

Constructing an Interactive
Multimedia Enabled Virtual Lab Learning Environment

On Vlab Platform

by

Qingyun Li

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved November 2014 by the
Graduate Supervisory Committee

Dijiang Huang, Chair
Hasan Davulcu
Partha Dasgupta

ARIZONA STATE UNIVERSITY

December 2014

ABSTRACT

Interactive remote e-learning is one of the youngest and most popular methods that is used in today's teaching method. WebRTC, on the other hand, has become the popular concept and method in real time communication. Unlike the old fashioned Adobe Flash, user will communicate directly to each other rather than calling server as the middle man. The world is changing from plug-in to web-browser. However, the WebRTC have not been widely used for school education.

By taking into consideration of the WebRTC solution for data transferring, we propose a new Cloud based interactive multimedia which enables virtual lab learning environment. Three modules were proposed along with an efficient solution for achieving optimized network bandwidth. The One-to-Many communication was introduced in the video conferencing and scalability was tested for the application. The key technical contribution is to establish a sufficient system that designed to utilize the WebRTC in its best way in educational world in the Vlab platform and reduces the tool cost and improves online learning experience.

DEDICATION

To My Beloved Family.

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my advisor and committee chair, Dr. Dijiang Huang, whose expertise and knowledge always inspire and motivate me to be an excellent student. Without his guidance and persistent help this dissertation will not be able to be completed in a timely manner.

I would also like to thank my committee members, Professor Partha Dasgupta and Professor Hasan Davulcu who gave me great feedback and advice on my thesis.

Also, I would like to thank everyone who is working in Secure Networking and Computing (SNAC) Lab for supporting me and making the research project possible.

TABLE OF CONTENTS

	Page
LIST OF TABLES.....	vii
LIST OF FIGURES.....	viii
CHAPTER	
1) INTRODUCTION.....	1
1.1 Motivation of the Research.....	3
1.2 Proposed Solution.....	5
2) VLAB BACK GROUND.....	7
2.1 Vlab System.....	7
3) TECHNICAL DETAILS.....	9
3.1 Concepts of HTTP.....	9
3.1.1 How HTTP Works.....	9
3.2 Concept of WebSocket.....	10
3.2.1 Advantages of WebSocket.....	11
3.3 WebRTC.....	12
3.3.1 WebRTC Work Flow.....	13
3.4 Benefits of Using WebRTC and WebSocket.....	16
3.5 Node js.....	17
3.5.1 Benefits of Node js.....	18
3.5.2 Node.js Use Cases.....	20
4) SYSTEM DESIGN.....	21
4.1 Overall Architecture.....	21
4.1.1 Access Control.....	22

CHAPTER	Page
4.1.2 Vlab Pages.....	22
4.1.3 Multimedia Learning WebPage.....	22
4.2 User Scenarios.....	22
4.2.1 Authentication.....	23
4.3 Multimedia Web Page.....	24
4.3.1 Control Bar.....	25
4.3.2 User List.....	26
4.3.3 Webcam Frame.....	26
4.3.4 Presentation Frame.....	27
4.3.5 Chat List.....	27
5) SYSTEM IMPLEMENTATION.....	29
5.1 Server.....	29
5.1.1 Server.js.....	29
5.1.2 Node Js.....	30
5.2 Authentication.....	30
5.2.1 Log In.....	31
5.2.2 Room Check Page.....	31
5.3 Multimedia Main Page Use Cases.....	32
5.3.1 Video Use Cases.....	32
5.3.2 Texting Use Cases.....	35
5.3.3 File Sharing Use Cases.....	36
5.3.4 Virtual Machine Sharing.....	38

CHAPTER	Page
5.3.5 GUI redesign.....	39
6) EVALUATION	41
6.1 System Advantages over Existing System.....	41
6.1.1 Multiple Peer Video Conferencing.....	41
6.1.2 File Sharing.....	41
6.1.3 License Free.....	42
6.2 Improvements on Bandwidth.....	42
6.2.1 One-on-one Video/Audio Conferencing.....	42
6.2.2 Many-to-Many Video/Audio Conferencing.....	43
6.2.3 One-to-Many Video/Audio Conferencing.....	43
6.2.4 Solutions for Many-on-many Bandwidth Usage.....	44
6.3 Improvements on Function Controls.....	46
6.3.1 Modules Separation.....	46
6.3.2 Control Functions for Single User.....	47
6.3.2 Control Functions for Single Video Frame.....	48
7) CONCLUSION.....	49
7.1 Conclusion of Current Work.....	49
7.2 Future Work.....	50
8) REFERENCES.....	51

LIST OF TABLES

Table	Page
1) 6-1 One-on-one Video Sharing Bandwidth Usage.....	40
2) 6-2 Many-to-Many Video Sharing Bandwidth Usage.....	41
3) 6-3 One-to-Many Video Sharing Bandwidth Usage.....	42

LIST OF FIGURES

Figure	Page
1) 2-1: Vlab System.....	8
2) 3-1: HTTP Work Flow.....	10
3) 3-2 Overhead Difference between Polling and WebSocket.....	11
4) 3-3 Response Difference between Polling and WebSocket.....	12
5) 3-4 WebRTC Support Growth.....	13
6) 3-5 WebRTC Diagram.....	14
7) 3-6 TURN And STUN Server.....	15
8) 3-7 Traditional Server And Node.js.....	19
9) 4-1 Overall System Design.....	21
10)4-2 Designed Username Check Page.....	23
11)4-3 Designed Room Check Page.....	24
12)4-4 Multimedia Web Page.....	25
13)4-5 Control Bar.....	25
14)4-6 User List.....	26
15)4-7 Video Frame.....	27
16)4-8 Chat List Area.....	28
17)4-9 One-to-One WebRTC Module.....	29
18)4-10 Many-to-Many WebRTC Module.....	29
19)4-11 One-to-Many WebRTC Module.....	30
20)5-1 Server.js Command In Node.js.....	32
21)5-2 Log In Page.....	33

Figure	Page
22) 5-3 Room Check Page.....	33
23) 5-4 Multimedia Main Page.....	33
24) 5-5 Video Chatting.....	34
25) 5-6 Text Communication	35
26) 5-7 File Choose.....	35
27) 5-8 File Receive Situation	36
28) 5-9 VM user scenarios	37
29) 5-10 GUI change	38
30) 6-1 many-to-many vs One-to-many bandwidth usage.....	43
31) 6-2 Frontend Bandwidth control algorithm.....	45
32) 6-3 One-to-many vs Bandwidth control solution.....	46
33) 6-4 Three Modules.....	47
34) 6-5 User List Buttons.....	47

CHAPTER 1

INTRODUCTION

Recently, interactive remote e-learning has become a popular topic in the educational world. Interactive remote e-learning depends highly on the internet technologies, which manages the learning system and make users develop the learning system online possible. Traditional study would be based on the in person communication first, which means every student need to show up in class at the same time and the same place. And all the students need to follow exactly the same class content and same schedule. Problems which arise are that study in the class can be categorized into several parts: First is that students in the same room will have to follow the same speed and same path to every other, which can do harm to the students who are advancing or falling behind. Second, traditional learning takes physical places, which costs more time and money compare to e-learning. Unlike learning online, people have to study in the class which is time consuming and more expensive. Third, e-learning is easy to adjust and update to specific needs for both students and instructors. The capability of internet makes website or applications easy to be modified over time. The main reason why more and more people choose e-learning is that it is reusable, easy to share the resources, capable to interoperability [1]. With all the advantages that e-learning has, it is becoming more popular than ever with the rising of internet.

Although online learning is getting popular nowadays, several problems still need to be fixed and improved. The most significant problem, which online learning has, is the video conferencing. Most systems that are applicable to users today are proprietary and usually they would require additional software or plug-ins to install before the usage,

which may require more licensing steps and more money to do[2]. Researchers have been working on this for a period of time but still no good applicable solutions were found until the emergence of WebRTC.

WebRTC, on the other hand, is under (W3C) rules, an API defined which supports browser-to-browser voice calling, chatting and P2P file sharing. It runs the supported web browsers without plug-ins. WebRTC is a latest and popular no plugin required online communication tool that gives us the opportunity to simplify steps to develop and use the e-learning tool [3]. By using it, we can simply erase all the software bundles we need to download before using the online learning application. It simply adds the functionalities of online chatting, video conferencing and real-time interacting to the pre-developed Web tools for students and instructors to use [2].

This project attempts to use the WebRTC real-time communication concepts on to Vlab platform. Vlab platform was first developed by Aniruddha Kadne (2010). The goal was to develop the Cloud based resource and service sharing platform for education usage virtual lab [4]. The system has been developed and redesigned for several times. It is now a fully functional Cloud system that is used for instruction purposes to students take Computer Lab Classes in Arizona State University [5].

By using the WebRTC real-time communication concepts in the Vlab system, we are going to provide our students and instructors an interactive multimedia enabled virtual lab learning environment, which requires no plug-in, high availability and easy to learn based on Cloud system. Therefore this research aims to develop a user-friendly, easy to extend, and easy to modify web user interface that can help user on learning online through the Vlab system by using WebRTC remotely totally.

1.1 Motivation of the Research

Online learning has become a very popular education service for the past few years. According to the news, more than 6.7 million students, which is about 32 percent of enrolled students, involved at least one online course through the university. This is a significant number in the amount of the students who join or become members of the group to online studying [6]. Even though we have large numbers of students taking online courses, there is still a serious problem that makes people less interested in e-learning. The main reason for that is the way it communicates through website is harder and less clear than taking in-class courses. This is especially true for computer engineers who have high possibility to involve in actual activities on coding, instant messaging and video conferencing. These functionalities, which traditional online study system does not provide, lead people who study high tech courses prefer in person class.

As time goes by, more and more people have the opportunities to online study. Arizona State University just provides fully accessible online degree for students in certain areas. With the growth in in-class students but limited class rooms ASU can provide, online learning becomes to a better option for all students. To achieve this, the change on the platform of e-learning is significant and cannot be delayed.

Right now most of the online learning applications focus on traditional learning strategies, which people can only see instructor and do only few communication and interaction with the instructor and other students. The complex of setting up the online learning system is also another problem. Students and instructors may not want to add plug-ins to their personal computers. Besides the bad quality of the video, people may not

like the design of the online study room at all. Because of these disadvantages, people have come up with some great ideas on developing a new platform to have better user experience and more stabilized system.

With the merge of the WebRTC, we see the future of interactive remote e-learning. As have been established under the W3C, more and more people are getting to know about this new web technology. It is very convincing that it has many advantages that flash or other online plug-ins does not have:

- First advantage over flash or skype is that WebRTC is open source and easy to code. We know most of the developers want to have some handles during the development cycles over the application. Developer can modify the UI and functionalities of the application to achieve the goal users wanted to have. But flash or skype does not give that opportunity. WebRTC is coded in JavaScript, which essentially the internet's most commonly used language. On the other hand, Flash is coded in Flex, which has very poor video and audio qualities[6].
- Second is that it does not only have the core technology for video conferencing, but also supports for cross-platform which includes but not limited to: Windows, Linux, Mac and Android [7]. This gives developer confidence to develop systems on any of the operation system with no concern about the cross-platform issues that applications usually have.
- Third, it allows web developers to develop media applications completely based on browsers [7]. There is no need for downloading any plug-ins, and developers will not need to handle the signaling process. All they need to do is to write a simple

JavaScript application. This saves lot of time and resource on developing the system, which eventually saves funding.

- Fourth, WebRTC allows users to not only use its own API, but also other JavaScript API and HTML5 open sources [7]. This gives a great opportunity for developer to develop the app in any way they want by using the existing commonly used APIs, which reduces lot of time on reproduce existing APIs.
- Last but not the least; WebRTC is a technology which allows user to establish real-time communication through websites. People do not have to have an application ready on the desktop in order to have a video conferencing in real-time with other people. They can simply input the URL of the website and start the meeting.

1.2 Proposed Solution

By taking into account the above advantages WebRTC has over the traditional online communication tools, we propose a new Cloud based interactive multimedia enabled virtual lab learning environment. This environment is designed to fully use the functionalities that WebRTC has to cooperate with the existing Vlab System to achieve the educational purpose. During the development, I will wrap up WebRTC functions we need for current and future development. The environment will be handled in one virtual machine inside the Vlab system. The server will be held by the Vlab system.

Administrator of the Vlab will be handling the functionalities of the environment.

The other important thing is to make it working with our Vlab System. To manage that, I will use the same access control system we are using for the Vlab. User will have the access to this environment from the Vlab webpage. Class instructors will be

responsible for managing the access control and teaching materials for the online class room. Both instructor and students will have real-time communication abilities during the usage of the environment. Students will talk, text and learn from the webpage, and instructors will be able to manage the course accordingly.

With this environment, Student and instructor will find the best solutions interact online.

CHAPTER 2

VLAB BACK GROUND

In this section I will provide the information about the Vlab platform. As described in previous chapter, our interactive multimedia enabled virtual lab learning environment will be accessed through Vlab. Vlab is the platform which established on top of the mobicloud system. It is capable of creating adaptable, changeable virtual machines in an isolated network system. This is easy to be used to help develop the e-learning system on top of it. In this section I will introduce the current vlab system.

2.1 Vlab System

The current system is developed based on OpenStack. This is the idea image for the system of the cloud.

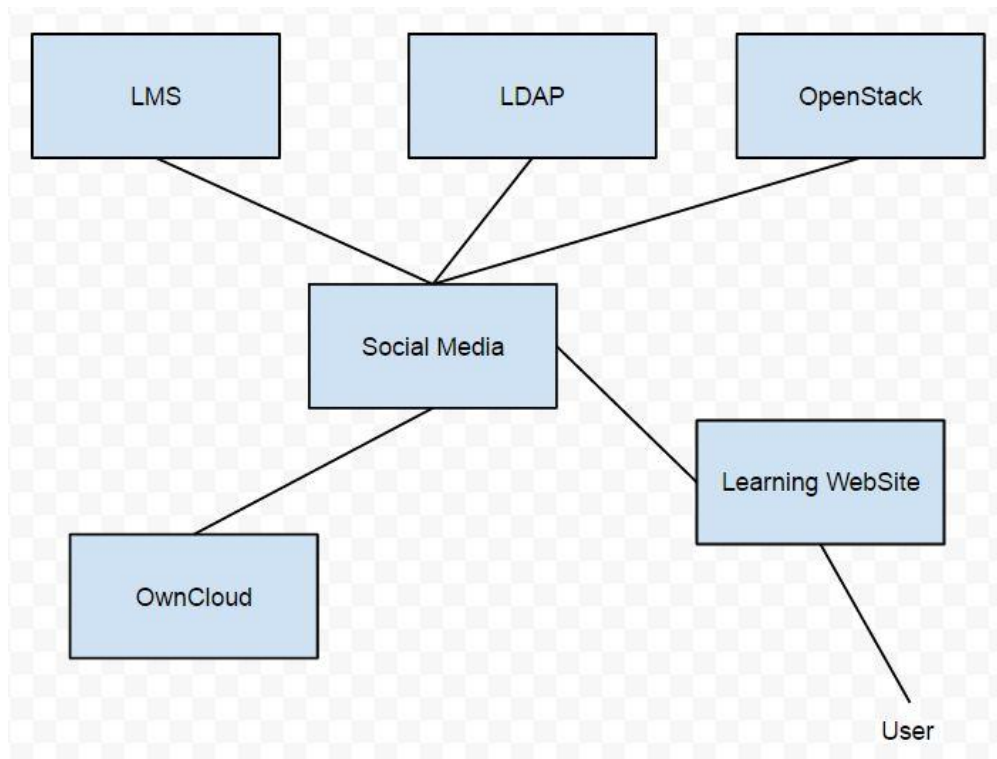


Figure 2-1: Vlab System.

2.2 Vlab Role

The functionalities that Vlab in this application are that Vlab run as the Cloud side server and give user the ability to interact. It provides the basic authentication and fundamental social media background.

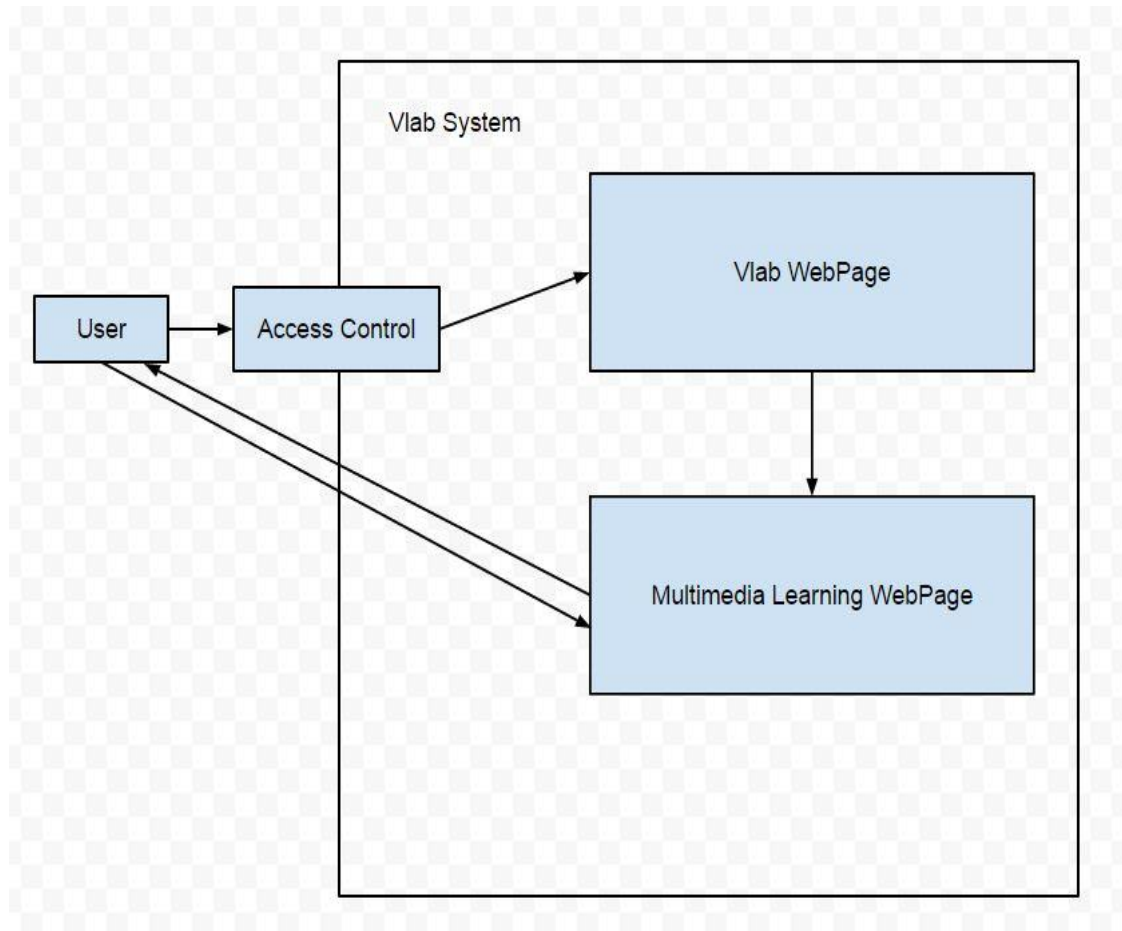


Figure 2-2: Vlab Role.

As we can see from above figure, in order to let the users use the Multimedia enabled virtual lab learning environment on Vlab Platform, They first will go through the access control. After authentication, they will be redirected to Vlab WebPage. In the page, there will be a link to redirect the user to the WebRTC page with another authentication again and select class.

2.2 Why Using Vlab System

The university has hundreds of students using the Vlab right now and the number is increasing fast. The Vlab takes virtualization techniques to implement a steady and reliable network system. By using the virtualization technology in Vlab, students will have full access to virtual world, which is important for online learning especially computer science students. Instructors who are taking the class will be able to teach students on the virtual machine and reach students instantly on the study.

CHAPTER 3

TECHNICAL DETAILS

In this section I will introduce the concepts of HTTP, websocket, WebRTC, the benefits of using WebRTC, and where and how we implement WebRTC.

3.1 Concepts of HTTP

HTTP, short for Hypertext Transfer Protocol, is an application protocol used for distributed, collaborative, hypermedia information system [8]. It is also the foundation of data communication for our internet.

3.1.1 How HTTP Works

How HTTP works is that first the web browser sends out a request to the server side using a TCP/IP connection. Server will send a response to the web browser after receiving the request [9]. During that time, session of the computer will be remembered on the server side and the session ID will be stored in local cookie. Once the response completed, the connection between server and browser will be cut and then browser can send another request to the server to do other things. Here is the illustration on how HTTP signaling works.

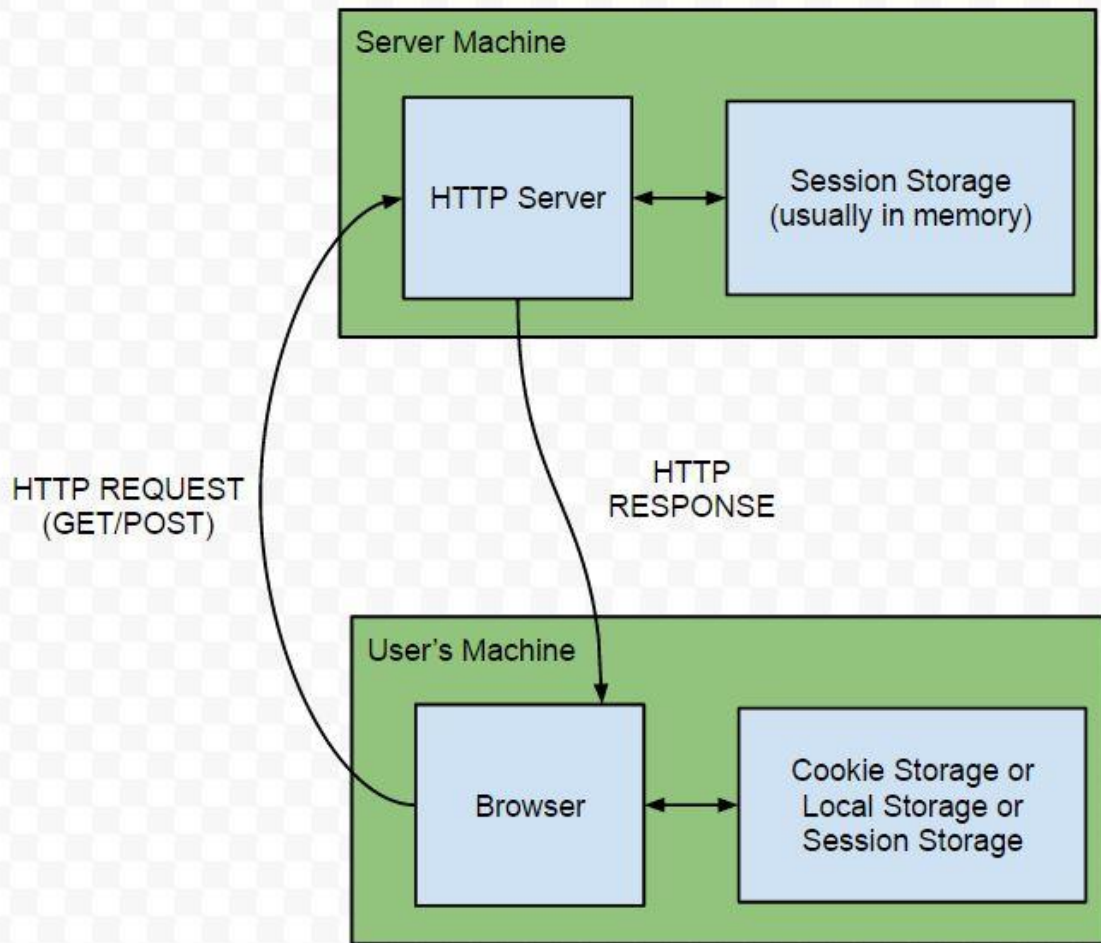


Figure 3-1: HTTP Workflow.

3.2 Concept of WebSocket.

WebSocket is the internet technology supported by HTML5 which is full-duplex communication channel over TCP connection between browser and server [10]. By using the WebSocket API, once we do the handshakes, the connection of the browser to server has been established. The tunnel will remain there. Data can be transformed from one side to another.

3.2.1 Advantages of WebSocket

There are two main advantages for WebSocket over traditional Socket. One is that the Header for WebSocket is much smaller than the traditional one. If we have small amount of data, this is not going to be a problem. But when we are making a video conferencing application, which contains large number of transactions, this will be a concern we need to pay attention to.

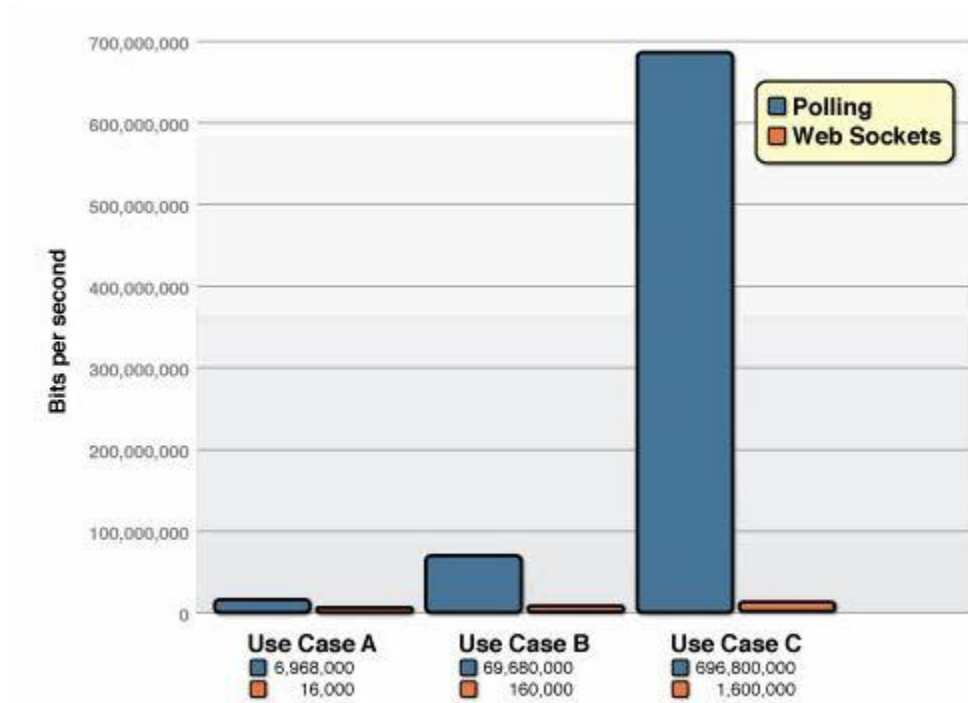


Figure 3-2 Overhead Difference between Polling and WebSocket [11].

Figure 3-2 shows the comparison between the socket polling and WebSocket. It depends on the number of users, which is 1,000, 10,000, and 100,000 with the head weight count. We can see that WebSocket is very efficient compare to traditional HTTP transaction.

The other advantage of WebSocket is that it will allow server to send response without getting request from the client side.

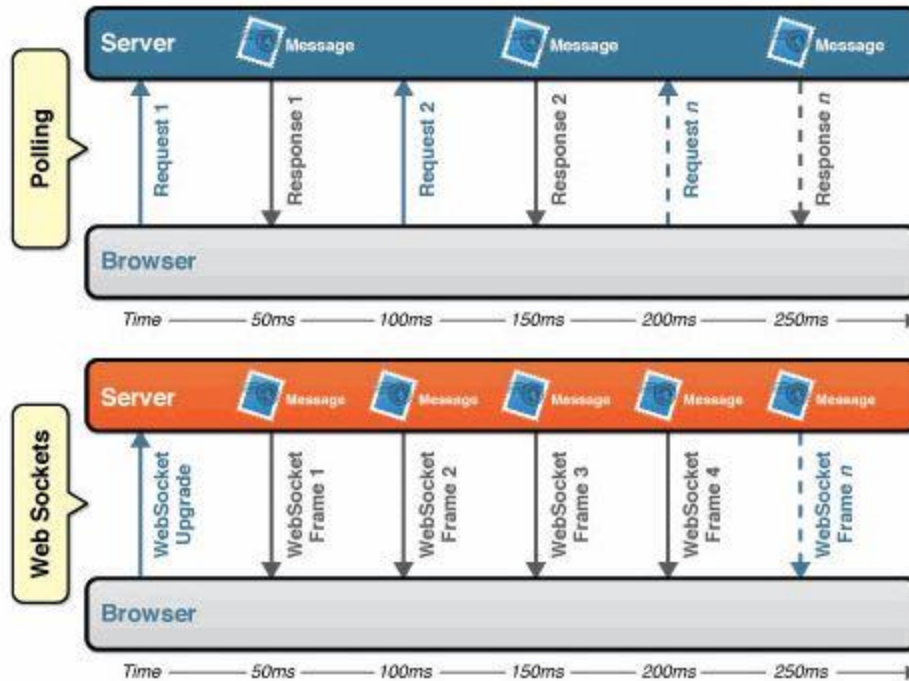


Figure 3-3 Response Difference between Polling and WebSocket [11].

What Figure 3-3 shows us is that unlike HTTP request response terminology, WebSocket opens up a tunnel for messaging once for all. Without sending request, server still send back response as needed in the application. This is important for real-time communication because we need to keep on updating the message between server side and the client side.

3.3 WebRTC

As discussed above, WebRTC is a new open source real-time communication application. It aims at developing real-time communication by using simple JavaScript and HTML5 through web browsers. It is the newest standard for real-time communication. As it grows rapidly, it is intend to replace the Flash. Here is the market share grows image to show how WebRTC market grows.

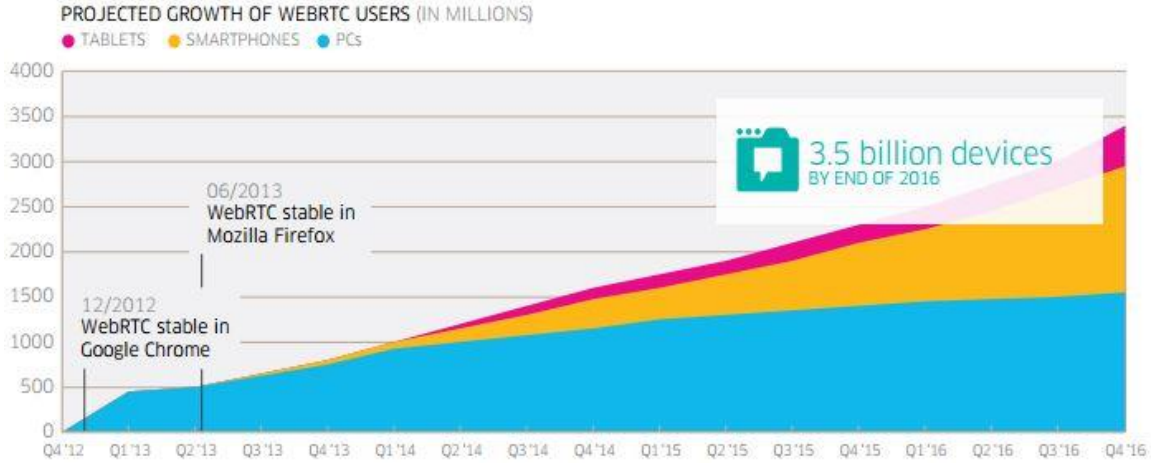


Figure 3-4 WebRTC Support Growth [7].

Figure 3-4 shows how quickly WebRTC supports grow in the recent years with PC, smartphones, and tablets. Unlike Flash which was coded in Flex and hard to do programming on, WebRTC is coded in JavaScript and easy to modify.

3.3.1 WebRTC Work Flow

Even though it aims at peer to peer connections, it does need servers to serve as an transmitter and message handler.

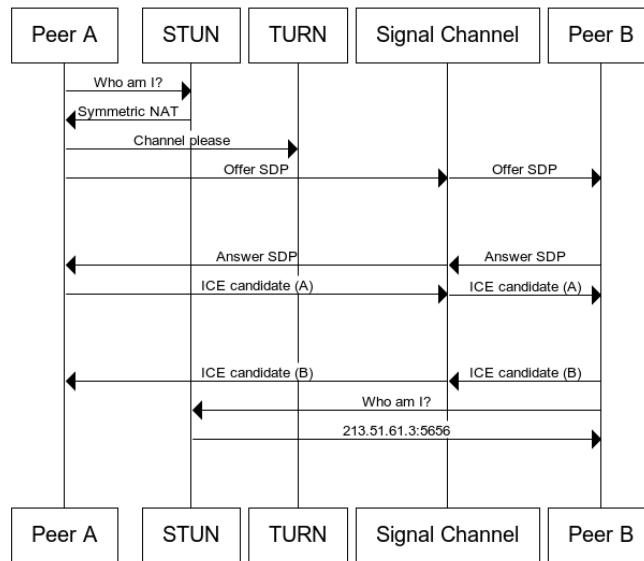


Figure 3-5 WebRTC Diagram [12].

Figure 3-5 shows a complete diagram for WebRTC on how it works among peers and servers.

- ICE stands for interactive connectivity establishment, which is a framework to make connections with peers.
- STUN, which is Session traversal Utilities for NAT, is a protocol to get users' public IP address.
- NAT, short for network address translation, which is used to give users' IP addresses.
- TURN (Traversal using relays around NAT) is meant to bypass the NAT restriction by connecting a TURN server and pass all information through the server. This is how users can find other people in other domain. Here is how TURN and STUN work together to get connection between two peers.

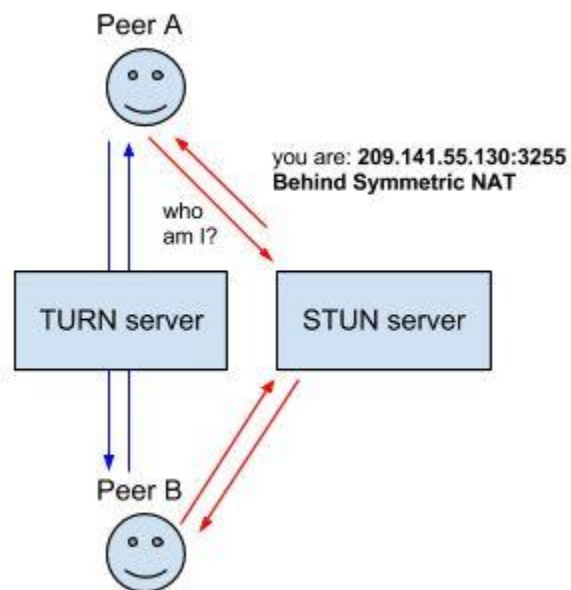


Figure 3-6 TURN And STUN Server [12].

- SDP (Session description protocol) is a protocol standard for multimedia data. So peers get each other's media data instead of the media itself.
- Signal Channel is the channel, which can run as the signal server that handles the communications to exchange data information before setting up the connection. It can be handled through URL.
- ICE candidate is known for its details of the exchange information for both peers.

First Peer A asks the STUN server for who he is to get IP address and session ID. STUN server gives user A his Symmetric NAT. Then Peer A goes to TURN server to set up an open channel request, offers his SDP to the WebRTC Signal Channel server. The server forwards A's SDP to Peer B, then Peer B sends out an answer SDP to Peer A. Once A gets the answer, he sends out his ICE candidate, to B and B sends his ICE candidate to A. Then B asks for the STUN server to get B's IP. As a result, the channel between A and B is established.

3.4 Benefits of Using WebRTC and WebSocket

The benefits of using WebRTC can be defined in two different ways, one is from the server side, and the other is from the user point of view.

From the server side, we can say that:

- It gives server to deliver a special real-time communication experience [13]. With the usage of WebRTC, every website can be the main control of them. Unlike the traditional Skype or Flash, which is defined by creators, developers will have complete control of WebRTC and the whole system to meet the experience they want this application to achieve. In this way, developers will have the most control of the

system. Furthermore, the experiences will now reflect a new meaning for communication values. The start of the communication can be passed through anywhere, not only the phone call, but any tool that can be evaluated by HTML5.

- WebRTC delivers simplicity, along with reliability and availability.

First, developers have no need to worry about certificates, plug-ins and other things.

The only thing they need to take into account is the browser. Today, the development of the platform matters on the complexity of the OS and so on. But WebRTC eliminates the concern of this part.

Second is client stateless. WebRTC is a stateless implementation. Its input is through browser and the media engine is controlled by the website side server. In this way the complexity of implementation of web application in WebRTC will be significantly dropped. Developer will feel easier by just using the WebRTC API.

- No dependence on service provider. Before WebRTC, people will need to download the client and get certificated, which means pay fees, then use the third party project. With WebRTC, there is no need to download anything. Service providers are those websites, so there is no need for developer to get a certified API to develop in.

Then, from the users' perspectives, the benefits are very similar:

- Website delivers the communication experience. So for any users, once they have the access to the website which has the WebRTC functionality, they can use the communication immediately. Unlike Flash, users will have unique experience depending on the website itself, other than the service provider. In this way, experience of the website can contain the experience of the communication.

- There is no need to download anything. In WebRTC, there is no need to download any clients' applications or plug-ins for any communication experience. With the emergence of tablet, cars, television, the plug-in or client downloading is almost impossible. With WebRTC, any website user visit will no need to download separate activity for communication in their life time.
- There is no registration fee. WebRTC is not a service but a standard. Unlike Skype, any users who use the website which contains WebRTC will not need to pay for extra cost for registration. Users are welcome to use real-time communication at any website which has WebRTC implemented.

3.5 Node js

It is an open source, cross-platform runtime environment for server-side networking applications. It is originally written in JavaScript [15]. It is designed to create real-time websites with push capability as a server side application [14]. It can also push technology over websocket. Furthermore it remains light weight and CPU efficiency by enabling the non-blocking, event-driven I/O. Lastly, it can also be beneficial to use across devices.

3.5.1 Benefits of Node js

The main benefits, which is the most significant advantage of node.js is that it is event-driven with one thread.

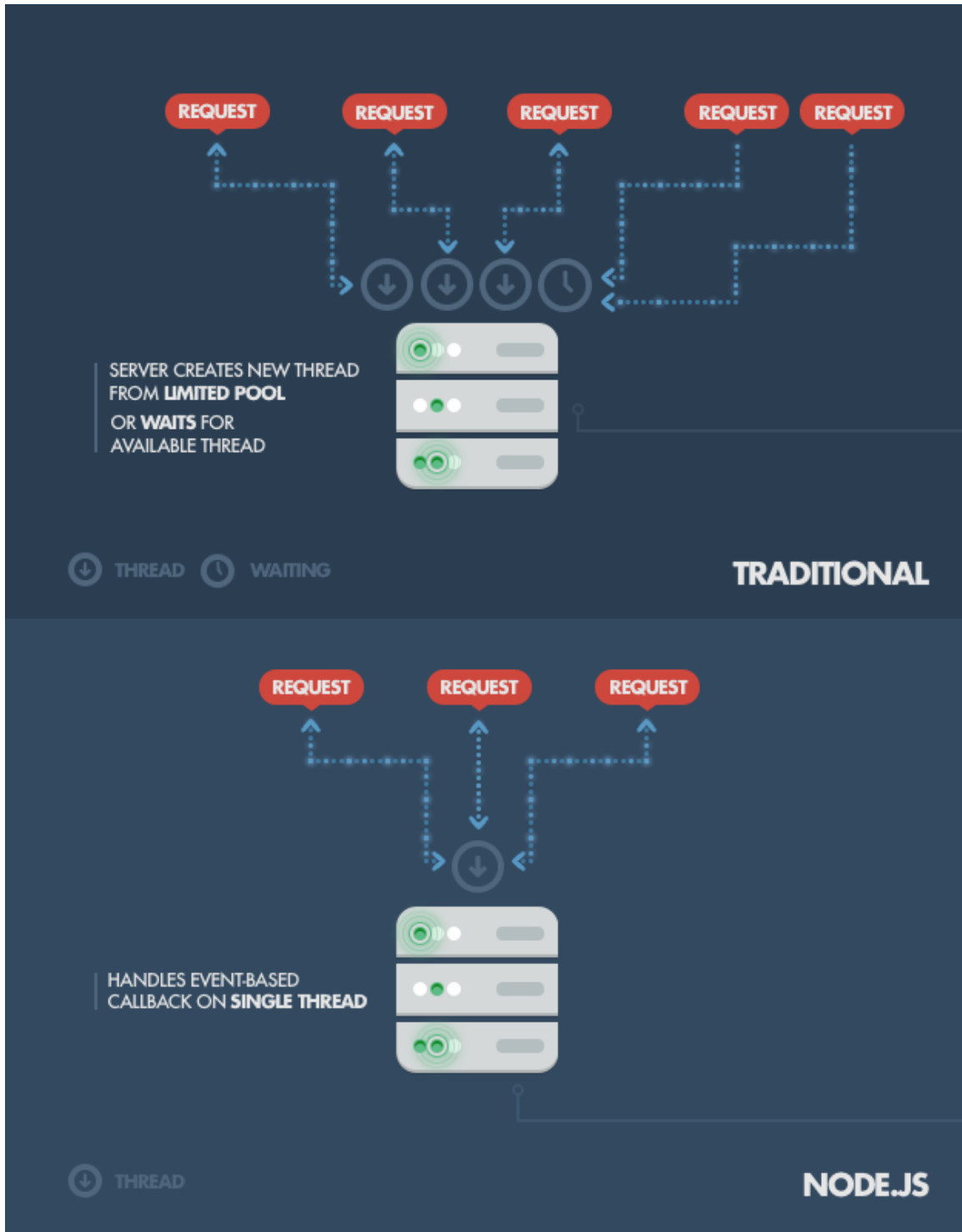


Figure 3-7 Traditional Server and Node.js [14].

From figure 3-7 we can see that traditional web server works in a way that creates different threads for different request on a limited pool. The result is that it will

eventually running out the space of the server RAM. The waiting time is significantly high for high volume of the request.

On the other hand, the node.js gives a solution for reducing the RAM usage by operating on an event-based callback on single-thread. By using this technology, no thread will be delayed and RAM usage will be reduced.

3.5.2 Node.js Use Cases

Since it is single-threaded, it is good to use Node.js on a server-side web application. Developers can write JavaScript from top-to-bottom, if the application does not require high CPU computation [14]. Also people can write a simple html page to be run on top of Node.js.

CHAPTER 4
SYSTEM DESIGN

This section will first discuss the overall architecture of the proposed interactive Multimedia enabled virtual lab learning environment on Vlab Platform. Then it will explain each component in the system and the description of user scenarios.

4.1 Overall Architecture

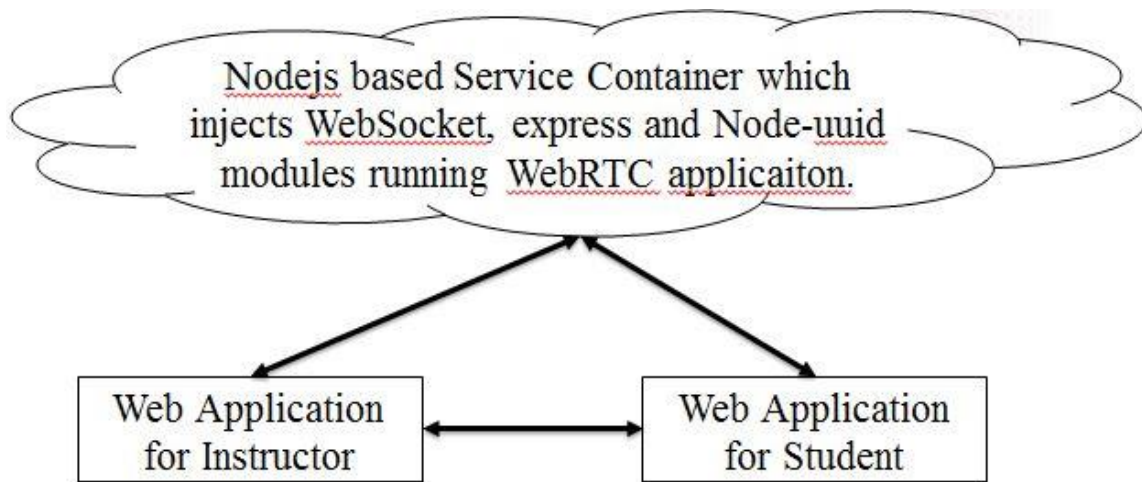


Figure 4-1 Overall System Design.

4.1.1 Multimedia Learning System Backend

The backend of the learning system was developed on top of Node.js. In this case, I use the Node.js as the base service container which injects the WebSocket, Express and Node-uuid modules running over WebRTC application.

4.1.2 Multimedia Learning Web Applications

Before getting into the WebPages, user will be checked their credential again for security purposes. Then they will choose the classroom that they have access, want to get into and go to that specific classroom. Inside the classroom, they will have all access with the WebRTC functionalities and can start the real-time communication right away.

4.2 User Scenarios

This section presents the user scenarios in the design prototype. The scenarios will show how each part works together in the Multimedia learning system, and in the educational way to achieve the purpose of this research.

4.2.1 Authentication

In this section, before getting into the multimedia page, user will need to get into the system to check their credentials. Here is the proposed login page.



The image shows a simple login form layout. On the left side, the text "Username:" is displayed. To its right is a rectangular input field for entering the username. Below the input field, centered horizontally, is a blue button with rounded corners containing the word "Confirm" in white text.

Figure 4-2 Designed Username Check Page.

Figure 4-2 is the page where a user gets to input his/her username and get checked by the LDAP server for security purposes. Once a user entered his/her username, s/he will be rerouted to the page shown below.



The image shows a web form with a light green border. On the left side, the text "Classes you have:" is displayed. To its right is a rectangular text input field with a light green border, containing the placeholder text "Class Name". Below the input field is a blue button with rounded corners and the word "Confirm" written in white text.

Figure 4-3 Designed Room Check Page.

Figure 4-3 checks user room (groups) that a user enters. Once s/he passes the authentication, s/he will join the classroom with his/her username on the list.

4.3 Multimedia Web Page

Here is the look for the page where user gets in with correct credentials and room information.

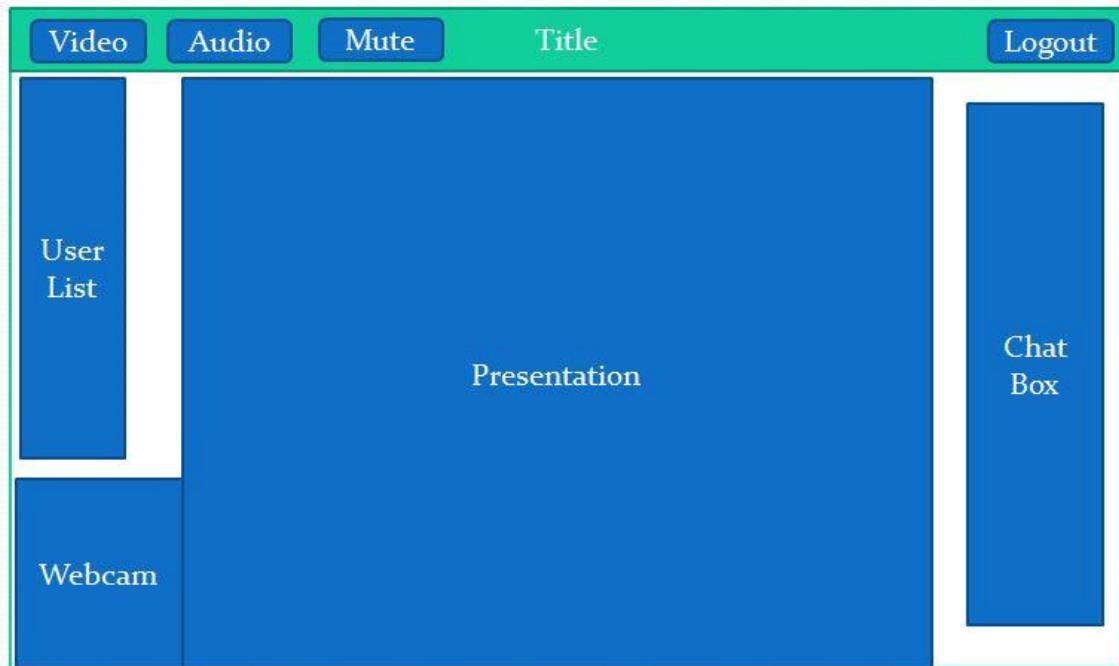


Figure 4-4 Multimedia Web Page (main page).

Figure 4-4 shows the main page for the Multimedia learning website. It can be categorized into several parts, naming: control bar, UserList, WebCam frame, presentation, and chat box.

4.3.1 Control Bar

The Control Bar is where users can control their own video control over the webpage. Here is the design image for Control Bar.



Figure 4-5 Control Bar.

Figure 4-5 shows what is included in the Control Bar. Users can open their local video, which contains both video and audio flag on. Thus in the same room, the other person will be able to see and hear what s/he is talking. Audio on the other hand, only has

the sound but no image. Mute means only the image but no sound. A lot of functionalities can be built on control bar so that users cannot only control their own media, but also other people's.

4.3.2 User List

User list is used to show all active users who join the same room and have ability to have communication with. Here is the idea of User List.



VMs	Users
 	<u>qingyunl</u> tom

Figure 4-6 User List.

Figure 4-6 shows the User List which contains all active users in the session in the same group with current users. VMs column shows the VM we are using for current study room.

4.3.3 Webcam Frame

This section shows the video area where people who joined the room can be seen in the frame for all users.



Figure 4-7 Video Frames.

People who join the room and open their video session will be seen in this area. Here is the way how people can have real-time communication with each other.

4.3.4 Presentation Frame

This part of the web page is used to present the class notes or user's virtual machine, PowerPoint, class notes. It can be used in learning and teaching and switched from video conferencing frame.

4.3.5 Chat List

Place where users who joined the room can share their thoughts and exchange their knowledge of the class.



Figure 4-8 Chat List Area.

Figure 4-8 shows that there are two ways of talking. The first is to use the everyone's tab. Any text sent from this tab will be sent to everyone active in the room. The second, user's click on the username in the User List, it will open up a new tab for that particular user to start a new conversation only among certain users.

4.4 WebRTC Modules Design

In this section, I will discuss about the WebRTC modules that I have designed for specific usage in the system.

4.4.1 One-to-One

One-on-One Modules was used in text conferencing and file sharing.

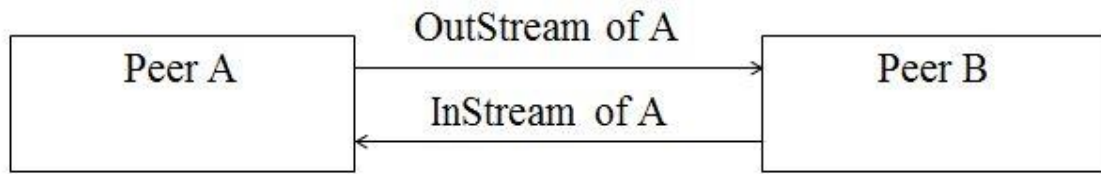


Figure 4-9 One-to-One WebRTC Module.

We can see that there is 1 input stream and 1 output stream for each peer. This module is used in both One-to-One text messaging and file sharing.

4.4.2 Many-to-Many

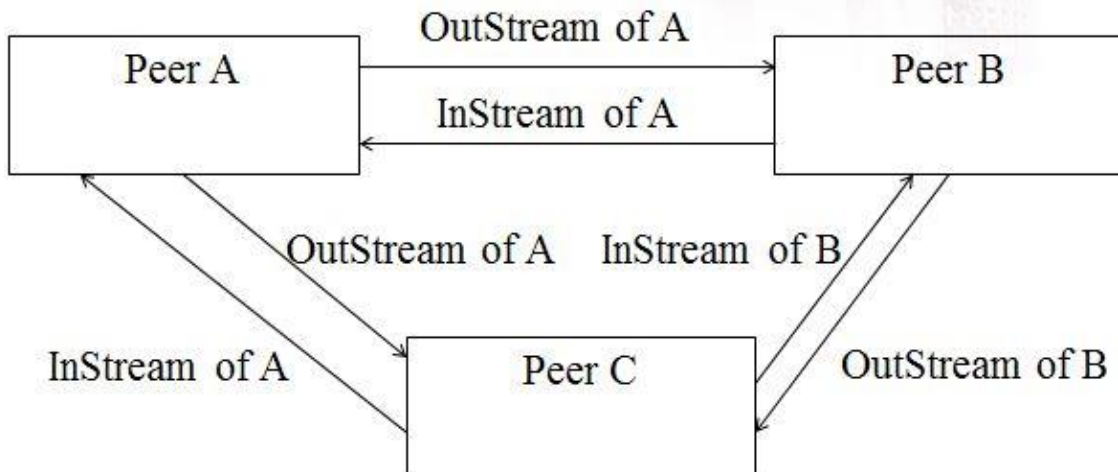


Figure 4-10 Many-to-Many WebRTC Module.

Many-to-Many WebRTC Module was used in video conferencing for this system. As shown in figure 4-10, if we have 3 users, then there will be 2 input streams and 2 output streams for each peer.

4.4.3 One-to-Many

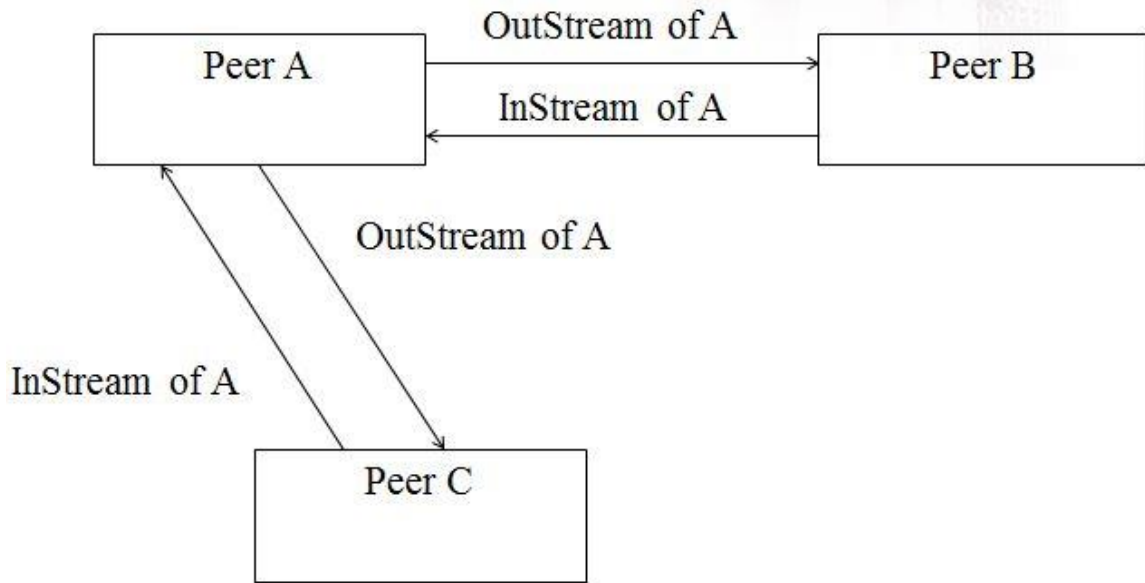


Figure 4-11 One-to-Many WebRTC Module.

Assume there are three users, peer A, B and C. If peer A is broadcasting to B and C, then he will create two output stream to B and C and receive two input stream from B and C. This module is used in broadcasting text conferencing; file sharing and Peer list add/remove update. This modules was used in Video conferencing. Since originally WebRTC only support peer to peer, I have to come up with thie One-to-Many to solve the bandwidth issues.

CHAPTER 5

SYSTEM IMPLEMENTATION

In this section I will demonstrate the implementation of an interactive multimedia enabled virtual lab learning environment on the Vlab Platform.

5.1 Server

On the one hand we need to run the html web page for the multimedia learning system, on the other hand, we also need to run a server as a signal exchange center for the browser and server.

5.1.1 Server.js

First we need to have a server running to translate all the messages and access controls.

```
var express = require('express');
var app = express();
var server = require('http').createServer(app);
var SkyRTC = require('skyrtc').listen(server);
var path = require("path");
var port = process.env.PORT || 3000;
server.listen(port);
app.use(express.static(path.join(__dirname, 'public')));
app.get('/', function(req, res) {
  res.sendFile(__dirname + '/index.html');
```



```
});
```

This is the part of the overall set up for the system, which listens for the port and makes all configurations for the websocket and etc.

The server.js also contains debugging sentences for users to debug each commit sent by other user. Here is one example in server.js

```
SkyRTC.rtc.on('new_connect', function(socket) {  
    console.log('Create New Connection');  
});
```

5.1.2 Node Js

Node.js is used for running the server.js. First we need to run the command: “npm install” in order to install all necessary modules required to run the application by Node.js. Then we go the server.js file directory to run the command: “node server.js”.

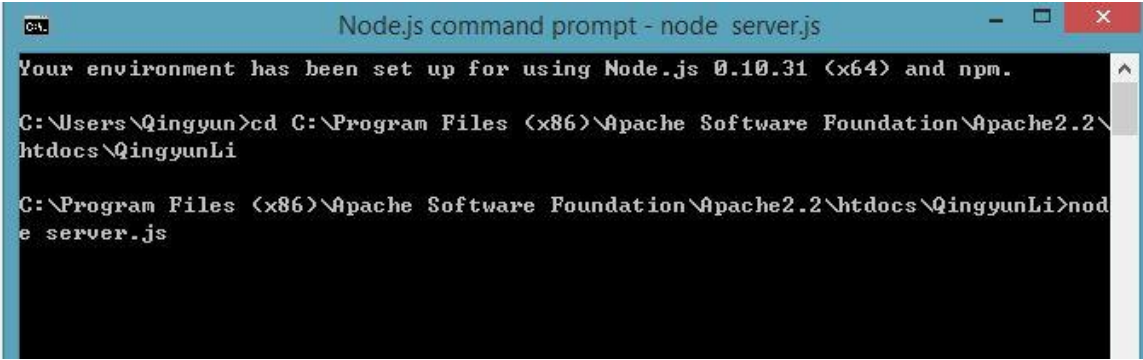
A screenshot of a Windows command prompt window titled "Node.js command prompt - node server.js". The window has a blue title bar with standard Windows window controls (minimize, maximize, close). The command prompt shows the following text: "Your environment has been set up for using Node.js 0.10.31 (x64) and npm." followed by a directory change command: "C:\Users\Qingyun>cd C:\Program Files (x86)\Apache Software Foundation\Apache2.2\htdocs\QingyunLi". The next line shows the execution of the server: "C:\Program Files (x86)\Apache Software Foundation\Apache2.2\htdocs\QingyunLi>node server.js". The background of the command prompt is black with white text.

Figure 5-1 Server.js Command In Node.js.

As we can see from Figure 5-1, server.js is up and running.

5.2 Authentication

Here is the first place where a user enters the system and will see what is going to happen next. In this section, a user will be authenticated and redirected to the correct room with his/her username.

5.2.1 Log In

On this page, the user will enter a username.

The figure displays two sequential screenshots of a login page. Both screenshots feature the heading "Please sign in" above a text input field, a "Remember me" checkbox, and a "Submit" button. In the top screenshot, the input field contains the placeholder text "Username". In the bottom screenshot, the input field contains the text "Tom".

Figure 5-2 Log In Page.

Figure 5-2 shows the log in page with username entered before and after. Once it gets checked, the user will be redirected to the room check page.

5.2.2 Room Check Page

On this page, the user will enter the room and be validated by the LDAP server.

The screenshot shows a form titled "Please Enter Room Name". It includes a text input field containing "CSE450", a "Remember me" checkbox, and a "Submit" button.

Figure 5-3 Room Check Page

Once the user gets checked into their room and the username matches the database, they will be redirected to the main page.

5.3 Multimedia Main Page Use Cases

This page is where users communicate and interact together.

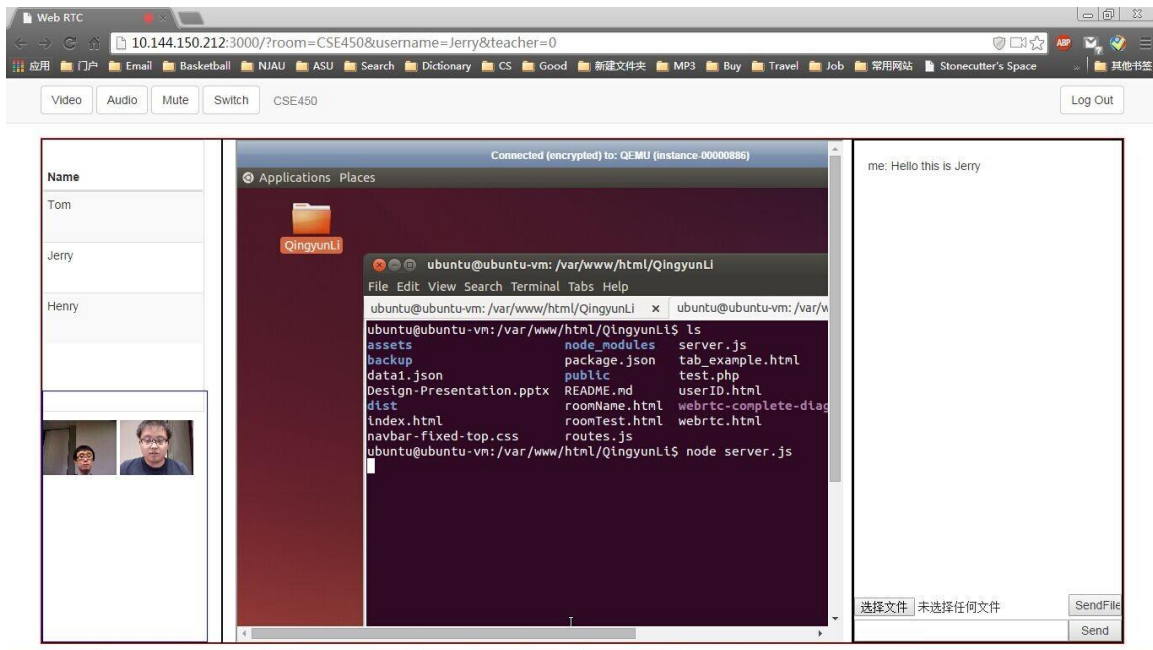


Figure 5-4 Multimedia Main Page.

Figure 5-4 shows the main page for the system. After authentication, the user will be redirected to this page and start real-time communication.

5.3.1 Video Use Cases

In the control bar, the user can choose either video with audio, only audio or only video.

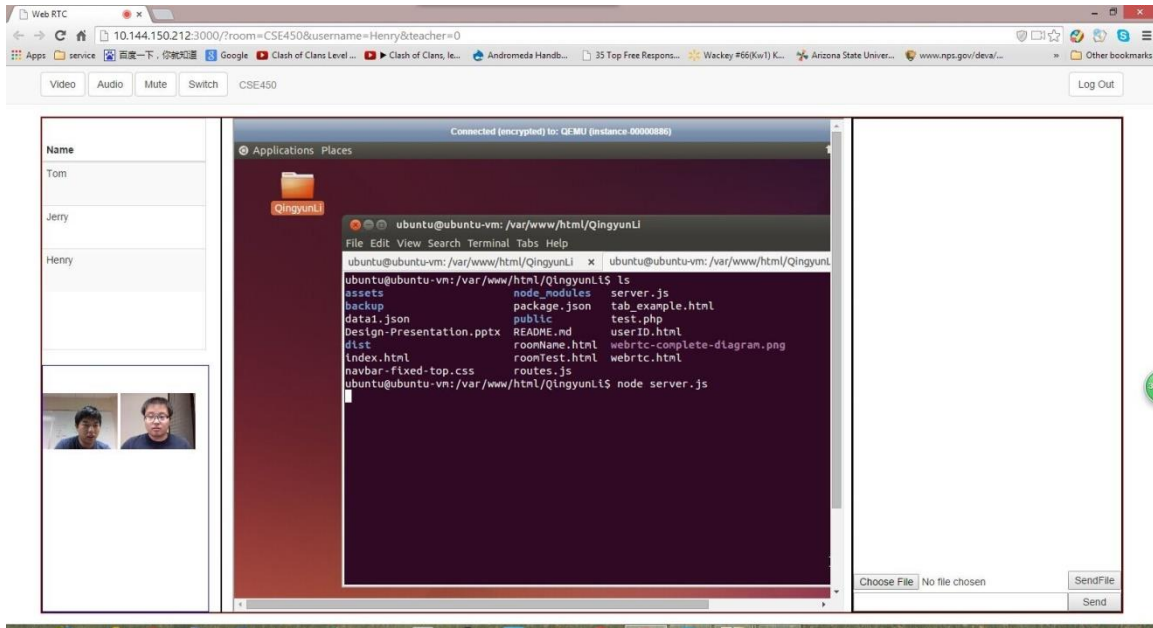


Figure 5-5 Video Chatting.

Figure 5-5 shows the case where users Jerry and Tom joined the room and opened up the video chatting function. The two pictures show two different users using two different computers to do video communication. Users can choose the audio only or video only chatting fashion.

5.3.2 Texting Use Cases

This main page also has functionality for text chat.

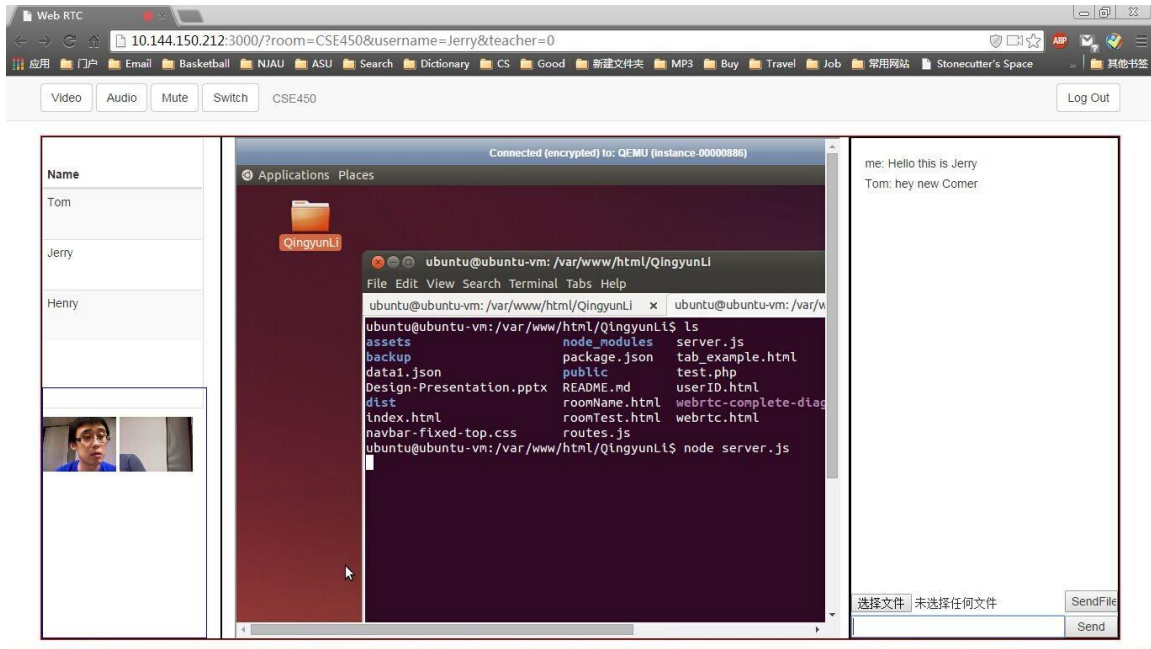


Figure 5-6 Text Communication.

Figure 5-8 shows the text communication between users Tom, Jerry and Henry. Jerry sent out the text message and Tom received the message from Jerry.

5.3.3 File Sharing Use Cases

The file sharing functionality gives both students and instructors the opportunity to share class notes during the meeting.

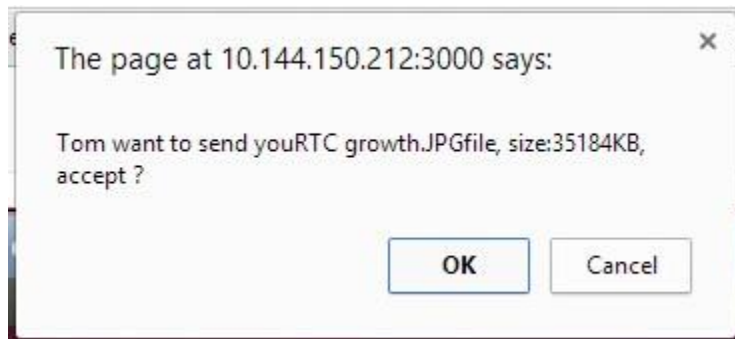


Figure 5-7 File Choose.

Figure 5-9 shows the user chooses a file from his own PC and ready to send it. Once the button is pressed, the notification window will pop up in the receiver's browser. Now the receiver has the ability to either accept or reject the file.

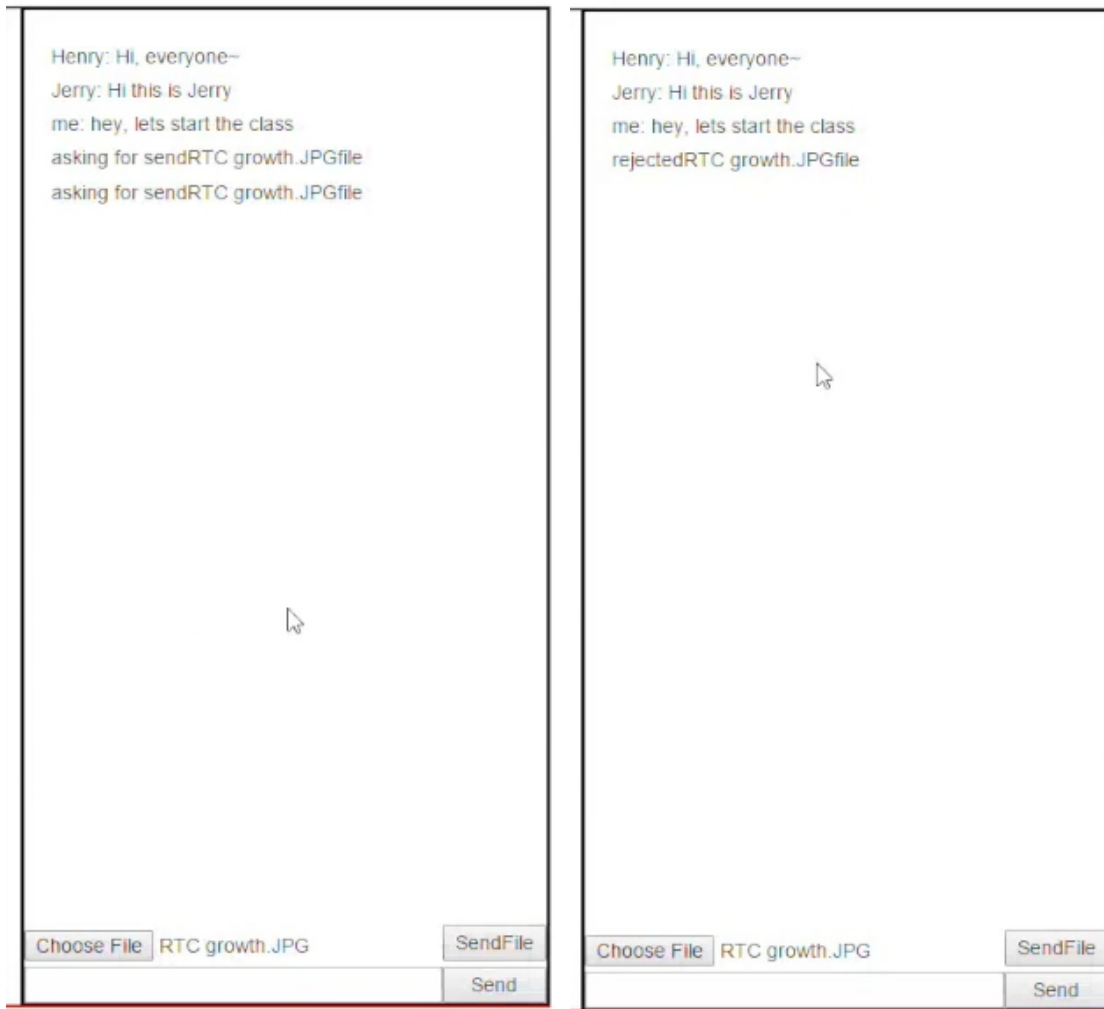


Figure 5-8 File Receive Situation.

Figure 5-10 shows two conditions when doing the file sharing. First, Tom sends out the file notification and Jerry rejected. The rejected message will be shown on the messaging block. Second, Jerry accepts the file, then file transferring start right after the click. Message of completion of sending file will show up in Tom's message block and

receiving percentage will be shown on Jerry's block. This is how File sharing works in this page.

5.3.4 Virtual Machine Sharing

This app's goal is to allow teachers and students enjoying video conferencing while studying, in which students use the command line command to retrieve the VM token info. Figure below shows user using the shared VM during the class, in current design all students in the class room will have access with the VM.

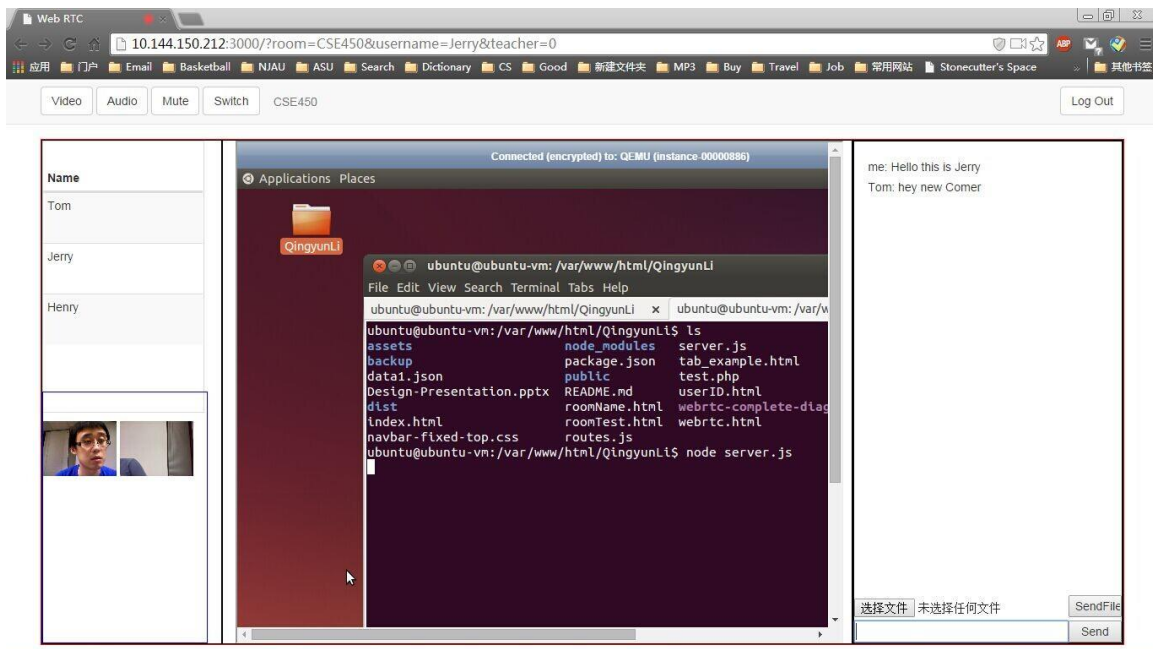


Figure 5-9 VM user scenarios.

5.3.5 GUI redesign

It is under development that the GUI of the Website can be redesigned. Right now only the switch between the user video and VM control frame is functional.



Figure 5-10 GUI change.

The above figure shows the change in the GUI.

CHAPTER 6

EVALUATION

In this section I will show the improvements and creativity that the new system has and why the new system is better than the old ones.

6.1 System Advantages over Existing System

In this section we will talk about the advantages that the system has over the existing online learning system, Bigbluebutton.

6.1.1 Multiple Peer Video Conferencing

The system design has its advantages over the existing design. In the existing design the real-time communication part is always one-on-one. People can only see one other person in the same room. This is against the purpose of multimedia online learning. Participants in the room want to share their thoughts but the video functionalities in existing tools do not allow users to do that. But in the new system, we can have multiple users can participant in video conferencing; people will have the ability to talk to each other and give them feedback right away.

6.1.2 File Sharing

In the existing tool Bigbluebutton created, they only have fundamental methods for text chatting, but there is no other functionality for things like file sharing. During real-time communication, people may want to share something during the class, but the old

system does not allow users to do that. In the new system, users can share their files to other people, both with one-to-one and one-to-many styles.

6.1.3 License Free

Most of the existing systems are developing online learning system on top of licensed tools, which means the tools are good but also very expensive. For college education, this high expense is not acceptable. On the other hand, WebRTC is completely open source and licensed free. Hence developers and users will not need to pay anything to Google for licensing, which saves schools a lot of money on development.

6.2 Improvements on Bandwidth

The development has been improved and the bandwidth usage has been significantly lowered. To see how it affects the system, we need to see it under different circumstances.

6.2.1 One-on-one Video/Audio Conferencing

In this scenario, there is a maximum of 4 RTP ports open for each user who is using the Video/Audio conferencing with another user.

User/Peer	RTP Ports	Usage
A	One RTP port for outgoing video	1MB
A	One RTP Port for incoming video	1MB
A	One RTP port for outgoing audio	1MB
A	One RTP port for incoming audio	1MB

Table 6-1 One-to-One Video Sharing Bandwidth Usage.

According to the table, we can see that each RTP port is using 1MB bandwidth. Thus, in total there is 4 MB bandwidth usage for each peer, which is acceptable for one-to-one communication.

6.2.2 Many-to-Many Video/Audio Conferencing

One-to-One video sharing bandwidth is acceptable, but it will not work in Many-to-Many case. Here shows why: For example, suppose we have 8 users using video conferencing simultaneously.

User	RTP ports	Bandwidth Usage
A	8 RTP ports for incoming video	8 MB
A	8 RTP ports for outgoing video	8MB
A	8 RTP ports for incoming audio	8MB
A	8 RTP ports for outgoing audio	8MB

Table 6-2 Many-to-Many Video Sharing Bandwidth Usage.

As we can see from Table 6-2, if we have 10 peers, the bandwidth for each user will be 40 MB. That is too high for users at home using the internet for online learning.

6.2.3 One-to-Many Video/Audio Conferencing

As is shown above, the Many-to-Many model is not feasible in our case since our instructors and students mainly study online at home, and their computers or devices can hardly meet the heavy requirements that Many-to-Many induces. To beat the above challenge, I implemented my own design, the One-to-Many module.

Given the example of 8 people, in this module, the instructor has 32 income and outcome RTP and students will only have 4 RTP since they only maintain a single connection with the instructor.

User	RTP ports	Bandwidth Usage
Instructor	8 RTP ports for incoming video	8 MB
Instructor	8 RTP ports for outgoing video	8 MB
Instructor	8 RTP ports for incoming audio	8 MB
Instructor	8 RTP ports for outgoing audio	8 MB
Student A	One RTP port for outgoing video	1MB
Student A	One RTP port for outgoing video	1MB
Student A	One RTP port for outgoing video	1MB
Student A	One RTP port for outgoing video	1MB

Table 6-3 One-to-Many Video Sharing Bandwidth Usage.

As we can see, using one-to-many video sharing, each student will remain at 4 MB usage and only the instructor will have the large 40 MB bandwidth usage. Figure 6-1 shows the bandwidth usage between the original WebRTC many-to-many and a modified One-to-Many module.

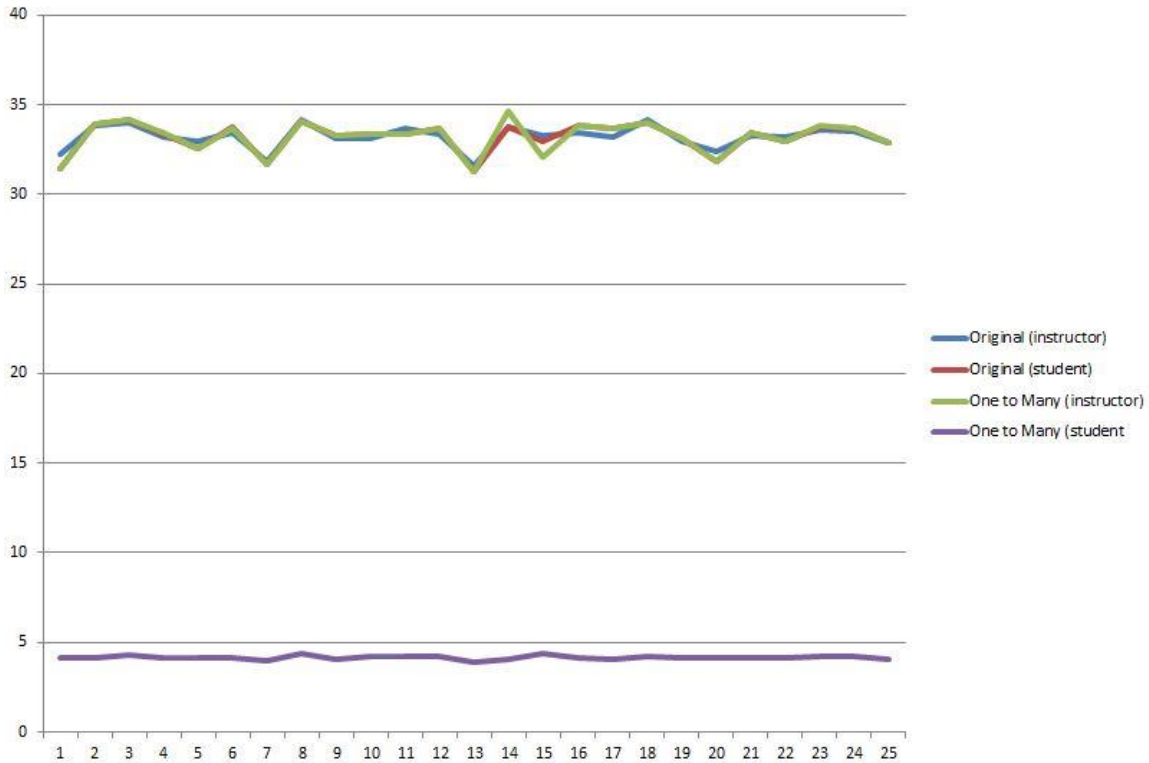


Figure 6-1 many-to-many vs One-to-many bandwidth usage.

As we can see from the above figure, if we have 8 peers, including one instructor and 7 students, the bandwidth usage for the many-to-many module will use almost 35 MB for both the instructor and the students. After the alteration, the usage for the one-to-many student becomes less than 5 MB, which is highly acceptable. However, the instructor's bandwidth is still too high to maintain. We came up with the following solution.

6.2.4 Solutions for Scalability Bandwidth Issues

In order to solve the bandwidth problem that caused by One-to-many real time video communication, we can up with the solution to adjust the bandwidth by the beginning when the web page was loaded. In another word, we inject bandwidth attributes ($b=AS$) in the session description.

```

audio = 50;
video = 256;

function bandwidthSet(sdp) {
    sdp = sdp.replace(/a=mid:audio\r\n/g, 'a=mid:audio\r\nb=AS:' + audioBandwidth
+ '\r\n');
    sdp = sdp.replace(/a=mid:video\r\n/g, 'a=mid:video\r\nb=AS:' + videoBandwidth
+ '\r\n');
    return sdp;
}

```

This is to set up the video bandwidth to 256 kilobits per second and audio to 50 by the beginning. To create peer connection, we set up these values.

```

pcCreateOfferCbGen = function(pc, socketId) {
    return function(session_desc) {
        session_desc.sdp = setBandwidth(session_desc.sdp);
        pc.setLocalDescription(session_desc);pc.setRemoteDescription(new
nativeRTCSessionDescription(sdp));
        pc.createAnswer(function(session_desc) {
            session_desc.sdp = setBandwidth(session_desc.sdp);
            pc.setLocalDescription(session_desc);

```

By setting up the peer connection we can control peer bandwidth by the very beginning.

On this many-to-many case again, if we have 8 peers, bandwidth usage for each peer becomes around 5 MB. Before doing this, I used the Yahoo API boomerang to

measure the network bandwidth and latency and then doing the calculation on the front end.

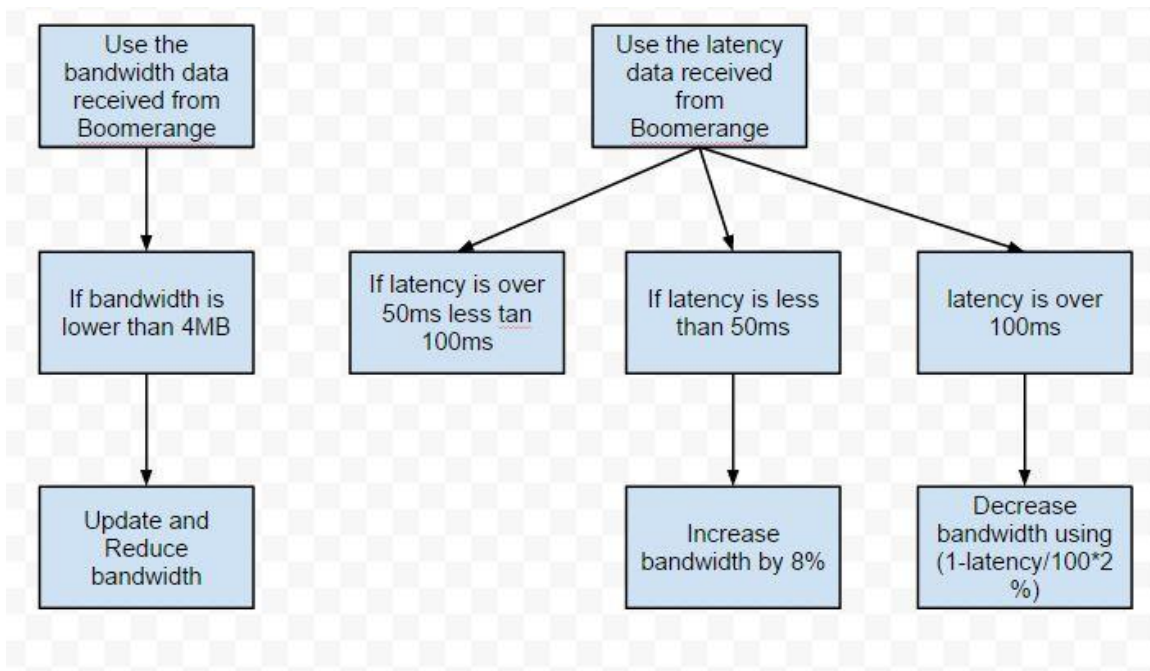


Figure 6-2 Frontend Bandwidth control algorithm.

Using the above algorithm before sending the video signal to peers, this will help estimate the bandwidth usage by frontend. Here is the result data for testing on the new bandwidth control solution.

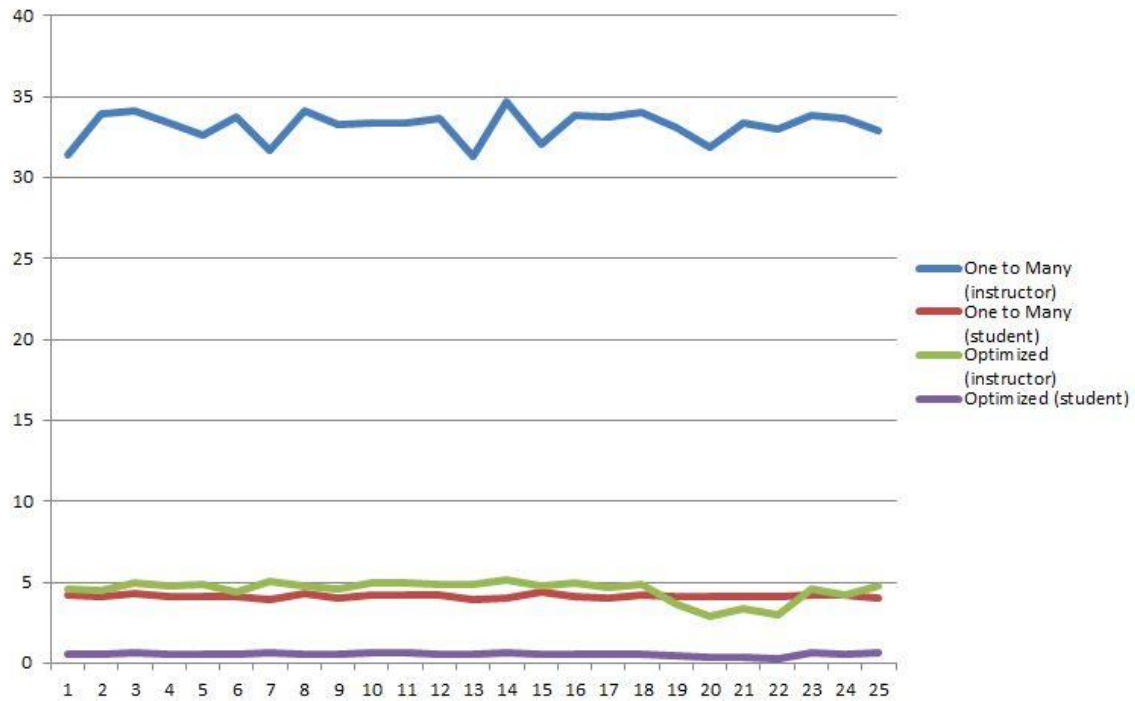


Figure 6-3 One-to-many vs Bandwidth control solution.

From the above figure we can see that student bandwidth goes down to less than 1 MB. This is highly recommended that user can use mobile devices to join the study. Also we can see that instructor’s bandwidth usage has been dropped significantly less than 5 MB. This is the solution for controlling the WebRTC bandwidth.

6.3 Improvements on Function Controls

After the second phase of the development, I have come up with some functionality updates on this system.

6.3.1 Modules Separation

In the old design, all video, virtual lab, text chat and file sharing are combined in one module in one web page. After the change, I have separated them into different

modules. One is video conferencing combined with virtual lab, the other one is chat together with file sharing.



Figure 6-4 Three Modules.

Right now users can hide and show the video conferencing, virtual lab and chat box as they wish. These functions are controlled by users themselves.

6.3.2 Control Functions for Single User

In order to make this application a better usability, we added more control functions to single user.



Figure 6-5 User List Buttons.

From the above figure we can see the user list has added two buttons, display and switch. Display is that when current user wants to see particular person's video, he can click the display button to display the person's video. Once the user is done watching, he can flick display again to hide the video for the person. This button is disabled if the user

does not open the video session and only open user's chat function. Switch button switches the particular user's video with the virtual lab frame. If current user wants to see particular user in a bigger frame, he can click this button to switch the person's video to the virtual lab frame. Thus, the current user will have the view for the particular person in a bigger frame.

6.3.2 Control Functions for Single Video Frame

After redesign the GUI of the page, we put the control function for other users to their own videos. Right now if the current user wants to mute the particular user, s/he can press the mute button on that video frame. The video frame can also become full screen now after the reprogram.

CHAPTER 7

CONCLUSION

In this chapter, the summary of the current work and the plan of future work will be discussed.

7.1 Conclusion of Current Work

The research has been done in this thesis was trying to move WebRTC into Vlab platform to form a new platform for instructors and students in Arizona State University with an interactive multimedia enabled virtual lab learning environment with ability to do real-time communication with audio and text in the cloud environment. To achieve this, this thesis attempt to implement the WebRTC into the website with interacts with Vlab, by using the vlab virtual machine and WebRTC together. In order for all student to have the opportunity to play around with the application, this website implementation will be shown to all students who takes interaction needed classes. To ensure the e-learning, we used WebRTC into the system to replace the old fashioned flash. In the environment we have audio, video, text communication and file sharing functionalities. These modules are separated and functional as plug-ins, which can be used by any other web applications. The system does not need any certificate, which means it is totally free to use by Arizona State University. The environment will serve as a service to serve all faculties working at ASU.

7.2 Future Work

Some extension work will need to be done. Application that uses modules of the multimedia learning system will need to be set up by students in our lab. The next goal to achieve for this environment is to set up more functionality which includes the UI design, personalized website, interaction with virtual machine in Vlab from UI itself and so on. It also need to be put into the cloud system to test the stability. Mobile side test will need to be done quickly in order to help people access our system by mobile devices.

REFERENCES

- [1] Lohmosavi, Vajihe, E-learning Ecosystem Based on Service-Oriented Cloud Computing Architecture, ISBN 978-1-4673-6489-8
- [2] "Using WebRTC to Transform Education." Using WebRTC to Transform Education. N.p., n.d. Web. 21 Oct. 2014.
- [3] WebRTC page from Wikipedia, <http://en.wikipedia.org/wiki/WebRTC>
- [4] A Kadne, Vlab: A Cloud based Resource and Service Sharing Platform for Computer and Network Security Education, Master Thesis, 2010.
- [5] Degree programs in Software Engineering, <http://www.gradschools.com/search-programs/software-engineering>
- [6] "WebRTC vs. Flash - Not Much of a Competition." SaleMove. N.p., n.d. Web. 21 Oct. 2014. <http://blog.salemove.com/webrtc-vs-flash-not-much-of-a-competition/>
- [7] "Webrtc Research." -. N.p., n.d. Web. 21 Oct. 2014. <http://blog.csdn.net/zz316316/article/details/8160679>
- [8] "Hypertext Transfer Protocol." Wikipedia. Wikimedia Foundation, 20 Oct. 2014. Web. 21 Oct. 2014.
- [9] Liang, W.h., and E. Oset. "B₀ and B_{s0} Decays into J/ψ F₀(980) and J/ψ F₀(500) and the Nature of the Scalar Resonances." Physics Letters B 737 (2014): 70-74. Web.
- [10] Web article for WebSocket from <http://zh.wikipedia.org/wiki/WebSocket>
- [11] "HTML5 Web Sockets: A Quantum Leap in Scalability for the Web." WebSocket.org. N.p., n.d. Web. 21 Oct. 2014. <http://www.websocket.org/quantum.html>
- [12] "Mozilla Hacks – the Web Developer Blog." Mozilla Hacks the Web Developer Blog RSS. N.p., n.d. Web. 21 Oct. 2014. <https://hacks.mozilla.org/category/webrtc/as/complete/>
- [13] "The Benefits of WebRTC." The Benefits of WebRTC. N.p., n.d. Web. 21 Oct. 2014. <http://www.webrtcworld.com/topics/from-the-experts/articles/319037-benefits-webrtc.htm>
- [14] "Why The Hell Would I Use Node.js? A Case-by-Case Tutorial." Toptal Engineering Blog. N.p., n.d. Web. 21 Oct. 2014. <http://www.toptal.com/nodejs/why-the-hell-would-i-use-node-js>

[15] "Node.js." Wikipedia. Wikimedia Foundation, 18 Oct. 2014. Web. 21 Oct. 2014.
<http://en.wikipedia.org/wiki/Node.js>

[16] "WebRTC auto bandwidth control" Toptal Engineering Blog. N.p., n.d.
http://blog.csdn.net/mahout_xb/article/details/8473833