IEICE TRANS. COMMUN., VOL.E107–B, NO.1 JANUARY 2024

PAPER

Quality and Transferred Data Based Video Bitrate Control Method for Web-Conferencing

Masahiro YOKOTA[†a)] ***and*** **Kazuhisa YAMAGISHI**[†b)], ***Members***

SUMMARY In this paper, the quality and transferred data based video bitrate control method for web-conferencing services is proposed, aiming to reduce transferred data by suppressing excessive quality. In web-conferencing services, the video bitrate is generally controlled in accordance with the network conditions (e.g., jitter and packet loss rate) to improve users' quality. However, in such a control, the bitrate is excessively high when the network conditions is sufficiently high (e.g., high throughput and low jitter), which causes an increased transferred data volume. The increased volume of data transferred leads to increased operational costs, such as network costs for service providers. To solve this problem, we developed a method to control the video bitrate of each user to achieve the required quality determined by the service provider. This method is implemented in an actual web-conferencing system and evaluated under various conditions. It was shown that the bitrate could be controlled in accordance with the required quality to reduce the transferred data volume.

key words: *web-conferencing, video bitrate control method, quality estimation*

1. Introduction

In recent years, the use of web-conferencing has drastically increased along with the promotion of telework. Many web-conferencing services are provided, such as Cisco Webex [1], Zoom [2], and Microsoft teams [3]. Since not only audio data but also video data are transferred in many cases, a large amount of data needs to be transferred to provide comfortable communication. However, an increase in the amount of transferred data leads to increased operational costs (e.g., data transfer cost for cloud service) and capital expenditures (e.g., the cost of communication equipment for on-premises). Therefore, to maintain the quality with which users are comfortable while reducing transferred data, service providers need to control the video bitrate of web-conferencing.

Web real-time communication (WebRTC) is a well-known technology for web-conferencing. WebRTC is standardized by the World Wide Web Consortium (W3C) [4] and the Internet Engineering Task Force (IETF) [5] and provides browser-based and mobile application-based real-time communication. WebRTC is widely used in web-conferencing, and many types of open source software (OSS) are provided [6]–[8]. In WebRTC, clients communicate peer-to-peer, so the processing load on the end-user's device increases when many users participate in the meeting. As a result, the number of users is limited.

To address the issue with the limitation of users, multipoint control unit (MCU) [9], and selective forwarding unit (SFU) are proposed. In MCU, the media data uploaded by all clients is synthesized at the MCU server and delivered to each client as one stream. This process reduces the number of processing streams for each client, thereby reducing the load on the client. On the other hand, the MCU server requires a high processing capability because video synthesis processing is needed. In SFU, the media data is delivered to each client without a synthesis process at the SFU server. The processing stream of the client is larger than that of the MCU, so the processing load of the client is increased. However, the processing capability of the SFU server is not required because the video synthesis process is not performed. Therefore, the equipment cost of SFU is lower than that of MCU, so SFU is widely used. Simulcast is one of the quality improvement methods at SFU. In simulcast, the sending client delivers multiple-quality videos to the SFU server, and the SFU server selects the suitable video bitrate for each received client in accordance with the received client's network condition (i.e., available bandwidth). Therefore, although quality can be improved, each client's uploaded data increases.

Many bitrate control methods for web-conferencing have been proposed [10]–[12]. Google congestion control (GCC) [10], [11] is one of the most common congestion control algorithms in WebRTC. GCC controls the quality of video streams on the basis of network conditions (e.g., jitter and packet loss rate). However, it may increase the amount of transferred data because excessive quality is realized in extremely high network conditions (e.g., high throughput and low jitter).

To address the above issues, the video bitrate needs to be controlled to the appropriate quality to reduce the amount of transferred data. To do this, a bitrate control method for SFU was proposed that considers video quality and transferred data [13]. The reasons for targeting SFU are as follows: 1) MCU can send multiple quality levels of synthesized video depending on the network conditions to each client. However, the hardware cost of the MCU server is high if many users' videos need to be synthesized, 2) Simulcast can improve/control quality by sending multiple quality levels of video from each client. However, upload data from each client is increased because each client needs to send multiple quality levels of video to the server, and 3) SFU does not

Manuscript received May 17, 2023.
Manuscript revised August 23, 2023.
Manuscript publicized October 13, 2023.
[†]The authors are with the Network Service Systems Laboratories, NTT Corporation, Musashino-shi, 180-8585 Japan.
a) E-mail: masahiro.yokota@ntt.com
b) E-mail: kazuhisa.yamagishi@ntt.com
DOI: 10.1587/transcom.2023EBP3079

need server resources and can send a suitable video bitrate depending on network conditions. Therefore, targeting SFU is the best way to control the quality and transferred data. In our proposed method, first, the service provider sets the required quality to avoid excessive quality. After the start of web-conferencing, the quality of each received video is estimated by using multimedia quality information such as bitrate, framerate, and resolution of all received streams for each sending client and display device type (e.g., laptop or smartphone). In addition, integrated quality on the screen (i.e., overall quality) is estimated by using the quality of each received video. On the basis of the overall quality, the video bitrate of the next interval (e.g., 1-second future) is determined, so that future quality approaches the required quality.

In the previous study [13], the evaluation conditions were limited. In other words, conditions under the bandwidth fluctuation condition and the influence of the various variables were not evaluated. Therefore, this study evaluates their viewpoints and shows the effectiveness of this method.

The remainder of this paper is structured as follows. In Sect. 2, the related work is summarized, and the novelty of this study is shown. The proposed method is described in Sect. 3. The evaluation set is shown in Sect. 4, and the results are shown in Sect. 5. Finally, the conclusion of this paper is presented in Sect. 6.

2. Related Work

In this section, literature is described to summarize issues that need to be addressed in this paper. Especially, the bitrate control algorithm and quality-estimation algorithm are described because these algorithms are needed to control the bitrate and quality.

2.1 Bitrate Control Algorithm

Transferred data (i.e., bitrate) is controlled in web-conferencing services such as Webex and Zoom. For example, the user cannot use a high-quality mode such as 1280 $\times$ 720p and 640 $\times$ 360p unless the user changes the default settings in Webex [1], and users can use a high-quality mode such as 1280 $\times$ 720p only in a specific case (e.g., a small number of users, joining users with specific licenses) in Zoom [2]. These methods can reduce the amount of transferred data. However, it cannot be provided even when the high-quality video is required, because the upper resolution was limited to reduce the transferred data volume.

Bitrate adaptation methods for WebRTC have been studied [10]–[12], [14], [15]. GCC [10], [11], which is one of the well-known methods, controls bitrate on the basis of jitter and packet loss rate. A lower bitrate is selected when the jitter or packet loss rate increases (i.e., congestion has occurred). Wu et al. proposed a method to solve the issue that GCC misjudges small delay fluctuations as congestion and fails to perform appropriate control in accordance with network quality [14]. Concretely, the bitrate control based on the jitter was not performed in accordance with the value calculated by inputting the packet loss rate and round-trip time (RTT) into the formula defined for each network line (i.e., Wi-Fi and 4G). With this control, quality degradation due to misjudgment of congestion is avoided, and quality is improved. Wang et al. proposed a scalable video coding (SVC) layer bitrate selection algorithm for multiparty interactive live video streaming to maximize the total quality of the receiver on the basis of the sender's uplink throughput, receiver's downlink throughput, and buffer occupancy [12]. Petrangeli et al. proposed a video bitrate control method for WebRTC-based remote teaching applications that maximizes the video bitrate transmitted to the receiver, taking into account the available bandwidth of the receiver [15]. This method can control the video bitrate in response to bandwidth fluctuations. Although these methods [10]–[12], [14], [15] can improve the quality, the amount of transferred data cannot be reduced because excessive quality is provided when the network quality is high enough.

Adaptive bitrate algorithms also have been extensively studied in video streaming services [16]. Previous studies of adaptive bitrate algorithms can be categorized into bandwidth-based [17]–[19], buffer-based [20], [21], and hybrid [22], [23]. In web-conferencing, a small buffer size is used to provide real-time communication. Therefore, the buffer-based and hybrid methods, which need a large buffer size, are not suitable for real-time communication.

The bandwidth-based approach estimates future network bandwidth and selects the bitrate on the basis of the estimated bandwidth [17]–[19]. Miller et al. proposed a bitrate control method to improve the quality of live streaming services [17]. This method uses short-time estimated throughput to select a bitrate at which the download success probability exceeds a threshold. This makes it possible to select a high bitrate while suppressing the rebuffering. Xie et al. proposed a video bitrate selection method based on the bandwidth estimation technique using physical layer information [18]. This method can improve video quality degradation by the effect of bandwidth fluctuation of long-term evolution (LTE). In this method, even in the case of a large bandwidth fluctuation, such as in the LTE network, the bandwidth can be estimated with high accuracy considering the information of the PHY layer, and using the estimated throughput, the bitrate is selected considering the video quality and rebuffering. Although these studies [17], [18] aimed to reduce rebuffering and improve quality, they cannot reduce the amount of transferred data. Therefore, the issue of transmitting excessive data in a high-bandwidth environment remains.

Li et al. proposed a bitrate control algorithm to solve the problem that network bandwidth estimation is sometimes not successful in an environment where multiple streams of HTTP adaptive streaming (HAS) are competing, resulting in poor bandwidth utilization and fairness [19]. Specifically, this method probes the network bandwidth while varying the data rate, thereby determining the video bitrate and the interval between segment download requests. This method can improve bandwidth utilization and fairness. However, as in previous studies, excessive data transmission remains an issue in high-bandwidth environments.

Some previous studies reduce the transferred data for video streaming services. Kimura et al. studied a bitrate control method for adaptive bitrate streaming services considering the quality and the amount of transferred data [24]. This method focuses on the fact that some customers value the amount of data over the quality, maintains the quality at the target quality set by the user, and minimizes the amount of transferred data. However, this method is targeted at video streaming services, so it uses the buffer occupancy information because it is difficult to apply the web-conferencing services that have high real-time property. In addition, this method controls one video stream, so it cannot be applied directly to web-conferencing services that need to control multiple streams.

As described above, the purpose of the previous bitrate control method for web-conferencing is to improve quality, but reducing transferred data is not considered. A bitrate control method for video streaming services considers the quality and transferred data but cannot be applied directly to web-conferencing services. In summary, a method to control the bitrate of multiple streams by considering the quality and transferred data without using buffer information has not been investigated.

2.2 Quality-Estimation Algorithm

The quality-estimation model can be categorized into media-based [25], [26], metadata-based [27], [28], and bitstream-based models [27], [29], in accordance with the input information. The media-based model takes pixel signal, the metadata-based model takes information such as bitrate, framerate, and resolution, the bitstream-based model takes bitstream to parse quantization parameters. Since the proposed method needs to process many web-conferencing streams simultaneously, it is not realistic to use media-based and bitstream-based models because each pixel needs to be analyzed in media-based models, and bitstream needs to be parsed in bitstream-based models. The use of the metadata-based model in quality estimation is reasonable and feasible because it does not need high computational power.

There are several studies on the quality of web-conferencing. These studies show that many factors such as audio encoding quality, video encoding quality, the fluctuation of video encoding quality, delay, and synchronization of video and audio quality [30]–[35]. Since client applications can control the video bitrate, this study focuses on a quality-estimation model that considers audio and video quality.

Some quality-estimation models for web-conferencing have been proposed [35], [36]. Hayashi et al. proposed a quality-estimation model for one-to-one web-conferencing based on audiovisual quality, delay, and media synchronization [37]. However, the audiovisual quality model considers the bitrate, framerate, and packet loss rate but not resolution. Current web-conferencing services change the video resolution on the basis of the bitrate, so the resolution change should be considered. Jana et al. proposed a video-quality-estimation model for web-conferencing for two types of mobile applications [36]. Trace data using the actual device are collected, and the quality-estimation model for each mobile application is constructed on the basis of end-user movement (i.e., a state of movement such that participants join a meeting stationary or while moving), network delay, packet loss rate, and bandwidth. Their proposed quality-estimation model does not consider framerate or resolution either. In addition, this model requires end-user movement information that can not be easily extracted from packets.

As shown here, no video-quality-estimation model for web-conferencing considers bitrate, framerate, and resolution. The quality-estimation model for video streaming services is similar to that of web-conferencing in terms of video and audio quality. Therefore, the quality-estimation model for video streaming services is also investigated.

Some metadata-based quality-estimation models for video streaming services have been proposed [27], [28]. The P.1203 mode 0 model is standardized by ITU-T. This quality-estimation model takes audio bitrate, video bitrate, framerate, resolution, stalling, and initial loading that affects the user experience. The device type is also considered because of the effect of the display size on video quality.

This model can take into account resolutions, which have not been considered in previous estimation models for web-conferencing and can be used in this study.

Yamagishi et al. also proposed the quality-estimation model, and its inputs and outputs are the same as those of the P.1203 mode 0 model [28]. A previous study found that the P.1203 mode 0 model and Yamagishi model were equivalent in terms of quality-estimation accuracy [38]. On the other hand, the Yamagishi model has a lower calculation cost than the P.1203 mode 0 model because its formula is simpler. Therefore, the Yamagishi model is more suitable as a quality-estimation model for web-conferencing, which requires faster processing.

Generally, since the metadata-based quality-estimation model cannot catch the codec dependency on quality, the model's coefficients need to be optimized per codec. To address this issue, Yamagishi et al. proposed a coefficient optimization method [39].

Web-conferencing allows multiple users to join the same conference at the same time. Then, unlike a video streaming service, it simultaneously receives and displays streams for all users. Therefore, the quality of users needs to be estimated by integrating multiple video streams. A previous study investigated the quality impact of the screen contrast effect, which is the difference in quality between videos displayed at the same time [40]. Schmitt et al. clarified that the contrast effect exists when the quality is different in multiple displayed streams [40]. They conducted a subjective evaluation experiment using crowdsourcing to evaluate the effect of the quality of individual streams on the overall quality. As a result, the contrast effect was confirmed, but it was smaller than the effect of the difference in encoding quality and situation on average.

In summary, the conventional web-conferencing quality-estimation model cannot be used for bitrate control because not all combinations of bitrate, framerate, resolution, and device type are considered. However, since these parameters are used in some quality-estimation models for video streaming services, they can be used if the coefficients of models are optimized for web-conferencing codecs (e.g., VP8 and VP9). In addition, it needs to be extended to a model that can evaluate the impact of multiple streams on video quality.

3. Proposed Method

Our proposed method controls the video bitrate so that the quality exceeds the required quality set by the service provider. Figure 1 shows a block diagram of our proposed method, and all processing steps are described in the following sections. The parameters set by a service provider are defined in Table 1. In addition, the variables used in this paper are summarized in Table 2.

3.1 Modules of Proposed Method

3.1.1 Parameter Setting Module

A service provider needs to set the required quality (R), three control parameters (i.e., a selectable set of bitrates (S), quality-estimation duration (T), and interval of bitrate control (I)) to a controller in advance. These parameters send to the directed-bitrate-estimation module (3.1.3) and are used to determine the upper bitrate for each user ($p_i(t)$).

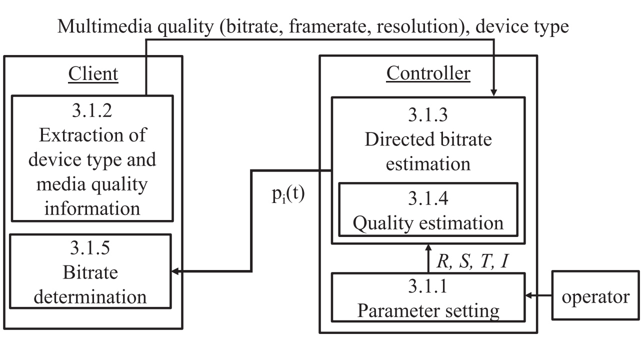


Fig. 1 Block diagram of proposed method.

Table 1 Parameters set by service provider.

parameter	description	example value
R	R represents the required quality that service provider want to provide to user. R is set with a mean opinion score (MOS) and takes a continuous value from 1 to 5.	3.5
S	S represents the selectable set of bitrates that the server can direct to the client.	128, 512 (kbps)
T	T represents quality estimation duration, which is the period of time considered in estimating quality.	60 (seconds)
I	I represents the interval of bitrate control.	1 (second)

3.1.2 Extraction of Device Type and Media Quality Information Module

After the web-conferencing starts, the user's device type (i.e., laptop (PC) or smartphone (SP)), which is derived from the User-Agent, is extracted at clients to control the quality of camera video (e.g., human's face) and audio. In addition, multimedia quality information (i.e., video and audio bitrate, framerate, and resolution) is extracted from WebRTC stats [41] at clients every 1 second and is sent to the controller. Note that although a screen-sharing stream (e.g., document and presentation materials) is used in web-conferencing, video bitrate of the stream is not controlled in this method because its bitrate is much lower than that of a camera video (i.e., showing each participant) if the variable bitrate (VBR) is used.

Table 2 Summary of the variables.

variable	description	example value
i	i represents the variable indicating specific user.	-
j	j represents the variable indicating specific stream.	-
N	N represents a number of users.	3
$ba_j(t)$	$ba_j(t)$ represents an audio bitrate in seconds per stream.	25 (kbps)
$bv_j(t)$	$bv_j(t)$ represents a video bitrate in seconds per stream.	250 (kbps)
$s_j(t)$	$s_j(t)$ represents a resolution in seconds per stream.	307200 (pels)
$r_j(t)$	$r_j(t)$ represents a framerate in seconds per stream.	30 (fps)
$A_j(t)$	$A_j(t)$ represents an audio quality in seconds per stream expressed as a MOS value from 1 to 5.	3.8
$V_j(t)$	$V_j(t)$ represents a video quality in seconds per stream expressed as a MOS value from 1 to 5.	3.2
$M_j(t)$	$M_j(t)$ represents an audiovisual quality in seconds of stream j expressed as a MOS value from 1 to 5.	3.5
$U_i(t)$	$U_i(t)$ represents an audiovisual quality in seconds of user i as calculated by $M_j(t)(j \in N, j \neq i)$, expressed as a MOS value from 1 to 5.	3.5
Q_i	Q_i represents long-term overall quality of user i, calculated by $U_i(t)$, expressed as a MOS value from 1 to 5.	3.5
$p_i(t)$	$p_i(t)$ represents the upper bitrate in accordance with R for each user as determined by the directed bitrate estimation module.	512 (kbps)
$g_i(t)$	$g_i(t)$ represents GCC bitrate that is determined by GCC in accordance with each user's network conditions.	256 (kbps)

3.1.3 Directed Bitrate Estimation Module

In this module, the upper bitrate for each user ($p_i(t)$) is selected from S so that the overall quality per user (Q_i) for T-seconds exceeds R and is directed to each client. A bitrate selected from S is input into the quality-estimation module (3.1.4) to determine the optimal $p_i(t)$ at which the estimated Q_i for T-seconds exceeds R. The quality-estimation module estimates Q_i for past $\frac{T}{2}$ seconds and future $\frac{T}{2}$ seconds from the current time. In the past duration, the multimedia quality collected from the client is utilized, and in the future, the selected bitrate from S is used for $\frac{T}{2}$-seconds. The detailed algorithm is described in Sect. 3.3. Finally, $p_i(t)$ is directed to each client by the receiver estimated maximum bitrate (REMB) [42]†.

3.1.4 Quality-Estimation Module

This module calculates Q_i using multimedia quality information (i.e., video and audio bitrate, framerate, and resolution) and device type (e.g., laptop or smartphone). Figure 2 shows an example of quality estimation for four users. i represents a specific user, j represents a specific video stream from the sending client to the receiving client, and N represents the number of users. First, audiovisual quality per stream ($M_j(t)$) is calculated using multimedia quality information and device type. Then, audiovisual quality per user ($U_i(t)$) is averaged over the quality of each stream displayed on the screen, excluding its stream ($M_j(t)(j \in N, j \neq i)$). Finally, Q_i of the estimated duration, which is the total of T-seconds from the past $\frac{T}{2}$ seconds to future $\frac{T}{2}$ seconds, is calculated from $U_i(t)$.

The detailed equations are described in Sect. 3.2.

3.1.5 Bitrate Determination Module

This module determines the video bitrate to achieve R. $p_i(t)$ that is calculated in the directed bitrate estimation module does not consider the network conditions (e.g., jitter, packet loss). Therefore, there is a possibility of stalling due to a directed high bitrate when network congestion occurs. Thus, GCC [10], [11] described in Sect. 2.1 is used to deal with changes in network conditions. GCC calculates the network condition based bitrate ($g_i(t)$) per user using jitter and packet loss. Each client compares the network condition based bitrates (i.e., $g_i(t)$) with the $p_i(t)$ and selects the lower one. In summary, the bitrate can be set in accordance with R to suppress excessive quality if there is enough bandwidth. On the other hand, when the network condition is poor, the GCC selects the appropriate quality in accordance with the network condition.

†The REMB [42] is used for notifying the bitrate that the receiver can receive in WebRTC. When a client receives a REMB message, it encodes the video at a bitrate that is lower than the upper bitrate described in REMB.

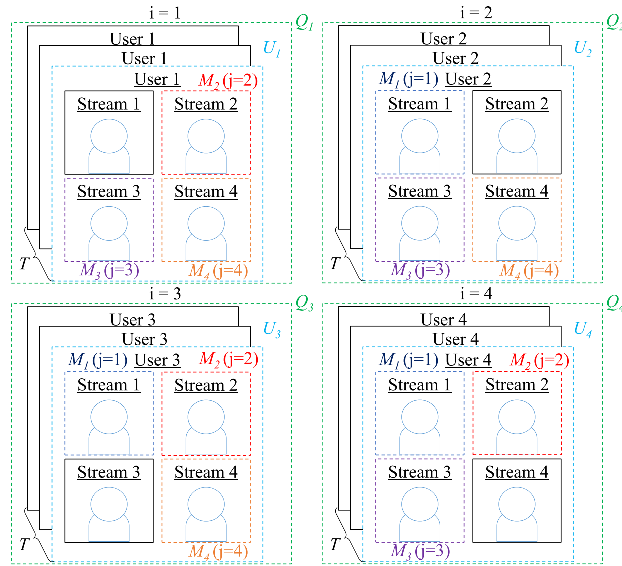


Fig. 2 Example of quality estimation.

3.2 Details of Quality-Estimation Model

This section details the quality-estimation model used in the quality-estimation module (described in 3.1.4). As described in 3.1.4, the audiovisual stream quality per stream ($M_j(t)$) is estimated from multimedia quality information. The audiovisual stream quality per user ($U_i(t)$) is estimated from that information and device type for the display user. Finally, Q_i for T-seconds is estimated as the overall quality.

Existing quality-estimation models [28] are used to estimate $M_j(t)$ and Q_i. First, the estimation model of $M_j(t)$ is shown below:

$$A_j(t) = a_1 + \frac{1 - a_1}{1 + (\frac{ba_j(t)}{a_2})^{a_3}}, \tag{1}$$

$$V_j(t) = X + \frac{1 - X}{1 + (\frac{bv_j}{Y})^{v_1}}, \tag{2}$$

$$X = \frac{4(1 - \exp(-v_3 \cdot r_j(t))) \cdot s_j(t)}{v_2 + s_j(t)} + 1, \tag{3}$$

$$Y = \frac{v_4 \cdot s_j(t) + v_6 log10(v_7 \cdot r_j(t) + 1)}{1 - e^{-v_5 \cdot s_j(t)}}, \tag{4}$$

$$M_j(t) = av_1 + av_2 \cdot A_j(t) + av_3 \cdot V_j(t) + av_4 \cdot A_j(t) \cdot V_j(t). \tag{5}$$

$A_j(t)$ represents j's audio quality per second calculated by an audio bitrate ($ba_j(t)$) (Eq. (1)). $V_j(t)$ represents j's video quality per second calculated by video bitrate ($bv_j(t)$), framerate ($r_j(t)$), and resolution ($s_j(t)$) (Eqs. (2)–(4)). $M_j(t)$ represents j's audiovisual quality per second. The coefficients $v_1 - v_7$ are set to different values for PC and SP to consider that the quality is different even if the same bitrate video is displayed due to the difference in the screen size in accordance with the device type. In addition, coefficients $v_1 - v_7$, $a_1 - a_3$ need to be changed by the codec. In this study, $v_1 - v_7$ are optimized for VP8, and $a_1 - a_3$ are optimized for opus. The details of the method of optimizing and the values of the coefficients used in this study are shown in 4.3. Coefficients $av_1 - av_4$ are not significantly affected by the codec, so the same values as in previous studies [38] are used.

Next, the estimation of $U_i(t)$ is described. In estimating $U_i(t)$, the quality of each video displayed on the same screen needs to be considered. In addition to each video quality, the differences in the video quality (i.e., contrast effect) and the screen size need to be considered. In a previous study [40], there is a case in which the contrast effect is not seen due to the effect of the difference in the situation and quality. Therefore, the contrast effect is not considered, but only each video quality and screen size are considered in this study.

The receiver's quality ($U_i(t)$) is calculated by a weighted average of $M_j(t)(j \in N, j \neq i)$ by weighting each display size. The equation is shown in Eq. (6).

$$U_i = \sum_{j=0, j\neq i}^{N} \left(\frac{ds_j}{\sum_{k=0,k\neq i}^{N} ds_k} M_j\right). \tag{6}$$

$U_i(t)$ represents the audiovisual quality per receiver i. $ds_j(t)$ shows the display size of j's video stream at the receiver. The receiver's own video and audio streams are excluded from the calculation of $U_i(t)$. This is because the audio and video streams of the receiver are not distributed from the server but are processed locally in many cases, and the video quality of a user's stream is very high.

Finally, the estimation of Q_i is described. Estimation of Q_i should consider the forgetting effect in addition to the short-time quality. Therefore, the models of previous studies considering their effects are utilized in this study [28], [40]. The formulas are shown in Eqs. (7)–(10). Coefficients of $t_1 - t_5$ are also not significant affected by the codec, so the same values as in previous studies [38] are used.

$$Q_i = \frac{\sum_{k=t-\frac{T}{2}}^{T} w_1(u) \cdot w_2(U_i(k)) \cdot U_i(k)}{\sum_t w_1(u) \cdot w_2(U_i(k))}, \tag{7}$$

$$w_1(u) = t_1 + t_2 \exp(u/t_3), \tag{8}$$

$$w_2(U_i(k)) = t_4 - t_5 \cdot U_i(k), \tag{9}$$

$$u = k/duration. \tag{10}$$

3.3 Details of Directed Bitrate Estimation

This section details the directed bitrate estimation method used in the directed bitrate estimation module (described in 3.1.3).

Our proposed algorithm is shown in Algorithm 1. First, set each user's bitrate to the lowest value in S (lines 1–4), then set the framerate and resolution to the latest values in the received multimedia quality information (i.e., the resolution and framerate at $t-1$ are used basically)[†]. $U_i(t)$ for $\frac{T}{2}$ seconds are estimated using these information. Because of this, $U_i(t)$ of future $\frac{T}{2}$ seconds are estimated to be the same value. Past $U_i(t)$ for $\frac{T}{2}$ seconds are estimated by using the multimedia quality information received by each client. Q_i for T-seconds of all users is estimated using these $U_i(t)$ (lines 6–8).

Algorithm 1 Video bitrate estimation method

1: sort $\boldsymbol{S}$ ascending
2: **for** all $j \in N$ **do**
3: temporary $p_j(t) \leftarrow S[0]$
4: **end for**
5: **while** $min(\boldsymbol{p(t)}) = max(\boldsymbol{S})$ **do**
6: **for** all $i \in \boldsymbol{N}$ **do**
7: calculate Q_i
8: **end for**
9: **if** $min(Q_i) \geq R$ **then**
10: break
11: **else**
12: **for** all $j \in N$ **do**
13: Increases the temporary $p_j(t)$ by one step in S
14: Calculate Q_i
15: **if** $min(\boldsymbol{Q}) \geq R$ **then**
16: **if** candidate $Q_i \geq$ average($\boldsymbol{Q}$) **then**
17: candidate $\boldsymbol{p(t)} \leftarrow$ temporary $\boldsymbol{p(t)}$
18: candidate $\boldsymbol{Q} \leftarrow$ temporary $\boldsymbol{Q}$
19: **end if**
20: **end if**
21: **end for**
22: **end if**
23: $\boldsymbol{p(t)} \leftarrow$ candidate $\boldsymbol{p(t)}$
24: **end while**

If the Q_i for T-seconds of all users exceeds R, the process finishes, and the currently set bitrate is directed. Concretely, whether minimum Q_i values exceed R is checked (lines 9–11).

If the Q_i for T-seconds of all users does not exceed R, combinations of bitrate where the Q_i of all users exceeds R are searched. Concretely, combinations are created that raise the bitrate of only one user by one step in S. The Q_i of each combination is estimated and verified to see if it exceeds R (lines 12–16). If there is a bitrate combination in which the Q_i of all users exceeds the R, the bitrate closest to the R is selected. If not, the algorithm is continued with the combination of the bitrates at which all users' average value of the Q_i is the highest as a tentative bitrate candidate (lines 17–18). An example of bitrate searching is shown in Fig. 3. When the bitrate of all streams reaches the maximum value in S, the process is also finished (line 5).

At the beginning of control, the quality cannot be estimated due to a lack of past multimedia quality information. Therefore, the initial quality information is exceptionally continued for the past $\frac{T}{2}$ seconds.

4. Evaluation Settings

This section describes the evaluation environment, the evaluation conditions, and the parameter sets used.

[†]If multimedia quality information is not being received due to a delay, information from earlier is used (e.g., $t-2$).

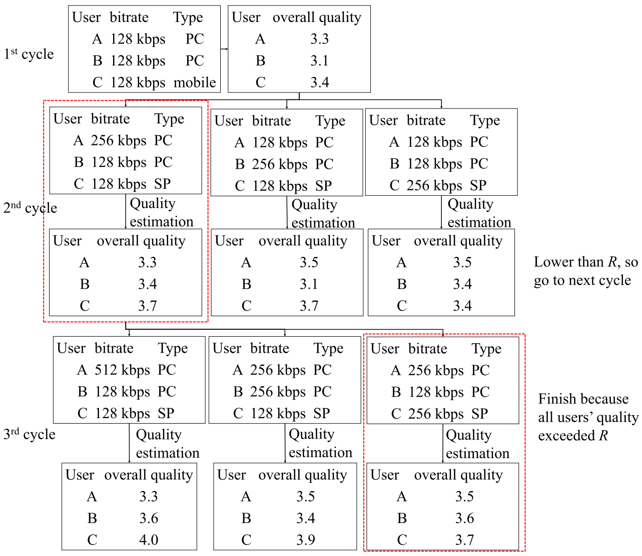


Fig. 3 An example of bitrate searching.

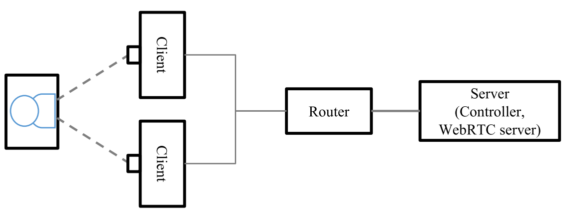


Fig. 4 Experimental environment.

4.1 Evaluation Environment

An overview of the experimental environment is shown in Fig. 4. The proposed method is implemented on the server, and Google Chrome is used as a client application. PC has a wired connection, and SP has a wireless connection to the server. In this evaluation, the video codec set VP8 that supports VBR and an audio codec set opus, two of the major codecs in WebRTC. All clients take a video of a person to simulate web-conferencing. To change the network condition, tc [43] is used to limit the bandwidth.

4.2 Evaluation Conditions

In this study, the proposed method is verified in eight scenarios. In scenarios 1–5, these evaluations are based on conditions that assume actual web-conferencing (e.g., changes in the number of users and network conditions). Scenarios 6–8 evaluate the impact of changing the parameters of this method (S, T, I).

As shown in Sect. 1, the purpose of the proposed method is to reduce the amount of transferred data while achieving R, so this perspective needs to be evaluated under various conditions (scenarios 1–5). First, the quality and the amount of transferred data are evaluated under three PCs with sufficient bandwidth conditions (scenario 1).

As described in Sect. 3.2, the coefficients of the quality-estimation model are changed for each device type, so it is possible to express the difference in quality due to the difference in screen size even if the same bitrate video is displayed on PC and SP. Therefore, the video bitrate required to satisfy R varies depending on the device type. Then, it is evaluated under the condition that it was joined by two PCs and one SP (scenario 2).

Since the number of users generally changes for each web-conferencing, it is necessary to evaluate whether the bitrate can be controlled without being affected by changes in the number of users. Two PCs are used to evaluate, and the results are compared with the results of scenario 1 (scenario 3).

Since throughput often fluctuates depending on the network usage, it is necessary to evaluate whether the R can be achieved in accordance with the changing network conditions. Three PCs are used in this scenario, where one PC's upstream bandwidth was limited by tc [43]. Sufficient bandwidth is maintained for the first 50 seconds, then the bandwidth is limited to 256 kbps for the last 10 seconds, and these conditions are repeated for 300 seconds (scenario 4).

Also, as described in Sect. 3.1.2, the proposed method does not control the screen-sharing stream, so it is necessary to evaluate whether the bitrate can be controlled without the influence of screen sharing. Three PCs are used in this scenario and share one PC's screen. It is evaluated under two bandwidth conditions: sufficient bandwidth condition (scenario 5-1) and the same bandwidth fluctuation condition as scenario 4 (scenario 5-2).

Furthermore, our proposed method has three parameters (S, T, and I). To determine what values to set for these parameters, Q_i is evaluated when the parameters are changed (scenarios 6-8).

First, the impact of changes in the number of S was evaluated (scenario 6). S is a set of selectable bitrates, and the number of S affects the computational time and fine bitrate changes. Therefore, the suitable number of S should be clarified in this study. It was evaluated by setting the number of S to 3, 5, 8, 16, 32, and 64. Selectable bitrates are $128, 512, 1024$ kbps when the number of S is 3; $128, 256, 512, 756, 1024$ when it is 5; $128 \times x(x = 1-8)$ when it is 8; $64 \times x(x = 1-16)$ when it is 16; $32 \times x(x = 1-32)$ when it is 32; and $16 \times x(x = 1-64)$ when it is 64. The bandwidth condition is made to be two patterns, the same as in scenario 5. A sufficient bandwidth condition is defined as scenario 6-1, and a fluctuating bandwidth condition is defined as scenario 6-2.

Next, the impact of changing T was evaluated (scenario 7). T is the quality estimation duration, and this value is one of the factors that affects the speed of response to changes in network conditions. Q_i is analyzed as T changes to 10, 20, 40, 60, and 120 to evaluate whether R can be achieved. The network condition is evaluated under the same fluctuating conditions as in scenario 4.

Finally, the impact of changing I was evaluated (scenario 8). I represents the interval of bitrate control, and this

Table 3 Evaluation conditions.

#	# of PC	# of SP	bandwidth fluctuation	screen share	S	T	I
1).	3	0	-	-	8 steps	60	1
2).	2	1	-	-	8 steps	60	1
3).	2	0	-	-	8 steps	60	1
4).	3	0	○	-	8 steps	60	1
5).	3	0	○/-	○	8 steps	60	1
6).	3	0	○/-	-	3/5/8/16/32/64 steps	60	1
7).	3	0	○	-	8 steps	10/20/40/60/120	1
8).	3	0	○	-	8 steps	60	1/5/10

value is also one factor that affects the speed of response to the change in network conditions. Then, the Q_i was evaluated when I was set at 1, 5, and 10 seconds. The network condition is evaluated under the same fluctuating conditions as in scenario 4.

Since most PCs and SPs have cameras with resolutions higher than high definition (HD), the resolution is set to HD at the maximum. For the same reason, the framerate is set to 30 fps at the maximum. Note that browser automatically changes the resolution and framerate depending on the video bitrate. Other conditions (device type, bandwidth condition, S, T, and I) are described in Table 3. In all scenarios, the evaluation time was set to 300 seconds.

4.3 Coefficients of Quality-Estimation Model

As described in Sect. 3.2, coefficients of the quality-estimation model (i.e., $a_1 - a_3$, $v_1 - v_7$) need to be optimized per codec (i.e., VP8 and opus) and device type (i.e., PC and SP). In this study, coefficients of the video-quality-estimation model (i.e., $v_1 - v_7$) were optimized on the basis of the method described in [39]. The coefficients of audio-quality-estimation model (i.e., $a_1 - a_3$) are optimized by referring to the previous study to optimize the coefficients of the video-quality-estimation model [39]. Concretely, the POLQA [44] is used instead of VMAF [45] for optimization.

The coefficients of video-quality-estimation models (i.e., $v_1 - v_7$) are optimized using processed videos that are generated using many video sources in web-conferencing. Thirty-five video sources (SRCs) of 10 seconds were prepared. The video sources show a man or woman who is speaking or listening. Each video source is encoded by VP8 (i.e., VBR mode). The detailed encoding settings are below: resolution: 240p, 360p, 720p, and 1080p, where the aspect ratio is 16:9, framerate: 15 and 30 fps, bitrate: 128, 256, 512, 1024, and 2560 kbps. By encoding 35 SRCs under 40 conditions, a total of 1400 processed videos (PVSs) were prepared, and the coefficients are optimized using these videos. The optimized coefficients of the model are listed in Table 4. Furthermore, to validate the model's accuracy, 20 different videos are prepared and encoded under the following conditions. The resolution, framerate: same conditions, bitrate: 192, 320, 448, 704, and 1536 kbps. Such validation data of 800 PVSs are prepared.

The results of the evaluation of the video-quality-estimation model are shown in Table 5. Root mean square error (RMSE) in the training data are 0.348 at the PC and 0.310 at SP, and RMSE for the validation data are 0.362 at the PC and 0.298 at SP. Since the RMSE is sufficiently low and the accuracy for validation data is almost the same as that for training data, this model can be used generically with sufficient accuracy.

Table 4 Coefficients of quality-estimation model.

	Value (PC)	Value (SP)
v_1	1.1305	1.3817
v_2	154007	43738
v_3	0.074261	0.12896
v_4	7.29E-05	2.02E-05
v_5	0.99697	0.99697
v_6	91.526	419.14
v_7	0.19429	0.010929

	Value (all device)
a_1	4.7907
a_2	8.1895
a_3	2.0665
av_1	0.62
av_2	0
av_3	0.61369
av_4	0.068487
t_1	0.006666
t_2	4.04E-05
t_3	0.15650
t_4	0.14318
t_5	0.023864

Table 5 Evaluation results of quality-estimation model.

			RMSE
video	train	PC	0.348
		SP	0.310
	validation	PC	0.362
		SP	0.298
audio	train	-	0.230
	validation	-	0.178

Coefficients of the audio-quality-estimation model (i.e., $a_1 - a_3$) are also optimized. Seventeen audio sources in web-conferencing are prepared, and they are encoded by opus. The encoding bitrate conditions are 5 (i.e., 10, 20, 30, 40, and 50 kbps). As a training dataset, 85 processed audios are generated, POLQA [44] value is used for audio-quality-estimation model optimization. Furthermore, to validate the model's accuracy, 9 different audio are prepared and 4 bitrate conditions are encoded (i.e., 15, 25, 35, and 45 kbps). In this way, 36 audio data are generated as the validation datasets. The optimized coefficients of the model are listed in Table 4. As shown in Table 5, RMSE was 0.230 in the training data and 0.178 in the validation data. These results indicate that a model with sufficiently high accuracy was constructed.

5. Results

This section shows the results under the evaluation condi-

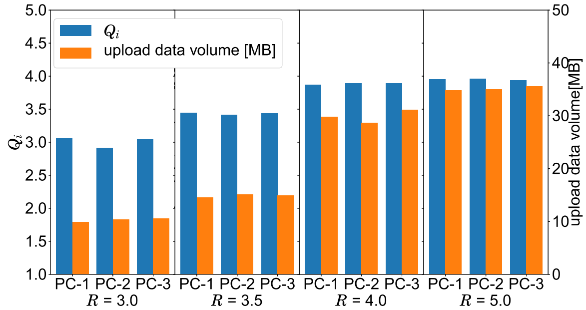


Fig. 5 The Q_i and upload data volume per user for scenario 1.

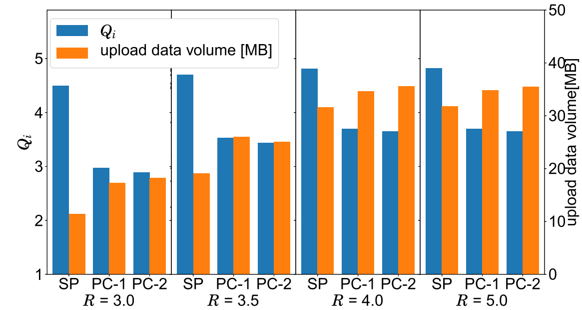


Fig. 6 The Q_i and upload data volume per user for scenario 2.

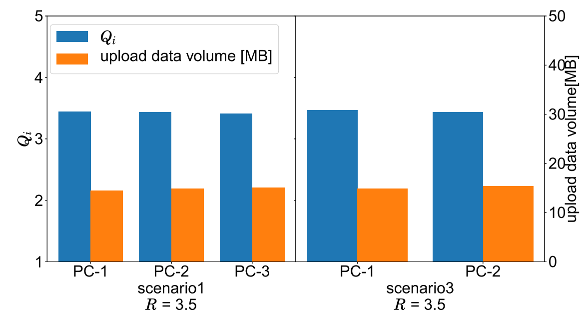


Fig. 7 The Q_i and upload data volume per user for scenario 3.

tions in Sect. 4.2.

In scenarios 1 and 2, how much the proposed method reduces the amount of data compared with GCC is evaluated. In this evaluation, due to the complexity of the implementation, GCC-equivalent control results are obtained by setting R to 5. Setting R to 5 means the maximum quality is achieved and the bitrate is not limited by $p_i(t)$. Therefore, since only $g_i(t)$ is used for control, it is equivalent to GCC, which is one of the commonly used existing methods. In scenarios 3 and later, how the proposed control behaves under various conditions is evaluated.

5.1 # 1: Control Excessive Quality and Reduce the Amount of Transferred Data

The Q_i for evaluation time (i.e., 300 seconds) and the upload data volume under scenario 1 are shown in Fig. 5. This graph shows the upload data volume per user and Q_i for evaluation time (i.e., 300 seconds) per R. The horizontal axis indicates the R. The blue bar shows the Q_i for evaluation time per R, and the left vertical axis shows the value. The orange bar shows the upload data volume per user, and the right vertical axis shows the value.

The amount of transferred data can be reduced by controlling the quality to approach R with the proposed method. When the R is set to 3.5, the Q_i is about 3.4, and the average upload data is about 15 MB. The upload data volume is reduced to 57% compared with setting the R to 5, which is similar to network condition based control without setting the required quality.

The result shows that our proposed method can keep the quality approximately at the R and reduce the amount of transferred data by suppressing excessive quality. However, the Q_i is about 3.4, which is lower than R (i.e., 3.5). This is because the bitrate indicated by $p_i(t)$ is an upper limit, and the actual bitrate is determined by the browser on the basis of this value, so the value is slightly smaller than R.

When R is set to 5, the Q_i for all users are about 3.95, which is lower than R. This is because $M_j(t)$ only goes up to about 4, even at a bitrate of 1Mbps, the maximum in S, when using the PC. The quality can be improved by raising the maximum bitrate of S, but the quality value of 5 cannot be achieved when using PC because the quality estimated by the video quality-estimation model saturates at about 4.3.

5.2 # 2: Impact of Changes Device Type

The results in scenario 2 are shown in Fig. 6. The axes of the graph and what each bar represents are the same as in Fig. 5.

When the R is set to 3.5, the upload data volume is reduced to about 31% compared with setting R to 5. The Q_i of each user are 3.44, 3.53, and 4.69. The only SP user is a high quality, and the quality of PC users is controlled around R. Since SFU can send only one quality of the video, if a bitrate is selected such that the PC user's Q_i exceeds R, the SP user's Q_i greatly exceeds R.

5.3 # 3: Impact of Changes in the Number of Devices

Figure 7 shows the Q_i and the upload data volume per users for scenario 1 (i.e., three PCs) and scenario 3 (i.e., two PCs). The axes of the graph and what each bar represents are the same as in Fig. 5.

There is no difference in the Q_i and upload data volume per user between scenarios 1 and 3. It is shown that this method is not affected by the number of users and that upload data volume increases depending on the number of users, where upload data volume per user is not changed.

5.4 # 4: Impact of Bandwidth Fluctuation

Figure 8 compares the Q_i and upload data volume for all users in scenario 1 (i.e., sufficient bandwidth) and scenario 4 (i.e., bandwidth fluctuation) when R is 3.5 and 4. The axes of the graph and what each bar represents are the same as in Fig. 5.

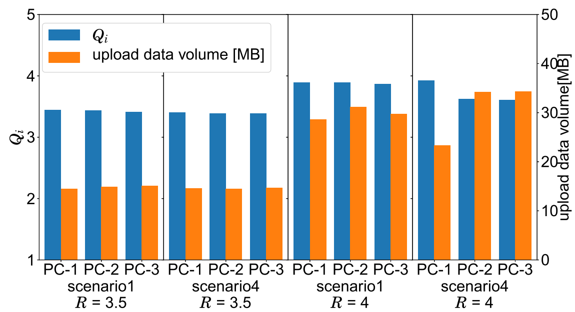


Fig. 8 The Q_i and upload data volume per user for scenario 4.

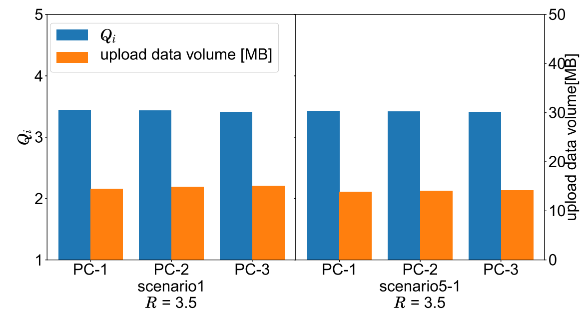
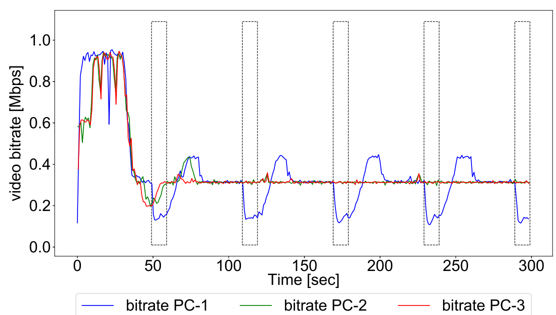
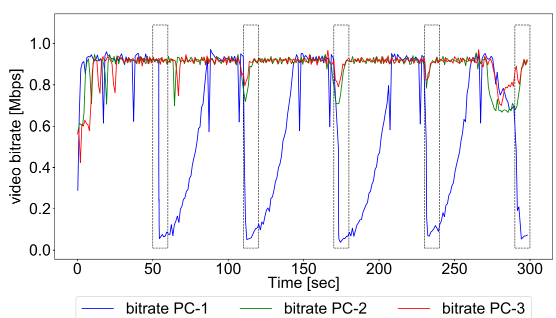


Fig. 10 The Q_i and upload data volume per user for scenario 5-1.



(a) $R = 3.5$



(b) $R = 4.0$

Fig. 9 Time series of bitrate, resolution and directed bitrate in scenario 4.

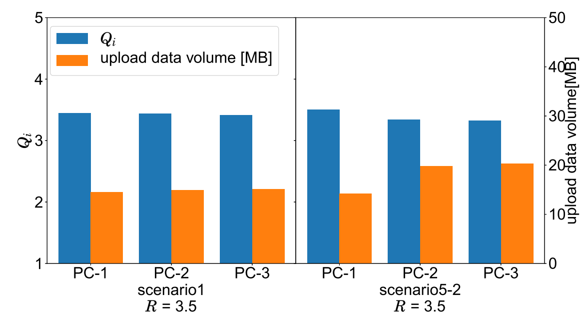


Fig. 11 The Q_i and upload data volume per user for scenario 5-2.

The results of $R = 3.5$ confirm that there is little difference between scenarios 1 and 4 and that the proposed method can be controlled to maintain the R even with bandwidth fluctuation. Figure 9(a) shows the time series of bitrate for each stream when $R = 3.5$ to confirm how the R is maintained by selecting the bitrate. The blue line is the bitrate of the bandwidth-limited user, and the bandwidth is limited near the section enclosed by the black dotted line. When the bandwidth is limited, the bitrate drops due to GCC ($g_i(t)$). After bandwidth limitation, the degraded quality due to bitrate degradation is improved by increasing the bitrate of its own stream (blue line). By controlling in this way, the quality can be improved efficiently. If the bitrate of one user ($i = 1$) drops, it affects the quality of others ($i = 2, 3$). Therefore, if the bitrate of the user ($i = 1$) is increased, it is sufficient to increase the bitrate of one stream. However, increasing the bitrate of $i = 2$ only improves the quality of $i = 3$. Therefore, increasing the bitrate of the user whose bitrate has not decreased is inefficient because the bitrate of multiple users must be increased.

When $R = 4$, the Q_i is lower than in scenario 1. The time-series data when the R is set to 4 are shown in Fig. 9(b). The maximum bitrate (i.e., 1024 kbps) is selected to achieve a MOS value of 4. The bitrate drops when the bandwidth is limited (i.e., section enclosed by a black dotted line), and the quality is gradually improved by GCC after the limit is lifted. However, since a bitrate higher than 1024 kbps cannot be selected, the degraded quality during bandwidth limit could not be improved, and Q_i decreased compared with the case without a bandwidth limit.

5.5 # 5: Unaffected by Screen Sharing

Figure 10 shows the results for scenario 5-1 (i.e., screen sharing) and scenario 1 (i.e., no screen sharing) when the bandwidth is sufficient, and R is set to 3.5. The axes of the graph and what each bar represents are the same as in Fig. 5.

No difference was found in Q_i and upload data volume regardless of screen sharing. It was shown that the proposed method could be controlled without being affected by the screen-sharing stream.

Figure 11 shows the results when the bandwidth fluctuates. The axes of the graph and what each bar represents are the same as in Fig. 5.

In the case of screen sharing, there is no difference in Q_i, but the upload data volume increased by about 1.2 times. To determine why the upload data volume increased, the time series of the bitrate for each stream are shown in Fig. 12. The

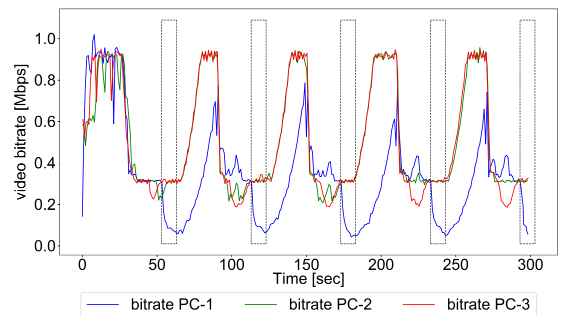


Fig. 12 Time series of bitrate, resolution and directed bitrate in scenario 5-2 (R = 3.5).

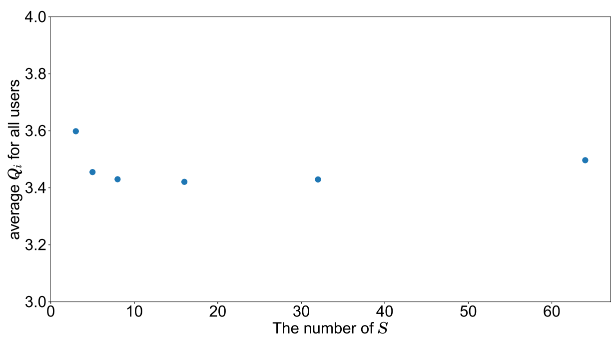


Fig. 13 The average Q_i for all users when changing S.

minimum bitrate while bandwidth is limited (i.e., section enclosed by a black dotted line) is lower than when there is no screen-sharing condition (shown in Fig. 9(a)). Since the bitrate of the screen-sharing stream is low, there is little effect when there is no bandwidth limitation. However, when there is a bandwidth limitation, the effect is large, so the video bitrate is lowered by GCC. A higher bitrate is required to improve the quality, which increases the upload data volume.

5.6 # 6: Impact of Changing S

Figure 13 shows the average Q_i for each number of S for scenario 6-1 (i.e., no bandwidth fluctuation and R is set to 3.5). The horizontal axis shows the number of S, and the vertical axis shows the average Q_i for all users.

The average Q_i exceeds R when the number of S is small (i.e., the number of S = 3) because the small number of selectable bitrates forced it to choose a bitrate higher than the optimal bitrate. When the number of S is more than 5, there is almost no difference in average Q_i. However, from the time series of bitrate when the number of S = 5 (Fig. 14), there is a larger fluctuation in the bitrate than when the number of S = 8 (Fig. 15). Since it is better for users to send as stable a bitrate as possible, the number of S should be 8 or more. On the other hand, increasing the number of S also affects the calculation time. Figure 16 shows the relationship between the number of S and the calculation time. The computation time increases linearly with the number of S. To reduce the processing load, as small a number of S as possible should be selected. From these results, it is reasonable to set the number of S = 8.

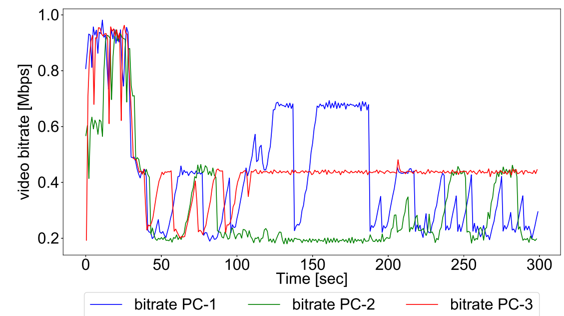


Fig. 14 Time series of bitrate, resolution, and directed bitrate when the number of S = 5.

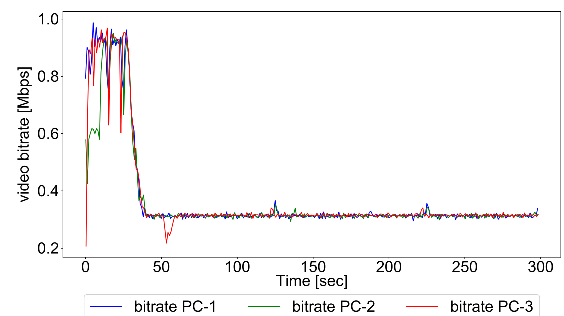


Fig. 15 Time series of bitrate, resolution, and directed bitrate when the number of S = 8.

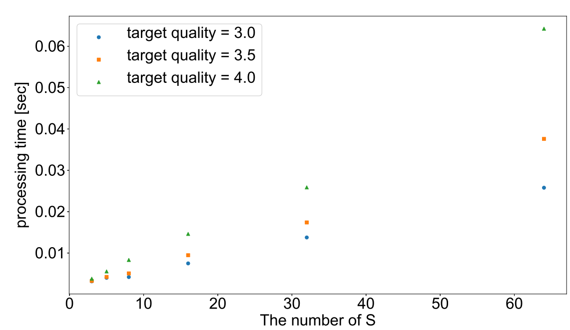


Fig. 16 The calculation time of scenario 6-1.

In scenario 6-2, the results were the same as in scenario 6-1. It was confirmed that the number of S should be 8 regardless of the presence or absence of bandwidth fluctuation.

5.7 # 7: Impact of Changing T

Figure 17 shows the average Q_i for each T value when R is set to 3.5. The horizontal axis shows T, and the vertical axis shows the average Q_i for all users.

It was found that the change in T value does not affect the average Q_i. To determine the reason, the time series of video bitrate is analyzed. Figure 18(a) and (b) show the time series of bitrate when T = 20 and T = 60, respectively. As shown in Fig. 18(a), when T is small, the quality is improved by increasing the bitrate of the stream without a bandwidth limit (PC-2 and PC-3) to improve the quality degradation

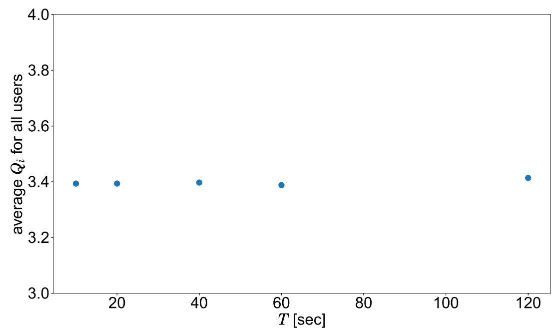


Fig. 17 The average Q_i for all users when changing T.

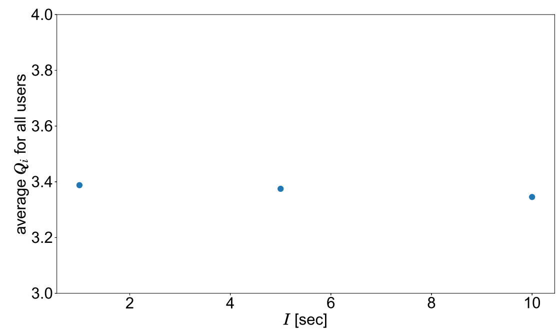
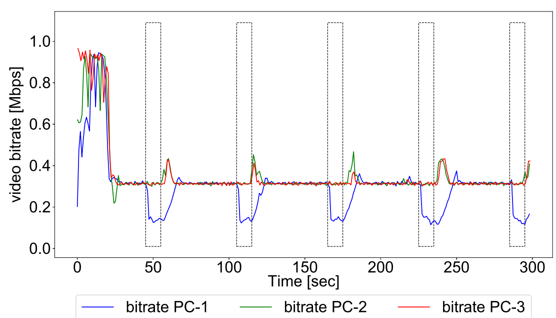
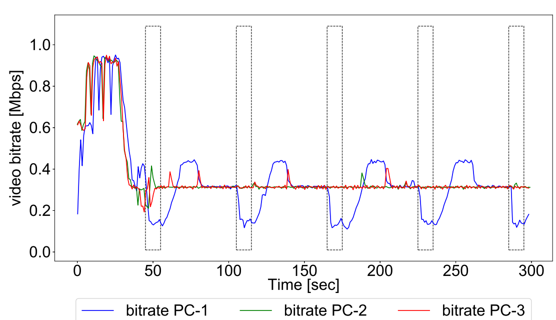


Fig. 19 The average Q_i for all users when changing I.



(a) T = 20



(b) T = 60

Fig. 18 Time series of bitrate/resolution/directed bitrate in scenario 7.

quickly. On the other hand, T is set bigger (Fig. 18(b)), and the bitrate of the bandwidth-limit stream (PC-1) is increased after the bandwidth limit is removed to improve the quality. Even if the value of T is changed, the quality can be improved by increasing the bitrate (the streams that increase the bitrate will change).

On the basis of this result, how the value of T should be set is considered. T represents quality estimation duration in the past and future. When T is large, past and future QoE estimation periods are longer and thus less susceptible to short-term QoE declines in past periods. If T is large when bandwidth limitation is applied, the bitrate is controlled to recover the quality degradation over a long period. On the other hand, a small value of T reduces the impact of past quality degradation over time, reducing the impact on quality estimates. Therefore, the quality improvement effect is limited.

From this, T should be set in accordance with the assumed network condition. However, since the bitrate can only be improved gradually due to GCC, the value of T should be somewhat high for multiple users to deal with network quality degradation at the same time.

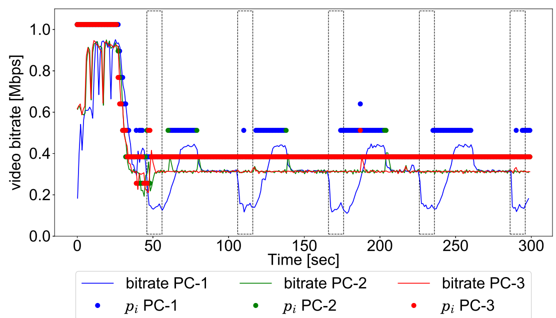
5.8 # 8: Impact of Changing the I

To verify the impact of changing the I, the average Q_i when the interval is set to 1, 5, and 10 seconds is shown in Fig. 19. The horizontal axis shows the I, and the vertical axis shows the average Q_i for all users.

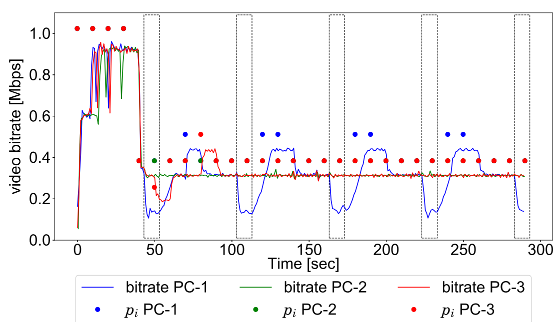
There is no difference in average Q_i even if the interval is changed. To determine the reason, the time series of video bitrate (solid line) and directed bitrate (marker) when the interval is 1 and 10 seconds are shown in Fig. 20(a) and (b). It can be seen that when I is 1 second, to direct high bitrate during bandwidth limit. On the other hand, If I is 10 seconds, it is after the bandwidth limit. Thus, it can be confirmed that when I becomes large, the direction to change the bitrate is delayed when bandwidth fluctuates. However, the bitrate increases gradually by GCC, so a high bitrate is directed during the increased bitrate even if the interval is long. As a result, the average Q_i is no different, regardless of the interval. As long as it is used with GCC, no problem occurs unless the control interval is extremely large.

6. Conclusion

In this paper, we proposed a method to control the video bitrate of each user to achieve the required quality. We implemented the proposed method in an actual web-conferencing system and evaluated it under various network and device conditions. It was confirmed that the video bitrate was controlled in accordance with the required quality under each condition, and the transferred data was reduced compared with the case where the required quality was not set. As a result, we showed that this method contributes to reducing operational costs by reducing the transferred data while providing users with the required quality set by the service provider. In this evaluation, the proposed method was compared with the condition in which the required quality (R) was set to 5 in some scenarios (control equivalent to GCC [10], [11]). It



(a) I = 1sec



(b) I = 10sec

Fig. 20 Time series of bitrate/resolution/directed bitrate on scenario 8.

would be interesting to compare the proposed method with control considering the quality as shown in [12], but since this evaluation was not a simulation evaluation, they could not be compared because application and server implementation are required Therefore, we want to compare and evaluate the proposed method and [12] in future work. Also, the proposed method targets the web real-time communication (WebRTC) selective forwarding unit (SFU) from the perspective of reducing the amount of transferred data. However, due to the promotion of remote work, web-conferencing is used in various places. In such a situation, the network conditions vary widely for each client. In WebRTC-SFU, the same bitrate is sent to all receiver clients, the quality of the users will suffer in this situation, and the simulcast technology will become necessary. Therefore, in the future, we will study the technology to improve the quality further and reduce the amount of transferred data by incorporating this technology into the proposed technology of simulcast and others.

References

[1] Cisco, "webex," Online: https://www.webex.com/, Available on March 3, 2023.

[2] zoom, "zoom," Online: https://zoom.us/, Available on March 3, 2023.

[3] Microsoft, "teams," Online: https://teams.com/, Available on March 3, 2023.

[4] World Wide Web Consortium, "WebRTC 1.0: Real-Time Communication Between Browsers," Online: https://www.w3.org/TR/webrtc/, Available on March 3, 2023.

[5] The Internet Enngineering Task Force, "Real-Time Communication in WEB-browsers (rtcweb)," Online: https://datatracker.ietf.org/wg/rtcweb/documents/, Available on March 3, 2023.

[6] Jitsi, "Jitsi," Online: https://meet.jit.si/, Available on March 3, 2023.

[7] BigBlueButton, "BigBlueButton," Online: https://bigbluebutton.org/, Available on March 3, 2023.

[8] A. Amirante, T. Castaldi, L. Miniero, and S.P. Romano, "Janus: A general purpose WebRTC gateway," Proc. 2014 Conference on Principles, Systems and Applications of IP Telecommunications (IPTComm), no.7, pp.1–8, Oct. 2014.

[9] M. Westerlund and S. Wenger, "RTP topologies," RFC 7667, Nov. 2015.

[10] S. Holmer, H. Lundin, G. Carlucci, L.D. Cicco, and S. Mascolo, "A Google congestion control algorithm for real-time communication," Internet-Draft, draft-ietf-rmcat-gcc-02, July 2016.

[11] G. Carlucci, L. De Cicco, S. Holmer, and S. Mascolo, "Congestion control for web real-time communication," IEEE/ACM Trans. Netw., vol.25, no.5, pp.2629–2642, Oct. 2017.

[12] Z. Wang, Y. Cui, X. Hu, X. Wang, W.T. Ooi, and Y. Li, "MultiLive: Adaptive bitrate control for low-delay multi-party interactive live streaming," Proc. 2020 IEEE Conference on Computer Communications (INFOCOM), pp.1093–1102, July 2020.

[13] M. Yokota and K. Yamagishi, "Quality-based video bitrate control for WebRTC-based teleconference services," Proc. IS&T Int'l. Symp. on Electronic Imaging: Image Quality and System Performance, pp.333-1–333-6, Jan. 2022.

[14] L. Wu, A. Zhou, X. Chen, L. Liu, and H. Ma, "GCC-beta: Improving interactive live video streaming via an adaptive low-latency congestion control," Proc. 2019 IEEE International Conference on Communications (ICC), pp.1–6, May 2019.

[15] S. Petrangeli, D. Pauwels, J. van der Hooft, J. Slowack, T. Wauters, and F. De Turck, "Dynamic video bitrate adaptation for WebRTC-based remote teaching applications," Proc. IEEE/IFIP Network Operations and Management Symposium (NOMS), pp.1–5, April 2018.

[16] A. Bentaleb, B. Taani, A.C. Begen, C. Timmerer, and R. Zimmermann, "A survey on bitrate adaptation schemes for streaming media over HTTP," IEEE Commun. Surveys Tuts., vol.21, no.1, pp.562–585, 1st Quart. 2019.

[17] K. Miller, A.K. Al-Tamimi, and A. Wolisz, "QoE-based low-delay live streaming using throughput predictions," ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), vol.13, no.1, pp.1–24, Oct. 2016.

[18] X. Xie, X. Zhang, S. Kumar, and L.E. Li, "PiStream: Physical layer informed adaptive video streaming over LTE," GetMobile: Mobile Comp. and Comm., vol.20, no.2, pp.31–34, Oct. 2016.

[19] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A.C. Begen, and D. Oran, "Probe and adapt: Rate adaptation for HTTP video streaming at scale," IEEE J. Sel. Areas Commun., vol.32, no.4, pp.719–733, April 2014.

[20] T.Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," Proc. 2014 ACM Conference on Special Interest Group on Data (SIGCOMM), vol.44, no.4, pp.187–198, Oct. 2014.

[21] K. Spiteri, R. Urgaonkar, and R.K. Sitaraman, "BOLA: Near-optimal bitrate adaptation for online videos," IEEE/ACM Trans. Netw., vol.28, no.4, pp.1698–1711, Aug. 2020.

[22] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over HTTP," Proc. 2015 ACM Conference on Special Interest Group on Data (SIGCOMM), pp.325–338, Aug. 2015.

[23] Z. Li, A.C. Begen, J. Gahm, Y. Shan, B. Osler, and D. Oran, "Streaming video over HTTP with consistent quality," Proc. 5th ACM Multimedia Systems Conference (MMSys), pp.248–258, March 2014.

[24] T. Kimura, T. Kimura, A. Matsumoto, and J. Okamoto, "BANQUET: Balancing quality of experience and traffic volume in adaptive video streaming," Proc. 2019 15th International Conference on Network and Service Management (CNSM), pp.1–7, Oct. 2019.

[25] A. Mittal, A.K. Moorthy, and A.C. Bovik, "No-reference image quality assessment in the spatial domain," IEEE Trans. Image Process., vol.21, no.12, pp.4695–4708, Dec. 2012.
[26] A. Mittal, R. Soundararajan, and A.C. Bovik, "Making a "completely blind" image quality analyzer," IEEE Signal Process. Lett., vol.20, no.3, pp.209–212, March 2013.
[27] ITU-T Rec. P.1203, "Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport," Oct. 2017.
[28] K. Yamagishi and T. Hayashi, "Parametric quality-estimation model for adaptive-bitrate-streaming services," IEEE Trans. Multimedia, vol.19, no.7, pp.1545–1557, July 2017.
[29] A. Raake, M.N. Garcia, W. Robitza, P. List, S. Göring, and B. Feiten, "A bitstream-based, scalable video-quality model for HTTP adaptive streaming: ITU-T P.1203.1," Proc. 2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX), pp.1–6, May 2017.
[30] D. Vučić and L. Skorin-Kapov, "QoE assessment of mobile multiparty audiovisual telemeetings," IEEE Access, vol.8, pp.107669–107684, June 2020.
[31] M. Schmitt, J. Redi, P. Cesar, and D. Bulterman, "1 Mbps is enough: Video quality and individual idiosyncrasies in multiparty HD videoconferencing," Proc. 2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX), pp.1–6, June 2016.
[32] G. Berndtsson, M. Folkesson, and V. Kulyk, "Subjective quality assessment of video conferences and telemeetings," Proc. 2012 19th International Packet Video Workshop (PV), pp.25–30, May 2012.
[33] N. Rao, A. Maleki, F. Chen, W. Chen, C. Zhang, N. Kaur, and A. Haque, "Analysis of the effect of QoS on video conferencing QoE," Proc. 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), pp.1267–1272, June 2019.
[34] K. Schoenenberg, A. Raake, and P. Lebreton, "Conversational quality and visual interaction of video-telephony under synchronous and asynchronous transmission delay," Proc. 2014 Sixth International Workshop on Quality of Multimedia Experience (QoMEX), pp.31–36, Sept. 2014.
[35] T. Hayashi, K. Yamagishi, T. Tominaga, and A. Takahashi, "Multimedia quality integration function for videophone services," Proc. 2007 IEEE Global Telecommunications Conference (Globecom), pp.2735–2739, Nov. 2007.
[36] S. Jana, A. Chan, A. Pande, and P. Mohapatra, "QoE prediction model for mobile video telephony," Multimed. Tools Appl., vol.75, no.13, pp.7957–7980, July 2016.
[37] K. Yamagishi and T. Hayashi, "QRP08-1: Opinion model for estimating video quality of videophone services," Proc. 2006 IEEE Global Telecommunications Conference (Globecom), pp.1–5, Nov. 2006.
[38] P. Lebreton and K. Yamagishi, "Transferring adaptive bit rate streaming quality models from H.264/HD to H.265/4K-UHD," IEICE Trans. Commun., vol.E102-B, no.12, pp.2226–2242, Dec. 2019.
[39] K. Yamagishi, N. Egi, N. Yoshimura, and P. Lebreton, "Derivation procedure of coefficients of metadata-based model for adaptive bitrate streaming services," IEICE Trans. Commun., vol.E104-B, no.7, pp.725–737, July 2021.
[40] M. Schmitt, D.C. Bulterman, and P.S. Cesar, "The contrast effect: QoE of mixed video-qualities at the same time," Qual. User Exp., vol.3, no.1, pp.1–17, May 2018.
[41] World Wide Web Consortium, "Identifiers for WebRTC's Statistics API," Online: https://www.w3.org/TR/webrtc-stats/, Available on March 3, 2023.
[42] H. Alvestrand, "RTCP message for receiver estimated maximum bitrate," Internet-Draft, draft-alvestrand-rmcat-remb-03, April 2014.
[43] B. Hubert, "tc - show / manipulate traffic control settings Linux Man Page," Online: https://man7.org/linux/man-pages/man8/tc.8.html, Available on March 3, 2023.
[44] ITU-T Rec. P.863, "Perceptual objective listening quality prediction," March 2018.
[45] Netflix, "vmaf," Online: https://github.com/Netflix/vmaf, Available on March 3, 2023.

Masahiro Yokota received his B.E. and M.E. degrees in Keio University, Kanagawa, Japan in 2009 and 2011. Since 2011, he has worked for NTT Laboratories in Tokyo, where has been engaged in the research of the control technology based on quality of experience.

Kazuhisa Yamagishi received his B.E. degree in electrical engineering from the Tokyo University of Science, Chiba, Japan, in 2001, and his M.E. and Ph.D. degrees in electronics, information, and communication engineering from Waseda University, Tokyo, Japan, in 2003 and 2013. He joined NTT Laboratories, Tokyo, Japan, in 2003. From 2010 to 2011, he was a visiting researcher with Arizona State University, Tempe, AZ, USA. His research interests include the development of objective qualityestimation models for multimedia telecommunications. He was a recipient of the Young Investigators' Award (IEICE) in Japan in 2007 and the Telecommunication Advancement Foundation Award in Japan, in 2008, the ITU-AJ Encouragement Awards in Japan, in 2017, and the TTC Award for Distinguished Service in Japan, in 2018.