International Journal of Multimedia and Ubiquitous Engineering
Vol.9, No.3 (2014), pp.11-20
http://dx.doi.org/10.14257/ijmue.2014.9.3.02

A Study on the Hybrid Web-based Real-Time Video Communication System

Tae-Hak Ban, Tae-Beom Cho and Hoe-Kyung Jung*

Department of Computer Engineering, PaiChai University, Korea
banth@pcu.ac.kr, tbcho@naver.com, hkjung@pcu.ac.kr

Abstract

Video communication technology of real-time using a Web browser has emerged. In this paper, we studied the real-time image communication technology of hybrid Web-based. In addition, we have implemented a real-time video communication systems of hybrid Web[1]-based design. In this system, it is possible to use the HTTP / Web Socket and RTC API via a Web server, in order to perform real-time video communication in all Web browsers. In all environments that are connected to the network, it is available at all terminals that you can use a Web browser and real-time video communication. In the future, this can be applied to a Settop-Box or Smart TV.

Keywords: *WebRTC, Hybrid Web, HTML5, Real-time Video Communication*

1. Introduction

Standard technology for real-time communications such as voice communication and video conferencing in a Web browser has been standardized through the RTCWeb (Real-Time Communication in WEB-browsers) working group of the IETF (Internet Engineering Task Force). In the meantime, the real-time multimedia communications in a Web browser, which depends on the application by each developer, have been carried out by a non-standard way. The efforts to standardize this are held on the IETF in collaboration with the W3C (World Wide Web Consortium). Accordingly, protocols and the API (Application Program Interface) requirements of the IETF and a standard API development of the W3C are on their way.

Recently, on Google, WebRTC [2] technology that provides user-to-user video communication services using only a Web browser has progressed, which does not use specific visual communication servers, applications and software. Unlike current commonly used video communication services, Hybrid web utilizing a Web browser has the advantage of using the existing Web, anywhere and anytime, if an Internet environment is established and visual communication service is available. A comparison of existing video communication program and WebRTC is shown in Table 1 below. In this paper, hybrid web-based video communication system was studied and implemented using HTML5 and JavaScript on the Web.

---

* Corresponding Author

ISSN: 1975-0080 IJMUE

Table 1. Comparison of video communications technology

	Existing Video Communication Program	WebRTC
Development Environment	Separate development program (tools)	HTML5 (JAVA Script)
Video Codec	H.264, FFMPEG, Xvid, etc	HTML5 built separately
Voice Codec	MP3, AAC, Ogg, etc	HTML5 built separately
Cost of Building	May be one million won ten million won~	Web access to an environment in which
Flexibility	System can be linked in the same	HTML5 any supported Web browser

A video communication system using existing web utilized html and php structures needed more storage capacity due to the use of the software such as a video or flash, active-x and plug-in's. A lot of side effects were found in terms of efficiency. But HTML5 [3, 4] that simply uses elements on Stream by specific tags has been verified excellent in several ways.

2. Paper Preparation

2.1. SRTP (Secure Real-Time Transport Protocol)

SRTP [5, 6] standard is RTP packet encryption technology to transmit voice and video traffic in the internet phone. It can pass encryption / decryption keys of the media stream for the session in the SDP (Session Description Protocol) message at both ends, where RTP packet encryption and decryption is possible. In order to encrypt or decrypt real-time data, AES (Advanced Encryption Standard) [7] algorithm is used in the SRTP.

2.2. ICE (Interactive Communication Environment)

ICE [8, 9] is a NAT (Network Address Translation) Traversal protocol for establishing UDP (User Datagram Protocol)-based multimedia stream sessions. The host at both ends using ICE protocol will establish the interactive session. ICE overcomes a non-standard NAT environment and provides an independent general-purpose NAT traversal solution in various network environments. Typically, the two ends in ICE want connection. These are connected indirectly through signaling channel of moderator generally called rendezvous server. The two ends exchange information needed for P2P connection to the session throughout rendezvous server. ICE provides enough information about the network environment to the host, and also helps find various routing routes so it can communicate with peer.

3. Web-based real-time video communication system design

In a Web-based, real-time video communications system design is divided into three main parts: video API, audio API, and P2P API. A separate codec and transport protocol were used.

Figure 1 Web-based, real-time video communication system design diagrams are divided into four major steps. First, the user requests the connection, and the user receives connection requests. Second, WebAPI uses a Web browser for Real-time video communication. Third, WebRTC API connects users in a web-based real-time video communications system. Fourth, APIs analyzes voice and video systems in a real-time video communications and transports Stream.

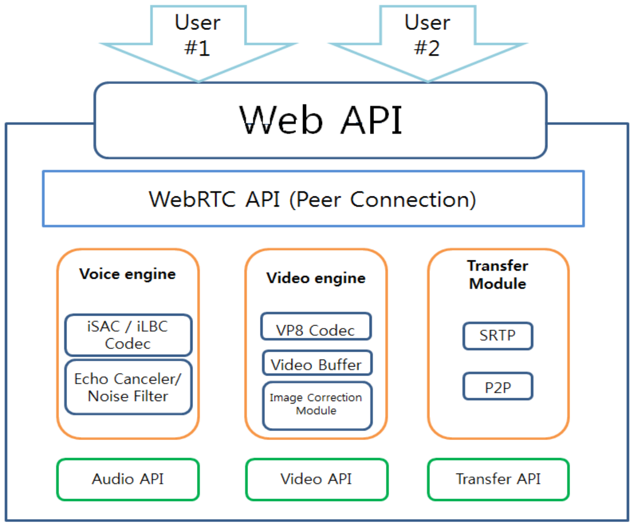

Figure 1. Web-based real-time video communication system design diagrams

3.1. Web-based, real-time video communications system configuration

Web-based video communication system consists of 2 web browsers and a web server. First, the browser communicates with the Web server. In this section, the communication medium is RTC API made of JavaScript and HTML between the browser and the RTC API communicates with the Web server applying HTTP and Web sockets.

The part receiving the requests takes the same way. Real-time video communication operates and works in a web browser: each web server receives its signals through a network communication path, and browsers transport video and audio through media path. Multimedia data such as voice and video in real-time multimedia communications on a Web browser communicates through a direct connection. In comparison, the signal path for the session connection between the browsers is designed that it may be worked through the separate server same as the web server. The Overall picture is shown in Figure 2 below.

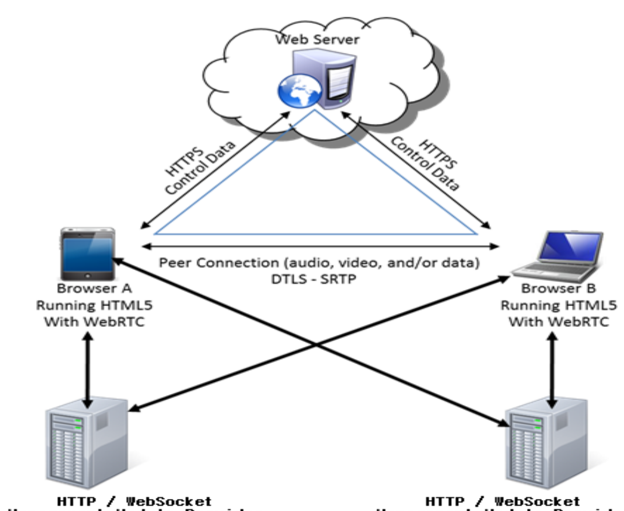

Figure 2. Web-based video communications systems operating configuration

3.2. Web-based video communication system design

In a Web-based video communication system, it was designed to distinguish the connection to the client and to the other. The following overall picture is shown in Figure 3.

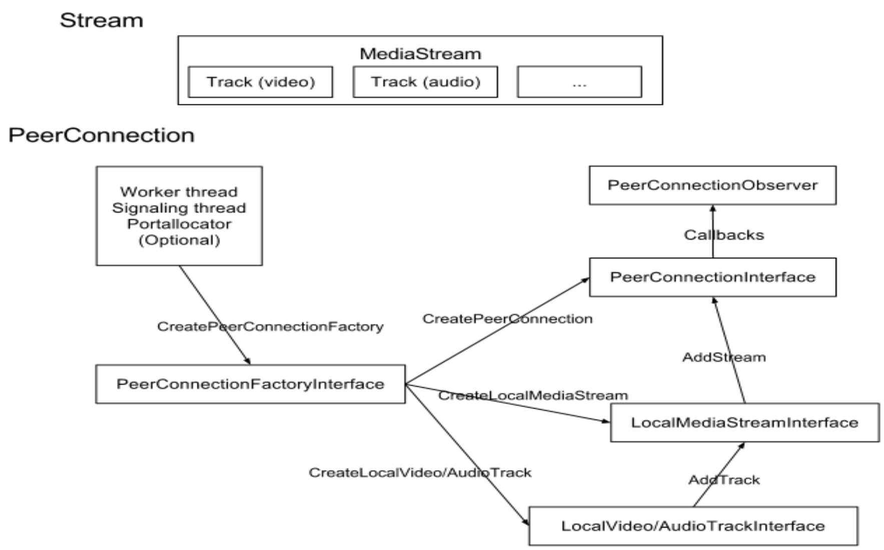

Figure 3. Web-based real-time video communications system sequence diagram

The part of Stream can be called the media stream. Media stream is divided into video track, audio track, and other device parts. Mainly used parts are video track and audio track. Peer connection section consists of 6 parts: Options, PeerConnectionFactoryInterface, Local Video/Audio Track Interface, Local Media Stream Interface, PeerConnectionInterface, and Peer Connection Observer. Option section is composed of the part connecting the communication between the thread and the client.

The part of PeerConnectionFactoryInterface is generated while proceeding to the step from options. And it can be regarded as a medium when the user requests a connection to the other. And 3 parts are created in the PeerConnectionFactoryInterface. First, the video and audio track interface is created, and media stream interface is created next. During this process, a track is created. Then, while generating a stream, PeerConnectionInterface is made. And finally it retransmits in PeerConnectionObserver.

3.3. Web-based video communication requesting system design

In Web-based video communication requesting system design consists of three parts: the user requesting the connection (PeerConnection), the user receiving a request (Remote Peer), and an application that connects two users (PeerConnectionObserver). Request’s connection, a task for connection from the application, is performed. It requests the other by notifying the results.

When the other receives a request, they are connected to each other through the visual communication. In this step, peer connection is created, and information on the connection and on tracks using video and audio for the tracks are generated. After this step, information on stream generated by users is retransmitted. The users send messages using an application, and the application requests a connection to a remote peer that receives the connection. When the remote peer sends a response message back to the peer connection, the application acts as the connecting medium and sends messages back to the user, executing the connection for user's communication. Thus, the real-time video communication continues to run. The overall system of activity diagram is shown in Figure 4.

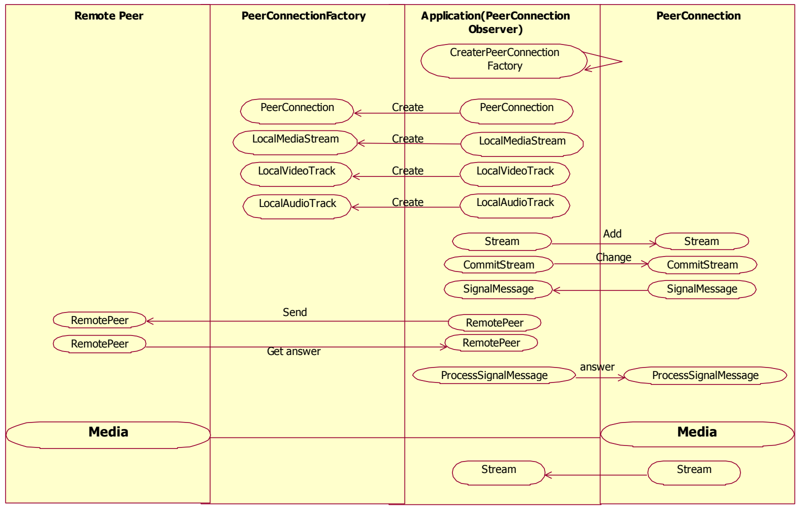

Figure 4. Connection request system activity diagram

Since HTML and JavaScript are used due to the nature of web-based real-time video communication systems, there are the video portion for videos on the web browser, source files for connecting each API and scons files for compiling. The classes for the server behavior are divided into the main and the library on the center of WebRTC. In the main, there are 7 parts. The first web, the second building, and the third are xmllite.

And this part includes the builder and parser for xml conversion and parsing, the contents of the content, elements related to the content and print for the output. The fourth is xmpp parts that are divided into the contents and modules, a handling part of the authentication protocol, xmpp clients and engines, and a task section to handle xmpp.

The fifth is a part conjunct with a talk part responsible for video communications, and this includes call part to start video communication, a channel for communication, codecs for video and audio, media stream, RTP use part, and video and audio clips. The sixth is a Base part which consists of elements for using the Web including a server and a client for HTTP, a handling part for message sending, and sockets and threads. The last is p2p part for transmission comprised of ice and turn that are one of the kinds of p2p, p2p transfer port and http portal connecting parts.

3.4. Web-based video communication response system design

Web-based video communication response system consists of three steps: the user requesting the connection (PeerConnection), the user receiving a request (Remote Peer), and the application connecting two users (PeerConnectionObserver). When accepting the request for the results of visual communication from a user, which requests the connection from a local user, the application analyzes the performed results on the response, notifies them to the user, and enables the mutual video communications. Once the remote peer (the user receiving the connection) accepts the request from the peer connection (the user requesting the connection), the other peer will pass the result to the response by an application. The application creates user connections to the requested results and passes them, thus creating the information about the user that requests the connection.

In this process, the generated information is to be delivered to the user as a message. Connection is established for the use requesting the connection, and information on the established connection and tracks using video/audio are generated. After this process, the information on Stream generated by a user is transmitted back. The user sends a message of communication information through an application and finally, this application sends back to the other connector receiving the connection request. At the end of this process, it enables two users to have the real-time video communication. Connection response system activity diagram is shown in Figure 5.

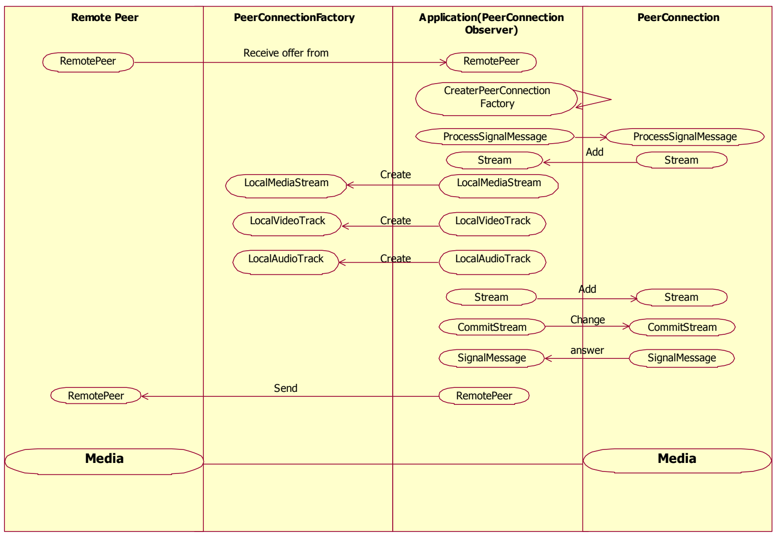

Figure 5. Connection response system sequence diagram

4. Web-based real-time video communication system implementation

4.1. Implementation of Web-based video communication request

Once one is allowed to answer whether the video device (web cam, camera) is connected to the user's computer on Chrome Web browser, the image of the user requesting a connection such as Figure 6 is displayed on a web browser.

Figure 6. Web-based real-time video communication request system running screen

HTML5 for the previous version of the HTML tags for the video and audio video, audio, tag link information, screen size, and color tables for HTML5, but the details need to specify the value of a specific tag and the picture in the configuration of the input.

4.2. Implementation of Web-based video communication response system

The implementation of requesting Web-based video communication system is the system configuration for the other users requested to connect visual communication system on a web browser. Once the user asked connection and put the other user's URL address of video communication system in the search box, it is interconnected to the other user's real-time video communication system by analyzing its IP address on the network automatically through HTTP. Figure 7 is interconnection screens between the user requesting the connection and the other user.

Figure 7. Web-based real-time video communication system interconnection screens

As shown in Figure 7, it enables users requesting a connection to have real-time video communication and to request it anytime to the other users if they are in an internet connected and a web browsing possible environment. There is no limit on time and place.

5. Conclusion

The conventional way of video communication service was a structure providing video communication between users through a specific server establishment, separate software and programs from each service provider. In order to configure video communication system, users had to purchase specific software or servers from video communication companies, use its service through the internal network in an already established environment, and set up specific programs for video communication.

Recently, however, WebRTC technology has progressed on its way of providing video communication service not using specific servers, applications and software but using only a web browser applied by HTML5. If the internet environment is built, real-time video communication using only a Web browser that does not require specific software, plug-in's, and Active-X is available at anytime and anywhere.

Accordingly, in this paper, we studied the real-time video communication technology utilizing the HTML5-based web browser. Based on this research, the design and implementation of real-time web-based video communication system were verified. In this system, real-time video communication between users on a web browser is possible by using RTC API and HTTP / WebSockets through web servers of different web browsers.

Further study includes a research that enables real-time video communication to all platforms of web environment on cloud computing, set-top boxes or Smart TVs, not on the current PC environment.

References

[1] Y. -S. Gwak, "Hybrid Web Solution System Based on the Internet Disk", Korea Institute of Information Technology, vol. 4, no. 6, **(2006)**, pp. 51-55.
[2] WebRTC, http://www.webrtc.org.
[3] S. -Y. Lee and K. -S. Park, "HTML5 and Smart Media Platforms", Journal of Korea Information and Communications Society, vol. 29, no. 10, **(2012)**, pp. 25-29.
[4] M. -W. Jo, J. -G. Park and G. -H. Kim, "Scene Composition Technology Based on HTML5 in Hybrid Broadcasting Environment", Korea National Engineering, vol. 18, no. 2, **(2013)**, pp. 237-248.
[5] Y. -C. Sin, G. -Y. Kim, M. -Y. Kim, J. -M. Kim, Y. -J. Won and J. -C. Ryu, "Performance Evaluation of VoIP Security Protocols", Institute of Information Security and Cryptology, vol. 18, no. 3, **(2008)**, pp. 109-120.
[6] J. -O. Park and K. -O. Oh, "A Study on Parallel AES Cipher Algorithm based on Multi Processor", Korea Computer and Information Science, vol. 17, no. 1, **(2012)**, pp. 171-181.
[7] H. -J. Seo and H. -W. Kim, "Network and Data Link Layer Security for DASH7", Journal of Information and Communication Convergence Engineering, vol. 10, no. 3, **(2012)**, pp. 248-252.
[8] SMOOTHWEB, http://www.smoothweb.com/ICE?sid=wlxwpxtf.
[9] G. Paravati, A. Sanna, F. Lamberti and L. Ciminiera, "An Adaptive Control System to Deliver Interactive Virtual Environment Content to Handheld Devices", Mobile Networks and Applications, vol. 16, no. 3, **(2010)**, pp. 385-393.

Authors

Tae-Hak Ban

He is received B.S. degrees in Department of computer engineering from Paichai University in 2011. And M.S. degrees in Department of Computer Engineering from Paichai University in 2013. And currently under Ph.D course in the Department of Computer Engineering at the Paichai University. His research interests are multimedia information processing, XML, next-generation Web, MPEG, Ubiquitous Computing.

Tae-Beom Cho

He is received M.S. degrees in Department of Information Communication from Paichai University in 1999. And Ph.D degrees in Department of Computer Engineering from Paichai University in 2010. His research interests are XML, mobile system, web security, multimedia, and database.

Hoe-Kyung Jung

He received the B.S degree in 1987 and Ph. D. degree in 1993 from the Department of Computer Engineering of Kwangwoon University, Korea. From 1994 to 2005, he worked for ETRI as a researcher. Since 1997, he has worked in the Department of Computer Engineering at Paichai University, where he now works as a professor. His current research interests include multimedia document architecture modeling, information processing, information retrieval, and databases.