Web-Based and Quality-Oriented Remote Collaboration Platform Tolerant to Severe Network Constraints

Yasuhiro MOCHIDA^{†a)}, Daisuke SHIRAI[†], and Tatsuya FUJII[†], *Members*

SUMMARY Existing remote collaboration systems are not suitable for a collaboration style where distributed users touch work tools at the same time, especially in demanding use cases or in severe network situations. To cover a wider range of use cases, we propose a novel concept of a remote collaboration platform that enables the users to share currently-used work tools with a high quality A/V transmission module, while maintaining the advantages of web-based systems. It also provides functions to deal with long transmission delay using relay servers, packet transmission instability using visual feedback of audio delivery and limited bandwidth using dynamic allocation of video bitrate. We implemented the platform and conducted evaluation tests. The results show the feasibility of the proposed concept and its tolerance to network constraints, which indicates that the proposed platform can construct unprecedented collaboration systems.

key words: Remote collaboration, Web-based system, Network constraint

1. Introduction

The globalization of business and the expansion of telecommuting are stimulating the demand for remote collaboration among spatially distributed users [1]. Bjørn et al. analyzed global software development companies and reported that the workflows have been successful [2]. In software development workflows, a total job can be segmented in a sequential, time-line manner, and thus each part can be processed by an individual worker. The efficiency depends on how the segmented jobs are combined into a serial process while sharing information and artifacts among these individuals or relay them from one to the next. Many well-designed coordination tools dedicated to specific workflows such as software development [3] and content creation [4] have been developed to assist management of segmented jobs. However, we consider that we can get higher creativity by gathering experts together on an artifact. Therefore, our research goal is to develop a real-time remote collaboration system where multiple people work on the same artifact at the same time.

Teleconference systems are basic examples of real-time remote collaboration tools and a vast range of approaches to them has been reported in previous studies [5]. Nakanishi et al. developed a teleconference system equipped with remote handshaking, which enhanced the feeling of being close to the remote partner [6]. FocalSpace guides participants' focus by diminishing unimportant video regions, which enables better memory

retention [7]. Due to the progress of audio/video (A/V) technologies, teleconference systems have achieved enough A/V quality for decision-making under the conditions that wide network bandwidths are available. The high-end teleconference products called telepresence systems provide whole room designs with prescribed arrangements of multiple HD videos, multichannel audio and furniture to guarantee the user experience [8], [9]. We also developed an extremely realistic teleconference system with triple 4K/60p videos and a 6-channel acoustic echo canceller that delivers nonverbal cues, such as micro-expressions [10].

These teleconference systems are designed for group decision making, so they are not optimized for cooperation, such as product design and movie creation where multiple users want to interact with artifacts at the same time. Google Docs is an online document editing system where distributed users can edit a document at the same time [11]. However, its use cases are very limited because it covers only a few basic document types. For remote movie creation, shared views of editing software and media assets are required. These shared tools also need to be operated or edited both by remote and local users. Although teleconference systems offer desktop sharing, they are optimized for static document presentation in a decision making phase. Thus, they are view-only for remote users and their video quality is not sufficient to check artifact details such as 3D model motion or movie previews.

For professional workflows, we need to consider that most experts have their own preference on specific software tools, so these currently-used work tools should be used even in the cooperation phase. It's ideal that these tools can be enhanced by their vendors to handle simultaneous access from multiple users; however, it is out of the user's control. Our approach is virtual enhancement by high quality desktop sharing and low latency remote operation, which can be a universal solution as long as the work tools are digital.

In designing remote collaboration systems for globally distributed users using common IP networks such as the Internet, it's essential for these systems to be able to handle constraints imposed by networks, namely transmission delays between distributed workplaces, packet transmission instability and limited bandwidth.

In this paper, we propose a web-based remote collaboration platform to construct a collaboration system that enables users to share currently-used work tools [12]. We call it REMote Collaboration Open Platform (REMOCOP). REMOCOP provides a framework to share work tools with

Manuscript received March 16, 2017.

Manuscript revised August 4, 2017.

Manuscript publicized January 19, 2018.

[†]The authors are with NTT, Yokosuka-shi, 239–0847 Japan.

a) E-mail: mochida.yasuhiro@lab.ntt.co.jp

DOI: 10.1587/transinf.2016IIP0024

a high quality A/V transmission module suitable for demanding remote collaboration, while maintaining the advantages of web-based systems. Thus, it achieves a level of quality that existing web-based collaboration systems are unable to provide. To address the problems caused by long transmission delay, a new data exchange method for web-based collaboration systems is also proposed [13]. In order to maintain remote collaboration quality in unstable networks, we propose a visual feedback method of audio delivery [14]. To make the most of the limited network bandwidth, we develop a JPEG2000 A/V transmission module that can control its bitrate instantly [15].

The rest of this paper is organized as follows. Section 2 presents background information related to remote collaboration platforms over common IP networks. Section 3 details the concept of REMOCOP and its approaches to handle network constraints. Section 4 describes the platform architecture. Section 5 explains obtained evaluation results. Finally, conclusions are drawn in Sect. 6.

2. Background

In this section, we present research background related to real-time remote collaboration using IP networks. In the early days of remote collaboration systems, they were hardware equipment or native software dependent on the operating system (OS). They then evolved to feature web-based and mobile-oriented systems independent of devices or OSs. However, existing web-based collaboration systems are unsuitable for demanding collaboration and not designed to handle network constraints.

2.1 Collaboration Platform

ConferenceXP [16] is a typical example of a collaboration platform that provides teleconferencing, desktop sharing and APIs to add customized functions. However, ConferenceXP-based systems must be Microsoft Windows native applications because ConferenceXP is dependent on Windows functions. The users are required to use Windows devices and install the collaboration software including its dependencies beforehand. In addition, tight coupling with Windows is unfavorable in terms of development efficiency, taking into account its utilization of existing resources.

Web-based collaboration systems implemented only with HTML5 and WebRTC, an A/V transmission framework embedded in web browsers, are attracting considerable attention because they don't require the installation of dedicated software and the users can use a variety of devices, including PCs, Mac devices and tablets [17]. In addition, there are a lot of libraries that accelerate web application development. Although WebRTC itself is a peer-to-peer communication framework, it is also able to support large group collaboration using A/V distribution servers [18]. Vclick is a platform for web-based collaboration systems composed of a web server and clients [19]. Vclick provides teleconferencing with WebRTC, screen sharing, white-board

sharing and a WebSocket-based data exchange framework to implement new functions. WeColLab is a platform for online laboratories, where students remotely interact with others and laboratory equipment, such as microscopes [20], [21]. It provides a framework to deploy different equipment using redirector servers in addition to a web server. WebRTC is suitable for mobile meetings on mobile devices, such as laptop computers or tablets and their narrow-bandwidth networks, because it highly compresses video bandwidth using inter-frame coding. However, WebRTC provides limited video quality and high latency, which means that it is unsuitable for the demanding cooperation described in Sect. 1.

2.2 Network Constraints

A. Long Transmission Delay

Clients of real-time collaboration systems are always exchanging data such as cursor position, drawing and remote control with each other in real time. The standardized protocol for real-time and bidirectional data exchange on web applications is WebSocket. RTCDataChannel included in WebRTC is another option of real-time communication, which allows peer-to-peer connection; however, it is still being standardized and thus has less web browser support than WebSocket. Moreover, as detailed in Sect. 4.2, peer-to-peer connection is not suitable for data exchange in large group collaboration. Therefore, we utilized WebSocket.

In globally distributed collaboration, data exchange among web applications tends to fall apart due to long transmission delay. In WebSocket, all data exchanges must be routed through a web server over TCP because web applications cannot be directly connected due to the web browser security policy. Thus, the web server distributes data received from one to the others. This limitation causes problems in data exchange both with remote clients and with local clients when there is a long transmission delay.

In data exchange with remote clients, WebSocket can cause an additional delay. Once a data packet is lost, the following packets are buffered and not processed until the lost packet is retrieved since WebSocket is a TCP-based protocol and the packet order must be preserved. If the network one-way delay is 100 ms, there will be a pause of more than 200 ms in the packet processing to wait for packet retransmission. If the pause appears when a user is moving a cursor to point at a picture, the other users cannot see what is pointed at.

Data exchange among local clients in web-based collaboration systems, such as SAGE2 [22], that have separated display clients and interaction clients in each location suffers from a problem. The reaction to the user input appears on the display clients with significant delay if the networks between the server and the clients have long transmission delay. This is also because all data exchanges must be routed through the web server, even though the clients in the same location want to directly communicate with each other.

B. Packet Transmission Instability

In unstable networks, teleconferencing is often disconnected due to the packet transmission instability such as high packet loss rate and delay jitter [23], [24], while the users are not always aware of the disconnection. Remote collaboration quality will be degraded if the users are unsure whether everyone is seeing and hearing the same thing or whether their voices are being delivered to the others. Two approaches have been used to deal with packet transmission instability. One approach is to utilize networks with guaranteed quality of service (QoS), such as leased lines. However, they are costly and it is difficult to guarantee QoS in mobile networks. The other approach is to provide feedbacks to enable the users to be aware of whether or not the systems and networks are working well. Some commercial products such as MeetingPlaza [25] are equipped with a voice level indicator on the display console to show whether the speakers' voices are being captured by a local microphone at a moderate level, but speakers cannot know whether the listeners are receiving their voices.

C. Limited Bandwidth

Generally, video transmission occupies most of the bandwidth consumption in remote collaboration. To reduce bandwidth consumption H.264 is widely used. Recently, H.265 is attracting a lot of attention because it achieves higher efficiency than that of H.264 in exchange for larger computation complexity. In high-end teleconference systems (e.g., Ref. [10]) intra-frame coding, such as H.264 (intra mode), H.265 (intra mode) and JPEG2000, is adopted to reduce coding latency.

3. Concept Proposal of REMOCOP

In this section, we propose the design concept of REMOCOP. REMOCOP provides high quality A/V in desktop sharing and teleconferencing suitable for demanding use cases, while maintaining the advantages of web-based systems. REMOCOP also provides functions to handle network constraints including long transmission delay, packet transmission instability and limited bandwidth.

3.1 Collaboration Model

We assume a collaboration model including workstation clients, display clients and interaction clients as depicted in Fig. 1. Distributed users are working on the same work tool running on a workstation that has a workstation client to share the desktop video and to accept remote operation. The work tool doesn't have to be designed for REMOCOP, which means that off-the-shelf and currently-used software can be used. The workstation can be anywhere such as an office, a server room or a data center. An interaction client provides such things as desktop sharing, teleconferencing, remote control, pointing, drawing, and asset sharing. The users manipulate interaction clients for operation. Display clients are view-only clients to provide desktop sharing and

Fig. 1 A collaboration model in REMOCOP.

teleconferencing with higher quality than that of interaction clients. Thus, display clients are assumed to be running on high performance machines in the office where wide bandwidth is available.

3.2 Web-Based and Quality-Oriented Collaboration

As mentioned in Sect. 2.1, web-based collaboration systems are very convenient. However, it's quite difficult to satisfy A/V quality requirements in demanding cooperation use cases. Our approach is combining a web application implemented only by HTML5 & WebRTC with a native A/V transmission module into a web-based and quality-oriented collaboration system. The web application provides the type of basic level service and it is used as an interaction client. Thereby, the users can join the collaboration using their own device without installation of dedicated software. The native A/V transmission module optimized for a particular OS is used in the display client to satisfy the requirements of A/V quality and latency because it has the potential to achieve higher performance than that of WebRTC.

The advantage of our approach is that different levels of service quality can be provided on the basis of a single web application in accordance with the user situation. For example, if the user is out of the office and only a mobile device and narrowband network are available, he/she joins the collaboration using the interaction client. On the other hand, if the user is in the office and both a high performance machine and wide bandwidth are available, the display client as well as the interaction client is used to achieve higher quality and lower latency compared to WebRTC.

3.3 Tolerance to Network Constraints

Considering the remote collaboration in the common IP networks, we added three functions to handle network constraints described in Sect. 2.2.

A. Data Exchange Tolerant to Long Transmission Delay

To address the two problems in real-time data exchange under long transmission delay, two approaches are adopted. To solve the first problem of the long pause, we focus on the characteristics of data exchanged in real-time collaboration systems. For example, even if a single position data of

pointer motion is lost, the user experience is not degraded significantly since it causes only a slight discontinuity in pointer motion. Hence, this kind of data requires low latency transmission rather than reliable transmission. In contrast, text should be transmitted without loss using packet retransmission. Taking these characteristics into consideration, our approach is providing two data paths to the remote clients with different characteristics: a low latency path without retransmission and a reliable path with retransmission. The second problem of delayed reaction to user input is caused by the transmission delay from an interaction client to the local display client through the web server. Our approach to this problem is providing a relay server placed in or close to each location in order to create a shorter path to the local display client. The local server terminates WebSocket connections and distributes data from a local interaction client both to the local clients and to the remote clients.

B. Audio Conferencing Tolerant to Packet Transmission Instability

Disconnection of teleconferencing due to the packet transmission instability degrades the collaboration quality significantly, especially when the users are communicating only with audio, which is a common situation in mobile conferences. In that case, disconnection might not be recognized by users because there are no other cues for it. Then, the speaker is not sure if the partners are really hearing the voices. Our approach is providing feedback with a visual indicator so that speakers can be aware of whether the listeners are really receiving their voices.

C. Effective Utilization of Limited Bandwidth

Our approach to fully utilize the limited network bandwidth is changing the network band allocation ratio among desktop sharing videos, teleconferencing videos, audio, text and other data dynamically depending on the priority at the moment. Audio is always important and enough bandwidth should be allocated. On the other hand, we consider that the user is not focusing on all videos at the same time and it is possible to reduce bitrate allocation of an unfocused video stream without deteriorating the user experience. For example, in the remote collaborative work using CAD, the

users sometimes edit the design and sometimes discuss it. During the editing process, more bitrate should be allocated for the desktop sharing video while the teleconferencing video should use more bitrate during the discussion. Thus, it is quite effective to rapidly control the network band allocation ratio between the videos of desktop sharing and teleconferencing.

4. Platform Architecture Design

To realize the concept of Sect.3, we designed REMOCOP system structure and implemented it. Three types of servers on the network, as depicted in Fig.2, are used. One is the centralized server that hosts a web application. The second type is the timecode server to integrate a web application and a native A/V transmission module in conjunction with an overlay manager in a display client (see Sect.4.1). The third type is the relay server to support data exchange among clients (see Sect.4.2). In Sect.4.3, feedback for audio delivery is presented. A newly-developed JPEG2000 A/V transmission module for high quality and effective utilization of limited bandwidth is presented in Sect.4.4.

4.1 Integrated Workspace

We propose a workspace integration method in the display client to satisfy A/V quality requirements in demanding use cases [12]. The main concept is creating a visually integrated workspace by overlaying the web application on the videos of desktop sharing and teleconferencing as illustrated in Fig.3. The web application provides the type of basic level service such as pointing and A/V transmission is delegated to the native module. The web application is running on a web browser with transparent background so that the videos behind it can be seen [26].

A. Spatial Alignment

The position of the native video is adjusted by the overlay manager, which communicates with the centralized server. Figure 4 illustrates the procedure. When a user requests desktop sharing or teleconferencing in an interaction client, the video position is notified to all clients by the

Fig. 2 A schematic diagram of system components and UDP data exchange paths.

centralized server. When an interaction client receives the position notification, it creates a WebRTC video object. On the other hand, when a display client receives the notification message, it makes the region transparent. The video position is also notified to the overlay manager and it adjusts the native video position. In this way, the web application and the native video are visually integrated.

In our prototype implementation, JavaScript Object Notation (JSON) is used as the message format. An example message for position notification is {"position": [X, Y, W, H]}. Note that X is horizontal offset, Y is vertical offset, W is width and H is height. To adjust the window position, overlay managers in display clients call SetWindowPos(), a Windows API, with the parameters provided by position notification messages. To handle large group collaboration, we use Kurento, a WebRTC media server capable of mixing, transcoding and copying WebRTC A/V streams [27]. WebRTC A/V streams are distributed to the clients through a Kurento server to save computation and traffic load in clients.

Furthermore, the users can control the work tool on the workstation desktop using the interaction clients. A user input onto the desktop sharing video in the interaction client is transmitted to a control manager in the workstation. Then, the control manager triggers the equivalent input event onto the work tool using the OS APIs. Thus, the work tool doesn't have to be designed for REMOCOP, which means that this method can deploy off-the-shelf and currently-used software.

B. Temporal Alignment

The timecode server provides timing information to temporally align the web application with teleconferencing

transmitted by the native A/V transmission module. Without any synchronization mechanism, they will become disjointed since each layer experiences different transmission delay factors, for example A/V coding and jitter. This means multiple events that happened at the same time in one location, such as someone explaining (audio) while pointing (pointer motion), will be presented separately in another location, which degrades understanding.

Here we propose a method that uses timecode servers to synchronize the web application with teleconferencing even if there is significant transmission delay jitter in a wide area network (WAN). One simple idea for synchronization is to insert a constant delay; unfortunately, this cannot deal with jitter. Our key idea is to extract a jitter-robust timecode from the video packet stream and use it to synchronize the web application. Timecode servers are placed in the video stream path as depicted in Fig. 5, assuming that the jitter in a location is negligible compared to video frame intervals.

At the sender's location, the timecode server extracts the timecode from the video stream and compensates the encoding delay so that the timecode corresponds to the frame that the camera is shooting. If no timecode is included in the video, the server can embed one in the video packet. The timecode is delivered to the interaction client in the same location through the relay server. Then, the interaction client embeds the timecode in media packet to be synchronized with teleconferencing, such as pointer motion.

In the same manner, the timecode server at the receiver's location extracts the timecode and compensates the decoding delay so that the timecode corresponds to the frame on the display at that moment. A compensated timecode is sent to the clients in the location. The timecode embedded in the media packet and the teleconferencing timecode from the same location enables the web application to synchronize itself with teleconferencing. Since the web application adjusts itself to teleconferencing, the native A/V transmission module doesn't have to be designed for REMOCOP. Thus, it is easy to deploy off-the-shelf A/V module in accordance with the requirements of the use case.

Fig. 3 Concept of overlay workspace integration.

Fig. 4 A schematic diagram of workspace integration.

Fig. 5 A schematic diagram of temporal alignment using timecode servers and data exchange using relay servers.

4.2 Data Exchange Using Relay Servers

Considering the data characteristics mentioned in Sect. 3.2, we propose a data exchange method for web-based collaboration systems using relay servers (Fig. 5) [12]. The main idea is converting WebSocket using UDP to send data over WAN and returning immediate reactions to user input using relay servers placed in each location.

The web application running in a browser is connected both to a relay server and to a common centralized server via WebSocket. The relay server is used to exchange data that should be transmitted without delay (Low latency path) and the centralized server is used to exchange data that should be transmitted without loss (Reliable path). In the former path, the relay server converts the protocol from WebSocket into UDP to exchange the data with the other relay servers over WAN. Although UDP transmission may drop some packets, deployment of forward error correction will improve the reliability. The developer using REMOCOP will select either TCP (WebSocket) or UDP path according to the data type because the requirement for data transmission is constant. In our prototype implementation, the low latency (UDP) path is used for the data requiring low latency transmission rather than reliable transmission, such as pointer motion. On the other hand, the reliable (TCP) path is used for the data requiring reliable transmission rather than low latency transmission, such as text chat.

The connection architecture among relay servers is full mesh and the connection architecture between a relay server and each client in the location is star topology. This connection architecture saves the computation and traffic load. If all clients are connected in full mesh, clients have to send the same data to all other clients, which would increase the computation and traffic load as the number of participants increases. Thus, peer-to-peer connection is not suitable for large group collaboration. This is the other reason we haven't used RTCDatChannel in data exchange.

The relay servers offer the other benefit of

Fig. 6 Visual feedback on an interaction client.

immediate reaction [13]. When the web server is far from a user, the user can feel delay between the input on the interaction client and its reaction on the display client. With relay servers, user input is delivered to the display client through a local relay server, which enables the display client to immediately react to the user input.

4.3 Feedback of Audio Delivery

Clients at both ends work in conjunction to provide feedback of audio delivery [14]. The listener client analyzes the average received audio level. If the level exceeds a threshold, the listener client sends a feedback message to the speaker client. When the speaker client receives the feedback, a visual effect is generated to indicate the speaker voices were received.

In accordance with this idea, we implemented audio conferencing using WebRTC in interaction clients as shown in Fig. 6. When a user speaks, a circle appears under the avatar, representing that the remote partner is receiving the audio. If the circle doesn't appear, it means the remote partner is not receiving the audio. The size of the circle is proportional to the received audio level. The feedback is sent 10 times per second. A desktop sharing is also implemented as shown at the top of Fig. 6.

4.4 JPEG2000 A/V Transmission Module with Instant Frame Rate Control

As mentioned in Sect. 3.2, we consider it is quite effective to rapidly control the bitrate of video streams. However, inter-frame coding such as H.264 or H.265 is not suitable for this purpose because it's difficult to control the bitrate instantly. In addition, the coding latency is usually as large as 10 frames or more, which is undesirable in teleconferencing or remote control of work tools. Intra-frame coding is more suitable in terms of low coding latency and instant bitrate control by controlling its frame rate.

There are several intra-frame codecs such as H.264 (intra mode), H.265 (intra mode) and JPEG2000. For high resolution image coding, JPEG2000 shows the best peak signal-to-noise ratio (PSNR) at low bitrates, although HEVC shows the best PSNR at higher bitrates [28]. As the image resolution increases, the boundary shifts to a higher

area. Therefore, JPEG2000 seems to be the most suitable intra-frame coding for remote collaboration with high quality video.

However, there is no existing JPEG2000 A/V transmission system that can control its frame rate instantly. In addition, no existing JPEG2000 systems are suitable for the overlay workspace integration introduced in Sect. 4.1. Existing JPEG2000 A/V transmission systems that can handle high definition videos are hardware-based and thus the video interface is a limited physical I/O, such as a serial digital interface (SDI) [29]. Using a hardware-based system in video overlay requires an expensive video composite system, which substantially raises the cost. Therefore, we developed a software-based JPEG2000 A/V transmission module with desktop input and output, which requires no additional equipment other than a generic PC for overlay workspace integration [15].

Table 1 shows the specifications of the JPEG2000 A/V transmission module. Since JPEG2000 coding can be parallelized and no reference data outside is required, it can be efficiently processed by GPUs. Currently, generic PCs are generally equipped with GPUs, which makes parallel computing cost lower than that for CPUs. Therefore, we adopted a Comprimato SDK [30] that off-loads coding on GPUs as a JPEG2000 codec engine.

4.5 Platform Architecture

Figure 7 shows the platform architecture of REMOCOP. The platform layer provides APIs and a JPEG2000 A/V transmission module to construct REMOCOP clients.

Table 1 System specifications.

Video input	3G/HD-SDI (Max 4 links) Desktop (Windows only)
Video output	3G/HD-SDI (Max 4 links) , Desktop
Framerate	60p, 59.94p, 60i, 59.94i, 30p, 29.97p, 30psf, 29.97p, 24p, 23.97p, 24psf, 23.97psf
Resolution	(Desktop I/O) Any (SDI) 4096 x 2160, 3840 x 2160, 2048 x 1080, 1920 x 1080
Wavelet filter	9/7
Color space	RGB 4:4:4 / 8bit, 10bit YCbCr 4:2:2 / 8bit, 10bit
Operating system	Linux 64bit Windows7 64bit

Fig. 7 Platform architecture of REMOCOP.

Overlay workspace integration, data exchange method with two paths and feedback of audio delivery are provided as overlay API, data exchange API and awareness API, respectively. Thereby, REMOCOP provides a framework to easily implement a web-based remote collaboration system that shares work tools with a high quality A/V transmission module. It is also easy to equip functions to handle network constraints. In this framework, the work tool to be shared doesn't have to be designed for REMOCOP, which means that off-the-shelf and currently-used software can be used as it is.

5. Evaluation

We implemented a prototype collaboration system using REMOCOP to confirm the feasibility of workspace integration concept. Subsequently, we conducted tests to evaluate our platform in terms of tolerance to long transmission delay and to packet transmission instability. Furthermore, the performance of our JPEG2000 A/V transmission module was evaluated in terms of the computation load and the response time to bitrate control.

5.1 Feasibility of Overlay Workspace Integration

We implemented a prototype collaboration system using REMOCOP to prove the feasibility of overlay workspace integration. Figure 8 shows a picture of the prototype. Our prototype system provides the users with a common operating environment to share pointing, images and desktop video. Users can move pointers, draw annotations on images and control the work tool running in a remote workstation through the interaction client.

The software-based JPEG2000 A/V transmission module was used for high quality desktop sharing and teleconferencing. The web application was synchronized with teleconferencing and the synchronization error was imperceptible. Desktop sharing of a workstation including CAD software achieved high quality enough to check the design details. Remote control of CAD software was comfortable because the control pointer immediately reacts to user input thanks to the relay servers.

Fig. 8 A prototype of REMOCOP-based collaboration system.

Fig. 9 A schematic diagram of test system for operation accuracy.

5.2 Tolerance to Long Transmission Delay

We conducted an evaluation to see how the immediate reactions with relay servers benefit the user operation using a test system shown in Fig. 9. In the test system, which was composed of interaction clients and display clients, the users can move images on the display clients by hand gestures captured by a Leap [31] controlled by an interaction client. The centralized server manages the image object position, processes the request for position change in order of arrival and notifies all the clients of the result to ensure consistency. If the user is far from the centralized server, the response to the request for position change is largely delayed. Therefore we used provisional feedback to display the expected object position immediately so that the users can manipulate the object comfortably. As illustrated in Fig. 9, while an object is being dragged, notifications of expected position (green arrow) are sent to a local relay server and they are distributed to the display client. Then, the display client generates provisional feedback from the notifications immediately. When the user stops dragging the object, the request for position change (red arrow) is sent to the centralized server. The centralized server processes the request and distributes the notification of position change to all clients (blue arrow). Only after the display client receives the notification of position change, the object position is updated. If there is no response to the request from the centralized server, the provisional feedback will disappear and the object position will be unchanged.

The four participants were asked to drag a clock image onto a target (the red X) 10 times in each trial. The green square is the expected position of the image. The target position was changed every time after dragging. Round-trip time of 240 ms was inserted by a network emulator to simulate long transmission delay similar to that between Tokyo and London, which gives a strong perception of latency in the actual reply from the centralized server. Two trials were performed under different conditions: “without relay server” and “with relay server.”

Figure 10 shows the averaged remaining distance between the image center and the target center. The t-test

Fig. 10 Remaining distance to the target. (*: $p < 0.05$)

result indicates that the p value is 0.03, which means there is a significant difference between them ($p < 0.05$). Thus, it can be said that the relay servers improved user operation accuracy under long transmission delay.

5.3 Tolerance to Packet Transmission Instability

We conducted an evaluation of tolerance to packet transmission instability in audio conferencing. We asked the eight participants (seven males and one female) to perform two tasks in pairs. Two trials were performed for each task under different conditions: “without feedback” and “with feedback.” After each trial they were asked to answer the statement “I felt like I could recognize whether my voice was being received by my partner or not.” Answers were made using a five-point scale where 1=strongly disagree, 2=disagree, 3=neutral, 4=agree and 5=strongly agree.

A. Dictation Task

The first task was dictation. One participant (the speaker) was asked to read out a short text with about 70 words while the partner was asked to transcribe the text. The test environment was set in two separate rooms so participants could talk to each other via the audio conferencing. To simulate real world unstable networks, audio disconnections of two or four seconds were inserted every 30 seconds. Figure 11 (a) shows the average time spent. The t-test results indicate the p value of 0.042, confirming the significant difference between the averages ($p < 0.05$). Figure 11 (b) exhibits the average scores obtained for the

Fig. 11 Results of dictation task. (*: $p < 0.05$, **: $p < 0.01$)

Fig. 12 Tangram puzzles.

Fig. 13 Results of puzzle task. (**: $p < 0.01$)

aforementioned statement. The t-test results indicate the p value is 0.003 ($p < 0.01$). Finally, Fig. 11 (c) represents the average number of errors in the transcribed text; the p value is 0.019 ($p < 0.05$). All of these result showed significant improvement by proposed method.

B. Puzzle Task

The second task was solving a tangram puzzle composed of seven pieces. From these pieces, a lot of different patterns, such as animals, people, etc., can be assembled. The speaker was given a target shape and was asked to orally describe the shape to the listener. The same test environment as that in the dictation task was used. Figure 12 (a) represents the initial shape and Fig. 12 (b) shows one of the target shapes. In accordance with the instructions from the speaker, the listener moved the pieces on PowerPoint shared with the speaker as shown in Fig. 6. Thus, the speaker could see the puzzle status. Figure 13 (a) shows the average times spent to complete the puzzle. The t-test indicates that the p value is 0.179, which is not a significant difference ($p > 0.05$). Figure 13 (b) shows the scores obtained for the aforementioned statement. The p values were 0.005, which means that there is a quite significant improvement ($p < 0.01$).

5.4 Instant Bitrate Control

We measured the computation load of our software-based JPEG2000 A/V transmission module using a PC whose specifications are shown in Table 2. Tables 3 and 4 respectively show the encoding and decoding loads. Note that the

Table 2 Test machine specifications.

CPU	Intel Xeon E5-2620 v3 (2 processors, 12 cores, 2.4 GHz)
Memory	32 GB
GPU	GTX980 (2048 CUDA cores, 1126 MHz) x 2
OS	Windows 7, 64bits

Table 3 Encoding computation load.

Encode	Idle	30p	60p
CPU	1%	5%	11%
GPU1	0%	53%	95%
GPU2	0%	7%	16%

Table 4 Decoding computation load.

Decode	Idle	30p	60p
CPU	1%	5%	10%
GPU1	0%	53%	98%
GPU2	0%	22%	25%

encoder input was desktop capture with 1920 x 1080 pixels of RGB 4 : 4 : 4 (8 bit) and was compressed to 1/20th. The CPU load is the average of 12 cores. GPU1 was used in coding processes while GPU2 was used in desktop output.

The CPU load increments were about 5% in 30p and 10% in 60p for both encoding and decoding, which means that less powerful CPUs are sufficient. On the other hand, the GPU1 load increments were about 50% in 30p and almost 100% in 60p for both encoding and decoding. These results tend to indicate that a high-performance CPU is not necessary although a high-performance GPU is required to

install this system.

We measured the video bitrate to find out the response time to frame rate control. At first, we started encoding with 20 fps. After that we sent a command to change the frame rate to 10 fps. Figure 14 illustrates the resulting bitrate transition. We confirmed that the bitrate converged to the half in 160 ms after the command was received. In addition, no visual errors were found at that time.

5.5 Discussion

The prototype implementation of REMOCOP-based collaboration system that achieved higher quality than WebRTC proved the concept of overlay workspace integration. In this prototype, A/V transmission and overlay workspace integration were processed by a generic PC, which shows the potential to reduce the system construction cost. The evaluation results in Sect. 5.2 show that the proposed method of data exchange using relay servers improved long distance communication, which was a tough situation for conventional web-based systems. From the results given in Sect. 5.3, it can be said that our feedback approach made the collaboration under unstable network more comfortable because the subjective rating was improved in both tasks. In the dictation task, our approach improved the efficiency in terms of time spent and errors although significant

Fig. 14 Bitrate transition resulting from frame rate control.

improvement was not obtained for the tangram puzzle scores. In the interview after the tests, some participants said that they were paying more attention to the desktop sharing of PowerPoint to see how the listener was moving the pieces rather than feedback of audio delivery. As a subject for future work, we should consider how to give the feedback to the speakers when they are focusing on tasks. The evaluation in Sect. 5.4 shows that our software-based JPEG2000 A/V transmission module is suitable for instant bitrate allocation and has the potential to make the most of the limited bandwidth. As a subject for future work, we should consider how to determine the optimal bitrate allocation between the desktop sharing video and teleconferencing video.

Table 5 is a comparison with other remote collaboration platforms introduced in Sect. 2.1, ConferenceXP, Vclick and WeColLab. Since ConferenceXP uses Windows native functions, it allows peer-to-peer connection and UDP transmission. However, the users are required to install a Windows application before the collaboration starts. Vclick and WeColLab don't require the installation of dedicated software in a user's device because they are implemented only with HTML5 and WebRTC. Additionally, WeColLab allows remote control of work tools although it requires redirector servers. The major advantages of REMOCOP compared to these platforms are high quality A/V support and tolerance to network constraints while the disadvantages are dependency on special A/V module in display clients and additional server equipment. Overall, REMOCOP offers the best experience for collaboration participants even under severe network constraints.

6. Conclusion

We proposed a web-based and quality-oriented remote collaboration platform tolerant to network constraints such as long transmission delay, packet transmission instability and limited bandwidth. To achieve higher A/V quality than conventional web-based systems can achieve, while

Table 5 Comparison of remote collaboration platforms. Red cells are advantages and white cells are disadvantages.

	ConferenceXP [16]	Vclick [19]	WeColLab [20] [21]	REMOCOP
Supports both high quality and low quality A/V				✓
No installation of dedicated software in user's device		✓ HTML5 and WebRTC	✓ HTML5 and WebRTC	(Interaction client) ✓ (Display client) Special module for high quality A/V
Remote control of work tools	✓		✓	✓
Tolerant to long transmission delay	✓ UDP transmission			✓
Tolerant to network instability				✓
Tolerant to bandwidth limitation		✓ Video bitrate control	✓ Video bitrate control	✓
No need for additional servers other than a web server	✓ Peer-to-peer connection	✓	Redirector servers	Relay servers and timecode servers

maintaining the advantages of web-based systems, we proposed overlay workspace integration to combine a web application and a native A/V transmission module. To handle long transmission delay we proposed a data exchange method using relay servers. To deal with packet transmission instability, we proposed a visual feedback method so that speakers can be aware of whether the listeners are really receiving their voices. To fully utilize the limited bandwidth by controlling video bitrate allocation, we developed a JPEG2000 A/V transmission module with dynamic frame rate control. We implemented a platform that incorporates the proposed functions. A prototype collaboration system using REMOCOP proved the feasibility of overlay workspace integration. Evaluation tests on tolerance to long transmission delay and packet transmission instability proved the improvement. Furthermore, the evaluation of the JPEG2000 A/V transmission module showed its potential for effectively utilizing limited bandwidth. Collectively, these results showed that our platform has the ability to construct unprecedented collaboration systems.

Acknowledgments

We are grateful to Hosoya Marcos Teruaki from Sao Paulo University for assistance in implementing and evaluating prototype systems during the internship in NTT Network Innovation Laboratories.

References

- [1] ITU, "Remote collaboration tools," ITU-T Technology Watch Report #5, March, 2008.
- [2] P. Bjørn, M. Esbensen, R. Jensen, and S. Matthiesen, "Does distance still matter? Revisiting the CSCW fundamentals on distributed collaboration," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol.21, no.5, pp.27:1–27:26, 2014.
- [3] Redmine, <http://www.redmine.org/>, accessed Feb. 28, 2017.
- [4] Shotgun, <https://www.shotgunsoftware.com/>, accessed Feb. 28, 2017.
- [5] I. Rae, G. Venolia, J.C. Tang, and D. Molnar, "A framework for understanding and designing telepresence," *Proc. 18th ACM conference on computer supported cooperative work & social computing, Vancouver, BC, Canada*, pp.1552–1566, 2015.
- [6] H. Nakanishi, K. Tanaka, and Y. Wada, "Remote handshaking: touch enhances video-mediated social telepresence," *Proc. 32nd annual ACM conference on Human factors in computing systems, Toronto, Ontario, Canada*, pp.2143–2152, 2014.
- [7] L. Yao, A. DeVincenzi, A. Pereira, and H. Ishii, "Focalspace: multi-modal activity tracking, synthetic blur and adaptive presentation for video conferencing," *Proc. 1st symposium on Spatial user interaction, Los Angeles, California, USA*, pp.73–76, 2013.
- [8] Cisco TelePresence, <http://www.cisco.com/>, accessed Aug. 1, 2017.
- [9] K. O'hara, J. Kenton, and J. Paay, "Blended interaction spaces for distributed team collaboration," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol.18, no.1, pp.3:1–3:28, 2011.
- [10] S. Kim, T. Nakachi, T. Fujii, S. Emura, and Y. Haneda, "Highly realistic collaboration system by using 4K multi JPEG2000 codecs and 6-channel acoustic echo canceller," *International Symposium on Intelligent Signal Processing and Communications Systems*, pp.452–457, Nov. 2012.
- [11] Google Docs, https://www.google.com/intl/en_us/docs/about/, accessed Feb. 28, 2017.
- [12] Y. Mochida, D. Shirai, and T. Fujii, "Novel Web-based Remote Collaboration System Architecture for Wider Use Cases," *Proc. 19th International Conference on Supporting Group Work*, pp.437–440, Nov. 2016.
- [13] M.T. Hosoya, Y. Mochida, D. Shirai, and T. Fujii, "Reduction on Latency Perception using REMOCOP in Web-based Remote Collaboration Systems and Gesture Application to Gesture UI," *IEICE Technical Report*, vol.115, no.496, pp.165–170, March 2016.
- [14] M.T. Hosoya, Y. Mochida, D. Shirai, and T. Fujii, "Visual Feedback in Audio Communication under Unstable Network Situations," *IEICE Technical Report*, vol.116, no.204, pp.53–58, Sept. 2016.
- [15] Y. Mochida, D. Shirai, and T. Fujii, "Software-based Low-latency Video Transmission System with Immediate Bandwidth Control for Remote Collaboration," *VRSJ Technical Report*, vol.22, no.TTS1-4, March 2017.
- [16] J. Beavers, T. Chou, R. Hinrichs, C. Moffatt, and M. Pahud, "The learning experience project: enabling collaborative learning with ConferenceXP," *Microsoft Research TechReport*, April 2004.
- [17] A. Johnston, J. Yoakum, and K. Singh, "Taking on webRTC in an enterprise," *IEEE Commun. Mag.*, vol.51, no.4, pp.48–54, April 2013.
- [18] W. Elleuch, "Models for multimedia conference between browsers based on WebRTC," *IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp.279–284, Oct. 2013.
- [19] K. Singh and J. Yoakum, "Vclick: Endpoint Driven Enterprise WebRTC," *2015 IEEE International Symposium on Multimedia (ISM)*, Miami, FL, pp.441–444, 2015.
- [20] M.A. Bochicchio and A. Longo, "A Multi-purpose Architecture for Collaborative Web Labs," *2009 Ninth IEEE International Conference on Advanced Learning Technologies, Riga*, pp.70–74, 2009.
- [21] M.A. Bochicchio, A. Longo, M. Zappatore, and D. Tarantino, "A collaborative multi-videoconferencing platform for online optical microscopy," *Proc. 2015 12th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, Bangkok, pp.143–151, 2015.
- [22] T. Marrinan, J. Aurisano, A. Nishimoto, K. Bharadwaj, V. Mateevitsi, L. Renambot, L. Long, A. Johnson, and J. Leigh, "SAGE2: A new approach for data intensive collaboration using Scalable Resolution Shared Displays," *10th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pp.177–186, Oct. 2014.
- [23] Y. Xu, C. Yu, J. Li, and Y. Liu, "Video Telephony for End-Consumers: Measurement Study of Google+, iChat, and Skype," *IEEE/ACM Trans. Netw.*, vol.22, no.3, pp.826–839, June 2014.
- [24] G. Carullo, M. Tambasco, M.D. Mauro, and M. Longo, "A performance evaluation of WebRTC over LTE," *12th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, pp.1–6, Jan. 2016.
- [25] MeetingPlaza, <https://www.meetingplaza.com/>, accessed Feb. 28, 2017.
- [26] NW.js, <https://nwjs.io/>, accessed Feb. 28, 2017.
- [27] L.L. Fernández, M.P. Díaz, R.B. Mejías, F.J. López, and J.A. Santos, "Kurento: a media server technology for convergent WWW/mobile real-time multimedia communications supporting WebRTC," *2013 IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, Madrid, pp.1–6, 2013.
- [28] Q. Cai, L. Song, G. Li, and N. Ling, "Lossy and lossless intra coding performance evaluation: HEVC, H. 264/AVC, JPEG 2000 and JPEG LS," *Proc. 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, pp.1–9, Dec. 2012.
- [29] D. Shirai, M. Kitamura, and T. Fujii, "4K Super High Definition Video Streaming System Using JPEG 2000 Codec," *IEICE Technical Report*, vol.107, no.244, pp.43–48, Sept. 2007.
- [30] Comprimato SDK, <http://comprimato.com/>, accessed Feb. 28, 2017.
- [31] Leap Motion, <https://www.leapmotion.com/>, accessed Feb. 28, 2017.

Yasuhiro Mochida received his B.E. and M.E. degrees from The University of Tokyo in 2009 and 2011. Since he joined NTT Laboratories in 2011, he has been engaged in research on remote collaboration including system architecture, network protocol, interface design and web technologies. He is a member of IEICE and VRSJ.

Daisuke Shirai received the B. Eng. degree in Electronic Engineering, the M. Eng. degree in Computer Science, and the Ph.D. in Media Design from Keio University, Japan, in 1999, 2001 and 2014. He joined Network Innovation Laboratories in 2001 and since then has been researching super-high-definition imaging systems and their transmission technologies. His current research interest is in Network Supported Collaborative Work, which centers on the unification of super-high-definition, robust video communication technologies and multimodal interaction. Dr. Shirai is now in NTT COMWARE. He is a member of IEICE.

Tatsuya Fujii received his B.S., M.S. and Ph.D. degrees in Electrical Engineering from the University of Tokyo in 1986, 1988, and 1991. He joined NTT upon receiving his doctorate. Since then he has been researching super-high-definition image communication networks. In 1996, he was a visiting researcher at Washington University in St. Louis, serving as a director of a digital cinema developing project. He is currently studying real-time 4K video communication system architecture, as well as reliable content distribution systems at NTT Network Innovation Laboratories. He is a member of IEICE, ITE of Japan and IEEE.