https://doi.org/10.2352/EI.2022.34.9.IQSP-333

Quality-based Video Bitrate Control for WebRTC-based Tele-conference Services

Masahiro Yokota, Kazuhisa Yamagishi; Nippon Telegraph and Telephone Corporation (NTT); Tokyo, Japan

Abstract

In this article, we propose a quality-based video bitrate control method for web real-time communication (WebRTC)-based tele-conferences. Video bitrate is controlled on the basis of quality of service (QoS) parameters such as delay and packet-loss rate in WebRTC. Therefore, the amount of transferred data may increase because media streams are transmitted at excessive quality levels depending on QoS conditions (e.g., the jitter and packet-loss rate are low). An increase in transferred data leads to higher operational cost (i.e., data transferred cost) and affects profitable growth. In the proposed method, quality desired by a service provider is set as TargetQuality, and the video bitrate of each stream is controlled aiming at TargetQuality, thereby suppressing the amount of transferred data while maintaining sufficient quality. The proposed method is implemented to an actual tele-conference system and is evaluated in terms of its effect at reducing the amount of transferred data. The results show that the amount of transferred data can be reduced by more than 40% by setting the value of TargetQuality appropriately.

Introduction

In recent years, the use of tele-conferences has drastically increased along with the promotion of telework. Since a large amount of video data needs to be delivered from multi-points, quality degradation occur due to network congestion. Therefore, improving a customer's quality is important for service providers. However, since providing excessive quality increases the amount of transferred data, the operational cost (i.e., data transferred cost) increases. To maintain quality and reduce the operational cost, a method should be developed that can control quality and the amount of transferred data.

Web real-time communication (WebRTC) is an important technology to provide tele-conference services. WebRTC has been standardized by the World Wide Web Consortium (W3C) and the Internet Engineering Task Force (IETF) and provides browser-based real-time communication. This technology is supported by many browsers (e.g., Microsoft Edge, Google Chrome, Mozilla Firefox, Safari) and is widely used. In WebRTC, clients basically communicate peer-to-peer, so the processing load on the end-user's device increases when many participants take part in the meeting. As a result, the number of participants is limited. To address the issue with the limitation of participants, multi-point control unit (MCU) [1] and selective forwarding unit (SFU) are proposed. In MCU, a MCU server merges the upload media data to one stream using a synthesis process and delivers it to all the clients. Since clients process one stream, the processing load of clients is minimized, but the load of the server cannot be ignored for the synthesis process. In SFU, a SFU server delivers the data without the media synthesis process, unlike in MCU. Therefore, the processing load of the server is not so high, but the load of client cannot be ignored to process many streams when there are many participants. Both server types have pros and cons, as described above. Since SFU is widely used, the control method for SFU is investigated in this study.

Google congestion control (GCC) [2, 3] 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). Therefore, when the network condition is good (i.e., the jitter and packet-loss rate are low), the video quality might become excessive and the amount of transferred data might increase.

The quality improvement for WebRTC has also been studied. Janczukowicz et al. proposed a priority control method by guaranteeing the expected bandwidth of the VP8 and Opus codecs [4]. They found that this technique improves the perceived quality. Wang et al. proposed a system of multi-party interactive live-streaming services to improve quality and latency. They addressed a many-to-many adaptive bitrate selection problem with the aim of maximizing quality by considering delay, stalling, and quality [5]. These methods are expected to improve quality but not reduce the amount of transferred data.

There are several studies aimed at reducing operational costs. Xhagjika et al. analyzed the server workload and media bitrate patterns derived from the WebRTC traffic. On the basis of their analysis, they proposed a resource allocation algorithm that distributes sessions among SFU servers to reduce server overload [6]. Petrangeli et al. studied a method to reduce encoding cost for a WebRTC-based remote teaching application. Their method introduces a conference controller and instructed the sender to encode fewer bitrate patterns than the number of receivers to reduce the encoding cost. In addition, it was found that the bandwidth could be efficiently utilized by varying the encoding bitrates in accordance with the bandwidth of the receiver [7]. These methods are expected to reduce server infrastructure costs, but that is not expected to reduce the transferred data volume.

Grozev et al. proposed an algorithm to selectively deliver the dominant speaker's video streams from audio activity. The results showed it reduced the central processing unit (CPU) usage and network bandwidth in a multi-party tele-conference [8]. Overall, this method can reduce the amount of transferred data, but it delivers streams with excessive quality in high throughput environments. Therefore, this method may transfer too much data per stream.

Some existing services are also trying to reduce transferred data by limiting the video resolution size. In Webex [9], the user cannot use a high-quality mode (i.e., the video resolution: 1280×720p and 640×360p) unless the user changes the default settings. Zoom [10] users can use a high-quality mode (i.e., the high video resolution: 1280 × 720p) only in a specific case (e.g., small number of users, joining users with specific licenses). Such limited upper quality can reduce the amount of data. However, these methods cannot basically select high-resolution video during a meeting, as mentioned above. As a result, if quality of service (QoS) levels (i.e., the jitter and packet-loss rate are high) are low in the beginning of the meeting and then QoS levels are improved, these approaches may limit quality improvement because the client cannot receive high-quality video depending on the settings.

On the other hand, such studies for video streaming services exist. Kimura et al. proposed a bitrate selection algorithm for adaptive bitrate streaming that maintains a user's setting target quality and minimizes the transferred data volume by focusing on the fact that the users are interested in not only quality but also the amount of data volume [11]. This method is targeted at single-stream services such as video streaming services and cannot be applied to multi-stream services such as tele-conferences. Therefore, such control methods for tele-conference have not been studied.

To maintain quality and reduce the amount of transferred data, a quality-based video bitrate control method for WebRTC is proposed. To address the issues mentioned above, quality desired by a service provider is set to a tele-conference system as TargetQuality (e.g., 3.5 in the quality range of 1 to 5). Note that although the important factors affecting the quality of tele-conference are the call establishment time, video quality, audio quality, audiovisual quality, and end-to-end delay [12], the delay is not investigated as a quality factor because the purpose of this study is to reduce the amount of transferred data while the quality is closer to TargetQuality. To control the quality on the basis of TargetQuality, the audiovisual quality a single stream is estimated by an existing quality-estimation method [13, 14]. Next, these qualities are merged as a single overall quality.

Then, a video encoding bitrate is set to each client so that the estimated overall quality maintains TargetQuality. The reason for controlling only the video bitrate is that the effect on the amount of data is large. On the basis of these procedures, the quality and the amount of transferred data are controlled.

To evaluate the effectiveness of this method, it is implemented on an actual tele-conference system, and the quality and the amount of transferred data are evaluated in various scenarios.

The remainder of this paper is structured as follows. First, the proposed architecture and method are described. Second, the evaluation environment is detailed, and then evaluation results are presented. Finally, the conclusions and future work of the paper are presented.

Proposed method

This section describes an algorithm for controlling quality and the amount of transferred data. The algorithm controls the video bitrate per stream on the basis of the estimated overall quality.

Figure 1 shows a block diagram of our proposed method, and each step of the processing can be summarized as follows.

1. Quality level required by a service provider, TargetQuality, is set to a controller in advance. Note that TargetQuality is not automatically set by the system.
2. After the tele-conference starts, the controller receives information of multimedia quality (i.e., resolution, bitrate, framerate) and device information (i.e., laptop or smartphone (SP)) from each client.
3. The controller estimates the quality per stream and merges these qualities into a single overall quality.
4. To estimate a suitable video bitrate of each stream on the basis of the estimated overall quality and the TargetQuality.
5. Controller sets the estimated video bitrate as the upper limit of the encoding bitrate for receiver estimated maximum bitrate (REMB) per client ($bitrate_{REMB}$).
6. The WebRTC server delivers RTCP messages for REMB to each client.
7. Each client also calculates the bitrate by GCC ($bitrate_{GCC}$), compares these bitrates (i.e., $bitrate_{GCC}$ and $bitrate_{REMB}$), and selects the lower one. In other words, the controller indicates the upper limit of the encoding bitrate. Each client encodes the video on the basis of the bitrate ($bitrate_{REMB}$) calculated by the controller when the QoS is sufficient. On the other hand, when QoS is not sufficient, the bitrate ($bitrate_{GCC}$) is controlled by GCC.

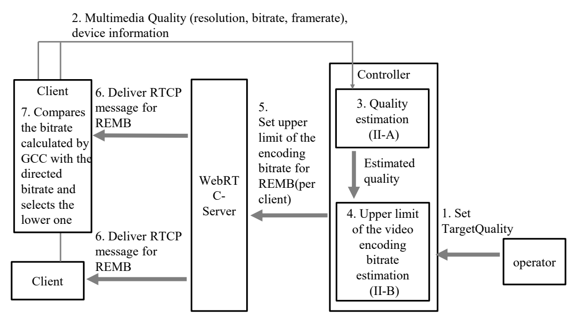

Figure 1. Overview of this method

Quality-estimation model

Several video quality-estimation models [14, 15] have been proposed. The quality-estimation accuracy of the model proposed by Yamagishi and Hayashi [14] has almost the same quality-estimation accuracy as the P.1203 model [15]. In addition, the processing load of the model [14] is lower than that of the P.1203 model because the model [14] has simpler mathematical equations than the P.1203 model. In the proposed method, it is assumed that many conferences need to be controlled at the same time, so the model proposed by Yamagishi and Hayashi [14] is used, which requires less computation

Quality, which ranges from 1 to 5 as a mean opinion score (MOS), is estimated by using quality-influencing parameters (e.g., bitrate). Audio quality (O21) is estimated as MOS with audio bitrate (b_a) as input (Eq. 1). Video quality (O22) is estimated as MOS with video bitrate (b_v), framerate (r), and resolution (s) as input (Eq. 2-4). The coefficients of video-quality-estimation models are optimized using video multimethod assessment fusion (VMAF) [16], where the coefficients are calculated for each device (i.e., laptop and SP) to take into account the effect of the difference in screen size.

The detailed procedure of generating coefficients of the model is described in Yamagishi et al. [13]. Put simply, many video sources are prepared, and then these sources are encoded by VP8 as a processed video. After that, VMAF is used to calculate the estimated quality using video sources and processed videos, where the VMAF value is converted to 1 - 5 because VMAF value ranges from 0 to 100. Finally, coefficients of the model are calculated using converted VMAF values.

Short-term audiovisual quality (O34) is estimated using O21 and O22 (Eq. 5) and long-term audiovisual quality (O35) is estimated using O34 (Eq. 6-9). Note that O21, O22, and O34 are calculated per second, and O35 is calculated per minute. The detailed equations are as follows:

$$O21 = a_1 + \frac{1 - a_1}{1 + (\frac{b_a}{a_2})^{a_3}}, \tag{1}$$

$$O22 = X + \frac{1 - X}{1 + (\frac{b_v}{Y})^{v_1}}, \tag{2}$$

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

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

$$O34 = av_1 + av_2 \cdot O21 + av_3 \cdot O22 + av_4 \cdot O21 \cdot O22, \tag{5}$$

$$O35 = \frac{\sum_t w_1(u) \cdot w_2(O34(t)) \cdot O34(t)}{\sum_t w_1(u) \cdot w_2(O34(t))}, \tag{6}$$

$$w_1(u) = t_1 + t_2 \exp(u/t_3), \tag{7}$$

$$w_2(O34(t)) = t_4 - t_5 \cdot O34(t), \tag{8}$$

$$u = t/duration. \tag{9}$$

The quality of each stream needs to be integrated into a single overall quality because the video of multiple participants is displayed on one screen. Therefore, overall quality per receiver is calculated by a weighted average of O34 per stream by weighting each display size. The equation is shown in Eq. 10-11. $O34_{streamx}$ shows the audiovisual quality per sender x's stream. $O34_{userx}$ shows the audiovisual quality per receiver x. ds_x shows display size of x's video stream at the receiving client. n is the number of participants, and N shows the set of participants. Q_x represents the overall quality per receiver x.

$$O34_{useri} = \sum_{j=0, j\neq i}^{n} (\frac{ds_j}{\sum_{k=0,k\neq i}^{n} ds_k} O34_{streamj}), \tag{10}$$

$$Q_i = O35(O34_{useri}). \tag{11}$$

In many cases, the receiver's own audio and video streams are not distributed from the server but are processed locally, and the video quality of a user's own stream is very high. Therefore, if a receiver's own stream is added to the calculation process, there is a difference with the displayed quality. To address this issue, the user's own video stream quality is not processed in the calculation of overall quality per user.

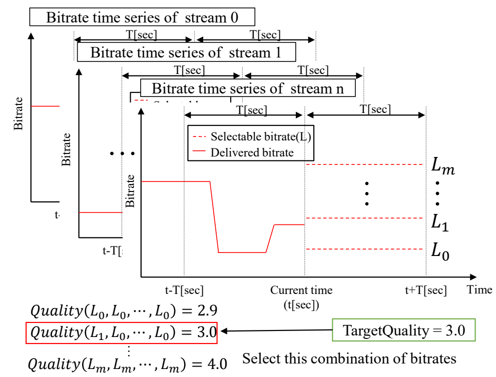

Figure 2. *Bitrate estimation method*

Upper limit of the video bitrate estimation

The upper limit of the video encoding bitrate estimation method is described in this section. Figure 2 shows the behavior of the method.

$O34_{stream}$ is estimated by using the information collected from each client. $O34_{user}$ is calculated from $O34_{stream}$ at the same time. Overall quality (Q_i) is estimated by $O34_{user}$ between T seconds before and T seconds after the current time (t). At this time, the information of the actual delivery quality (i.e., bitrate, resolution, framerate) is used to estimate the $O34_{stream}$ before the current time and the value of $O34_{stream}$ after the current time is estimated by assuming that one bitrate selected from the selectable bitrate (L) continues for T seconds. At this time, the value of resolution and framerate use the latest value. Selects the lowest bitrate that Q_i of all users calculated in the way exceeds TargetQualty. The selectable video bitrate is usually a continuous value, but in that case, the computational complexity is enormous. Therefore, the selectable bitrate is preset in this article.

The algorithms for implementing the above process are described below. All combinations of bitrate per stream need to be calculated to find the optimum bitrate, so the number of calculations is n times the number of L in the worst case. Therefore, the computational complexity is large when the number of users and the number of L increase. Limiting the number of participants is a serious problem for a tele-conference system, and the smaller the number of L, the more difficult it is to select a suitable bitrate. To solve this issue, this article focuses on the quality characteristic that the quality is improved more by increasing bitrate when the bitrate is lower and proposes a bitrate estimation method that increases the bitrate of a stream with a lower bitrate. The method is detailed in Algorithm 1. First, the bitrate for each user is set to the smallest value among L (lines 1-4), and then all users' Q_i is calculated (lines 6-8). If the minimum Q_i values exceed TargetQuality, the calculation ends (lines 9-11). Otherwise, the stream with the lowest O34 is selected (line 12). The reason the stream with the lowest value of O34 is selected is that the improvement effect by increasing bitrate is greater when O34 is low. A one step larger bitrate in L is selected for the stream (line 13). Repeat this step, and the processing is finished if overall quality (Q_i) for all participants exceeds TargetQuality or the bitrate for all streams selects the largest value of L.

Algorithm 1 Video bitrate estimation method

```
sort L accending
for all i ∈N do
    br_i ← L[0]
end for
while min(br) = max(L) do
    for all i ∈N do
        Calculate Q_i
    end for
    if min(Q) ≥ TargetQuality then
        break
    else
        Select smallest O34 Stream(i)
        one step larger bitrate in L
    end if
end while
```

Implementation

This section describes an implementation of the proposed method. The multimedia quality information collected by a client is derived from the information of WebRTC stats and device information is derived from UserAgent (Figure 1-2). In this evaluation, the WebRTC stats' sampling interval is one second. In the bitrate direction shown in Figure 1-6, the maximum bitrate is directed by using REMB message [17]. The client compares the bitrate calculated by GCC ($bitrate_{GCC}$) with the directed bitrate ($bitrate_{REMB}$) and selects the lower one.

Evaluation settings

This section describes evaluation settings for controlling the quality and effect of reducing the amount of transferred data by the proposed method. First, the quality-estimation model used in this assessment is explained, and then the evaluation conditions are described.

Coefficients of quality-estimation model

In this section, the coefficients of the quality-estimation model optimized by the method described in the Quality-estimation model section are shown. Thirty-five video sources (SRCs) that last 10 seconds each are used by taking into account the spatiotemporal information. The video sources show a man or woman who is talking 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 aspect ratio is 16:9, framerate: 15 and 30 fps, bitrate: 128, 256, 512, 1024 and 2560 kbps. In other words, total conditions are 40 (i.e. 4 resolutions $\times$ 2 framerates $\times$ 5 bitrates). As a training dataset, 1400 processed videos (PVSs) are generated. The optimized coefficients of the model are listed in Table 1. Root mean square error (RMSE) in the training data was 0.348 at laptop and 0.310 at SP, indicating that a model with sufficiently high accuracy is constructed.

To validate the model, 20 SRCs different from the training data are prepared. Each SRC is encoded by VP8. The bitrate values (i.e., 192, 320, 448, 704, and 1536 kbps) were changed from the training data, where the same resolution and framerate were used. As a validation dataset, 800 PVSs were generated. RMSE for the validation data was 0.362 at laptop and 0.298 at SP. This model was found to perform sufficiently because RMSE for the training data is almost the same as that for the validation dataset.

Table 1: Coefficients of quality-estimation model

	Value (laptop)	Value (SP)
v_1	1.130524	1.381678
v_2	154006.9	43737,49
v_3	0.074261	0.128961
v_4	7.29E-05	2.02E-05
v_5	0.99697	0.99697
v_6	91.52606	419.1394
v_7	0.194293	0.010929

	Value (all device)
a_1	4.964967
a_2	16.4606
a_3	2.08184
av_1	0.62
av_2	0
av_3	0.613691
av_4	0.068487
t_1	0.006666
t_2	4.04E-05
t_3	0.156498
t_4	0.14318
t_5	0.023864

Evaluation conditions

The purpose of this study is to reduce the amount of transferred data while bringing the quality close to TargetQuality that a service provider requires. To verify the effectiveness under various possible conditions, the number of participants (i.e., 3 or 5), terminal type (i.e., laptop and SP), and bandwidth limitations were varied as described in Table 2. TargetQuality is varied among [3, 3.2, 3.35, 3.5, 4, 5], and selectable bitrate L sets 128 (kbps) $\times x (x = 1\text{-}8)$ in all scenarios.

The maximum value of L is set to 1024 (kbps) because it is common as a bitrate of high-quality video. In addition, T is set to 30 seconds, and evaluation time per session is 300 seconds. When TargetQuality is set to 5, the behavior of a video bitrate control scheme is similar to the GCC because the video bitrate is not limited by TargetQuality.

Scenario 1 confirms how much the amount of data can be reduced under the high throughput condition. Three laptops are used in Scenario 1. Scenario 2 verifies whether the impact of the device type on quality affects the reduction of the transferred data. Two laptops and one SP are used in Scenario 2. Scenario 3 confirms whether the impact of the network condition on the quality affects the reduction of the transferred data, especially, one of the participants under a narrowband network. When applying the bandwidth limitation, the upstream bandwidth is limited at the client and only one client is limited. Three laptops are used in Scenario 3. Scenario 4 identifies how much the amount of transferred data can be reduced depending on the number of participants. Five laptops are used in Scenario 4. In the evaluation, the average value of overall quality for all participants during the evaluation and the total amount of audio and video stream upload data for all participants are used.

Table 2: Evaluation conditions

# of Scenario	# of laptop	# of SP	bandwidth limit
1.	3	0	-
2.	2	1	-
3.	3	0	512 (kbps)
4.	5	0	-

The experimental system is constructed to verify the effectiveness of the proposed method. The proposed method is implemented to the server, and Google Chrome is used as a client application. A laptop has a wired connection to the server and a SP has a wireless connection. Each client takes a video of an image of a moving person on a video monitor. In this evaluation, VP8 is used as video codec and Opus is used as audio codec, which are two of the major codecs in WebRTC. Also, the video is encoded by variable bitrate (VBR) encoding.

Results

This section shows the results under the conditions previously described. Figure 3 shows the total amount of upload data and average overall quality for TargetQualities of each scenario. The horizontal axis indicates the scenario number and TargetQuality. The blue bar shows the average overall quality of all participants for TargetQualities of each scenario, and the value is shown by the left vertical axis. The orange bar shows the total amount of upload data volume, and the value is shown by the right vertical axis.

Scenario 1

In Scenario 1, how much the transferred data can be reduced is investigated depending on the TargetQuality (3, 3.2, 3.35, 3.5, 4, 5), where throughput is always high enough. Depending on the TargetQuality, the total amount of upload data of all users is reduced and average overall quality is degraded because a smaller video bitrate is selected. When the TargetQuality is set to 3.35, the total amount of upload data of all users could be reduced by 42% while suppressing the degradation in the quality to about 0.13, compared with the case where the TargetQuality is set to 5 (i.e., the behavior of the proposed control scheme is similar to the GCC as mentioned above). In other words, the reduction of the large amount of data is obtained by the small-scale quality degradation (i.e., 0.13). However, the value of TargetQuality is further reduced to 3, the total amount of upload data is reduced by 73%, but the quality is degraded to 2.98. Therefore, the TargetQuality needs to be set considering the balance between the operational cost and quality.

Next, the result of setting TargetQuality to 5 is focused on. Quality was 3.46, which is smaller than TargetQuality. The client is directed to set the maximum video bitrate that is 1024 (kbps) and encode video around 950 (kbps) accordingly. The resolution is 640×480, and the framerate is set to 30 (fps) at this time. When a laptop is used, the resolution is 640×480, framerate is 30 (fps), and bitrate is 950 (kbps), and the O22 is 3.21. Therefore, average of overall quality was 3.46. Note that higher resolution (e.g., 1280 × 720 and 1920 × 1080) needs to be selected if O22 is increased.

Scenario 2

In Scenario 2, two laptops and one SP are used to investigate the impact of devices on the quality. The total amount of upload data has the same values as in Scenario 1. However, the average overall quality in Scenario 2 is higher than that in Scenario 1. The quality of a SP is higher than that of a laptop because of the impact of the screen size on the quality when the same bitrate is used. Therefore, this phenomenon is observed.

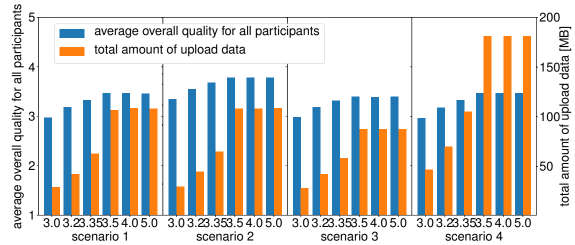

Figure 3. *The total amount of upload data and average quality when changing the TargetQuality for each scenario*

Scenario 3

In Scenario 3, how the network condition affects the quality and the amount of transferred data is investigated. The average overall quality is 3.40 when the TargetQuality is set to 3.5 in Scenario 3 while the average overall quality is 3.47 in Scenario 1. The reason the overall quality is lower in Scenario 3 than in Scenario 1 is that one video bitrate becomes low due to bandwidth limitation, as shown in Figure 4(b). When the TargetQuality is set to a small value (e.g., TargetQuality = 3.2, 3), the bitrate and overall quality are the same as in Scenario 1.

In Figure 4(a), the bitrate degraded temporarily at around 50 and 250 seconds. On the other hand, the maximum bitrate is selected for laptop with no bandwidth limitation in Figure 4(b). This is because, a laptop with bandwidth limitation cannot select appropriate video bitrate, and the other laptop selects a high bitrate to cover the quality degradation. Such quality improvement cannot be achieved by existing applications that limit the upper quality but can be achieved by controlling the bitrate on the basis of the quality.

Scenario 4

Finally, the extent to which the average of overall quality is affected depending on the number of participants is investigated. It is shown that the average overall quality is not affected by the number of participants and that the amount of transferred data increases depending on the number of participants, where the amount of transferred data per participant is not changed.

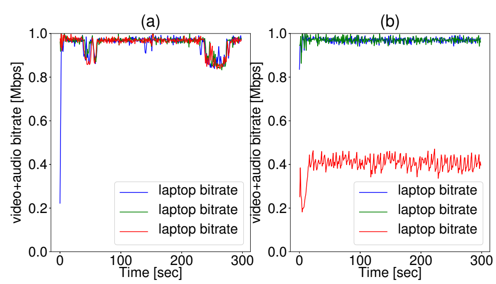

Figure 4. *Time-series video+audio bitrate when the TargetQuality set to 3.5. (a) Scenario 1 and (b) Scenario 3*

Discussion

Four simple scenarios were evaluated in this article. The results show that the amount of transferred data can be reduced while maintaining quality.

The maximum value of L is set to 1024 (kbps), but the quality value only increases to about 3.5 in Scenario 1. Video quality must be a HD video image and 2.5 Mbps or higher for the quality value to exceed 4.0 for the model. The above bitrate and resolution conditions will be evaluated in the future. The amount of transferred data is large when a higher bitrate is set, so the increase in the amount of transferred data will be evaluated.

Next, the selectable bitrate (L) is set in eight types, but it is not evaluated whether this number is optimal. Therefore, the optimum number of L and the optimum setting bitrate need to be investigated.

In the video bitrate estimation method, to reduce the amount of processing, the proposed method preferentially increases the bitrate for lower bitrate streams. However, as shown in the results in Scenario 2, the bitrate estimation does not take into account the information of devices. Since the quality can be improved effectively by selecting the bitrate considering the quality, a bitrate estimation method based on the quality should be considered. However, such an algorithm requires more computation, so it is necessary to evaluate not only quality improvement but also computational complexity and also determine whether it can be applied as an actual system.

As shown in the results in Scenario 3, the system can control the video bitrate in accordance with the available bandwidth by GCC when there is a participant with narrow band. However, a scenario in which the bandwidth changed during the teleconference was not included. It is assumed that the quality can be improved by selecting a higher bitrate after the bandwidth is improved when the quality is lowered due to the bandwidth limitation. Such scenarios will be evaluated in future investigations.

In Scenario 4, the effect of the number of participants was evaluated, but a reinvestigation is required considering the above algorithm changes.

TargetQuality is a very important parameter for reducing the amount of transferred data. In this study, TargetQuality is manually set by a service provider considering the balance of operational cost and quality. However, this value should be set automatically and appropriately on the basis of the cost to date. Such an automatic setting method is future work.

Conclusion

In this article, we proposed a quality-based video bitrate control method for web real-time communication (WebRTC)-based tele-conferences. We implemented the proposed method and evaluated the amount of transferred data and quality in several scenarios. The results show that the amount of transferred data can be reduced while maintaining quality if a suitable quality is selected.

The method was evaluated only in simple scenarios to assess its fundamental effect in this article. In the future, it needs to be evaluated under more complicated conditions such as changing the bandwidth over time in accordance with an actual communication environment. Also, the optimum value of some parameters (i.e., L) will be evaluated, and a bitrate estimation method based on quality will be examined to improve accuracy. Furthermore, it is necessary to consider how to set the value of TargetQuality automatically.

References

[1] M. Westerlund and S. Wenger, “RTP Topologies (RFC 7667),” in https://rfc-editor.org/rfc/rfc7667.txt

[2] S. Holmer, H. Lundin, G. Carlucci, L. De Cicco and S. Mascolo, “A Google Congestion Control Algorithm for Real-Time Communication,” in https://datatracker.ietf.org/doc/draft-ietf-rmcat-gcc/

[3] G. Carlucci, L. De Cicco, S. Holmer and S. Mascolo, “Congestion Control for Web Real-Time Communication,” *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 2629–2642, 2017.

[4] E. Janczukowicz, A. Braud, S. Tuffin, A. Bouabdallah and J. Bonnin, “Evaluation of network solutions for improving WebRTC quality,” *IEEE SoftCOM 2016*, Sept 2016.

[5] 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,” *IEEE INFOCOM 2020*, Aug 2020.

[6] V. Xhagjika, Ò. D. Escoda, L. Navarro and V. Vlassov, “Media streams allocation and load patterns for a WebRTC cloud architecture,” *IEEE NOF 2017*, Nov 2017.

[7] 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,” *IEEE/IFIP NOMS 2018*, July 2018.

[8] B. Grozev, L. Marinov, V. Singh and E. Ivov, “Last N: Relevance-Based Selectivity for Forwarding Video in Multimedia Conferences,” *ACM NOSSDAV 2015*, Mar 2015.

[9] “webex,” in https://www.webex.com/.

[10] “zoom,” in https://zoom.us/.

[11] T. Kimura, T. Kimura, A. Matsumoto and J. Okamoto, “BANQUET: Balancing Quality of Experience and Traffic Volume in Adaptive Video Streaming,” *IEEE CNSM 2019*, Feb 2020.

[12] B. García, M. Gallego, F. Gortázar and A. Bertolino, “Understanding and estimating quality of experience in WebRTC applications,” *Computing*, vol. 101, no. 11, pp. 1585–1607, 2019.

[13] K. Yamagishi, N. Egi, N. Yoshimura and P. Lebreton, “Derivation Procedure of Coefficients of Metadata-based Model for Adaptive Bitrate Streaming Services,” *IEICE Transactions on Communications*, vol. E104-B, no. 7, pp. 1–14, 2021.

[14] K. Yamagishi and T. Hayashi, “Parametric Quality-Estimation Model for Adaptive-Bitrate Streaming Services,” *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1545–1557, 2017.

[15] ITU-T Recommandation P.1203, “Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport,” *ITU-T*, 2017.

[16] Netflix, “VMAF,” in https://github.com/Netflix/vmaf.

[17] H. Alvestrand, “RTCP message for Receiver Estimated Maximum Bitrate,” in https://datatracker.ietf.org/doc/draft-alvestrand-rmcat-remb/

Author Biography

Masahiro Yokota received his bachelor’s (2009) and master’s (2011) degrees in science from Keio University, Tokyo, Japan. Since joining NTT Laboratories in 2011. His work has focused on the control technology based on quality of experience.

Kazuhisa Yamagishi received his B.E. degree in electrical engineering from the Tokyo University of Science 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. Since joining NTT Laboratories in 2003, he has been engaged in the development of objective quality-estimation models for multimedia telecommunications.