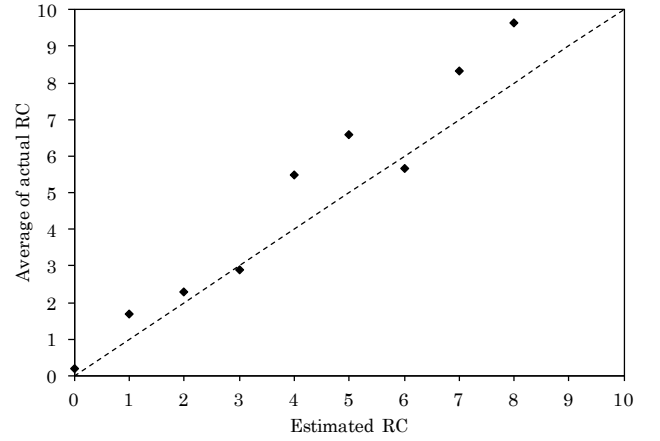
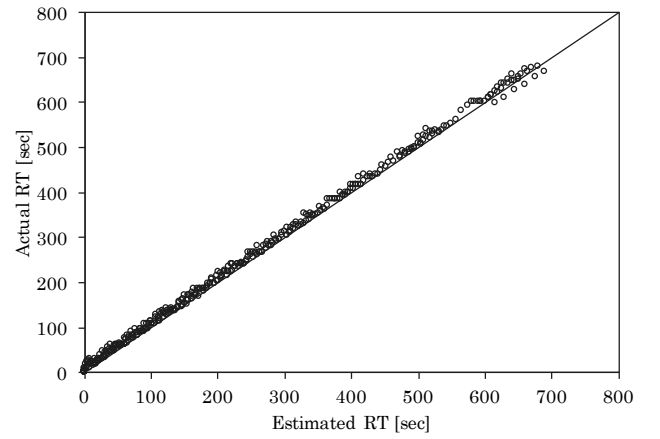
Fig. 18: Comparison of estimated RC and measured RC 

Fig. 19: RT evaluation in simulation

TABLE IX: Parameter ranges used in equation

Symbol	Parameter
T	50 ... 2050
T_{std}/T	0.0 ... 1.2
Th_{stop}	0.0 ... 5000
Th_{resume}	$(Th_{stop} + 500) \dots (Th_{stop} + 15500)$
Th_{init}	$(Th_{resume} + 500) \dots (Th_{resume} + 40500)$

due to the implementation limitation. the collected data do not distinguish user triggered rebuffering events, such as a seek operation done by a user, from network-quality triggered events. We could not collect the user seek operation data due to implementation limitation.

Next, we evaluated the RT estimation part. Unfortunately, we could not collect the actual RT data in the field due to the limitation of the application implementation again. Therefore, we conducted a simulation test.

First, we set each parameter by using the least-square method to minimize the error between the method and data generated in the range given in Table VI. Next, we evaluated the accuracy of our method by using a wider range of parameters. We randomly generated as many as 10,000 samples

within the ranges in Table IX.

We show the simulation results in Fig. 19. The PCC was 0.999, and RMSE of RT was 5.95.

From the above two evaluations, we can confirm the accuracy of the method for estimating RC and RT .

VI. FIELD TRIAL

In this section, we explain the results of a large-scale field trial. In this field trial, QUVE was applied to the Niconico video service operated by Dwango Co., Ltd. This service is one of the biggest video streaming service in Japan and has more than 50 million registered users [27].

We applied QUVE to the in-service application, collected video viewing quality feedback data from all iOS and Android users and conducted recommendation of the encoding conditions to all Android users. All Niconico users could watch videos freely and QoE was calculated with our estimation model described in Subsec. IV-D. To further analyze the effectiveness of QUVE, we gathered actual RC from user devices in addition to the result feedback data described in Fig. 2.

We conducted an A/B test for the Android users by dividing them into two groups: one with the encoding condition recommendation with our framework (*Proposed*) and the other without recommendation (*Default*). As it is an in-production service policy, we cannot describe the complete behavior of the bitrate selection of *Default*, but we can note that *Default* chose relatively lower bitrates for mobile users. Another point of behavior is that the bitrate selection of *Default* did not depend on the user environment, e.g., whether a user was in a congested area or not, which was adopted in *Proposed*.

In this field trial, about 1.4 billion quality feedback data was stored and 1,053,134 feedback data during an 18-day period was evaluated. This is because QUVE needs sufficient amount of feedback data to recommend the appropriate encoding condition.

A. QoE evaluation

First, we compared the QoE of each group. Figure 20 shows the cumulative distribution function (cdf) of the QoE. The x-axis is the QoE starting with a MOS value of 1 and ranging to 5. The y-axis is the cdf of the MOS. These results indicate that *Proposed* achieved a higher QoE distribution than that of *Default*.

To clarify why the QoE improved, we further analyzed the encoding bitrate and rebuffering ratio². Figure 21 shows the selected encoding bitrate distribution for each group. The reason the cdf value for the bitrate of 3000 kbps was under 1.0 is that there are videos that were prepared with only one higher-bitrate format. This result shows *Default* selected a lower bitrate compared with *Proposed*. More precisely, 95% of viewings were delivered in a bitrate lower than 500 kbps for *Default*. In terms of rebuffering, Table X shows the rebuffering ratio for each group. These results indicate that *Proposed* caused higher rebuffering ratio than *Default*. It should be noted

²ratio of viewing data that suffered from one or more rebuffering events

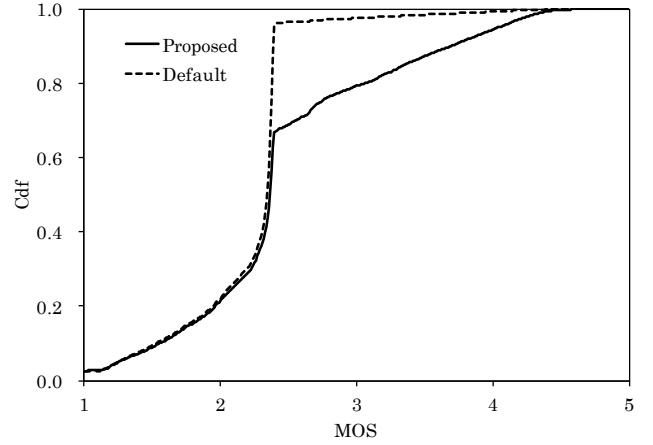


Fig. 20: QoE distribution for each group

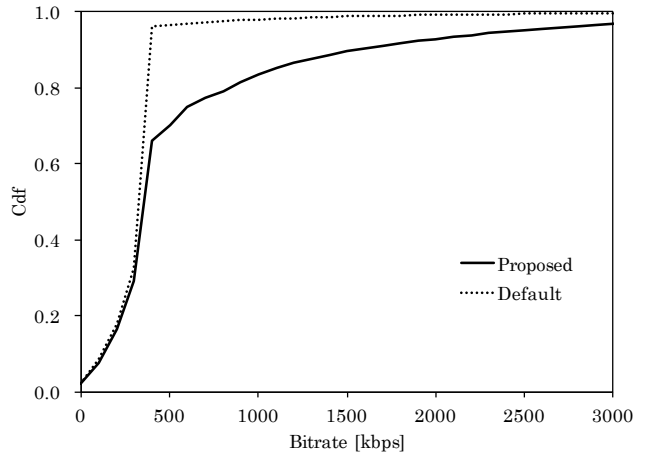


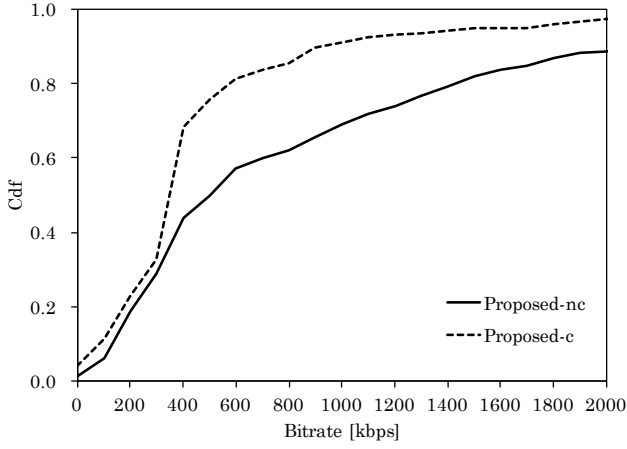
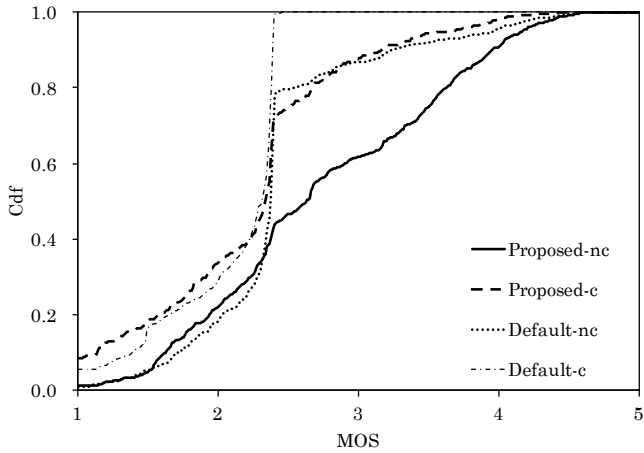
Fig. 21: Bitrate distribution for each group

that the ratio of both groups seemed to be higher than the actual ratio. This is because the video-seek operations were counted as a rebuffering event due to the implementation limitation.

From these analyses, it seems that *Default* achieved a lower rebuffering ratio because of a lower bitrate selection. However, according to the encoding-aware QoE-estimation model (Eq. 1), the QoE was limited below 2.7 for 500 kbps or lower encoding bitrate even without rebuffering. This means that almost all the QoE values of *Default* were lower than those of the QoE delivered with, for example, over a 1400 kbps bitrate and 4 sec of rebuffering (see Fig. 6). On the other hand, *Proposed* selected a bitrate of over 500 kbps for 30% of viewings. This indicates that the encoding-aware QoE value of *Proposed* tended to be higher than that of *Default*. However, a higher rebuffering ratio was exhibited in *Proposed*, as shown in Table X. Therefore, taking these analyses into account, in *Proposed*, it seems that the QoE gain from the higher bitrate exceeded the QoE loss of the higher rebuffering ratio.

TABLE X: Rebuffering ratio for each group

Group	Rebuffering ratio
<i>Default</i>	9.68 %
<i>Proposed</i>	12.77 %

Fig. 22: Distribution of bitrate of *Proposed* in congested and non-congested situationsFig. 23: Distribution of QoE of *Proposed* in congested and non-congested situations

B. Effectiveness of congestion-aware bitrate selection

Second, to confirm the effectiveness of bitrate selection depending on the stored network congestion information, we further evaluated the most and least congested situations. We analyzed one month of average throughput data for each combination of CID and time slot and extracted the upper 10% as non-congested and lower 10% as congested situations.

Figure 22 and 23 show the cdf of bitrate and QoE of *Proposed* for both situations, respectively. In these figures, the dataset gathered from *Proposed* in non-congested situations is denoted as *Proposed-nc*, and that in the congested situations is denoted as *Proposed-c*. In the same manner, in Fig. 23, the dataset gathered from *Default* in non-congested situations is denoted as *Default-nc*, and that in the congested situation is

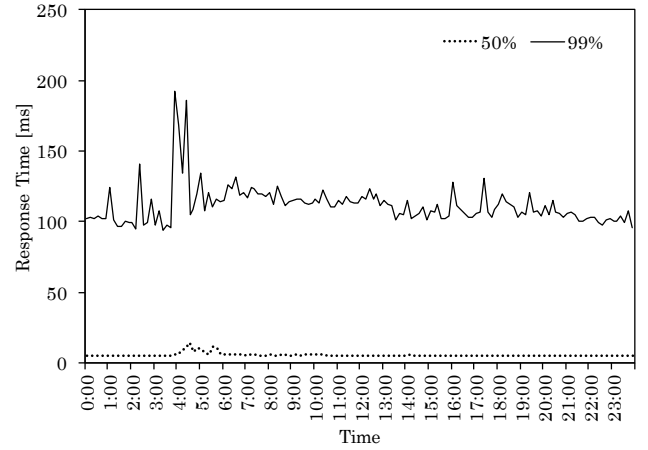


Fig. 24: Response time transition of QUVE server

denoted as *Default-c*.

Figure 22 shows that *Proposed* selected a higher bitrate when the user was in the non-congested situation and selected a lower bitrate when the user was in congested situation. These results indicate that our framework selects the bitrate reflecting the congestion situation. In addition, even in the congested situation, *Proposed* selected bitrates higher than 500 kbps in about 30% of viewings. This ratio was higher than that of *Default*, as shown in Fig. 21. Figure 23 indicates *Proposed* achieved higher QoE distribution in both congested and non-congested situations than that of *Default*.

From these analyses, it is confirmed that *Proposed* did not blindly select a higher bitrate, but rather selected the bitrate that maximizes QoE considering the congestion. By taking into account congestion, our framework achieved higher QoE in both congested and non-congested situations.

C. Overhead

Third, we evaluated the overhead of QUVE.

Figure 24 shows the median and 99 percentile of the response time series of the QUVE server for every 10 min. We used the data of one day's response time in the collected data set.

This results illustrate that the QUVE server can process almost all the requests within 200 msec and half the requests within 20 msec. According to ITU-T Recommendation P.1201 [28], a startup delay of less than 4.29 sec does not affect QoE. Therefore, the field trial also proved that the overhead by introducing QUVE is small and ignorable from the perspective of video-streaming QoE.

VII. RELATED STUDIES

A. Video-quality Maximization

We describe studies related to video-quality maximization.

In certain studies on ABR, forthcoming available throughput was estimated on the basis of the traffic-receiving status of the ongoing session and the video bitrate for chunks was chosen ahead accordingly [29] [30]. In another ABR study, the

amount of received data in the playback buffer was used and the forthcoming chunks' receiving bitrate was selected [31]. These studies have a common issue in that the throughput or buffer status could not be estimated before receiving video traffic for a certain amount of time, so these studies cannot be applied for selecting the initial receiving bitrate. Therefore, many ABR-streaming method implementations start with the lowest bitrate and gradually switch to better bitrates. Another common issue with these implementations is that they can involve bitrate fluctuation that may lead to user QoE degradation [32].

Aquarema [14], [15] is a centralized QoE management framework for video streaming service in the wireless mesh network. With this framework, a network element manipulates resource allocation corresponding to the client device request to increase the QoE of application services. *Rocket optimizer* [16] is also a centralized QoE management solution for video streaming service. This solution selects an appropriate encoding condition and applies it by transcoding the video content. However, this framework is implementable only in a managed network environment and not feasible in every end-to-end path in the Internet.

B. QoE Estimation

We describe the current QoE-estimation model for CBR-based video-streaming services.

A no-reference model that does not depend on information of the original video is used for real-time QoE monitoring and QoE calculation. The model estimates QoE on the basis of network information, such as packet loss ratio and bandwidth, and application characteristics, such as the encoding bitrate and packetization scheme. It can be further classified as a media layer model and packet-layer model depending on the base information. Our complete QoE-estimation model is classified as a packet layer model that uses packet header information obtained in networks.

Packet layer models include V_q [17], *CAPL* [18], and *G.1071* [19] standardized by ITU-T. The V_q is a very straightforward model, in which bitrate-based quality is calculated first. The final quality value is calculated using the bitrate-based quality and packet-loss effect. The *CAPL* calculates the frame quality from the temporal complexity estimated using information such as P-frame and I-frame, and from the bitrate value. The frame-quality and packet-loss information are used for calculating the final quality value. The *G.1071* derives the bitrate, framerate, and I-frame average bytes using the packet-header information and estimates the encoding quality of a video. Then, the number, average duration, and average interval of rebuffering events are used to calculate the degree of quality degradation. The QoE is calculated by subtracting the quality degradation degree of rebuffering from the encoding quality.

In *G.1071*, the QoE models are built for each resolution condition individually. Hence, we cannot compare the QoE between videos that have different resolutions. Another problem is that the standardized model is only applicable for very short video content, such as 10 sec. We found a non-linear

relationship of QoE between *Dur* or rebuffering conditions, so we could not use this model, for example, by calculating the QoE of longer content using the average of the split chunks' QoEs. Therefore, we had to construct a new QoE-estimation model to support a wider range of content duration and resolution.

C. QoE Parameter Estimation

We describe studies on estimating required parameters to calculate QoE. As in Sec. V, to calculate QoE, we have to estimate *RC* and *RT*.

For estimating *RT*, Liu et al. [10] adopted an approach to use the past quality information for cities. With this approach, we need a sufficient amount of data for each player implementation and a buffer control algorithm. On the other hand, our QoE parameter-estimation method can be applied to any player implementation by only modifying the coefficients.

Methods related to network throughput estimation are divided into active and passive measurement.

With active measurement methods, packets for measurement are transmitted in a row. By manipulating the transmission interval and packet size, the amount of measurement packet traffic can be controlled for each time period. If the transmitting traffic exceeds the available throughput, some of the packets are buffered in the middle of the network, and arriving packets are delayed. By using these characteristics, the available network throughput can be estimated [33], [34]. With these methods, the measurement time is necessary before the data-packet transmission. This measurement period increases the waiting time before video playback starts, which leads to user dissatisfaction and them leaving the service [11], [35].

The passive measurement methods use in-service data traffic to estimate future available throughput. Therefore, they cannot usually use real or near-real time measurement data when starting to use a network service. The network-quality-estimation part of our method uses the past traffic data of different users.

VIII. CONCLUSION

We proposed QUVE, a framework for maximizing the user QoE of video-streaming services. The framework consists of two key components: a QoE-estimation model and QoE parameter-estimation method. It recommends the best encoding conditions depending on the time slot in a day and the user location in the network. We demonstrated our framework in a large-scale field trial for Niconico video service and confirmed its effectiveness for QoE enhancement.

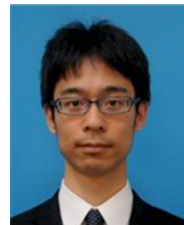
In the field trial, we applied QUVE to a CBR-based video-streaming service. However, QUVE is also expected to be effective for ABR-based video-streaming by recommending not only the initial encoding condition but also the mid-stream encoding conditions. Our future work is extending QUVE to ABR-based services and demonstrating its effectiveness.

ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers and our colleagues for their valuable comments and feedbacks. We also thank Internet Multifeed Co. for providing the excellent network connectivity, which we used in the field test.

REFERENCES

- [1] "Cisco Visual Networking Index (VNI) : Forecast and Methodology, 2014-2019," <http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html>.
- [2] A. Zambelli, "IIS smooth streaming technical overview," *Microsoft Corporation*, vol. 3, p. 40, 2009.
- [3] Apple Inc., "HTTP Live Streaming Overview." [Online]. Available: <https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/StreamingMediaGuide/Introduction/Introduction.html>
- [4] I. Sodagar, "The mpeg-dash standard for multimedia streaming over the internet," *IEEE MultiMedia*, no. 4, pp. 62–67, 2011.
- [5] "Niconico video service (In Japanese)," <http://www.nicovideo.jp/>.
- [6] M. Salem, P. Djukic, J. Ma, and M. Hawryluck, "QoE-aware joint scheduling of buffered video on demand and best effort flows," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*, Sep 2013, pp. 1893–1898.
- [7] P. Casas, M. Seufert, and R. Schatz, "YOUQMON: a system for on-line monitoring of YouTube QoE in operational 3G networks," *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 2, pp. 44–46, Aug 2013.
- [8] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hofeld, and P. Tran-Gia, "A Survey on Quality of Experience of HTTP Adaptive Streaming," *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, pp. 469–492, Sep 2015.
- [9] P. Casas, M. Seufert, F. Wamser, B. Gardlo, A. Sackl, and R. Schatz, "Next to You: Monitoring Quality of Experience in Cellular Networks From the End-Devices," *IEEE Transactions on Network and Service Management*, vol. 13, no. 2, pp. 181–196, June 2016.
- [10] X. Liu, F. Dobrian, H. Milner, J. Jiang, V. Sekar, I. Stoica, and H. Zhang, "A case for a coordinated internet video control plane," in *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*. ACM, Aug 2012, pp. 359–370.
- [11] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang, "Developing a predictive model of quality of experience for internet video," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 339–350, Aug. 2013.
- [12] S. Floyd, "TCP and Explicit Congestion Notification," *SIGCOMM Comput. Commun. Rev.*, vol. 24, no. 5, pp. 8–23, Oct. 1994.
- [13] M. Kühlewind, S. Neuner, and B. Trammell, "On the state of ECN and TCP options on the Internet," in *International Conference on Passive and Active Network Measurement*. Springer, Mar 2013, pp. 135–144.
- [14] B. Staehle, M. Hirth, R. Pries, F. Wamser, and D. Staehle, "Aquarema in action: Improving the YouTube QoE in wireless mesh networks," in *Internet Communications (BCFIC Riga), 2011 Baltic Congress on Future*. IEEE, Feb 2011, pp. 33–40.
- [15] F. Wamser, D. Hock, M. Seufert, B. Staehle, R. Pries, and P. Tran-Gia, "Using buffered playout for QoE-oriented resource management of YouTube video streaming," *Transactions on Emerging Telecommunications Technologies*, vol. 24, no. 3, pp. 288–302, Sep 2013.
- [16] "Opera Rocket Optimizer," <http://www.operasoftware.com/products/operators/rocket-optimizer>.
- [17] K. Yamagishi and T. Hayashi, "Parametric packet-layer model for monitoring video quality of IPTV services," in *IEEE International Conference on Communications, 2008. ICC'08*. IEEE, May 2008, pp. 110–114.
- [18] F. Yang, J. Song, S. Wan, and H. Wu, "Content-adaptive packet-layer model for quality assessment of networked video services," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 6, pp. 672–683, Jul 2012.
- [19] ITU-T Recommendation G.1071, "Opinion model for network planning of video and audio streaming applications," Jun 2015.
- [20] ITU-T Recommendation P.910, "Subjective video quality assessment methods for multimedia applications," Apr 2008.
- [21] ITU-T Recommendation P.800.2, "Mean opinion score interpretation and reporting," May 2013.
- [22] ITU-T Recommendation J.247, "Objective perceptual multimedia video quality measurement in the presence of a full reference," Aug 2008.
- [23] E. LTE, "Evolved universal terrestrial radio access (e-utra) and evolved universal terrestrial radio access network (e-utran) (3gpp ts 36.300, version 8.11.0 release 8)," *ETSI TS*, vol. 136, no. 300, p. V8, Dec 2009.
- [24] J. J. Ramos-Muñoz, J. Prados-Garzon, P. Ameigeiras, J. Navarro-Ortiz, and J. M. López-Soler, "Characteristics of mobile youtube traffic," *IEEE Wireless Communications*, vol. 21, no. 1, pp. 18–25, 2014.
- [25] F. Wamser, P. Casas, M. Seufert, C. Moldovan, P. Tran-Gia, and T. Hossfeld, "Modeling the youtube stack: From packets to quality of experience," *Computer Networks*, Mar 2016.
- [26] T. Kawano, K. Takeshita, and H. Yamamoto, "Video stall count estimation model of progressive download-based video service (in japanese)," *IEICE Society Conference*, vol. 2014, no. 2, p. 258, 2014.
- [27] "Niconico premium members reach 2.5 million. Registered members reach 50 million - CNET Japan (In Japanese)," <http://japan.cnet.com/news/service/35068480/>.
- [28] ITU-T Recommendation P.1201, "Parametric non-intrusive assessment of audiovisual media streaming quality," Oct 2012.
- [29] Y. L. X. Yanling, W. Baolin, "A Network-Adaptive SVC Streaming Strategy with SVM-Based Bandwidth Prediction," vol. 3, no. 3, Jun. 2014.
- [30] T. Oshiba and K. Nakajima, "Quick and simultaneous estimation of available bandwidth and effective UDP throughput for real-time communication," in *Computers and Communications (ISCC), 2011 IEEE Symposium on*, Jun 2011, pp. 1123–1130.
- [31] T. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A Buffer-based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service," in *Proceedings of the 2014 ACM Conference on SIGCOMM*, ser. SIGCOMM '14. ACM, 2014, pp. 187–198.
- [32] F. Bronzino, D. Stojadinovic, C. Westphal, and D. Raychaudhuri, "Exploiting Network Awareness to Enhance DASH Over Wireless," *CoRR*, vol. abs/1501.04328, 2015.
- [33] T. Oshiba and K. Nakajima, "Quick end-to-end available bandwidth estimation for QoS of real-time multimedia communication," in *Computers and Communications (ISCC), 2010 IEEE Symposium on*, Jun 2010, pp. 162–167.
- [34] H. Wang, K. Lee, E. Li, C. Lim, A. Tang, and H. Weatherspoon, "Timing is Everything: Accurate, Minimum Overhead, Available Bandwidth Estimation in High-speed Wired Networks," in *Proceedings of the 2014 Conference on Internet Measurement Conference*, ser. IMC '14. ACM, 2014, pp. 407–420.
- [35] S. Krishnan and R. Sitaraman, "Video Stream Quality Impacts Viewer Behavior: Inferring Causality Using Quasi-experimental Designs," in *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, ser. IMC '12. ACM, 2012, pp. 211–224.



Takuto Kimura Resercher, Communication Quality Group, Communication Traffic & Service Quality Project, NTT Network Technology Laboratories. He received the B.E., M.E. degrees in information and computer sciences from Tokyo Institute of Technology, Tokyo, in 2011 and 2013, respectively. He joined NTT in 2013 and has been engaged in the research of the network management.



Masahiro Yokota received his Master's Degree from Keio University Japan. He is currently a member of researcher at NTT Network Technology Laboratory. His current research focuses of network management technologies, and quality of experience assessment.



Arifumi Matsumoto Senior Research Engineer, Communication Quality Group, Communication Traffic & Service Quality Project, NTT Network Technology Laboratories. He received his B.S. and M.S. in information and computer science from Kyoto University, Kyoto, in 2002 and 2004. He joined NTT in 2004. He has been working on the designing architecture, engineering, and standardization of IP networks.



Takanori Hayashi Manager of the Communication Quality Group, Communication Traffic & Service Quality Project, NTT Network Technology Laboratories. He received his B.E., M.E., and Ph.D. in engineering from the University of Tsukuba, Ibaraki, in 1988, 1990, and 2007. He joined NTT in 1990 and has been engaged in the quality assessment of multimedia telecommunication and network performance measurement methods. He received the Telecommunication Advancement Foundation Award in Japan in 2008.



Kei Takeshita received his Master's Degree from Osaka University Japan in 2008. He is currently a member of researcher at NTT Network Technology Laboratory. His current research focuses of network management technologies, quality of experience assessment, and architecture of future network.



Taichi Kawano received his B.E. and M.E. degrees in Engineering from the University of Tsukuba, Ibaraki, in 2006 and 2008. He joined NTT Laboratories in 2008. He has been engaged in subjective and objective 2D/3D video quality assessment for Internet protocol television(IPTV) services. He received the Young Investigators' Award(IEICE) in Japan in 2011.



Kazumichi Sato received his B.S. and M.S. in information science from Tokyo Institute of Technology in 2006 and 2008. He joined NTT Laboratories in 2008 and has been engaged in DNS traffic analysis and network security. He is a member of IEICE.



Hiroshi Yamamoto Senior Research Engineer, Communication Traffic & Service Quality Project, NTT Network Technology Laboratories. He received the B.S. and M.S. in information and computer science from Waseda University, Tokyo, in 1999 and 2001, respectively. He joined NTT Service Integration Laboratories (now NTT Network Technology Laboratories) in 2001. He has been working on the architecture and performance evaluation of IP networks and web applications. He is a member of IEICE (Institute of Electronics, Information and

Communication Engineers).



Kohei Shiomoto is Senior Manager of Communication & Traffic Service Quality Project, NTT Network Technology Laboratories, NTT, Tokyo, Japan. He joined the Nippon Telegraph and Telephone Corporation (NTT), Tokyo, Japan in April 1989. He was engaged in research and development of ATM networks in NTT Laboratories. From August 1996 to September 1997 he was engaged in research in high-speed networking as a Visiting Scholar at Washington University in St. Louis, MO, USA. Since September 1997, he had been engaged in research and development in the areas of IP/GMPLS networking, IP and optical networking at NTT Network Innovation Laboratories and NTT Network Service Systems Laboratories. From April 2006 to June 2011, he lead the IP Optical Networking Research Group in NTT Network Service Systems Laboratories. He was involved in standardization of GMPLS in the IETF. Since July 2011, he has been leading the traffic engineering research group in NTT Service Integration Laboratories. Since July 2012, he has been leading Communication & Traffic Service Quality Project of NTT Network Technology Laboratories, NTT, Tokyo, Japan. He received the B.E., M.E., and Ph.D degrees in information and computer sciences from Osaka University, Osaka in 1987 1989, and 1998, respectively. He is a Fellow of IEICE, a Senior Member of IEEE, and a member of ACM.



Kenichi Miyazaki Platform Division Manager, Dwango, Co., Ltd. He joined Dwango as a mid-career employee in December 2009. He has developed many Niconico services such as niconico-jikkyou*1 and niconico-denwa*2 as well as Niconico's back-end systems. Since October 2014, he has been in charge of the development of the entire Niconico system. *1 niconico-jikkyou is a communication service enabling viewers to post comments on current broadcast services such as TV programs. *2 niconico-denwa is a communication service enabling

live video distributors to talk on the phone with viewers. This service was terminated in Dec. 2014.