IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 31, 2020, accepted January 21, 2021, date of publication February 23, 2021, date of current version March 16, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3061488

SDN Based in-Network Two-Staged Video QoE Estimation With Measurement Error Correction for Edge Network

SHUMPEI SHIMOKAWA[1], YUZO TAENAKA[2], (Member, IEEE), KAZUYA TSUKAMOTO[3], (Member, IEEE), AND MYUNG LEE[4], (Senior Member, IEEE)

[1]Graduate School of Computer Science and System Engineering, Kyushu Institute of Technology, Iizuka 820-8502, Japan
[2]Graduate School of Science and Technology, Nara Institute of Science and Technology, Ikoma 630-0192, Japan
[3]Department of Computer Science and Electronics, Kyushu Institute of Technology, Iizuka 820-8502, Japan
[4]Grove School of Engineering, The City College of New York, New York City, NY 10031, USA

Corresponding author: Shumpei Shimokawa (shimokawas@infonet.cse.kyutech.ac.jp)

This work was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant 18H03234, in part by the National Institute of Information and Communications Technology (NICT) under Grant 19304, and in part by NSF, USA, under Grant 1818884 and Grant 1827923.

ABSTRACT Network resource management is one of the key technologies needed to ensure that multiple applications in edge networks provide reliable and stable performance. Although throughput has previously been seen as the primary network performance metric, recent applications do not focus on throughput alone. Instead, Quality of Experience (QoE) is attracting significant attention as an indicator of network resource management performance because it allows a wide variety of applications to be compared within a single metric. In this study, we tackle QoE measurements for a video streaming service as a way to evaluate QoE-based network management. However, there are several problems related to measuring QoE. For example, in-network components are difficult to measure because QoE is normally measured at end-points, and several properties that are deeply related to application settings are required for those calculations. Additionally, the measurements set forth in the International Telecommunication Union's ITU-T G.1071 standard require a certain duration, which is too long for network resource management evaluations. Therefore, this paper proposes a two-staged in-network QoE estimation method for video flows that can resolve these issues. In the first stage, we focus on producing a fast and rough QoE estimate to start forwarding the arriving flow onto an appropriate route as soon as possible. Next, the second stage is designed to produce precise QoE estimations based on careful long-duration measurements. In both stages, the proposed method uses a parameter estimation process that converts in-network information to end-point information for QoE calculations by following ITU-T G. 1071 and corrects measurement errors reducing QoE calculation errors to the greatest extent possible. Through experimental evaluations, we then demonstrate that the QoEs of all flows can be maximized by selecting appropriate routes based on the predicted QoE at the first stage, and that the accuracy of the QoE estimation at the second stage can be improved in real-time even when packet losses occur.

INDEX TERMS Error correction, network measurement, OpenFlow, RTP, software defined networking, quality of experience, video streaming.

I. INTRODUCTION

Network traffic is increasing significantly due to the spread of rich video-based contents such as movie streaming, vlogs, and even newscasts. According to a report by Cisco Systems Inc., [1], video streaming services have grown to become a key component of the Internet. Since this trend has resulted in several problems that must be resolved, including alleviation of huge traffic amounts and reductions to communication latency, contents tend to be delivered from near proximity to their end-users, such as local data centers or edge servers [2]. In such local content delivery models, network resource

The associate editor coordinating the review of this manuscript and approving it for publication was Wuliang Yin.

management is one of the key technologies needed to ensure that multiple applications in edge networks provide reliable and stable performance.

Additionally, although previous studies have examined throughput-oriented resource management [3]–[5], providing good network performance for some applications is particularly difficult because not all emerging applications focus solely on throughput. Thus, a network resource management evaluation that is based on another simple and useful metric is required. One candidate metric is Quality of Experience (QoE), which indicates application performance at an end host. Because QoE enables us to compare a wide variety of different application flows on a network with a single common metric, it is expected to be useful for resource management on networks where a multiplicity of applications coexist.

Ideally, the fundamental QoE concept is the user's perception of an application [6], which means its score should be based on subjective user evaluations. However, since such scoring is not applicable to network management, several other ways of measuring QoE without human perception have been proposed and standardized [7], [8],[9]. If we feed the QoE scores of every flows to a network management system, it is possible to optimize network resources (network routes) that maximize the QoEs for those same flows.

Video streaming is a representative application that does not rely solely on throughput and QoE measurements regarding it have been studied extensively for decades [10]. However, there are several problems when attempting to apply QoE measurements to network resource management. First, in-network components cannot measure QoE because QoE is an end-point performance metric, and standards such as the International Telecommunication Union's ITU-T G.1071 require end-point measurements. Second, several of the parameters needed for QoE calculations cannot be collected within a network because they are deeply and directly related to the properties of the displayed video. Third, QoE measurements normally require a certain duration after a designated flow begins, which imposes delays that make it difficult to promptly arrange network resources for newly arriving flows. Fourth, since video streaming is particularly sensitive to packet losses, the calculated QoE score is significantly decreased if even a minor measurement error occurs.

To enable QoE scores to be used for resource management and resolve the above-mentioned problems, we propose a two-staged in-network QoE estimation method for video flows. For ''in-network'' realization, we employ OpenFlow [11] software-defined network (SDN) technology, which enables us to collect statistical information for each flow from network switches in between end hosts. To adapt the QoE measurement method, which is originally standardized for end-point measurements, to in-network measurements, we created a way to convert in-network measurement results to application video parameters. Additionally, to fill the time gap between the start of communication and the QoE estimation, we employ a two-staged QoE estimation mechanism in which we first obtain a roughly QoE value based solely on the information collected at the start of a newly arrived flow, and then precisely estimate QoE based on careful long-duration measurements in the second stage.

The remainder of the paper is organized as follows. Section II introduces works related to QoE based network management and discusses the advantages of our proposal. In Section III, we explain a standardized QoE calculation method for video streaming and conduct a theoretical survey for that model. Section IV introduces our proposed method in detail, while evaluations are discussed in Section V. Finally, Section VI concludes our study.

II. RELATED WORKS

QoE-driven networks have previously been the topic of numerous papers [12]–[20]. For example, in the paper [12], the authors studied a network-wide QoE-fairness method that operates by dynamically allocating network resources for heterogeneous applications. As a result, they improved the quality of video streaming and file downloading by considering video buffering time and throughput. However, since that method was specialized for cases involving coexisting video and file download traffic, it did not result in a common metric that would be applicable to any other applications. This is an important consideration because using QoE directly as a network management metric would eliminate the need to change management policies to support each application.

Separately, papers [13]–[19] calculated QoE by parameters obtained from the application layer. However, since application layer information is often encrypted between end-to-end hosts, the retrieval of information deeply relating to application layer from in-network components such as routers is inherently difficult. Accordingly, to calculate QoE on in-network components, a way to estimate these application parameters based on the available network parameters is required. Additionally, even though other papers such as [17]–[19] have introduced ways to measure application parameters, they assume that each of the network components directly collect and analyze video packets for the QoE score calculations. In contrast, instead of directly measuring every packet, our method uses SDN functionalities to collect statistics for each application flow (flow level) in order to calculate its QoE scores. Additionally, we propose a method for estimating video properties that considers realistic deployment conditions.

In the paper [20], the authors used a neural network to estimate QoE scores for video flows. Their neural network learned only network parameters, such as packet loss rates (PLRs) and the length of burst losses, in the access point (AP), and then allocated flow priorities based on the estimated QoEs. However, they used subjective QoE test user responses as the neural network labels even though subjective QoE evaluations are highly dependent on individual perceptions. In contrast, our method adopts an objective way to apply ITU standardized QoE parameters, which means it does not depend on individual perceptions.

Moreover, most of the previous studies assumed that the parameters needed for QoE calculations would be measured precisely. However, in realistic scenarios, errors often occur in the in-network component measurements. Therefore, in this paper, we first investigate how measurement errors occur through the experiments in a real environment, and then identify the key parameters that significantly affect QoE values.

III. QoE VIDEO STREAMING CALCULATION

The previously mentioned ITU-T G.1071 [7], which is a major standard for use in video QoE calculations, is intended to make it possible to calculate video QoEs using network measurement results. Accordingly, while it would be ideal to employ this QoE measurement standard in our work, the G.1071 standard also requires both network parameters to be obtained via measurements, and the application parameter video settings to be collected by video images from the application. These information also must be from 8-second measurement. Furthermore, while the QoE score is normally calculated using a combination of video and audio metrics, we omit the audio data in our method and focus solely on the video metric to simplify the QoE calculation model. Note that QoE estimation results are judged on a range from 1 to 5.

TABLE 1. G.1071 target video settings.

Category	Lower resolution	Higher resolution
Protocol	RTSP over RTP	MPEG2-TS over RTP
Video codec	H.264, MPEG-4	H.264
Resolution	QCIF (176×114), HVGA (480×320)	SD(720×480), HD (1280×720,1920×1080)
Video bitrate (bps)	QCIF:32 - 1000 k, HVGA: 192 - 6000 k	SD:0.5 - 9 M, HD:0.5 - 30 M
Video frame rate	5 - 30 fps	25 - 60 fps

G.1071 consists of two resolution categories: high- and low-resolution (Table 1). Herein, we use high-resolution video because there is now more demand for that category than low-resolution video. The QoE calculation model is denoted as shown below:

$$Q_{\mathrm{score}} = 1.05 + \frac{3.85}{100Q_{100}} + (7.0\times10^{-6})Q_{100}(Q_{100}-60)\times(100-Q_{100}) \quad (1)$$

$$Q_{100} = 100 - Q_{\mathrm{app}} - Q_{\mathrm{net}} \quad (2)$$

$$Q_{\mathrm{app}} = Ae^{BP} + C + (De^{EP} + F) + G \quad (3)$$

$$P = \frac{10^6 v_{\mathrm{br}}}{v_{\mathrm{r}} v_{\mathrm{fr}}} \quad (4)$$

$$Q_{\mathrm{net}} = H\log(IJ+1) \quad (5)$$

$$J = v_{c1}\exp\{\frac{v_{\mathrm{pl}} v_{c2}(v_{c3}-v_{c4})}{v_{c4}(v_{c5}v_{\mathrm{bl}}+v_{c6})+v_{\mathrm{pl}}}\} - v_{c1} \quad (6)$$

Q_{score} is the QoE score range from 1 to 5, and is calculated from Q_{100}, which is QoE score range from 1 to 100, via Eq. (1). In this study, we use Q_{score} as the QoE value. The $A \sim I$ parameters are fixed and defined in G.1071. Only B and E have negative values, while all others have positive values. Additionally, the video bitrate v_{br} [bps], resolution v_{r} [pixel], frame rate v_{fr} [fps], and packet loss concealment (PLC) are pre-determined as the video setting parameters. On the other hand, the PLR v_{pl} [%] and the average number of consecutive packet losses v_{bl}, which is the average length of the burst losses, are parameters that need to be measured at the end hosts. Here, it should be noted that $v_{c1} \sim v_{c6}$ in the Eq. (6) have different values depending on the PLC function chosen in the application settings. A PLC is an error correction method for damaged video frames occurring from packet losses with two functions: freezing (which only ignores losses) and slicing (which tries to correct the losses).

From this model, we can see that the PLR has the largest impact on the video QoE. Table 2 shows the relationship between the PLR and QoE, as calculated by the model. All of the parameters except the PLR are fixed to a standard definition (SD) video with a bit rate of 2.5 Mbps, a frame rate of 30 fps, slicing PLC with one slice/frame, and an average number of consecutive packet losses of one. In this case, even 0.1% packet losses (two lost packets per second on the SD video) have a significant impact on the QoE score. Therefore, we can conclude that PLR is a key video QoE parameter.

TABLE 2. PLR impact on QoE.

PLR (%)	QoE
0	4.654314
0.01	3.333041
0.1	1.874826
1	1.390250

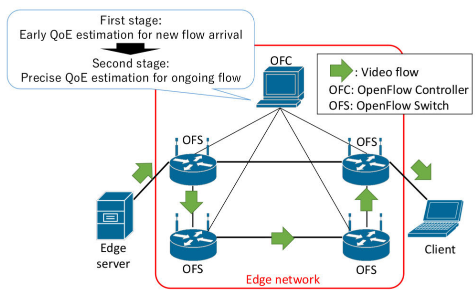

FIGURE 1. Target network model.

IV. TWO-STAGED IN-NETWORK QoE ESTIMATION

To perform QoE estimations in an in-network fashion, we assume the use of an SDN (OpenFlow)-enabled edge network, i.e., a network positioned in between two end hosts who communicate by video streaming, as shown in Figure 1. There are two routes with different network characteristics. A single OpenFlow Controller (OFC) collects statistical information from any OpenFlow Switches (OFSes) located on either route. It then uses that information to determine appropriate transmission routes for each flow by directing flow entries towards the appropriate OFSes. On this network, we define a flow by 5-tuple, which refers to a unit of video streaming flow, and estimate the QoE score for each flow based on the flow statistics collected by the OFC.

To apply the G.1071 standard to our QoE calculation model, we essentially need to measure six parameters observed at an end host: the PLR, the average number of consecutive packet losses, the video bitrate, the video resolution, the frame rate, and the PLC. However, as a means of contributing to network management, end-point measurements are not a realistic technique for real Internet services. Accordingly, we virtually convert statistical information acquired at the OFC (in-network measurement results) to end-point measurement results. Furthermore, the G.1071 QoE calculation requires relatively long (eight-second) measurements, which is unsuitable for network management operations that must allocate network resources to every flow upon arrival. Therefore, we divide the QoE estimation process into two stages. In the first stage, a QoE score is promptly but roughly estimated to select a temporary routing path for flow transmission at the arrival time. Then, in the second path, a precise QoE score for the ongoing flow is estimated. Since SDN-based measurement errors occur in both stages, we employ error correction techniques for each. These two stages, including the details of several techniques mentioned above, will be described below.

A. FIRST STAGE: EARLY QoE ESTIMATION FOR NEW FLOW ARRIVAL

When a new video flow arrives at the SDN-enabled edge network, its QoE must be measured as soon as possible so appropriate network resources can be allocated. This means we must make the initial new flow QoE estimate based on limited information because detailed flow information is not yet available. To calculate the QoE via Equation (1), we must first acquire network parameters (i.e., the PLR and the average number of packet losses) and application (video) parameters (i.e., bitrate, resolution, frame rate, and PLC) on the OFC. However, since most of these information types require measurement results that are unavailable at the start of a flow, this stage focuses on producing prompt but rough QoE estimations that can be used to immediately start the flow on an appropriate route. The second stage then gathers information until a sufficient quantity is available for a precise QoE estimation.

1) EARLY NETWORK PARAMETER ESTIMATION

Network parameters required for QoE include the PLR and the average number of consecutive packet losses. However, since it is impossible to obtain the PLR for a flow before it starts, we begin by assuming that the PLR for all flows on the same data link will be the same, and thus regularly measure the PLR of existing traffic. Since these measurements take place before a flow arrives, we can use the pre-measured PLR for calculating the QoE of a newly arrived flow.

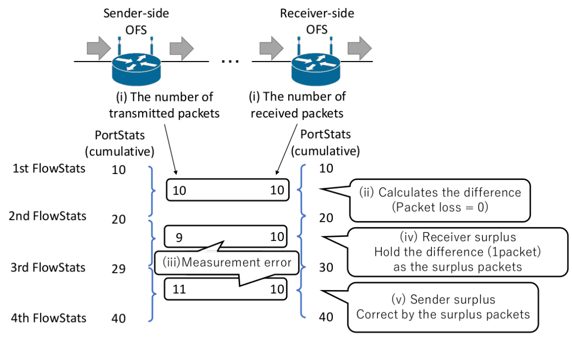

FIGURE 2. Example of the error correction using receiver surplus packets.

Additionally, since network conditions may be different among routes, we measure the PLR for each route.

We regularly measure the PLRs of existing traffic using the port statistics (PortStats) of each OFS, which are collected from the OFC on a request basis. Since the PortStats include the cumulative number of transmitted and received packets on each network interface (not each flow) of a designated OFS, the OFC obtains the number of transmitted and/or received OFS packets by subtracting the number from two consecutive PortStats acquired during a certain period (see Figure 2 (i)). The OFC takes the difference of the number of packets transmitted from one side of the OFS (sender-side OFS) and the number of packets received at the other side of OFS (receiver-side OFS) (see Figure 2 (ii)) — this difference is treated as the number of lost packets — and then calculates the PLR of one-hop (between directly connected sender- and receiver-side OFSes) by dividing the number of lost packets by the number of transmitted packets. After the PLRs of every hop are calculated, the OFC calculates the end-to-end PLR, which refers to the PLR between two OFSes accommodating two end hosts, by multiplying the PLR of every hop on a route.

The above procedure may cause measurement errors because PortStats cannot be collected between two OFSes at the exact same time (see Figure 2 (iii)). However, since packet losses are the primary factor affecting video QoE, we can try to correct the errors with an additional mechanism. More specifically, since the OFC is set up to perform the measurement for handling new flows at any time, we can employ a simple error correction in the time domain. The measurement errors are categorized for two cases: when the number of received packets at the receiver-side OFS is larger than the transmitted packets at the sender-side OFS (receiver surplus), and when the number of transmitted packets at the sender-side OFS is larger than the number of received packets at the receiver-side OFS (sender surplus). The receiver surplus, which is obviously a measurement error, often causes a subsequent sender surplus error; while the sender surplus is most likely to be caused by packet losses and measurement errors.

As the first stage of the QoE estimation focuses on promptness while allowing for a certain degree of reduced preciseness. We only use this stage to correct receiver surplus measurement errors. More specifically, the OFC continuously collects and holds receiver surplus packets when a sender surplus occurs. These packets are held cumulatively because a sender surplus not only occurs on an immediate subsequent measurement. For example, in the case of Figure 2, a +1 receiver surplus occurs first (see Figure 2 (iv)) and the sender surplus occurs later. Since we have one surplus packet that can be used to correct a subsequent sender surplus at the start of the sender surplus, (see Figure 2 (v)), it can be used to correct measurement errors caused by the receiver surplus.

Regarding another necessary parameter, the average number of consecutive packet losses, the OFC cannot recognize consecutiveness because it only allows us to collect statistics. Since, according to G.1071 model, the QoE's worst scores occur when the average number of consecutive packet losses is one, we always use that value in this paper.

2) EARLY APPLICATION PARAMETERS ESTIMATION

It is difficult to discern application video settings properties by in-network methods, and it is even more difficult when a new video flow first arrives because there is no information about the flow at all. That is why it is necessary to collect any and all information as soon as possible before estimating application parameters. To accomplish this, the OFC quickly but tentatively, forwards a newly arriving flow onto the route with the largest residual bandwidth and use statistics but promptly switches the flow onto a more appropriate (better QoE) route once it obtains sufficient information.

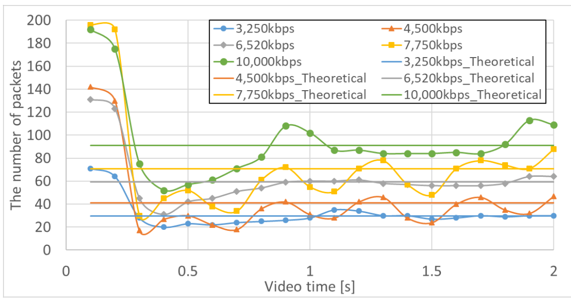

FIGURE 3. The number of video flow packets transmitted every 0.1 seconds.

Once on the tentative route, we measure the new flow throughput because we already know its video bitrate must relate to throughput. Figure 3 shows the number of transmitted packets of a video flow plotted for every 0.1 seconds. Note here that, in this experiment, we are only transmitting a single video flow on a stable network, and that "Theoretical" refers to the theoretical number of packets that can be transmitted every 0.1 seconds in each video bitrate. As throughput at the beginning is significantly different from the theoretical value due to video buffering, we cannot use instantaneous throughput for bitrate estimations. Instead, we use the number of transmitted bytes for a set period. The correlation between the number of transmitted bytes for 0.9 seconds and the video bitrate shown in Figure 4 indicates that this time period

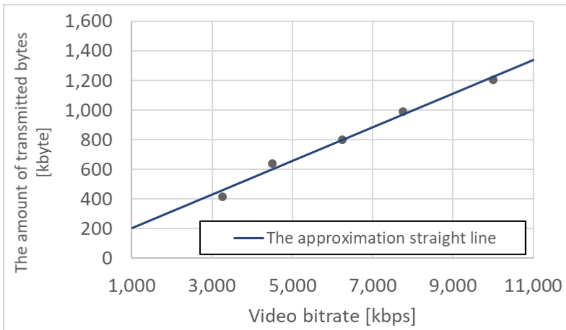

FIGURE 4. The correlation between the number of transmitted bytes for 0.9 seconds and video bitrate.

is sufficient to identify the video bitrate. The reason why 0.9 seconds was selected as the measurement period is that it and the operational margin (0.1 seconds) are most suitable for switching to the second stage of QoE estimation, which requires the stable throughput behavior that starts around one second after the flow begins. Therefore, to obtain the video bitrate from the number of transmitted bytes, we use the approximation line of Figure 4,

$$v_{\mathrm{br}} = 8.7903b - 7.7067 \times 10^2 \tag{7}$$

where v_{br} [kbps] is video bitrate and b [kbyte] is the number of bytes transmitted at 0.9 seconds from the start of new video flow.

For collecting b, we use the flow statistics (hereafter, FlowStats) that any OFC can obtain on a request basis. In this process, after the OFC forwards sends a new flow to a tentative route, the OFC waits 0.9 seconds, and then collects FlowStats by specifying the new flow. Since the obtained FlowStats include the cumulative number of packets and the cumulative number of bytes handled by the OFS, we use that byte total in Equation (7).

TABLE 3. Recommended video bitrates for resolution and frame rates.

Resolution	Frame rate [fps]	Video bitrate [kbps]
SD (720×480)	24,25,30	2,500
SD (720×480)	48,50,60	4,000
HD (1280×720)	24,25,30	5,000
HD (1280×720)	48,50,60	7,500
HD (1920×1080)	24,25,30	8,000
HD (1920×1080)	48,50,60	12,000

Other parameters, video resolution and frame rate, are converted from the video bitrate. We use YouTube recommendations for the resolution, frame rate, and video bitrate video settings, as shown in Table 3 [21]. We then modified Table 3 and obtained Table 4 to define the video settings boundaries, which are the mean of the video bitrate of each setting and the maximum frame rate. As a result, we can estimate the resolution and frame rate based on the video bitrates shown in Table 4.

For the PLC application parameter, we use a fixed value because it is impossible to acquire or estimate that value

TABLE 4. Resolution and frame rate estimates based on video bitrates.

Video bitrate [kbps]	Resolution	Frame rate [fps]
- 3,250	SD (720×480)	30
3,250 - 4,500	SD (720×480)	60
4,500 - 6,250	HD (1280×720)	30
6,250 - 7,750	HD (1280×720)	60
7,750 - 10,000	HD (1920×1080)	30
10,000 -	HD (1920×1080)	60

in-network without decoding the video data. Therefore, we use one slice per frame as PLC, which results in a lower QoE score than other PLC parameter types.

3) ROUTE SELECTION BASED ON ESTIMATED QoE

After estimating all parameters, the OFC calculates the expected QoE for each route, which is based on the estimated QoE score obtained if the target flow follows those routes, after which it switches the flow from the tentative route to the route with the highest estimated QoE. This method allows us to effectively use the time gap between the flow arrival and when the QoE is more precisely estimated by the second stage procedure.

B. SECOND STAGE: PRECISE QoE ESTIMATION FOR ONGOING FLOW

Since the OFC collects flow statistics for a longer period in the second stage, information precision can be improved significantly over that possible in the first stage. Since the G.1071 standard requires eight-second measurements, we adopted that time period for our measurements. This can also help produce higher precision of video bitrate and PLR estimates. Note that the average number of consecutive packet losses, frame rate resolution, and PLC, are obtained in the same manner in the first stage because they do not depend on the measurement period.

In the second stage, we use the most recent recorded eight seconds to obtain the throughput of each flow. More specifically, once the first stage selects an initial transmission route (i.e., one second after the flow starts), the OFC starts collecting FlowStats from the OFSes accommodating both end hosts at one-second intervals. In this throughput measurement, we use the statistics obtained from the OFS that is nearest to the host sending the video stream. Along with a periodic collection of FlowStats, the OFS calculates the average throughput for the latest eight seconds and then, after reducing the amount of control headers, uses that value as the video bitrate. This is because the video streaming throughput converges around the video bitrate one second after a flow starts, as can be seen in Figure 3.

The flow PLR is calculated using statistics obtained from the OFSes accommodating both end hosts. The basic idea is based on the same concept used by the first stage, which is observing differences in the number of packets sent on the sender-side OFS and received on the receiver-side OFS. The differences are the measurement section. In the first stage, we measure the PLR hop-by-hop to calculate candidate routes. However, in the second stage, we only calculate the PLR on a route if the flow is complete, which means this stage only uses statistics from neighboring (or directly connecting) end host OFSes.

As with the first stage, measurement errors occur due to the same reason. However, the measurement errors in this stage occur in the FlowStats instead of the PortStats used in the first stage. Since the second stage focuses on producing a more precise QoE estimation, a more sophisticated packet loss error correction mechanism is needed. In the following subsection, we describe measurement error characteristics and the measurement error correction method that uses those characteristics.

1) MEASUREMENT ERROR CHARACTERISTICS

Since the differences between the number of sent and received packets on the sender- and receiver-side OFSes are caused by either packet losses or measurement errors, we need a method to differentiate packet losses from measurement errors in order to precisely calculate the PLR.

Since the OFC collects measurement statistics periodically, we assume that the error quantity per measurement and error measurement recovery delays, which refer to the time between an error occurrence and its compensation, possess some similar tendencies. To investigate this point, we conducted an experiment in which we transmitted a single video flow using the settings in Table 5 for 1508 seconds on a stable OpenFlow-controlled network. In this experiment, there were two directly connected OFSes with connected end hosts. We performed the same measurement with the second stage and detected 243 measurement errors. Here, it should be noted that no packet losses occurred in this experiment.

TABLE 5. Video settings of Section IV-B1.

Codec	H.264
Resolution	SD (720 × 480)
Frame rate	30
Video bitrate	Fixed 2.6 Mbps

Figure 5 shows the frequency and error quantity per measurement trial results. Since measurement errors were classified into two types, sender and receiver surpluses, we analyze them separately. In Figure 5, we can see that there is no bias between the sender and receiver surpluses and that 96% of measurement errors are within the quantity of one packet. From this result, we can say that the maximum measurement error quantity is one packet.

Figure 6 shows the result for measurement errors recovery delays. Measurement errors result from the time gap in collecting FlowStats from the two OFSes. Since both sender and receiver surpluses are definitely due to measurement errors in this experiment, we can measure delays until each surplus has been compensated for. There are two recovery delay types, sender surplus first and receiver surplus later, and vice versa. Since we have already determined that the possibility

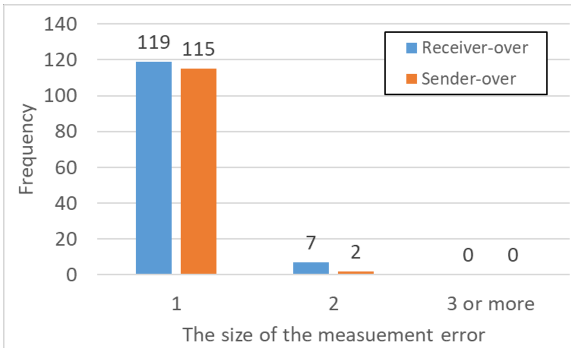

FIGURE 5. Frequency and error quantity per measurement.

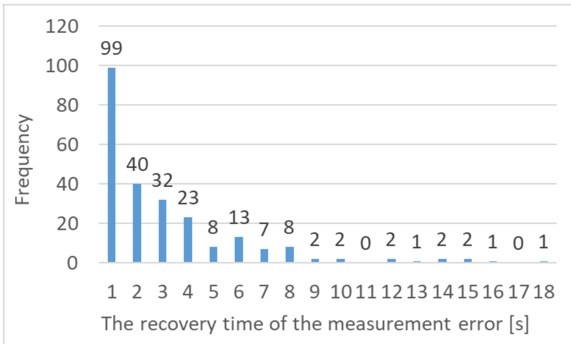

FIGURE 6. The frequency of the measurement error recovery time.

of causing sender and receiver surpluses is almost same (see Figure 5), we can count both together. In Figure 6, which shows a recovery delay and frequency histogram, we can see that the shorter the time from the occurrence of errors, the greater the frequency of recovery. Since 95% of the measurement errors have been recovered within eight seconds, we can conclude that any surpluses remaining for eight or more seconds are packet losses. From these points, we convert Figure 7 to a cumulative distribution and fit plots with $y = \frac{A}{x+B} + C$ (Figure 7). The approximate curve is

$$F(t) = \begin{cases} 0 & (t \leq 0) \\ \frac{-2.45300}{t+1.98451} + 1.23839 & (0 < t < 8) \\ 1 & (8 \leq t), \end{cases} \tag{8}$$

where t is the time from the occurrence of a measurement error.

Summarizing the investigation of measurement errors, we can find the following two characteristics.

- The quantity of measurement error is one packet.
- The recover delay follows the curve of Equation (8).

2) ERROR CORRECTION

Next, we present a measurement error correction method for counting packet losses that is based on the measurement error analysis results. Since two types of errors occur in our measurements, we correct each error type via a different method.

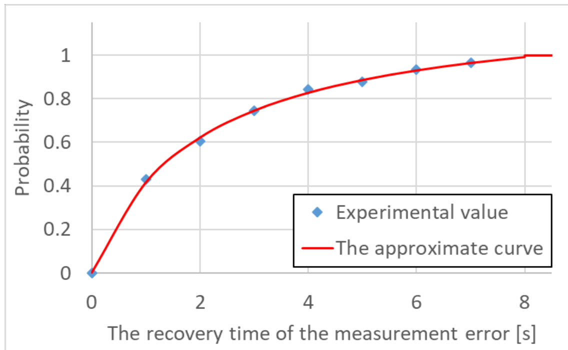

FIGURE 7. The cumulative distribution of the measurement error recovery time.

First, we focus on sender surplus error correction. Sender surplus errors come in two types (packet loss and measurement error) but the type cannot be identified when the surplus occurs. Nevertheless, we need to decide its type and obtain the exact number of packet losses to calculate QoE at this point. For this decision, the two characteristics identified in the previous section can be useful. If the number of sender surplus packets is two or more, we can treat that number minus one as a packet loss although the remaining packet might be still a measurement error. We then calculate the possibility of measurement errors for the remaining packet based on the fact that the expected value of the packet loss gets larger with the passage of time (see. Figure 7). That is why we calculate the packet loss fraction in relation to the measurement error possibility (Eq. (8)), i.e., calculating $F(t)$, and adding the calculated value from $F(t)$ to the number of packet losses. When the time after surplus occurrence exceeds eight seconds, it is handled as a 100% packet loss, i.e., we add one packet to the packet loss number. On the other hand, if the sender surplus is compensated for by the arrival of another receiver surplus, we do not treat the sender surplus as lost.

In the case of a receiver surplus, a measurement error can be clearly identified because the number of received packets on a receiver-side OFS must be equal to or less than the number of sent packets on the sender-side OFS. This is why we use the subsequent sender surplus to compensate for the receiver surplus, even though the sender surplus used here might still be due to packet losses. In such cases, the next send-surplus caused by measurement error will be treated as packet losses if they are not recovered within eight seconds. Hence, although a short delay may occur, actual packet loss can be detected.

V. EXPERIMENTAL EVALUATION

In this section, we evaluate the two-stage in-network QoE estimation method in a real environment. To evaluate the preciseness of estimated QoE, we compare with the true value measured by the end hosts based on G.1071.

A. EVALUATION FOR EARLY QoE ESTIMATION ON THE FIRST STAGE

Since the aim of this experiment is to show the effectiveness of the early QoE estimation, we began by comparing the estimated QoEs with the true values obtained from the end hosts, and then showed that the QoE-based route selection successfully improved the QoE of all flows transmitted on the SDN-enabled network. For these evaluations, we prepared two scenarios with different purposes.

1) COMPARATIVE METHOD

We use a throughput-oriented route selection scheme [3] that works on the same SDN-enabled network used this study as a comparative method. This method avoids packet losses for newly arriving flows by temporarily forwarding the flow onto the route with the largest residual bandwidth because its required throughput is unknown at arrival. After the OFC measures the flow throughput by exploiting FlowStats, the OFC redirects the flow to the route with the smallest, but still sufficient, residual bandwidth. This throughput-oriented mechanism is designed to ensure network resources are used as efficiently as possible.

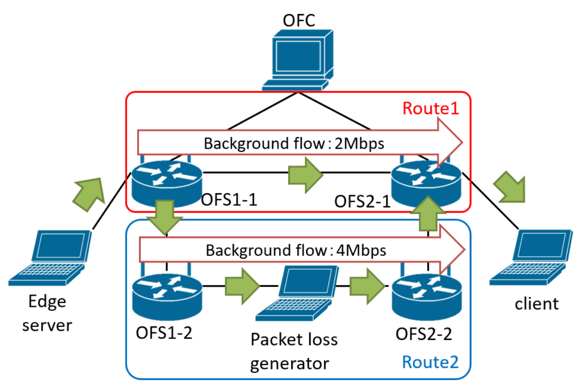

FIGURE 8. Environment for early estimation experiment.

2) EXPERIMENTAL SETTINGS

The experimental topology is shown in Figure 8. Regarding OpenFlow, we used Trema as the OFC controller software and installed Open vSwitch on each OFS. All devices were connected via a 100 Mbps Ethernet. We placed a personal computer (PC) running the Linux operating system (OS) between OFS 1-2 and OFS 2-2 to generate 1% random packet losses. Next, we defined the route from OFS 1-1 to OFS 2-1 as Route 1 and from OFS 1-2 to OFS 2-2 as Route 2. In addition, we prepared two background traffic flows, a 2 Mbps flow through Route 1 and a 4 Mbps flow through Route 2, to create an imbalanced residual bandwidth situation between the two routes. We then evaluated how the QoEs of all flows varied when several flows, including video streaming flows, were transmitted from the edge server to the client.

3) SCENARIO 1: EVALUATION IN CASE OF ONLY VIDEO FLOW

In this scenario, we show how the proposed method improves the QoE of the incoming video streaming flow. We also evaluate the accuracy of the estimated PLR, video bitrate, and QoE. To accomplish this, we transmitted a video flow from the edge server (PC 1) to the client (PC 2) and then evaluated the early QoE estimation and route selection results for the first stage. In this experiment, two kinds of video (3,250 kbps and 10,000 kbps) were transmitted, and the scenario was repeated nine times.

TABLE 6. The true QoE value after the route selection in Scenario 1.

Video bitrate	Proposed method			Comparative method		
	Median	Max.	Min.	Median	Max.	Min.
3,250 kbps	4.395	4.395	1.317	1.311	1.328	1.285
10,000 kbps	4.391	4.391	1.302	1.321	1.345	1.307

Table 6 shows the true QoE value after the first stage route selection. When the comparative method is used, even though a newly arriving flow is first transmitted on Route 1, which has the largest residual bandwidth, video flow is later switched to Route 2 to utilize the maximum available residual bandwidth on that route. However, after switching, the video flow QoE drops drastically to around 1.3 because of Route 2 packet losses. In contrast, in our proposed method, the video flow is continuously transmitted on Route 1, even after the route selection based on estimated QoE. Hence, the median and maximum values of QoE are around 4.3, which is good quality. However, the minimum QoE value is around 1.3. This is because PLR measurement errors, which cannot be corrected in Section IV-A1, degrade the estimated QoE. As a result, switching the video flows to Route 2 causes packet losses and degrades the QoE.

TABLE 7. Estimated and true PLR value in Route 1.

Video bitrate	Estimated value			True value
	Median	Max.	Min.	
3,250 kbps	0%	1.622%	0%	0%
10,000 kbps	0%	3.125%	0%	0%

To investigate the PLR estimation, Tables 7 shows the estimated and true PLR values in Route 1. Although no packet losses occur, the estimated PLR in the maximum value case is overestimated by 1% more than the true value, which means it degrades the QoE score. This gap, which was caused by failing to correct the errors introduced in Section IV-A1, occurs because the first stage of the proposed method only corrects receiver surplus measurement errors. However, the problem is resolved by the detailed QoE estimation for the ongoing flow.

Next, focusing on the video bitrate shown in Table 8 and 9, we can see that the estimated video bitrate for 3,250-kbps video is almost the same as the true value, while it is lower

TABLE 8. Estimated and true value of video bitrate in Route 1.

Video bitrate	estimated value			True value
	Median	Max.	Min.	
3,250 kbps	3,128 kbps	3,317 kbps	3,057 kbps	3,500 kbps
10,000 kbps	5,339 kbps	5,872 kbps	4,086 kbps	10,000 kbps

TABLE 9. The estimated and true value of QoE in Route 1.

Video bitrate	estimated value			True value
	Median	Max.	Min.	
3,250 kbps	4.720	4.726	1.286	4.395
10,000 kbps	4.732	4.747	1.393	4.395

than the true value for a 10,000-kbps video. This is because the number of bytes measured in the OFC is limited due to its processing delay. However, the difference between the true value and median/maximum estimated QoE score is only around 0.3, which is almost the same as the true value because the video bitrate has less impact on the QoE.

These results demonstrate that the first stage of the proposed method can improve video QoE by switching to a more appropriate route based on the estimated QoE. Although there are cases in which the estimated accuracy declines, this does not present a problem because the second stage of the proposed method can accurately estimate QoE and modify the flow route based on the corrected results.

4) SCENARIO 2: EVALUATION IN CASE OF MULTIPLE FLOWS

In this scenario, we demonstrate the effectiveness of the first stage of our proposed method in case of multiple new flow arrivals and using the estimated QoE for route selection. For this evaluation, we transmit three flows in the order of Video 1, Video 2, and the File Transfer (FT) flows at 5-second intervals. The bitrate of both Video 1 and 2 is set to 2,500 kbps while the FT is a simple TCP file transfer. Since we assume that OFC can precisely estimate QoE of FT based on throughput, we use the true value of the QoE calculated following the method described in [22] for FT route selection. Although we repeated this experiment nine times, we will limit our discussion to the median of the obtained experimental values.

Figure 9 shows time-series results for the true QoE value when the comparative method is used. Although a newly arriving video flow is first transmitted on Route 1, which has the largest residual bandwidth, the flow is then switched to Route 2 based on the throughput-oriented mechanism. As a result, the Video 1 QoE is high at the flow arrival time but drops drastically after route selection due to packet losses on Route 2. For Video 2, since the comparative method performs the same flow management as used with Video 1, the flow QoE declines. Finally, for the FT, the comparative method initially selects Route 1 but does not change the route because the FT cannot be transmitted on Route 2 without packet losses. As a result, the FT could maintain high QoE but both video flows could not when the comparative method was used.

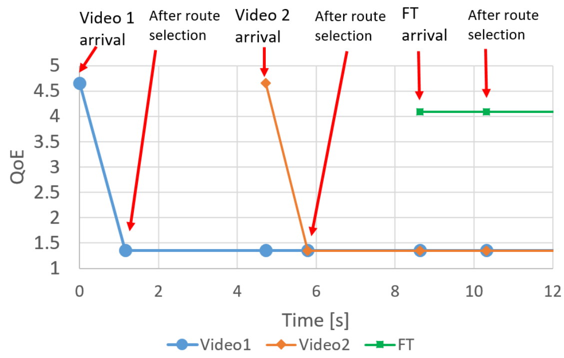

FIGURE 9. Scenario 2 QoE transition using the comparative method.

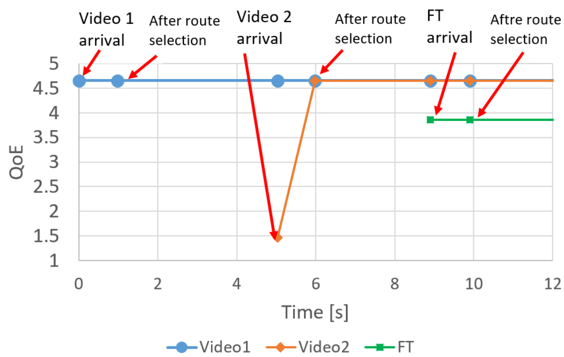

FIGURE 10. Scenario 2 QoE transitions using the proposed method.

Figure 10 shows the time-series results for the true QoE value when the proposed method is used. Here, it can be seen that Video 1 is first transmitted on Route 1, as with the comparative method, but is kept on the same route because the estimated Route 1 QoE is determined to be the best. Upon arrival, Video 2 is first transmitted on Route 2 because that route has the largest residual bandwidth at that time. Thus, the Video 2 QoE is low at arrival, but since the proposed method switches Video 2 to Route 1 in accordance with its QoE estimate, the QoE clearly improves one second after the flow starts. Finally, the FT is first transmitted on Route 2, which has the largest residual bandwidth and does not change routes because the FT's QoE is not affected by the few losses that occur. Hence, the FT QoE remains around 3.5. Taken together, these results show that the proposed method can keep the QoEs of every flow at an excellent level. We also note that our first stage QoE estimation is useful for QoE-based route selection and can improve QoEs for multiple new flow arrivals.

However, the proposed method still faces limitations. For example, it does not consider that the QoE of existing flows may drop due to the transmission of a new flow. Moreover, unlike the comparative method, the proposed method does not prepare a route with large residual bandwidth, which means that the possibility of bandwidth scarcity at flow arrivals is somewhat higher than that of the comparative method.

B. EVALUATION FOR PRECISE QoE ESTIMATION ON THE SECOND STAGE

This experiment will show the effectiveness of QoE estimations for ongoing flows during the second stage. More specifically, we will compare our proposed method using not only the true value but also the Section IV-A1 error correction of the comparative method. For this evaluation, we prepared three experimental scenarios with different evaluation purposes.

1) EXPERIMENTAL SETTINGS

Since we only estimate the QoEs of ongoing flows in the second stage, we an experimental topology with a single route, as shown in Figure 11. The other settings used are the same as those used in Section V-A2. In this experiment, video streaming is transmitted from the edge server to the client. The video was made using the H.264 codec with SD (720 x 480), 30 fps, and 2.5 Mbps CBR of video bitrate.

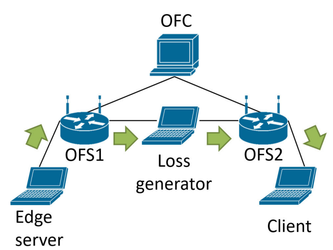

FIGURE 11. Experiment environment for detailed estimation.

2) SCENARIO 1: NO PACKET LOSS

In this scenario, we compared the preciseness of the proposed estimation without packet losses. Figure 12 shows the PLR results and the accumulated values of receiver surplus packets of the comparative method, and Figure 13 shows the QoE time-series results. In Figure 12, it can be seen that the PLR of the comparative method is between 17 and 24 seconds larger than the true values. This is because the OFC treats the measurement error as packet losses because there are no accumulated surplus packets and the sender surplus occurs at 17 seconds. In addition, since the estimated QoEs are calculated by using the data for the latest eight seconds, the estimated QoE of the comparative method is significantly smaller than the true value for eight seconds (Figure 13). Although the estimated QoE of the comparative method after 25 seconds is almost the same as the true value, the receiver surplus that is actually the recovered sender surplus at 17 seconds is added to the receiver surplus packets (at 19 seconds of Figure 12), and used to correct other errors (at 23, 56, and 58 seconds of Figure 12). Therefore, if packet losses occur after 19 seconds, they will be modified by mistake, which exacerbates the difference between the true value and the estimated QoE of the comparative method.

On the other hand, although measurement errors occur at 17 and 23 seconds in the results obtained with the proposed method, the difference between the estimated QoE and the true value is smaller than that between the true value and the comparative method (Figure 12 and 13). This is because the proposed method updates the PLR every second by exploiting the cumulative distribution function (Section IV-B1, Figure 7). In addition, the period at which the measurement error affects the QoE is shorter than the comparative method because the PLR is updated to zero when the measurement error recovers, as seen at the 18- and 23-second marks of Figure 12. As a result, the proposed method has better estimation preciseness than the comparative method in a no packet loss environment.

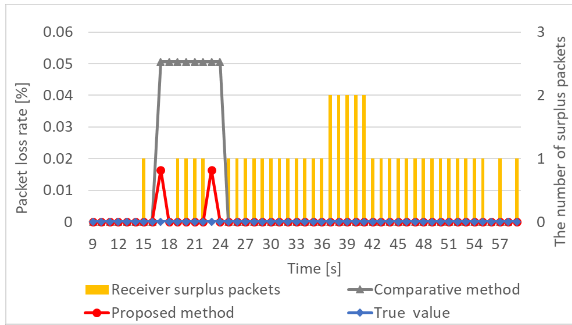

FIGURE 12. PLR results and the receiver surplus packets of the comparative method in a no packet loss environment.

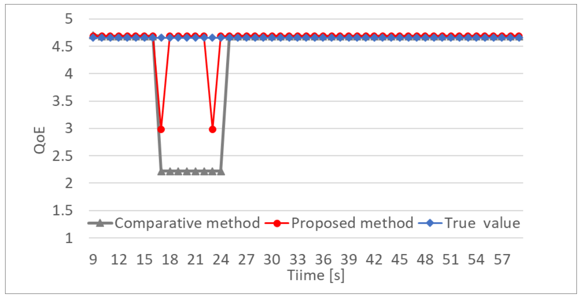

FIGURE 13. QoE in a no packet loss environment.

3) SCENARIO 2: 0.1% PACKET LOSS

At a preset PLR of 0.1%, Figure 14 shows the PLR result and the accumulated values of receiver surplus packets when using the comparative method, while Figure 15 shows the time-series QoE results. These figures show that the packet loss is incorrectly modified using the receiver surplus packets and that the estimated PLR is 0% at from nine to 17 seconds and from 29 to 33 seconds. Thus, the estimated QoE of the comparative method is larger than the true value because there is an insufficient distinction between packet losses and measurement errors.

In the proposed method, we exploit the cumulative distribution function to distinguish between packet losses and measurement errors. The difference with the true value decreases over time (from nine to 13 seconds as shown in Figure 14). At from nine to 17 seconds and from 29 to 33 seconds, where the packet loss is modified by mistake in the comparative

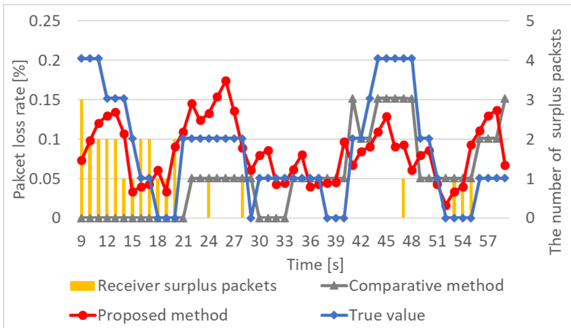

FIGURE 14. PLR result and the receiver surplus packets of the comparative method at 0.1% packet losses.

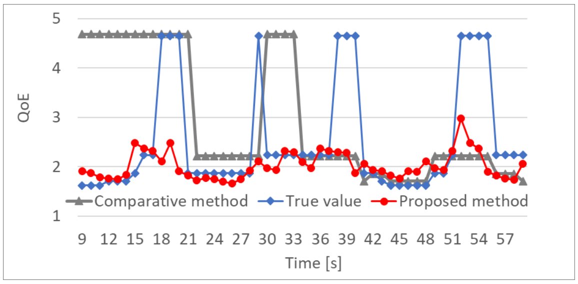

FIGURE 15. QoE results at 0.1% packet loss.

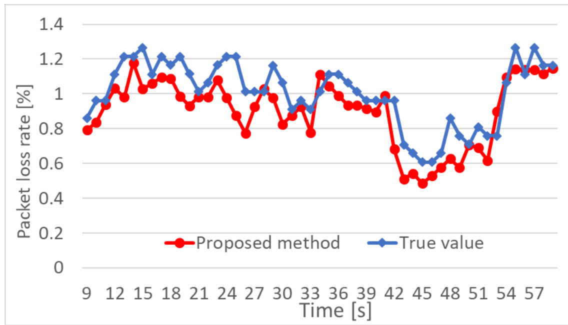

FIGURE 16. PLR results in 1% packet loss environment.

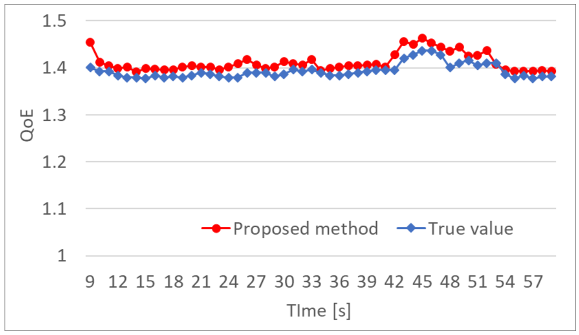

FIGURE 17. QoE results in 1% packet loss environment.

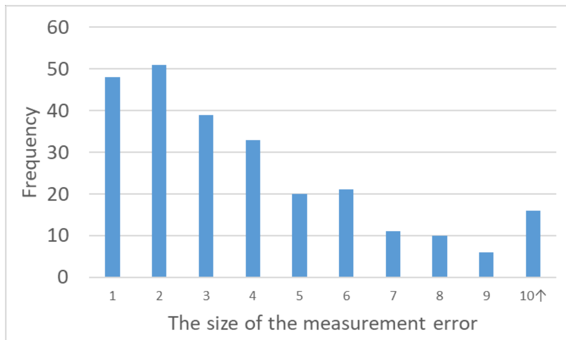

FIGURE 18. Measurement error recovery time frequency in the case of 64 Mbps.

method, the estimated PLR of the proposed method has a difference of less than 0.13% from the true value and the QoE difference is less than 0.5. This indicates that the proposed method has a good level of accuracy. On the other hand, even though the true value of the PLR temporarily becomes to 0% at 18, 29, 38, and 52 seconds in Figure 14, the proposed method does not follow that trend because it takes a short time (at most eight seconds) to determine whether it is encountering a packet loss or a measurement error. While this means that the proposed method cannot find a temporary recover of the network condition in a short period, we can still say that the proposed method not only has good estimation performance but also provides a high level of stability because using network control to recover the temporal network condition may lead to worse consequences.

4) SCENARIO 3: 1% PACKET LOSS

At 1% packet loss, Figure 16 shows the PLR result and receiver surplus packets of the comparative method, while Figure 17 shows the QoE time-series results. Note that we only compared the proposed method with the true value because a receiver surplus does not occur due to numerous packet losses and the comparative method does not work at all.

Since the proposed method requires a little time to determine one packet as a packet loss, the estimated PLR is slightly smaller than the true value. This means that the estimated QoE is larger than the true value, but the difference is at most 0.05. From the Scenario 2 and Scenario 3 results, we can see that the proposed method provides good performance even in a packet loss environment.

C. DISCUSSION

We used several variables with fixed values in Equation (1) and found that some of those had a relatively large impact on the QoE score. Although the largest factor affecting the QoE score is the PLR, the average number of consecutive packet losses is the second most influential factor. When we measured this parameter in the network, we could further improve our estimated QoE scores. One potential future improvement might be a collaboration between wireless and conventional network technologies. In that scenario, as the wireless technology monitors frame-level losses, the network may produce better QoE estimates with lower measurement results if the

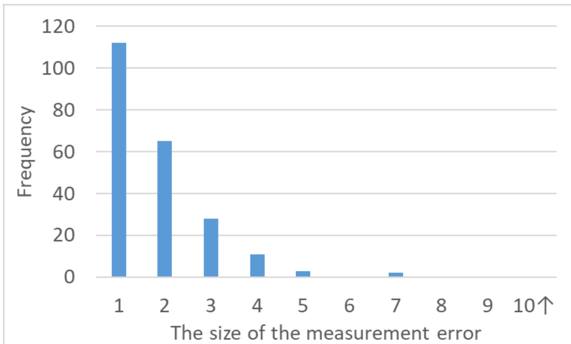

FIGURE 19. Measurement error recovery time frequency in the case of 24 Mbps.

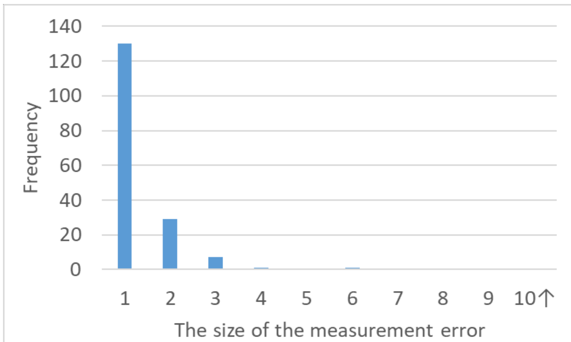

FIGURE 20. Measurement error recovery time frequency in the case of 12 Mbps.

wireless and conventional network measurements could be synchronized. However, the fact that those network types run on a different time granularity is likely to be an obstacle to applying this solution.

As the second stage measurement errors characteristics discussed in Section IV-B1 may change depending on the environment, we investigated the environmental adaptability of those characteristics. Figure 18, 19 and 20 shows the quantity of measurement errors in the cases of 64, 24 and 12 Mbps of video flow. Here, we can see that larger video bitrates result in larger quantities of measurement errors. In the case of transmitting a video with a large bitrate, the quantity of measurement error is not one packet, so the second stage error correction cannot be used. To extend that error correction for a larger bitrate, a method for differentiating packet losses from measurement errors based on resolution is necessary, which means it will be important to find a way to identify the resolution.

As the time gap of FlowStats collection timing is one millisecond at most on a stable control plane of an OpenFlow network, the theoretical maximum measurement error quantity per millisecond can be theoretically calculated as follows:

$$e_{\max} = v_{\mathrm{br}}/0.008s, \tag{9}$$

where $e_{\max}$ [packet] is the maximum quantity of the measurement error per millisecond, v_{br} [bps] is the video bitrate, and s [byte] is packet size. Note that we assume a 1500-byte packet size. In the case of $v_{\mathrm{br}} = 12$, $p_{\max}$ became one packet, which means the measurement error quantity is one packet in less than 12 Mbps, which is the recent streaming video recommended resolution for 1080p on YouTube and has thus been adopted in our proposed method.

VI. CONCLUSION

In this paper, we focused on QoE measurements from the viewpoint of network resource management. Although QoE score measurements of all transmitting flows by network components in an edge network are essential to achieving efficient QoE-driven network resource management, the following four problems must be addressed. First, in-network components cannot directly measure QoE because it is normally measured at an end-host. Second, some application parameters needed for QoE measurement such as video image information cannot be collected by in-network components. Third, since the QoE calculation standardized by ITU requires relatively long-duration measurements, quick network resource management is difficult. Fourth, video QoE scores suffer significantly even from a few measurement errors.

To address these problems, we estimated QoE inside a network by exploiting a two-stage method. The first stage is an early QoE estimation for new arrival flows. In this stage, we focus on a prompt but rough QoE estimate that is used to forward the flow onto an appropriate route as soon as possible. We could correct measurement errors promptly by correcting only the sender surplus which is found easily. To estimate the application parameters early, we exploited the difference between the number of transmitted bytes for 0.9 seconds in order to avoid video buffering. The second stage is a more precise QoE estimation for the ongoing flow that is produced by using stable and reliable statistics observed for a longer period. By exploiting the measurement error characteristics, we dynamically updated the estimated QoE once each second.

We then conducted experiments to evaluate each of the two stages. The results of those experiments showed that the first stage had sufficient accuracy to perform initial route selection and that it could maintain a high QoE. In the second stage, we showed that our proposed method could estimate QoEs precisely with/without packet losses.

Since our study treats video applications following ITU-T G.1071 standard only, other video standards and applications need to be considered. If other video standards require other parameters except for PLR to calculate their QoE, a measurement way and its error correction in terms of these parameters should be considered. On the other hand, our study has a potential to be adapted for other applications employing PLR because it can precisely correct PLR error. We will tackle applying the specific model and evaluation for other video standards and their applications.

In this study, single OFC estimates QoE for all transmitting flows on an edge network. Hence, although we assume that our method performs on the edge network with a relatively small scale, scalability issue will be raised when the number of transmitting flows and hosts is increased. We will work on evaluation of the number of flows and devices that can be accommodated and extending scalability.

The next step of this study will be flow control based on the estimated QoE. If we switch a flow onto a different route where various flows already coexist, the QoEs of the existing flows on that route may be degraded. Therefore, we need to develop a new flow control (resource management) method that can consider the QoE score of not only a specific flow but also all the transmitting flows in the entire edge network. To solve this problem, we will attempt to apply machine learning (ML) to flow control because it is believed that ML might be able to select an appropriate route that can further improve long-term as well as short-term QoEs.

REFERENCES

[1] *Cisco Annual Internet Report (2018–2023) White Paper*. Accessed: Dec. 10, 2020. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[2] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—A key technology towards 5G," Eur. Telecommun. Standards Inst. (ETSI), Sophia Antipolis, France, White Paper 11.11, 2015, pp. 1–16.

[3] M. Tagawa, Y. Taenaka, K. Tsukamoto, and S. Yamaguchi, "A channel utilization method for flow admission control with maximum network capacity toward loss-free software defined WMNs," in *Proc. 14th Int. Conf. Netw. (ICN)*, Feb. 2016, pp. 118–123.

[4] H. Xu, X.-Y. Li, L. Huang, H. Deng, H. Huang, and H. Wang, "Incremental deployment and throughput maximization routing for a hybrid SDN," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1861–1875, Jun. 2017.

[5] Y. Lu, B. Fu, X. Xi, Z. Zhang, and H. Wu, "An SDN-based flow control mechanism for guaranteeing QoS and maximizing throughput," *Wireless Pers. Commun.*, vol. 97, no. 1, pp. 417–442, Nov. 2017.

[6] *Vocabulary for Performance, Quality of Service and Quality of Experience*, Standard Recommendation ITU-T P.10/G.100 Amendment 1, Jun. 2019.

[7] *Opinion Model for Network Planning of Video and Audio Streaming Applications*, Standard Recommendation ITU-T G.1071, Nov. 2016.

[8] *Parametric Bitstream-Based Quality Assessment of Progressive Download and Adaptive Audiovisual Streaming Services Over Reliable Transport*, Standard Recommendation ITU-T P.1203, Oct. 2017.

[9] *QoE Factors in Web-Browsing*, Standard Recommendation ITU-T G.1031, Feb. 2014.

[10] W. Robitza, A. Ahmad, P. A. Kara, L. Atzori, M. G. Martini, A. Raake, and L. Sun, "Challenges of future multimedia QoE monitoring for Internet service providers," *Multimedia Tools Appl.*, vol. 76, no. 21, pp. 22243–22266, Nov. 2017.

[11] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008.

[12] T. Zinner, M. Jarschel, A. Blenk, F. Wamser, and W. Kellerer, "Dynamic application-aware resource management using software-defined networking: Implementation prospects and challenges," in *Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, May 2014, pp. 1–6.

[13] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race, "Towards network-wide QoE fairness using openflow-assisted adaptive video streaming," in *Proc. ACM SIGCOMM Workshop Future Hum.-Centric Multimedia Netw.*, Aug. 2013, pp. 15–20.

[14] A. Khan, L. Sun, and E. Ifeachor, "QoE prediction model and its application in video quality adaptation over UMTS networks," *IEEE Trans. Multimedia*, vol. 14, no. 2, pp. 431–442, Apr. 2012.

[15] H. Nam, K.-H. Kim, J. Y. Kim, and H. Schulzrinne, "Towards QoE-aware video streaming using SDN," in *Proc. IEEE Global Commun. Conf.*, Dec. 2014, pp. 1317–1322.

[16] V. Joseph and G. D. Veciana, "NOVA: QoE-driven optimization of DASH-based video delivery in networks," in *Proc. IEEE INFOCOM-IEEE Conf. Comput. Commun.*, Apr. 2014, pp. 82–90.

[17] S. Petrangeli, J. Famaey, M. Claeys, S. Latré, and F. D. Turck, "QoE-driven rate adaptation heuristic for fair adaptive video streaming," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 12, no. 2, pp. 1–24, Mar. 2016.

[18] A. Farshad, P. Georgopoulos, M. Broadbent, M. Mu, and N. Race, "Leveraging SDN to provide an in-network QoE measurement framework," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2015, pp. 239–244.

[19] O. Dobrijevic, A. Kassler, L. Skorin-Kapov, and M. Matijasevic, "Q-POINT: QoE-driven path optimization model for multimedia services," in *Wired/Wireless Internet Communications*. Cham, Switzerland: Springer, May 2014, pp. 134–147.

[20] J. Seppanen and M. Varela, "QoE-driven network management for real-time over-the-top multimedia services," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2013, pp. 1621–1626.

[21] YouTube. *Recommended Upload Encoding Settings*. Accessed: Dec. 4, 2020. [Online]. Available: https://support.google.com/youtube/answer/1722171?hl=en

[22] S. Thakolsri, S. Khan, E. Steinbach, and W. Kellerer, "QoE-driven cross-layer optimization for high speed downlink packet access," *J. Commun.*, vol. 4, no. 9, pp. 669–680, Oct. 2009.

SHUMPEI SHIMOKAWA received the B.E. degree in computer science from the Kyushu Institute of Technology (KIT), Japan, in 2019. He has been the Student with the Graduate School, KIT, since April 2019. His research interest includes resource management of computer networks.

YUZO TAENAKA (Member, IEEE) received the D.E. degree in information science from the Nara Institute of Science and Technology (NAIST), Japan, in 2010. He was an Assistant Professor with The University of Tokyo, Japan. He has been an Associate Professor with the Laboratory for Cyber Resilience, NAIST, since April 2018. His research interests include information networks, cybersecurity, distributed systems, and software defined technology.

KAZUYA TSUKAMOTO (Member, IEEE) received the D.E. degree in computer science from the Kyushu Institute of Technology (KIT), Japan, in 2006. From April 2006 to March 2007, he was a Japan Society for the Promotion of Science (JSPS) Research Fellow with KIT. In 2007, he was an Assistant Professor with the Department of Computer Science and Electronics, KIT, and then has been an Associate Professor with the Department of Computer Science and Electronics since April 2013. His research interests include performance evaluation of computer networks and wireless networks. He is currently a member of ACM, IPSJ, and IEICE.

MYUNG LEE (Senior Member, IEEE) received the B.S. and M.S. degrees from Seoul National University and the Ph.D. degree from Columbia University. He is currently a Professor with the Department of Electrical and Computer Engineering, The City University of New York. His current research interests include secure V2X communications, edge cloud resource management, the IoT, machine learning, and stochastic computing application to intrusion detection. He publishes extensively in these areas and hold over 25 U.S. and international patents. His research group developed the first NS-2 simulator for IEEE 802.15.4, a standard NS-2 distribution widely used for wireless sensor network research. He co-received three best paper awards. He actively contributed to international standard organizations IEEE (TG chairs for 802.15.5 WPAN Mesh and 802.15.8 PAC) and ZigBee. He is also a Past President of KSEA.