eJAICT

ISSN 2355 1771

JOURNAL OF APPLIED INFORMATION, COMMUNICATION AND TECHNOLOGY

Published by:
Research Center of Technology and Innovation
Academic Research and Community Services
Swiss German University

Research of Web Real-Time Communication - the Unified Communication Platform using Node.js Signaling Server

Florensia Unggul Damayanti

Department of Information Technology, Swiss German University, Tangerang 15143, Indonesia

Article Information

Received: 5 August 2018
Accepted: 7 September 2018
Published: 3 October 2018
DOI: 10.33555/ejaict.v5i2.54

Corresponding Author:

Florensia Unggul Damayanti
Email: florensia.damayanti@gmail.com

ISSN 2355-1771

ABSTRACT

The way of people communicate has been changed from generation. With rapid innovation of technology and the improvement of computer technology, smartphone, tablet and etc., the traditional communication is confronted. However, unified communication born and Web Real-Time Communication as a part of the innovation of communication technology, enhance the way peoples share their thought, feeling and information, in real-time and different place. This research offers an educational foundation and applied research of WebRTC, to leverage unified communication, through WebSocket node.js as the signalling protocol. A system of web real-time communication developed at the end of research to give an understanding and view of the opportunity given by WebRTC technology, such as leverage online distance-learning with a real-time communication, healthcare video conferencing, online video conference, self-service customer care and corporate online meeting.

Keywords: *Real-Time Communication, Unified Communication, Node.js, Web-Socket, Peer-to-Peer Communication*

1. Introduction

Web Real-Time Communication (WebRTC) is an Application Programming Interface definition sign up by W3C (World Wide Web Consortium) which support peer-to-peer communications for audio/voice call, video call, chat and peer-to-peer file sharing without plugins (Altanai, 2014). This is an open-source project for browsed-based real-time communication released by Google. However, unified communication (UC) is about the integration of communication tools such as presence technology, IP phone, and synchronous communication which occurs real-time in a different place, to help people exchange ideas (Rouse, M). Generally, communication occurred on the desktop, known as CTI (Computer Telephony Integration). Therefore, communication-based on browser or mobiles application is challenging. WebRTC enables rich, high-quality RTC applications to leverage the unified communication. It because WebRTC support browser, mobile platform and IoT devices and permit them all share and exchange information through a set of protocols. This paper emphasis on research of the web real-time communication for unified communications as the next generation of browser communications. In the end of this research, the system developed and implemented to prove capabilities of WebRTC on unified communication such as desktop sharing, file sharing, video conferencing, voice call, presence and room chatting or messaging.

2. Related Research

Jian and Lin, 2015, conduct research and implement video conferencing use WebRTC API in an international conference on Computer Sciences and Automation Engineering. They implement the Video call using WebRTC technologies on mobiles devices. Sheetal, 2015, also do research and review of WebRTC as the next generation of web-based communication. Their research covers the architecture of WebRTC, various protocol and some impact of WebRTC. This research complement their research of WebRTC, which proves other abilities of WebRTC technologies, to power UC such as desktop sharing, messaging/chatting, file transferring using Data Channel and real-time video/audio calling using peer connection APIs.

3. Research Approach

3.1. The Unified Communication on WebRTC Architecture

This research uses the main object of WebRTC API component to deliver rich multimedia communications. The components are *GetUserMedia*, *PeerConnection* and *DataChannel*. They have an important role to construct real-time communication on the browsers. The Get User Media APIs, one of WebRTC components, responsible for capturing the input media use for peer-to-peer communication, for instance, microphone and camera on computers (Manson, R., 2013).. Whereas, the whole thing activities about media transportation handle by Peer Connection API (Jian, C., Lin Z., 2015).. These API take care of media sending and receiving, SDP negotiation, handling packet losses and other network issues, implementations of codec and NAT traversal (Sergiienko, A., 2014).. The API wraps it in a simple package, with the purpose of easy to adding media streams into a peer connection, negotiating capabilities and NAT traversal. In order to ease developing the unified communication web application, this research uses an open-source framework that binds WebRTC APIs. The framework is EasyRTC by Prologic. This framework putting up on top of WebRTC and evolving W3C/IETF standard for real-time communication of audio, video, and data in a straight line between web browsers. The core architecture is shown in fig.1.

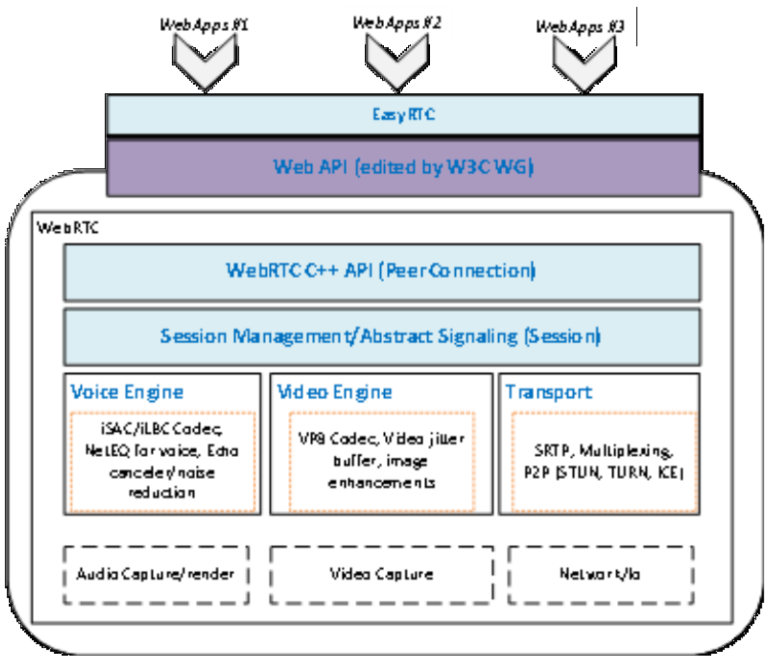

WebApps #1
WebApps #2
WebApps #3
EasyRTC
Web API (edited by W3C WG)
WebRTC
WebRTC C++ API (PeerConnection)
Session Management/Abstract Signaling (Session)
Voice Engine
iSAC/iLBC Codec, NetEQ for voice, Echo canceler/noise reduction
Video Engine
VP8 Codec, Video jitter buffer, image enhancements
Transport
SRTP, Multiplexing, P2P (STUN, TURN, ICE)
Audio Capture/render
Video Capture
Network I/O

Fig. 1. WebRTC Application Architecture using EasyRTC APIs.

3.2 System Design

There are two type application models of WebRTC. They are Triangular model and Trapezoid model. In the WebRTC Triangular model, peer-to-peer communication between browsers are running a web application that downloaded from a different web server, whereas in the WebRTC Triangular model, the web applications downloaded from the same web server. This model is the most common WebRTC scenario (Sergiienko, A., 2014).. This research implements Triangular model to develop unified communication based on WebRTC. The flow diagram as shown in fig.2, proceed through the following steps:

1. Peer A as the initiator connects to the node.js signalling server and generate the signalling channel.
2. When the initiator getting the user’s approval, the initiator gets access to the user’s media devices. It could be a microphone or/and camera.
3. Peer B as the joiner connect to the node.js server and joins the signalling channel.
4. Once the joiner getting the user’s approval and get access to the user’s media devices, a message is sent to Peer A and trigger the negotiation step:
    a. Peer A creates a *PeerConnection* and enhances the local stream to *PeerConnection,* creates SDP offer and sends it to Peer B through the node.js server.

   b. Upon received SDP offer, Peer B copies the behaviour of Peer A by creates a *PeerConnection* object, adds the local stream to *PeerConnection* and build a SDP answer to be sent back to Peer A.
5. The node.js server (signalling server) exchange network reachability data in the form of ICE protocol candidate addresses.
6. Once Peer A receives the Peer B's answer, the negotiation procedure is finish. Both peers communicate by using their own *PeerConnection* objects. The data channel also exists in this objects to exchange text-message.

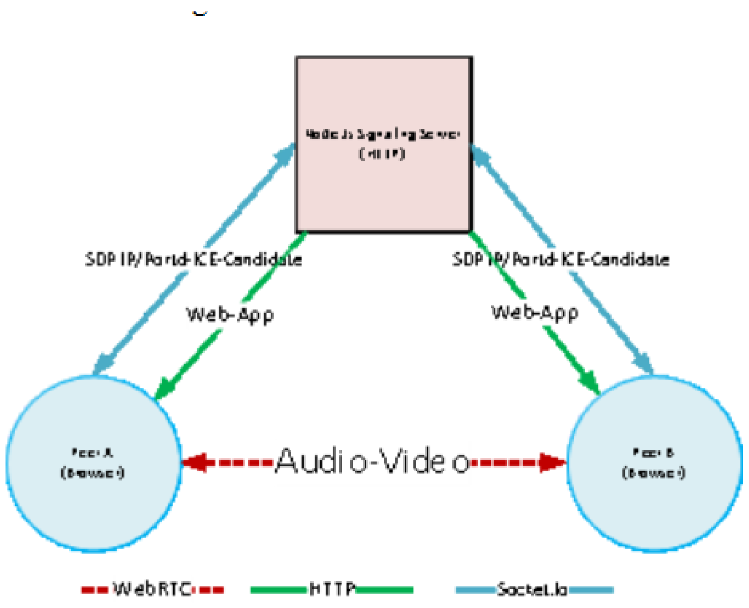


Fig. 2. WebRTC applications with Triangular Application Model

The WebRTC web application written as a combination of HTML5, JS, and CSS. However, the signalling server is written in node.js. Both browsers are connected to the node.js server using socket.io connections and exchange information through node.js server. Socket.io is a JavaScript library that runs on the client-side browser and server-side library for node.js server. Node.js is powerful software platform which permits users to develop a scalable server-side application with JavaScript easily (Loreto,S., Romano S.P., 2014).The WebRTC web application and web browsers communicate using the standardised WebRTC API through, to control and exploit the real-time browser function, such as querying browser abilities and get a notification which generates by the browser. While the interaction between browsers and web server privately-run over web sockets. The real-time multimedia communication process is shown in fig. 3.

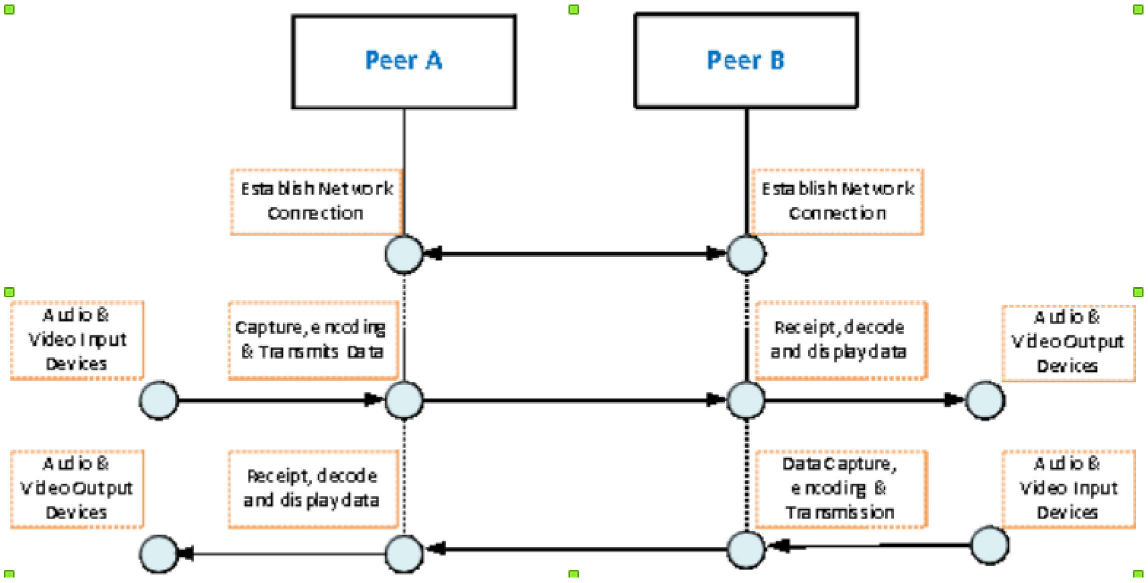

Peer A
Peer B
Establish Network Connection
Establish Network Connection
Audio & Video Input Devices
Capture, encoding & Transmits Data
Receipt, decode and display data
Audio & Video Output Devices
Audio & Video Output Devices
Receipt, decode and display data
Data Capture, encoding & Transmission
Audio & Video Input Devices

Fig. 3. (a) The real-time multimedia communication process.

Develop a real-time multimedia communication between browsers require video technology, audio technology and network communication technology. Audio and video input-output device used to get and play multimedia information. Once input device collects audio and video from Peer A, the data must be encoded and transmitted. Whereas, the other peer, the peer B need to accept the data and decode it so that it can be displayed. The transportation between browsers needs a stable network connection to guarantee the data transmission because there is a huge number of data need to be transmitted.

3.3 WebSocket Signaling

Any communication mechanism which can use to exchange Session Description Protocol (SDP) between peers can be used for signalization (Loreto,S., Romano S.P., 2014). SDP describe the peer's parameters (Dutton). It is protocol to describe multimedia communication session for session announcement, invitation and negotiation. SDP does not send the media data itself, otherwise, it is used for negotiation between peers of media types, formats, resolution, codecs, encryptions and other associated properties (Dutton). The signalling is mechanism use to detect peers presence, exchange offer/answer between peers as well as exchange ICE candidates. This research uses WebSocket protocols as the signalling protocol. Other web application may use different standardised signalling protocols such as XMPP and SIP. The WebSocket API is easy, lightweight and fast to utilise bi-directional communication within web applications (Thakkar S., Thakkar K., Bhanushali B., Arora A., Chawla D., 2015). The WebSocket API server can send and receive a message directly to browsers which make this simple to offer/answer signalling that needs to setup WebRTC communication. Else, a single WebSocket API server can support very large number of clients. The WebSocket connection establishment process is integration WebSocket service in HTTP server, open the port and listen to the requests. Otherwise, the browser creates WebSocket object according to the server address and port. After the handshake, the connection will be made, and the browser can push signal data to server and vice versa. The mechanism is shown in fig.4.

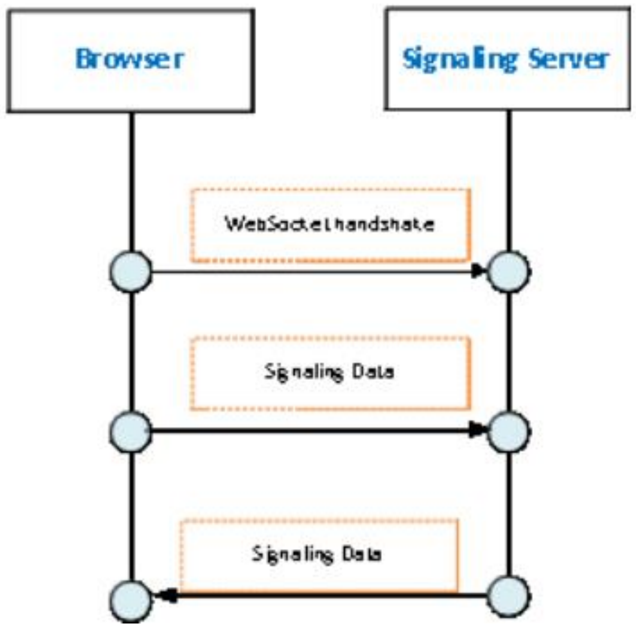


Fig. 4. (a) WebSocket connection establishment process.

3.4 Real-Time Data Transmission

Data transmission between server and browser need to timeliness and low latency, therefore WebRTC uses UDP (Manson, R., 2013).. Unified communication based on WebRTC also need the capability to recognise each other on the network. If peers located in intranet without firewall or NATs, each peer can query its OS for its IP address. In the real world, usually there are many firewalls or NATs between peers, therefore ICE agent that colligates TURN, STUN is used to penetrate firewalls and NATSs in WebRTC (Khan M). Each connection object of WebRTC contains an ICE agent which responsible for collect local IP, port tuples and queries an external STUN server to retrieve the public IP and port tuples of the peers as shown in fig.5.

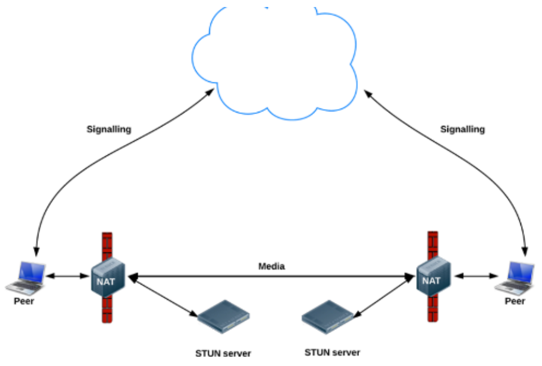
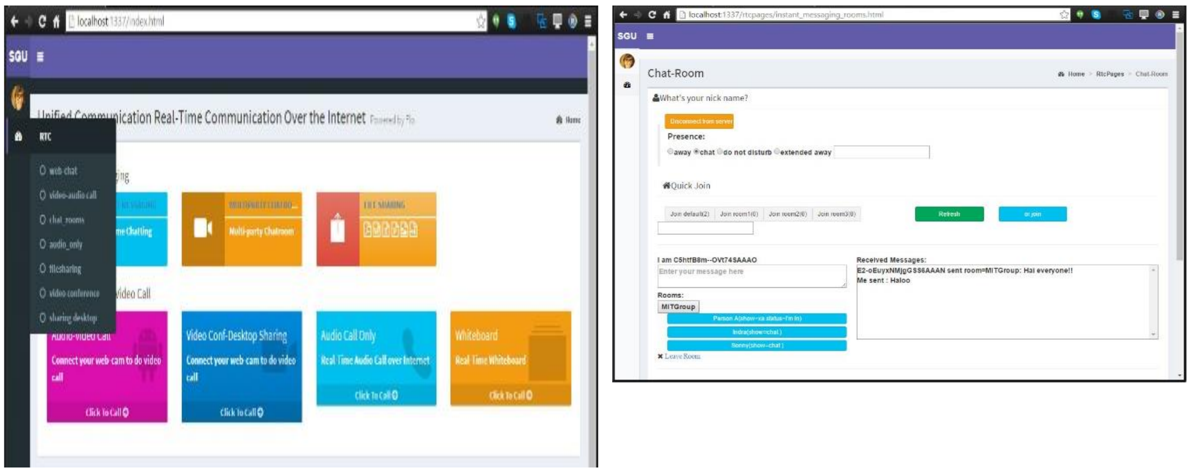


Fig. 5. (a) WebRTC Real-Time Data Transmission using STUN server.

4. Implementation and Analysis

This research implements a simple unified communication-based WebRTC using node.js signalling server. This system support p2p connections after signalling negotiation, therefore users can perform real-time communication such as chatting, video conferencing, audio conferencing (click-to-call), and file transfer. Additionally, if this system hosted on the internet, this system support browser-mobile devices and supports multimedia real-time interaction between the smartphone. The system screenshots of UC page and chat-room page shown in fig. 6.



I am C5htfB8m--OVt74SAAAO

Received Messages:
E2-oEuyxNMjgGSS6AAAN sent room=MITGroup: Hai everyone!!
Me sent : Haloo

Fig. 6. (a) Screenshot of Unified Communication's page; (b) Screenshot of chat room.

The screenshot of video-audio call/video conferencing shown in fig.7. In this scenario, Person A (Peer A) and Person B (Peer B) login to the same web server (triangular model), connect via WebSocket and Communicate each other via camera and microphone on their computer.

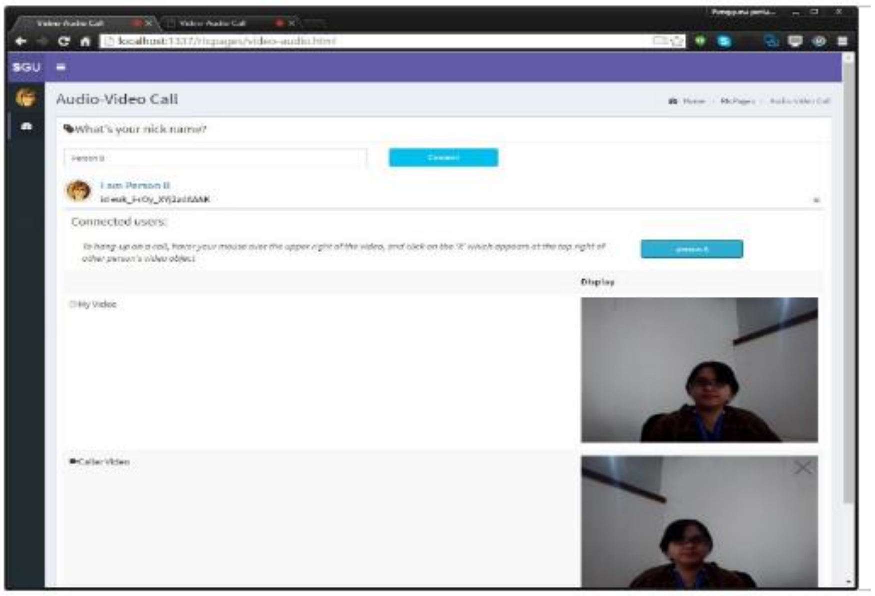
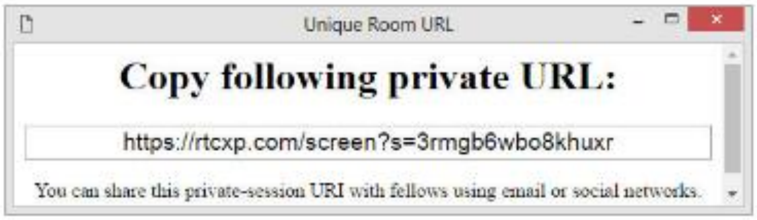


Fig. 7. Video Conferencing.

The screenshot of desktop sharing shown in fig.9. In this scenario, Person A (Peer A) share her desktop to Person B (Peer B). The WebRTC API sharing media via WebRTC desktop Sharing (chrome plugins). As soon as Peer A access WebRTC desktop Sharing, then browser show private session URI, as shown in fig.8, therefore Peer B and the other peer that allowed accessing the URI can see the Peer A's desktop. In this scenario, peer A sharing PowerPoint Presentation as shown in fig.9 (b).

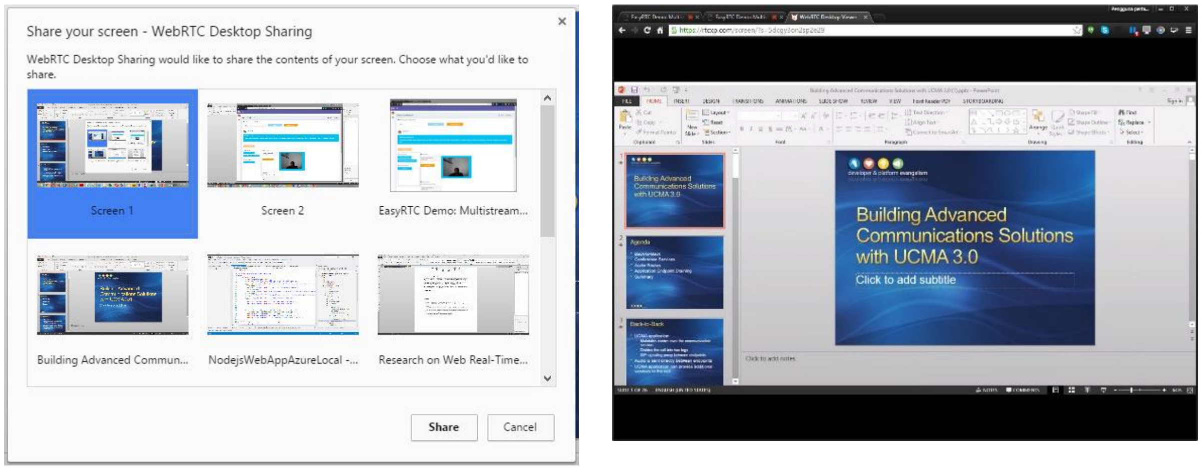

Unique Room URL

Copy following private URL:

https://rtcxp.com/screen?s=3rmgb6wbo8khuxr

You can share this private-session URI with fellows using email or social networks.

Fig. 8. Private session URL.


Share your screen - WebRTC Desktop Sharing

WebRTC Desktop Sharing would like to share the contents of your screen. Choose what you'd like to share.
Screen 1
Screen 2
EasyRTC Demo: Multistream...
Building Advanced Commun...
NodejsWebAppAzureLocal -...
Research on Web Real-Time...
Share
Cancel
Building Advanced Communications Solutions with UCMA 3.0
Click to add subtitle
Click to add notes

Fig. 9. (a) WebRTC desktop Sharing; (b) Private session URI open on Browser.

This system implement HTML5, EasyRTC Client API, JavaScript and CSS on the client side, otherwise the HTTP server develop using node.js Express module and configure Google's public test server as STUN server. The analysis result is the system running on the HTTP server have low-cost hardware, real-time communication function running well in the whole communication process. This system running well using Google Chrome version 51.0.2704.103 m (64-bit) and Firefox version 47.0.

5. Future Scope

The way people communicate are dynamic, therefore modification WebRTC and traditional communication using PSTN are challenging, and research about it can be valuable. WebRTC as peer-to-peer communication on browser also produce prospects to IoTs (Internet of Things), since many new endpoint devices become browsers. Another research opportunities are real-time media (video-audio) recording and securing WebRTC data transmission. Any research about the enhancement of WebRTC will be valuable for society.

6. Conclusion

The web and internet technologies have transformed communication. WebRTC offers to take this transformation a step further. The free, open-source project enable the browser to exchange information in real-time using simple JavaScript APIs. This research proposes solutions to enrich unified communication through WebRTC APIs using WebSocket protocol and node.js for signalling server, therefore real-time communication between browsers can be completed. In the end of research, the system builds to verify capabilities of WebRTC on unified communication such as desktop sharing, file sharing, video conferencing, voice call, presence and room chatting or messaging.

References

Altanai, (2014). ”WebRTC Integrator’s Guide: Successfully build your own scalable WebRTC infrastructure quickly and efficiently”, Packt Publishing.

Dutton,” WebRTC in the real world: STUN, TURN and signalling”. Retrieved July 12, 2016, from http://www.html5rocks.com/en/tutorials/webrtc/infrastructure/.

Jian, C., Lin Z., (2015). “Research and Implementation of WebRTC Signaling Via WebSocket-based fro Real-time Multimedia Communications”,5th International Conference on Computer Sciences and Automation Engineering,

Khan M., WebRTC Signaling Concepts ® Muaz Khan. (n.d.). Retrieved July 12, 2016, from https://www.webrtc-experiment.com/docs/WebRTC-Signaling-Concepts.html.

Loreto,S., Romano S.P., (2014). “Real-Time Communication with WebRTC peer-to-peer in the Browser”, O’Reilly Media, Inc.

Manson, R., (2013). ”Getting started with WebRTC, explore WebRTC for real-time peer-to-peer communication”, Packt Publishing.

Rouse, M.,” Unified communications (UC)”. Retrieved July 12, 2016, from http://searchunifiedcommunications.techtarget.com/definition/unified-communications.

Sergiienko, A., (2014). ”WebRTC Blueprints”, Packt Publishing.

Thakkar S., Thakkar K., Bhanushali B., Arora A., Chawla D., (2015). “Research and Review of Web RTC- the Next Generation of Web-based Communication”, International Journal Of Advanced Research in Computer Science and software Engineering 5, 6.