



P2P AUDIO AND VIDEO CALLING APPLICATION USING WEBRTC

Hairudin Abdul Majid, Azurah Abu Samah, LizawatiMi Yusuf, DewiNasien and Tek Loon Cheah

Faculty of Computing, UniversitiTeknologi Malaysia, Skudai, Johor Bahru, Malaysia

E-Mail: hairudin@utm.my

ABSTRACT

This paper discussed the internet access had become primary issues among UniversitiTeknologi Malaysia (UTM) students and it causes difficulty in discussion for tasks and assignment. Thus, this proposed P2P audio and video calling application using local area network of UTM and it does not require any installation of extra plug-ins. The application reduces the students data usage and reduce the cost of perform audio and video calling. The features of the proposed system are provide a real-time audio and calling application for students to interact with each other; and able to let each other chatting with text and transferring files to each other.

Keywords: WebRTC, P2P, real-time, HTML5.

INTRODUCTION

Internet access has always become a primary issue discuss among Universiti Teknologi Malaysia (UTM) students. Students in UTM always require internet access to perform online discussion through audio and video calling tool like Skype, Google Hangout and etc. All of these applications require internet access.

Web Real-Time Communications (WebRTC) is a set of Javascript Application Programming Interfaces (APIs) and it enables web developers to develop Real Time Communication (RTC) features into their web-based application without bother any complexities of plugins. Google launched an open source project for web-based real-time communication and known as WebRTC at May 2011 [1]. However, there are a lot of applications in current market like Skype, Google Hangout, FaceTime and most of the applications are using client-server architecture. Users connect using an agent; this agent could be mobile phone, workstation, and other hardware or software application. Then, the agent connects to a central server. However, client-server architecture using in application will increase a lot of system cost such as configuration and maintenance. Peer-to-peer (P2P) architecture is better as it is scalable and reliable than client-server architecture as single nodes failure will not affect against the whole system. Besides, WebRTC system consists of web servers, browsers with different operating system, workstations, tablets, mobile phone. In addition, WebRTC caninteroperate with Session Initiation Protocol (SIP), Jingle and Public Switched Telephone Network (PSTN). Due to its interoperability with VoIP and other video communication system like SIP, Jingle, and PSTN;WebRTC is the best options in implementing P2P audio and video calling application in UTM.

RELATED WORKS AND OBJECTIVES OF THIS STUDY

Communication problem among UTM students

Interview had been carried out with UTM students to understand the problem. Students unable to access to Internet due to the inconsistent and limited Internet service provided. Thus, students are unable to communicate with their group members to discuss on assignments and tasks. Due to transportation problem, it is inconvenience for all the students to gather together and perform discussion.

Therefore, in order to resolve the issue, video and calling application run using the local area network necessary to be implemented.

Existing studies for audio and video calling application

Audio and video calling application resolved a lot of problems in the world and it became a very important application in both personal use and business. Forpersonal use, people tend to communicate with their friends and family that are far away from each other. While for business usage, it ease the company operations as virtual meeting can be created with the audio and video calling applications. There are three existing systems found, which is Skype, Google Hangout and FaceTime.

All of these applications almost have the same features like group audio calling, group video calling, text-chatting and file sharing. The most popular used is Skype. However, each of them has their own drawbacks where Skype have low quality of audio transmission, while Google Hangout is only available to the Google+ users. FaceTime is available to Apple users only. Lastly, all of them are not open source which causes integration of the technology to big organization is difficult and not favoured by organization.



PROPOSED SYSTEM

Overview of proposed system

Figure-1 indicates an overview of the proposed P2P audio and video calling application function. This proposed system provides direct real-time connection with remote peers without any server. In order to initiate the connection, users have to go through the signaling server by identify and locate the remote peers. After the session established, two-way direct connection can be activated without go through signaling server.

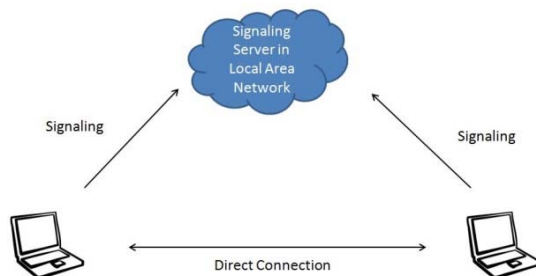


Figure-1. Overview of proposed system.

The proposed system is assumed to be used by two students at one session. Students can use their own workstation or any other devices as long as it contain HTML5-enabled browser and the browsers supported WebRTC. Two procedures required before two students can start communicate with each other. Firstly, caller have to setup the call session and a unique token for the call session will be generated which is specific URL. Secondly, caller just has to share the URL to his friend and paste the URL link into the browser and the call session will be established.

Proposed system function

In this system, there are seven use cases for students. Figure-2 show the use case diagram of the system. In order to use the system, students have to login to the system by using their ACID ID and password. Once users login into the system, users will have two choices, it is either create an audio call or create a video call. In order to create audio call, browser will request access to the user's microphone; while if it is video call, browser will request access to both user's microphone and camera. If the request to access is block, the communication will not be able to setup. After the two ways communication established, users can perform audio chat only if audio call is chosen while video chat can see the people through camera and chat through microphone. Besides, users also can communicate through text chatting and transferring file to each other.

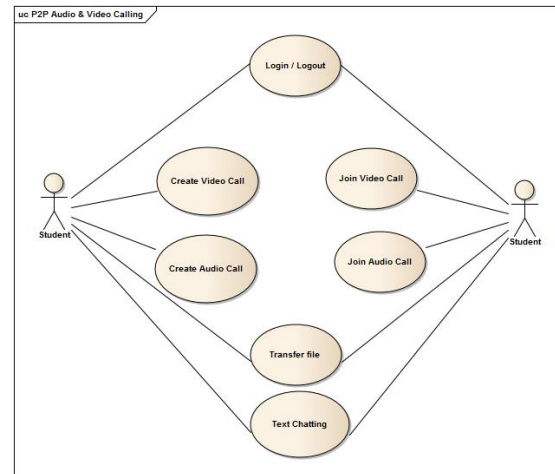


Figure-2. Use case diagram of proposed system.

Setup signaling server

The signaling server is setup using Websocket and Nodejs. WebSocket is full-duplex and single-socket connection. With WebSocket, the Hypertext Transfer Protocol (HTTP) request from browser becomes a single request to open the WebSocket connection. Thus, this single request feature reduces the latency between server and client as the server does not have to wait for the request from client [2]. Client is able to communicate with server at any time. Besides, WebSocket provides better performance and more efficient when it comes to real-time communication. Before WebSocket architecture exists, real-time communication between HTTP is very complicated to maintain. However, WebSocket simplify the communication between client and server. In order to develop a web application, HTTP is definitely needed. However there were major deficiencies for HTTP to support real-time communication. The implementations of WebSocket address all the major deficiencies and allow buildingahigh quality real-time web application as the asynchronous and bidirectional patterns connection [2].

WebSocket enables web conferencing and reduces communication overhead. Efficiency of the communication between web servers and clients increase. The WebSocket willbe running with Node.js. Node.js also known as Node and it is a server-side JavaScript environment. It is a platform for building fast, scalable network applications easily. As shown in Figure-3, we declared some useful library from NodeApplication Program Interface (API) which we had use in our application. They were HTTP, File System and WebSocket. Then some general variable like the port number and clients were defined. The port number is the port number for the application to run. Clients is the variable for store the client who using the application.



```
//API function
var http = require("http");
var fs = require("fs"); //File System library
var websocket = require("websocket").server;
//General variable
var port = 1111;
var clients = [];
//The list of call token
var webRTC_call_token = {};
```

Figure-3. Overview of proposed system.

Before any direct peer-to-peer connection, Amy need to know whether the Bob is in the network that is reachable and get the permission of willing to connect to each other. In order to achieve that, Session Description Protocol (SDP) is implemented. As shown in Figure 4, in order to initiate the peer connection, Amy will start gather the local information and create an SDP offer and send through signaling server or signaling channel to Bob. Bob will respond with the SDP answer to the initiator through the signaling server.

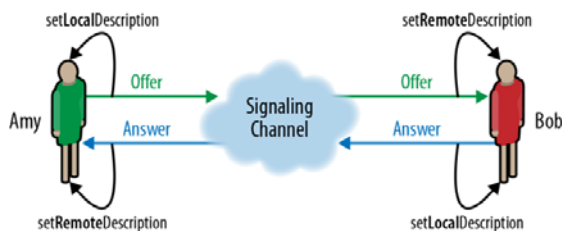


Figure-4. Work flow of SDP.

Implementation of audio and video calling

Audio and video calling function were two separate functions. However, both of these two functions were using the same coding. The difference between these two functions was the streaming elements and request access to your computer. For audio calling, the function just request to your microphone and stream the audio to the other peer while video calling request both camera and

microphone and stream video and audio together. The video calling function coding was shown at Figure-5. When user gets access to the audio or video stream from local workstation, user will have to stream the local stream to the other peer and accept the streaming from other peer so we can interact with each other.

```
function setup_video() {
  get_user_media(
    {
      "audio": true,
      "video": true
    },
    function (local_stream) {
      console.log("new local stream added");
      connect_stream_to_src(local_stream,
        document.getElementById("local_video"));
      document.getElementById("local_video").muted =
        true;
      console.log("local stream added to
        peer_connection to send to remote peer");
      peer_connection.addStream(local_stream);
      local_stream_added = true;
    },
    log_error
  );
}
```

Figure-5. Coding snippet of call WebRTC API.

EVALUATION OF PROPOSED SYSTEM

The prototype system has tested through black box testing and user acceptance test.

Black box testing

Black box Testing is the technique that tests the functionality of the application without understanding of how the internal structure working [3]. The testers should only know what the expected output is when the tester gives an input. The table shown all the possible input, expected output and the result of whether the real output were similar with the expected output. Figure-6 shows the result of black box testing of Video Calling Page, Audio Calling Page, Text Chat Page and File Transfer Page.

Testing statement	Events	Expected Output	Result
Video Calling Page	Browser will appear the pop-up box to request for permission to access the microphone and camera. There were two options which is allow and cancel.	If click allow, users will be given an unique token for the video calling session to share with friend. Otherwise, users will not be able get the token for video calling.	Passed
Audio Calling Page	Browser will appear the pop-up box to request for permission to access the microphone. There were two options which is allow and cancel.	If click allow, users will be given an unique token for the audio calling session to share with friend. Otherwise, users will not be able get the token for audio calling.	Passed
Text Chat Page	User enter message into the textbox and press enter.	The message typed by user will be added into the text box.	Passed
Text Chat Page	User left blank in the textbox and press enter.	The message 'Please type a message' will pop-up.	Passed
File Transfer Page	User select more than one file.	Prompts pop-up messages 'Please select one file only at a time'.	Passed

Figure-6. Black box testing result.



User acceptance test

User Acceptance Test (UAT) also known as beta testing consists of a set of processes to verify the developed system works for the user. The purpose of conduct UAT is to seek approval or agreement from the system users, which is student in UTM Johor Bahru. This process is the last process in system testing before the system deploy to production environment. The prototype of P2P Audio and Video Calling application was tested by real users of the system in this test.

Figure-7 shows that the summarized result of the user acceptance test and the comments on each function in the system.

There were also extra comments other than the comments in table. The extra comments related to the system flows, the system flows and the process was complicated if without the user manual and guidance of people when first time using the system.

No.	Acceptance requirement	Critical	Test result	Comments
1	The system must be able to execute audio calling function.	Yes	Accept	The transmission of audio was not clear enough and sometimes the sound echo effect cause the irritated sound broadcast from speaker.
2	The system must be able to execute video calling function.	Yes	Accept	The transmission of video was good and the video flow was smooth. However, the video stream quality depends on the camera quality that your devices own.
3	The system able to execute text chat function.	No	Reject	The text chat function was extra useful when the audio transmission quality was low. The text sending with each other was very fast. It does not lag at all.
4	The system must be able to execute the file sharing function.	Yes	Accept	The file sharing was very useful for discussion purpose. However, file sharing only available when audio and video calling was performed. It would be better if the file sharing can be standalone.
5	The system must be able to let valid user login.	Yes	Accept	The login is easy and simple. However, if the system is going to be implemented in UTM, it is better to just open to everyone to use it without any login needed.

Figure-7. User acceptance test result.

CONCLUSIONS

The development of this project was follow the phases that pre-determined based on the methodology used. It started with initial project planning which defined all the information of the project and what need to be achieved so the project development able to run smoothly. During the project development phase, several objectives that defined at the beginning of the project had achieved:

- The objective for design of P2P Audio and Video Calling application had been established through compare the system existing in the market. Through the study and analysis of the related system, system architecture design, database design and interface design had developed.
- The objective for development of P2P Audio and Video Calling application using WebRTC had also achieved. After the design of the application

completed, development phase carried out and it involve server-side and also client-side coding.

- The objective for testing of P2P Audio and Video Calling application had achieved in this project as well. After the development of the application completed, several people were recruited in assist of black-box testing on each function in the system. Furthermore, user acceptance test also carried out in order to gather feedback on the system function and system flows before the application ready for deployment.

In future study, several areas could be improved in this application. Firstly, also is the most vital one is the limitations of number of users in one video calling session. If the number of user increases for video conferencing purpose, it will encourage more users to use this application. There were still a lot more function can be



implemented with WebRTC technology like video broadcast which enable lecturer to teach online. Thus, it is more advisable for future improvement is more towards video broadcast and increase more file type that support in file sharing.

REFERENCES

- [1] Alexandru C. 2014. Impact of WebRTC (P2P in the Browser). inB. Stiller *et al.* (Eds.). Internet Economics VIII. Technical Report. Department of Informatics, University of Zurich.
- [2] Vanessa W., Salim F. and Moskovits P. 2013. The definitive guide to HTML5 WebSocket. New York: Apress.
- [3] Beizer B. 1995. Black-box testing: techniques for functional testing of software and systems. Canada: John Wiley and Sons.