WebRTC-based Multi-View Video and Audio Transmission and Its QoE

Yuki Maehara and Toshiro Nunome
Department of Computer Science and Engineering, Graduate School of Engineering,
Nagoya Institute of Technology, Nagoya 466–8555, Japan
maehara@inl.nitech.ac.jp, nunome@nitech.ac.jp

***Abstract*—In this paper, we implement a Multi-View Video and Audio (MVV-A) transmission system utilizing WebRTC media channel, which employs UDP-based transmission into Web technologies, to enhance QoE under large delay. According to viewpoint change requests, this system switches audiovisual streams. We compare QoE with MVV-A transmission using MPEG-DASH, which employs HTTP/TCP, through a subjective experiment with various network conditions. As transmission methods utilizing MPEG-DASH, we treat single viewpoint transmission and simultaneous transmission of all the viewpoints. As a result, we find that the MVV-A transmission with WebRTC achieves higher QoE than that with MPEG-DASH under large delay.**

***Index Terms*—WebRTC, MPEG-DASH, MVV, Streaming, QoE**

I. Introduction

Web-based streaming services, in which a user watches video and audio through a Web browser, have been widely used. For the methodology of video and audio streaming, adaptive streaming such as MPEG-DASH (Dynamic Adaptive Streaming over HTTP) [1] is the mainstream.

In such services, for traversing firewalls and NAT (Network Address Translation), HTTP/TCP is usually employed [2],[3]. Meanwhile, WebRTC (Web Real-Time Communication) appears as a new mechanism to realize real-time communications between Web browsers with UDP [4]. The mechanism enhances the possibility of Web.

The Internet is best-effort. Thus, network delay, delay jitter, and packet loss can occur according to network congestion. They cause short and long pauses of output. The pauses degrade not only QoS (Quality of Service) but also QoE (Quality of Experience) [5]. QoE is the perceptual quality of users. The ultimate goal of the network services is the provision of high QoE.

As a multimedia service over the Internet, MVV (Multi-View Video) [6] has become popular. In MVV, users can select a viewpoint from multiple viewpoints. In this study, we deal with MVV-A, which is MVV accompanied by audio [7].

Reference [8] evaluates a trade-off between viewpoint change delay and video quality of the new feed through a subjective experiment. However, the users in this experiment do not change viewpoint because viewpoint change delay, video quality, and timing of viewpoint change occurrence are determined in advance. Thus, the study does not assume practical usage.

In [9], Maehara and Nunome evaluate a simultaneous transmission method for MVV-A transmission with MPEG-DASH. They employ two transmission methods. One is the simultaneous transmission method, which requests video data for all the viewpoints simultaneously. The other is the selected single viewpoint transmission method; it requests the data of viewpoint selected by the user. They conduct a subjective experiment and compared their QoE. As a result, under the highly loaded network condition, the users prefer the simultaneous transmission method because of ability to change viewpoint immediately.

The experiment in [9] does not consider large network delay. TCP employs acknowledgment and ARQ (Automatic Retransmission reQuest) mechanisms. In networks with large delay, the mechanisms cannot work well, and then the performance of TCP may degrade. It affects QoS and QoE of MVV-A transmission over HTTP/TCP.

In this study, we employ WebRTC for MVV-A transmission. References [10], [11], and [12] deal with QoE issues on WebRTC. In [10], the effect of CPU, resolution, and display size on QoE in a teleconference over WebRTC is evaluated by means of a subjective experiment. Reference [11] investigates the relationship between video pauses and QoE on WebRTC-based video communications through statistical information obtained by Google Chrome. In [12], an online questionnaire is performed to assess the effect of audio quality, image quality, video frame loss, among others on QoE. However, all of the studies are for single viewpoint video.

This study aims to QoE enhancement of MVV-A transmission with WebRTC, in which UDP is employed as the transport protocol, under large delay conditions. We implement an MVV-A transmission system by means of WebRTC media channel. We then evaluate QoE through a subjective experiment.

The rest of this paper is structured as follows. Section II introduces the MVV-A system with WebRTC. Section III explains the method of the experiment. Section IV presents experimental results. Section V concludes this paper.

II. WebRTC-based MVV-A system

In the MVV-A system with WebRTC, the audio and video streams are transmitted through the WebRTC media channel. The channel employs SRTP (Secure Real-time Transport Protocol)/UDP.

The users can watch contents from four viewpoints while selecting a viewpoint arbitrarily. Audio and video data are stored on the server beforehand. The client requests the chosen viewpoint by the user to the server, and the server transmits the requested viewpoint data to the client.

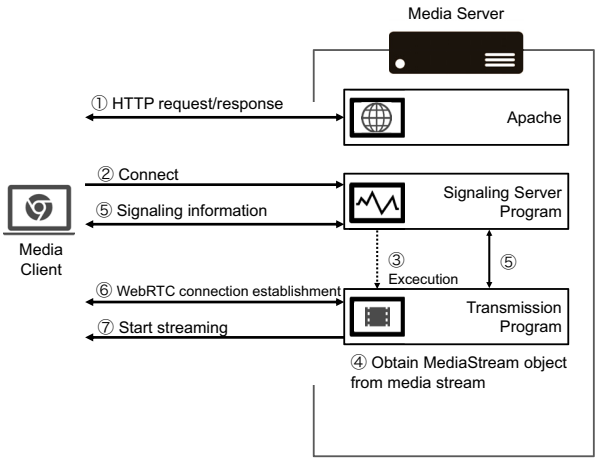

Fig. 1. MVV-A system with WebRTC

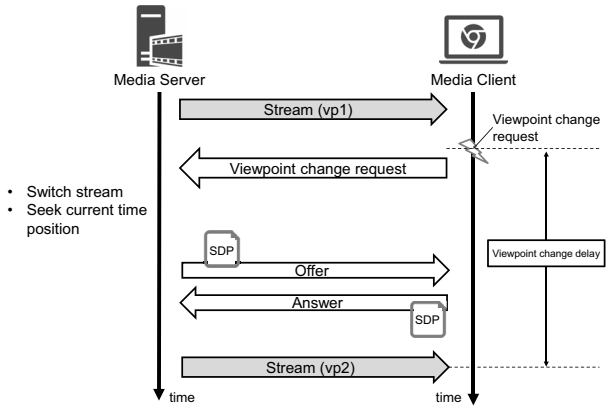

Fig. 2. Sequence of viewpoint change

To implement our MVV-A system, we utilize Electron [13], which is an opensource library for building cross-platform applications. Figure 1 shows the MVV-A system model by means of WebRTC. ① Media Client gets an HTML file from Media Server. ② It then connects to Signaling Server Program according to the description in the HTML file. When the connection is established to Media Client, ③ Signaling Server Program executes Transmission Program. ④ Transmission Program generates a MediaStream object of media to be transmitted by means of a JavaScript API named captureStream(). After obtaining the MediaStream object, ⑤ Transmission Program establishes a WebRTC connection to Media Client via Signaling Server Program. ⑥ After the establishment, ⑦ Media Client can output the media stream sent from Transmission Program through the WebRTC media channel.

Figure 2 shows a sequence of viewpoint change. We utilize removeStream() and addStream() methods for viewpoint change in WebRTC. The removeStream()/addStream() removes/adds a transmission stream on WebRTC connection. When these methods are executed, Media Server informs removal/addition of the stream through Offer/Answer and exchanges SDP (Session Description Protocol). Media Client counts the output time after starting the output of initial viewpoint's video. When the viewpoint change occurs, Media Client informs a requested viewpoint and the current time position at occurring the viewpoint change request as a viewpoint change request message via a WebRTC data channel, which employs SCTP as the transport protocol. On receiving the request message, Media Server changes the stream by using removeStream() → addStream(). On transmitting the requested viewpoint, Media Server adjusts the output time position to the received one through the viewpoint change request message for continuous output at Media Receiver.

In this paper, we do not refer to the existence of NAT for simplicity. However, we can use several techniques for NAT traversal such as STUN (Session Traversal Utilities for NAT) [14].

III. Methodology of Experiment

A. Experimental system

Figure 3 depicts an experimental network in this study. The OS of Media Server is CentOS 7, and that of Media Client is Windows 7. The two Routers are Alcatel Lucent (formerly RiverStone Networks) RS3000. All the links are 100 Mbps duplex Ethernet.

Media Server sends the audio and video of viewpoints to Media Client. Load Sender sends the load traffic with HTTP/TCP to Load Receiver according to requests generated by Webstone 2.5, which is a Web server benchmark tool. Webstone creates Web client processes on Load Receiver by simulating the activity of multiple clients. Both Load Sender and Media Server in MPEG-DASH are Apache 2.2. Netem, which is laid out between Media Server and the left-side Router, is a PC installed network emulator. Netem delays packets through Media Server and the Router.

We employ Google Chrome as the Web browser for playing the audio and video in Media Client. We employ WebM as a container format for the audio and video streams. The video and audio encoding formats are VP8 and Vorbis, respectively. We compare the WebRTC transmission with the MPEG-DASH single viewpoint transmission method and the MPEG-DASH simultaneous transmission method; the latter two methods are employed in [9].

Table I shows the specifications of video and audio. In MPEG-DASH, we encoded the video into four types: 200 kbps (image size 426×240 pixels), 500 kbps (640×360 pixels), 1000 kbps (854×480 pixels) and 1500 kbps (1280×720 pixels). The 200 kbps video is used as the simultaneous video; it is only for the simultaneous transmission of unfocused viewpoint. The 500, 1000 and 1500 kbps videos are changed seamlessly depending on the load condition of the network.

For WebRTC, we only employ the video of 1500 kbps. WebRTC automatically controls the video frame rate and video image quality according to the network conditions. We obtain QoS through the WebRTC Statistics API.

B. QoE assessment method

In the subjective experiment, an assessor watches a toy train which runs on plastic rails with changing the viewpoint;

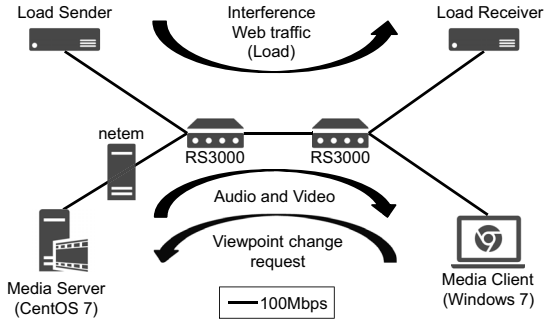

Fig. 3. Network configuration

TABLE I
SPECIFICATIONS OF VIDEO AND AUDIO

	item	value
video	codec	VP8
	frame rate [fps]	30
	GOP length	30
	bitrate [kbps] (image size [pixels])	200 (426×240), 500 (640×360), 1000 (854×480), 1500 (1280×720)
	container format	WebM
audio	codec	Vorbis
	bitrate [kbps]	32
	channel	mono
	sampling rate [kHz]	8

it is one of the two contents employed in [9]. The camera arrangement is shown in Fig. 4. Although we need to assess various types of contents, this study employs the content as the first step. In addition, the difference between the contents on the assessment results is not large in [9].

In this study, we perform a multidimensional QoE assessment. We present audio and video to the users in a combination of experimental conditions as a stimulus. Table II shows adjective pairs for evaluating each stimulus. The adjectives are classified into five categories: response, video, audio, synchronization, and overall quality. Abbreviated names from r1 to o1 are attached to the pairs of polar terms.

The assessors evaluate each criterion with the rating scale method. The rating scale provides a numerical indication of the perceived quality and is expressed as a single number in the range 1 to 5. The worst grade (score 1) means the negative adjective (the left-hand side one in each pair), while the best

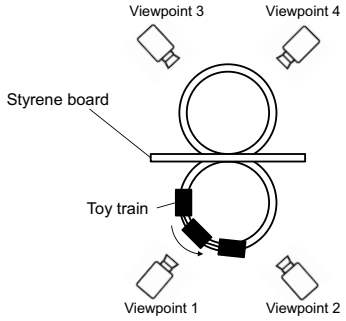

Fig. 4. Camera arrangement

TABLE II
ADJECTIVE PAIRS FOR QOE ASSESSMENT

category	adjective pairs
response	r1: slow - fast r2: unstable - stable
video	v1: rough - smooth v2: blurred - sharp
audio	a1: artificial - natural
synchronization	s1: out of synchronization - in synchronization
overall	o1: bad - excellent

TABLE III
GRADES IN "R1: VIEWPOINT CHANGE RESPONSE IS SLOW - FAST"

score	grade
5	fast
4	a little fast
3	moderate
2	a little slow
1	slow

grade (score 5) represents the positive adjective (the right-hand side one). The middle grade (score 3) is neutral. For example, each grade is defined for "r1: viewpoint change response is slow - fast" as shown in Table III. Finally, we calculate the MOS (Mean Opinion Score), which is the average of the rating scale scores for all the assessors, for each criterion.

For the number of client processes in Webstone 2.5 at Load Receiver, we employ 0, 30, and 50. The delay in Netem is set to one of the three constant values: 0 ms, 50 ms (assuming delay from Japan to the U.S.A.) or 100 ms (assuming delay from Japan to Europe). The duration of an experimental run is 25 seconds. After each experimental run, the assessor evaluates the seven criteria. The number of stimuli is 29; they are two dummy stimuli and the combinations of the three methods, the three values of Web client processes, and the three values of constant delay.

The assessors are 15 male students in their twenties. ITU-T Rec. P.911 describes that at least 15 subjects should participate in the experiment [15]. Thus, we employed the 15 assessors. On the other hand, we need to evaluate with more assessors; it is a future study issue. Before the experiment, the assessors have practiced without delay and load.

IV. EXPERIMENTAL RESULT

In the figures, the abscissa is the combination of the number of Web client processes and the additional delay value. We also show 95 % confidence intervals. In the legend, "DASH single" means the MPEG-DASH single viewpoint transmission method, "DASH simultaneous" represents the MPEG-DASH simultaneous transmission method of all the four viewpoints, and "WebRTC" is the proposed transmission method with WebRTC.

A. Application-level QoS

1) Average viewpoint change delay: Figure 5 shows the average viewpoint change delay. It is the average time between when the user clicked viewpoint change button and when video for the changed viewpoint is displayed.

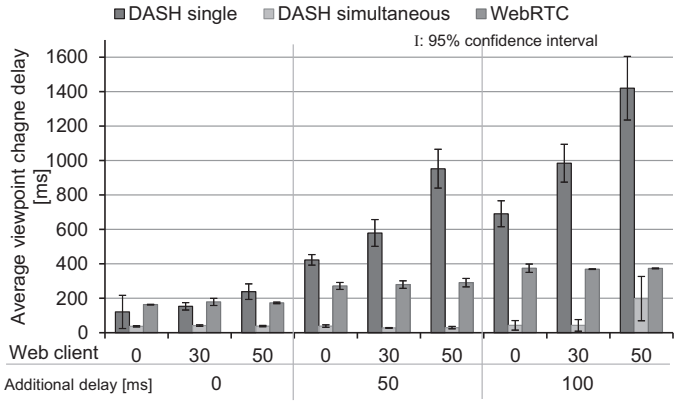

Fig. 5. Average viewpoint change delay

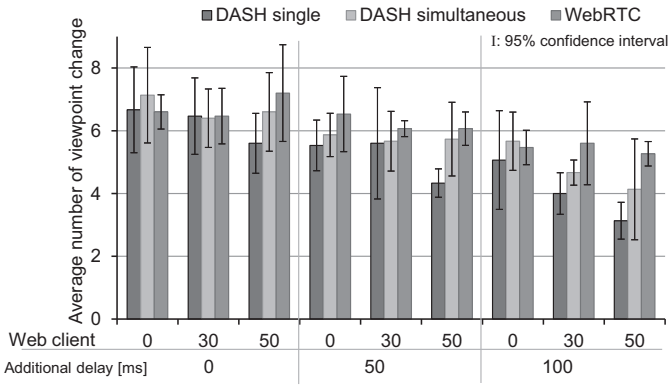

Fig. 6. Average number of viewpoint change

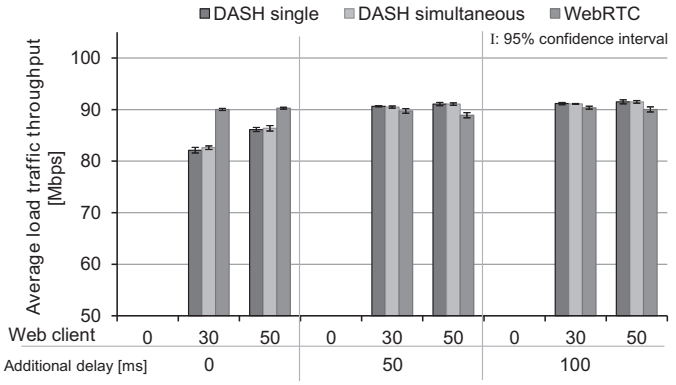

Fig. 7. Average load traffic throughput

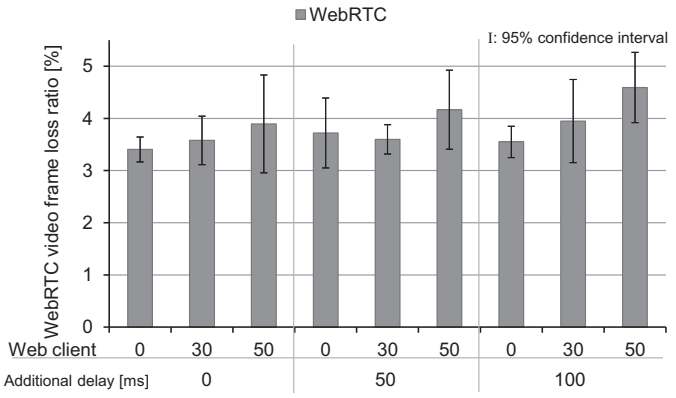

Fig. 8. WebRTC video frame loss ratio

We notice in Fig. 5 that the viewpoint change delay in WebRTC is longer than that in the MPEG-DASH simultaneous transmission method and is shorter than that in the MPEG-DASH single viewpoint transmission method. In addition, the number of Web clients scarcely affects the delay in WebRTC, and the 95 % confidence interval is also small. Hence, we can confirm that the user can change the viewpoint with a constant response in the WebRTC-based transmission. In WebRTC, i.e., UDP-based transmission, the interference traffic causes packet loss. On the other hand, the delay does not become large because of no retransmission mechanism.

2) Average number of viewpoint changes: We show the average number of viewpoint changes in Fig. 6. It represents the average of the number of viewpoint changes during an experimental run.

We find in the figure that the number of viewpoint changes tends to decrease as the delay and the load traffic increase. This is because the net viewing time for the user decreases as the initial delay and the viewpoint change delay increase. As for the number of viewpoint changes in WebRTC, the amount of load traffic does not affect the number of viewpoint changes largely.

3) Average load traffic throughput: Figure 7 shows the average load traffic throughput in each experimental run, i.e., 25 seconds. The average load traffic throughput in WebRTC is the same as or larger than those in the MPEG-DASH methods. This implies that WebRTC does not steal a large amount of bandwidth from the other flows although WebRTC utilizes UDP as the transport protocol.

4) WebRTC video frame loss ratio: Figure 8 depicts the WebRTC video frame loss ratio. It is the ratio of the number of video frames which cannot be decoded to the number of video frames which should be decoded during the MVV-A transmission with WebRTC. As for the MPEG-DASH methods, there is no frame loss because they use TCP.

We see in the figure that the loss ratio without load traffic is about 3 %. This is because of the viewpoint changes. As we explained in Fig. 2, when Media Server receives the viewpoint change request, it discards the stream of old viewpoint and adds the stream of the new viewpoint. The discarding/adding processes cause the video frame loss at Media Receiver.

In addition, we notice that the loss ratio increases as the load traffic increases. This is due to the network congestion.

B. QoE

We show the QoE assessment results in Figs. 9 through 15.

1) Viewpoint change response is slow - fast: We notice in Fig. 9 that WebRTC has the higher MOS value of "r1: viewpoint change response is slow - fast" than the MPEG-DASH single viewpoint transmission method, but it has smaller MOS value than the MPEG-DASH simultaneous transmission method. As the load traffic increases, the MOS value in

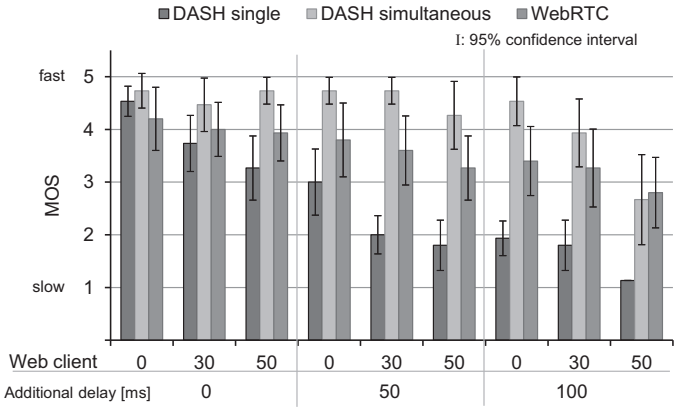

Fig. 9. MOS (r1: viewpoint change response is slow - fast)

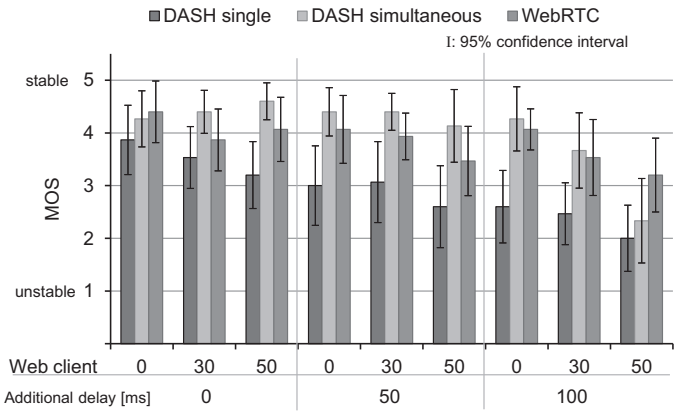

Fig. 10. MOS (r2: viewpoint change is unstable - stable)

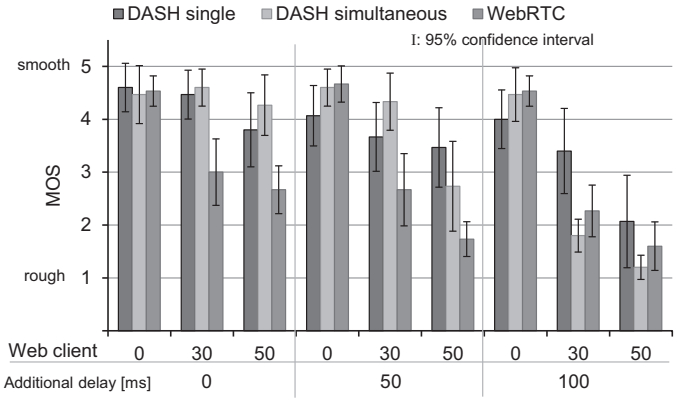

Fig. 11. MOS (v1: video is rough - smooth)

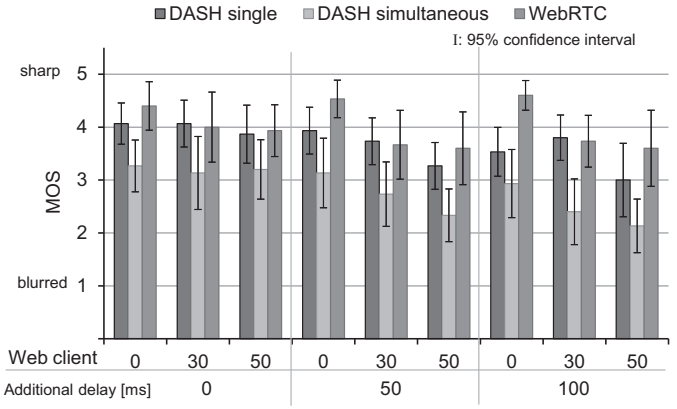

Fig. 12. MOS (v2: video is blurred - sharp)

WebRTC tends to decrease, although the viewpoint change delay does not increase in Fig. 5. This is the effect of small pauses owing to video packet loss.

2) Viewpoint change is unstable - stable: We see in Fig. 10 that the result of "r2: viewpoint change is unstable - stable" is almost the same tendency as that of "r1: response is slow - fast" in Fig. 9.

3) Video is rough - smooth: We notice in Fig. 11 that WebRTC tends to have the smaller MOS value of "v1: video is rough - smooth" than the other methods. This is because WebRTC suffers packet loss under heavy traffic condition. When the load traffic is small, the MOS value in WebRTC is not so different from that in the MPEG-DASH methods.

4) Video is blurred - sharp: In Fig. 12, we see that WebRTC has the larger MOS value of "v2: video is blurred - sharp" than the other methods or approximately the same MOS value as the MPEG-DASH single viewpoint transmission method. Although all the methods degrade the video image quality at the viewpoint change, the user hardly notices instantaneous image quality degradation in WebRTC. In addition, the MOS value in WebRTC decreases as the interference traffic increases. This is because the user is affected by the disturbance of smooth output due to packet loss.

5) Audio is artificial - natural: We find in Fig. 13 that the MOS value of "a1: audio is artificial - natural" decreases as the load traffic increases in WebRTC. This is because of audio packet loss. The audio packet loss is easily noticeable for the users. Thus, QoE will enhance by reliable transmission of audio stream; an application of WebRTC data channel is an idea for this.

6) Out of synchronization - in synchronization: In Fig. 14, we can observe that the MOS value of "s1: out of synchroniza-

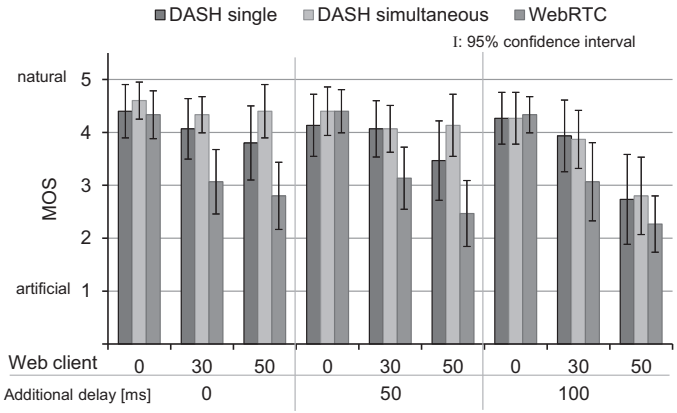

Fig. 13. MOS (a1: audio is artificial - natural)

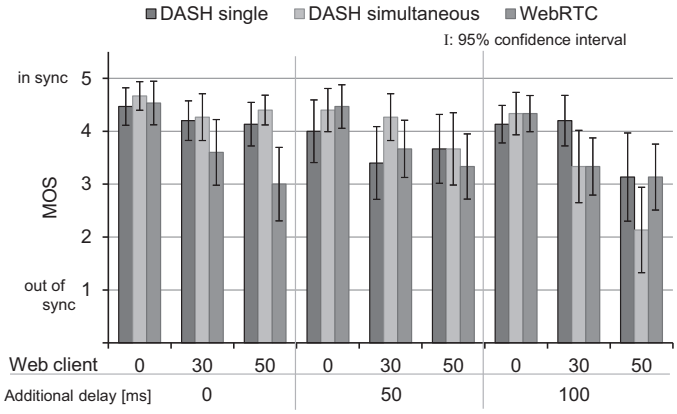

Fig. 14. MOS (s1: out of synchronization - in synchronization)

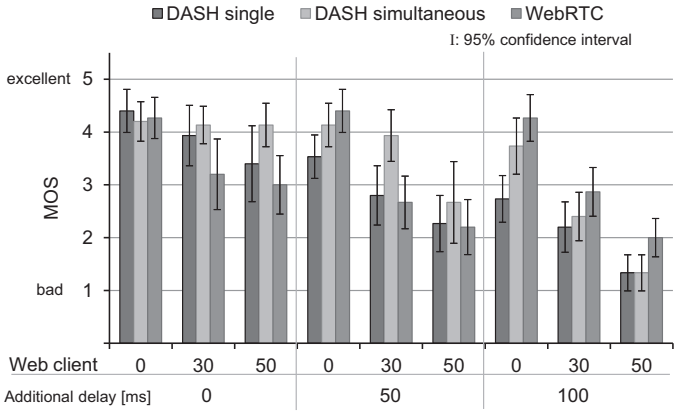

Fig. 15. MOS (o1: bad - excellent)

tion - in synchronization" in WebRTC decreases as the load traffic increases. This is because of audio and video packet loss; they cause the instance in which only audio or video is output.

7) Bad - excellent: We find in Fig. 15 that WebRTC achieves the high MOS value of "o1: bad - excellent" with no load traffic irrespective of additional delay. On the other hand, as the load traffic increases, the MOS value decreases.

When the additional delay is 100 ms, the MOS value in WebRTC is the highest among the three methods irrespective of the load traffic. Under the delay condition, TCP cannot perform well in the methods with MPEG-DASH. Then, the long pauses of video output occur. On the other hand, WebRTC employs UDP, and then it does not suffer such the large pauses.

Table IV shows the correlation coefficients between "o1: bad - excellent" and the other adjective pairs in descending order in the MVV-A transmission with WebRTC. We can observe in the table that the overall quality mostly correlates with video smoothness and audio naturality. Thus, in the WebRTC-based MVV-A transmission, the effect of packet loss can mainly affect the overall quality.

V. Conclusions

This paper proposed the Web-based MVV-A system using WebRTC for QoE enhancement under large delay conditions.

TABLE IV
Correlation coefficient with o1 in WebRTC transmission

adjective pair	coefficient
v1: video is rough - smooth	0.837
a1: audio is artificial - natural	0.694
s1: out of synchronization - in synchronization	0.573
v2: video is blurred - sharp	0.566
r2: viewpoint change is unstable - stable	0.512
r1: response is slow - fast	0.501

We compared the proposed method with the MPEG-DASH single viewpoint transmission method and the MPEG-DASH simultaneous viewpoints transmission method through the subjective experiment; the latter two methods are employed in [9]. As a result, we found that the WebRTC-based MVV-A system can achieve higher QoE than the systems based on MPEG-DASH. This is because WebRTC can output audio and video without larger pauses than those in MPEG-DASH. However, WebRTC suffers packet loss under heavily loaded conditions because it employs UDP as the transport protocol.

In future work, we need to assess QoE with other contents. We should also devise a mechanism for mitigating the effect of packet loss. The employment of WebRTC data channel can be a solution.

References

[1] ISO/IEC 23009-1, "Dynamic adaptive streaming over HTTP (DASH) Part1: Media presentation description and segment formats," May 2014.
[2] Cisco WhitePaper, "HTTP versus RTMP: Which way to go and why?," 2011.
[3] A. C. Begen, T. Akgul and M. Baugher, "Watching video over the web, part I: Streaming protocols," IEEE Internet Computing, vol. 15, no. 2, Mar./Apr. 2011.
[4] "Webrtc," *https://webrtc.org/*
[5] ITU-T Rec. P.10/G.100, "Vocabulary for performance, quality of service and quality of experience," Nov. 2017.
[6] I. Ahmad, "Multi-view video: Get ready for next-generation television," Proc. IEEE Distributed Systems Online, vol. 8, no. 3, art. no. 0703-o3006, Mar. 2007.
[7] E. Jimenez Rodriguez, T. Nunome and S. Tasaka, "QoE assessment of multi-view video and audio IP transmission," IEICE Trans. Commun., vol. E92-B, no. 6, pp. 1373-1383, June 2010.
[8] N. Staelens, P. Coppens, N. V. Kets, G. V. Wallendael, W. V. Broeck, J. D. Cock and F. D. Truck, "On the impact of video stalling and video quality in the case of camera switching during adaptive streaming of sports content" Proc. IEEE QoMEX 2015, pp. 1-6, May 2015.
[9] Y. Maehara and T. Nunome, "Multidimensional QoE assessment of a simultaneous transmission method in multi-view video and audio transmission with MPEG-DASH," Proc. IEEE CIT 2017, pp. 101-106, Aug. 2017.
[10] D. Vucic and L. Skorin-Kapov, "The impact of mobile device factors on QoE for multi-party video conferencing via WebRTC," Proc. ConTEL, pp. 1-8, July 2015.
[11] D. Ammar, K. D. Moor, M. Xie, M. Fiedler and P. Heegaard, "Video QoE killer and performance statistics in WebRTC-based video communication," Proc. IEEE ICCE 2016, pp. 429-436, Sep. 2016.
[12] J. B. Husić , S. Baraković and A. Veispahić, "What factors influence the quality of experience for WebRTC video calls?," Proc. MIPRO, pp. 428-433, May 2017.
[13] "Electron," *https://electron.atom.io/*
[14] J. Rosenberg, R.Mahy, P. Matthews, and D. Wing, "Session traversal utilities for NAT (STUN)," RFC 5389, Oct. 2008.
[15] ITU-T Rec. P.911, "Subjective audiovisual quality assessment methods for multimedia applications," Dec. 1998.