

**INTERVIEW VIDEO CONFERENCING SYSTEM ASSISTED BY
ARTIFICIAL INTELLIGENCE SIGN LANGUAGE INTERPRETER**

By

**MUHAMMAD SHAFIQ BIN AHMAD RAZMAN
2017724867**

This thesis was prepared under supervision of project supervisor, Nor Azylia Binti Ahmad Azam. It was submitted to the Faculty of Computer and Mathematical Sciences and it is a part of fulfilment of the requirements for the degree of Bachelor of Computer Science (Hons.) Data Communication and Networking.

Approved by

.....
NOR AZYLIA BINTI AHMAD AZAM
Project Supervisor

JANUARY 2, 2020

STUDENT DECLARATION

I certify that this thesis and project is the result of my own work. Any idea, findings or quotation from the work of other people from the journals and articles, published or otherwise are fully acknowledged accordance with the standard of referring regarding to the practices of the discipline.

.....
MUHAMMAD SHAFIQ BIN AHMAD RAZMAN
2017724867

JANUARY 2, 2020

ACKNOWLEDGEMENT

Praises to Allah because of His blessings, I was able to complete this thesis within the time given. First, I would like to thank to my supervisor, Madam Nor Azylia Binti Ahmad Azam, my CSP 600 lecturer, Madam Noor Ashitah Binti Abu Othman and my CSP600 lecturer, Sir Hafifi Supir for their guidance to make this Final Year Project report success.

Next, I also would like to give a special thanks to my family for always being supportive and taking concern in this project.

Lastly, I would to thank to all my friends for spending their time to lend me a help in completing this thesis. I am so very grateful to have the advice from these people in order to make this project successful.

ABSTRACT

Video Conferencing is a common system that we used. Through it we can communicate with other people face to face with real time interactions in a distance. However, deaf people are having trouble with communicating with other people in a distance who did not understand sign language. There are several Artificial Intelligence systems that can translate sign language into text. However, it is not integrated with the video conference yet. Hence, this project is implemented to integrate the two systems, video conference and AI system that translate sign language into text in hope that it will be enhanced the deaf people community live.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1	Background of Study	1
1.1.1	Video Conference.....	1
1.1.2	Artificial Intelligence	2
1.1.3	Sign Language.....	2
1.2	Problem Statement	2
1.3	Objectives	3
1.4	Project Scope	4
1.4.1	Users.....	4
1.4.2	Sign Language.....	4
1.4.3	Software	4
1.4.4	Application Programmable Interfaces (APIs).....	4
1.4.5	Hardware	4
1.5	Significance	4
1.6	Summary	5

CHAPTER 2: LITERATURE REVIEW

2.1	Types of Video Conference Systems	6
2.1.1	Telepresence Video Conferencing	6
2.1.2	Room-Based Video Conferencing	7
2.1.3	Desktop Video Conferencing.....	8
2.2	Network Topologies of Multiparty Video Conferencing.....	9
2.2.1	Mesh Network Topologies.....	9
2.2.2	Star Network Topologies	10
2.3	Video Conference Protocols.....	11
2.3.1	Real-Time Transport Protocol (RTP).....	12

2.3.2	Session Initiation Protocol (SIP).....	12
2.3.3	Real Time Streaming Protocol (RTSP).....	12
2.3.4	Real Time Messaging Protocol (RTMP)	13
2.4	STUN, TURN, and Signalling Servers	13
2.5	WebRTC.....	14
2.6	WebRTC-based Video Conferencing API	14
2.7	OpenTok.....	15
2.8	Jitsi.....	16
2.9	Video Codec	16
2.9.1	VP8.....	16
2.9.2	VP9.....	17
2.10	Gesture Recognition	17
2.11	Machine Learning.....	17
2.12	Deep Learning.....	18
2.12.1	Convolutional Neural Network (CNN).....	18
2.12.2	K-Nearest Neighbour (KNN) Algorithm	18
2.13	Summary.....	19

CHAPTER 3: METHODOLOGY

3.1	Methodology of the Project.....	20
3.2	Information Gathering.....	21
3.3	Requirement Analysis	22
3.3.1	Hardware Requirements.....	22
3.3.2	Software Requirements	23
3.4	System Design.....	24
3.4.1	User Flowchart.....	25
3.4.2	Topology	26
3.5	Development and Implementation	26

3.5.1	Database Setup	26
3.5.2	Code Segment of Firebase Implementation	27
3.5.3	Code segment of OpenTok.js Implementation.....	28
3.5.4	Code segment of Tensorflow.js KNN Implementation	29
3.6	Summary	29
CHAPTER 4: EVALUATION		
4.1	Evaluation Scope	30
4.2	Network Evaluation.....	30
4.2.1	Hosting Server.....	30
4.3	Network Evaluation Environment.....	31
4.3.1	Network Latency Evaluation	31
4.3.2	Packet Loss Evaluation	34
4.3.3	Bitrate Evaluation.....	36
4.3.4	Summary of Findings and Analysis of Network Evaluation	37
4.4	KNN Model Accuracy Evaluation	38
4.5	Functionality Test.....	41
CHAPTER 5: LIMITATIONS, RECOMMENDATIONS AND CONCLUSION		
5.1	Limitations and Recommendation.....	45
5.2	Conclusion.....	45

LIST OF FIGURES

Figure	Page
Figure 2.1 Telepresence Topology.....	7
Figure 2.2 Room Based Topology	8
Figure 2.3 Point to Point Desktop Video Conference Topology	9
Figure 2.4 Mesh Network Topology	10
Figure 2.5 Star Network Topology	11
Figure 2.6 Popular Video Conference Protocol	11
Figure 2.7 Comparison of delay(ms) between RTSP/H263 and RTMP/H264	13
Figure 2.8 OpenTok Client and Server Flow	15
Figure 2.9 KNN Graph Visualization	19
Figure 3.1 Project Methodology	20
Figure 3.2 User Flowchart.....	25
Figure 3.3 Network Topology	26
Figure 3.4 Firebase Cloud Firestore Structure	27
Figure 3.5 Firebase Implementation.....	27
Figure 3.6 Getting User Role	28
Figure 3.7 Code Segment of OpenTok	28
Figure 3.8 Training Image Function	29
Figure 4.1 Result of Ping Command.....	30
Figure 4.2 User Information for the Session.....	31
Figure 4.3 Latency (ms) for User 1	32
Figure 4.4 Latency (ms) for User 2	32
Figure 4.5 Latency (ms) for User 3	33
Figure 4.6 Latency (ms) for User 4	33
Figure 4.7 Packet Loss (%) for User 1	34
Figure 4.8 Packet Loss (%) for User 2	34
Figure 4.9 Packet Loss (%) for User 3	35
Figure 4.10 Packet Loss (%) for User 4	35
Figure 4.11 Bitrate (kbps) for User 1	36
Figure 4.12 Bitrate (kbps) for User 2	36
Figure 4.13 Bitrate (kbps) for User 3	37
Figure 4.14 Bitrate (kbps) for User 4	37

LIST OF TABLES

Table	Page
Table 3.1 Descriptions of the Phases.....	21
Table 3.2 Hardware Requirements	23
Table 3.3 Software Requirements	24
Table 4.1 Maximum Value of Each Network Performance Parameter.....	37
Table 4.2 KNN Model Accuracy Test.....	38
Table 4.3 Speech Synthesis Testing	41
Table 4.4 Functionality Test.....	41
Table 5.1 Limitations and Recommendations	45

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
API	Application Programming Interface
ASL	American Sign Language
CNN	Convolutional Neural Network
DLTS	Data Transport Layer Security
HTTP	Hypertext Transfer Protocol
KNN	K-Nearest Neighbour
MCU	Multipoint Control Unit
MPVC	Multiple Peers Video Conference
P2P	Peer-to-Peer
RTMP	Real Time Messaging Protocol
RTP	Real Time Protocol
RTSP	Real Time Streaming Protocol
SIP	Session Initiation Protocol
SRTP	Secure Real Time Protocol
STUN	Session Transversal Utilities for NAT
WebRTC	Web Real Time Communication
XMPP	Extensible Messaging and Presence Protocol

CHAPTER 1

INTRODUCTION

This chapter provides the understanding about the project's overview. The project background of the study, problem statement, objectives, scope and significant will be described in detail in this chapter. This chapter also gives an overview of the expected outcome of the project. Besides, it also helps to manage the ideas and flows of the project in conducting out researches and writing conclusions.

1.1 Background of Study

Today, the uses of video conferencing among various platforms rise from years to years. Cisco has predicted that by 2021, video type traffic will be more than 80% of all Internet traffic. While IP video traffic is going to be account for 82% of all IP traffic globally for both business and consumer by 2022, up from 75% in 2017. Global IP video traffic between 2017 and 2022 will boost four times at 29% CAGR (Cisco Visual Networking Index: Forecast and Trends, 2017–2022 White Paper, 2019). Therefore, with the high percentage of video traffic type, a video conference system must be reliable to accommodate the rising demand from the users.

Artificial Intelligence, on the other hand, is a branch of computer science. It tries to understand the nature of intelligence and create a new intelligence model that can communicate equally with human intelligence (Lawrence, Gonzalez & Harris, 2016). Artificial Intelligence is also widely understood as smart machine science and engineering, particularly smart computer programs (Garichev & Vedyakhin, 2018).

1.1.1 Video Conference

Video conference refers to a virtual meeting between two or more parties at different locations by using a computer network. Several parties can share a live camera image of themselves while they talk. The uses of video

conferencing include holding routine meetings, negotiating business deals and interviewing job candidates. The video conferencing systems are playing a crucial role in communicating the decisions and the work progress between the teams and the organizational hierarchy of the large companies (Feng & Wang, 2015).

1.1.2 Artificial Intelligence

Artificial intelligent (AI) is the simulation of human intelligence processes by machines, especially in computer systems. In simpler words, it makes a system to behave, react and think like a human. It makes machines learn from experience, adjust to new inputs and perform a human-like task. Machines are trained to accomplish specific tasks by processing large amounts of data and recognizing patterns in the data, which means that it heavily relies on the data itself. There are many uses and applications of AI. Usages of AI are expert systems, speech recognition and machine vision. With those applications of AI, it grows every day at a furious rate. According to Adobe in 2018, only 15% of enterprises are using AI as of today, but 31% are expected to add it over the coming 12 months.

1.1.3 Sign Language

People communicate with each other to exchange information, deliver messages, and express their thoughts and feelings. However, people with deafness have difficulties in communicating with others. They usually use sign language to communicate. The National Centre for Health Statistics of America estimates that 28 million Americans (about 10% of the population) have some degree of hearing loss. The natural language of around 500,000 deaf people in the US and Canada is American Sign Language (ASL). However, sign languages are not universal, and they are not mutually intelligible (David, 2018).

1.2 Problem Statement

People with hearing and speaking disabilities use sign language as a way of communicating with each other, but not every person knows sign language, hence, the result is lack of communication and isolation (Ahmed, Idrees, Abideen, Mumtaz & Khalique, 2016). Furthermore, they also have difficulties

in communicating with each other in a long distance. One of the solutions proposed by (Prateek, Jagadeesh, Siddarth, Smitha, Hiremath & Pendari, 2018) is a dynamic tool that can that can translate sign language into alphabets and words. However, the tool cannot work as a telecommunication medium such as video conference application to able the deaf communicate in a long distance.

There are many video call communication applications today, whether they are open-source or paid software (Alimudin & Muhammad, 2018). Although there are free available multiparty video conference services, they are lacking in value-added services like multi-platform and user-friendly for the deaf. For the deaf to communicate with people who do not understand sign language by using video conferencing, it is hard for the other parties to communicate with them. An interpreter is needed to translate the sign language into a spoken language.

For business and individuals alike, video conferencing has become the preferred method for long-distance communication (Rao, Maleki, Chen, Chen, Zhang, Kaur & Haque, 2019). Therefore, a video conference system should be necessary, and an Artificial Intelligence sign language interpreter should assist in overcoming these issues.

1.3 Objectives

- 1) To design and develop a web-based system that can host a group video conference of 4 people for an interview session.
- 2) To implement a sign language interpreter algorithm that converts sign language into text and audio using Artificial Intelligence (K-Nearest Neighbour) algorithm and Speech Synthesis.
- 3) To evaluate the network performance of the video conference latency, bitrate and packet loss.

1.4 Project Scope

Project scope defines the boundaries and limitations of the project. In this project, it involves various aspects which are target user, sign language, software, hardware, and APIs being used.

1.4.1 Users

The main target users of this system include managers and Human Resource Department of organizations and the deaf whom are involved in an online interview session. Other than that, family that have a deaf family member that wants to communicate with each other in the long-distance.

1.4.2 Sign Language

As stated in 1.1.2, sign languages are not universal, and they are not mutually intelligible. Hence, the system will take user input regardless of the version of the sign language.

1.4.3 Software

The host on deaf person will act as a server, while others are clients. They will use internet browser such as Chrome or Firefox to access the system.

1.4.4 Application Programmable Interfaces (APIs)

Two main open source APIs that will be used are OpenTok and Tensorflow.js

1.4.5 Hardware

Both server and clients need a computer, webcam or camera, monitor. A broadband connection is required for all users.

1.5 Significance

The project will benefit deaf people. They do not have to bring their translator which may be their family or friend, in order to translate the sign language to the people who do not understand sign language. Besides, deaf people can have an online video interview when applying for a job without a human interpreter. The deaf candidate does not have to travel to the venue that has been set for the interview, while the organization can save time and space for the interview preparation.

The deaf also can communicate with their family members and friends who do not know sign language. Instead of having them to meet physically, they can communicate with each other without human interpreter. Another benefit is that the people can learn the sign language indirectly from the deaf. Face-to-face meetings are overtaken by video conferences to reduce transport-related costs and impact on the environment and improve the efficiency of the organization (Jørgensen, Jeong & Toftum, 2017).

1.6 Summary

With the high percentage of video type traffic, it indicates the that video conference plays an important role. It enables people to ‘meet’ virtually with different locations. Video conferencing is not limited to interview session only, but other applications such as telemedicine and attending a meeting virtually.

While Artificial Intelligence (AI) can mimic the behaviours, actions and decisions like a human being. The technological advancement in AI lead to the wide usage of it among enterprises. One of the usages of the AI is expert systems, computer vision and speech recognition.

On the other hand, deaf people use sign language as a communication medium. This may limit their daily routine if they are communicating with the non-sign language user. An interpreter may be needed as they want to attend the interview with a company. A system that can host an interview session and could translate sign language into words and audio could reduce both parties’ cost (interviewer and interviewee) in conducting an interview session.

CHAPTER 2

LITERATURE REVIEW

This chapter will provide better understandings into the related areas of the project. Some of the fields that will be covered are the protocols involved, network topology and APIs used to develop the project. On the other hand, the AI algorithm that will be implemented is Deep Learning Convolutional Neural Network Classifier and K-Nearest Neighbour Classifier are also be covered. Besides, the chapter will help to make decisions on the techniques and algorithms and methodologies that will be used in developing this project.

2.1 Types of Video Conference Systems

In video conferencing systems, they come in various types. In short, they are telepresence video conferencing (Nalamwar, Kalhapure & Khatake, 2016), room-based systems (Rixe, Carter, Sheng, Spector, Doering, Chien & Joshi, 2018), and desktop video (Sorokin & Rougier, 2017).

2.1.1 Telepresence Video Conferencing

Telepresence video conferencing is dedicated to a broad audience. Virtual telepresence is the idea that people in geographically separate locations feel like meeting face to face (Onishi, Tanaka & Nakanishi, 2016). The current telepresence system is built based on open standard protocols. Standard video communication systems can be hard to set up, challenging to use, and in quality often unsatisfactory (Luevano, Lara & Quintero, 2019). It still encounters many difficulties in interoperating with other systems. One of the reasons for this problem is that the current telepresence system does not have the standardized mechanism to describe and negotiate the way of using multiple media stream on the media flow.

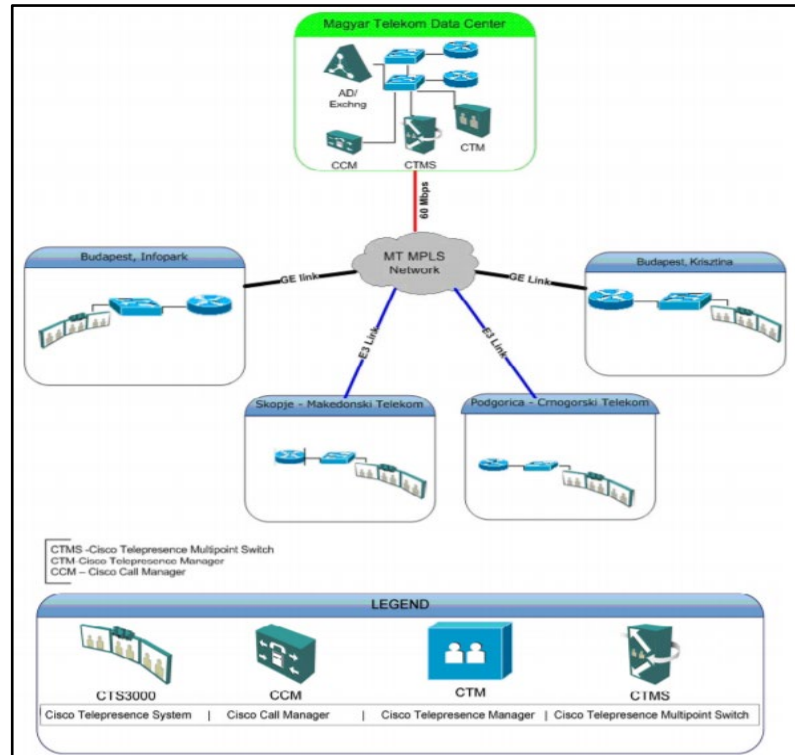


Figure 2.1 Telepresence Topology

(Source: Cisco Telepresence Implementation for Telekom's Corporate Requirements)

Figure 2.1 shows the topology of a telepresence video conference. The Magyar Telekom's Data Centre manages the telepresence networks CCM, CTM and CTMS via the company's Multiprotocol Label Switching network.

2.1.2 Room-Based Video Conferencing

It offers a variety of configurations and generally scalable to support small, medium and large meeting rooms. One of the features available is multi-screen support, the capability to project content from laptops or mobile devices, and cameras focusing on active speakers in a room. While audio specifications often vary, some vendors offer beam-forming microphones to reduce background noise or other audio improvements (Lazar, 2016).

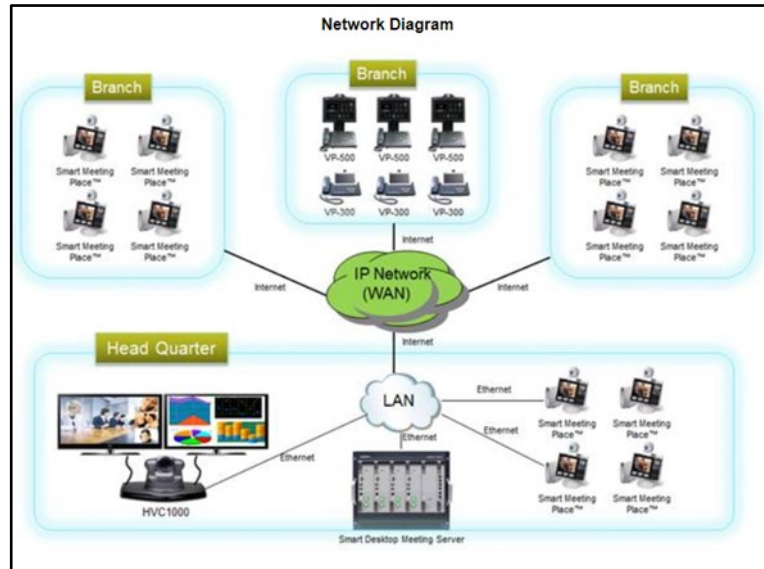


Figure 2.2 Room Based Topology

(Source: www.addpac.com)

Figure 2.2 indicates the topology of a room-based video conferencing. IP Network with WAN connection is used to connect between multiple networks (branches) and Head Quarter. Smart Desktop Meeting Server acts as Multipoint Control Unit (MCU) to manage inter-network and intra-network connections.

2.1.3 Desktop Video Conferencing

Desktop video conferencing is a type of video conferencing in which all components of the hardware and software system are included in a desktop computer. One of the features of desktop conferencing is it allowing synchronous interactions (Lakhal & Khechine, 2016). Furthermore, desktop video conferencing is a core component of unified communications and web conferencing services in the business world (Rouse, 2016).

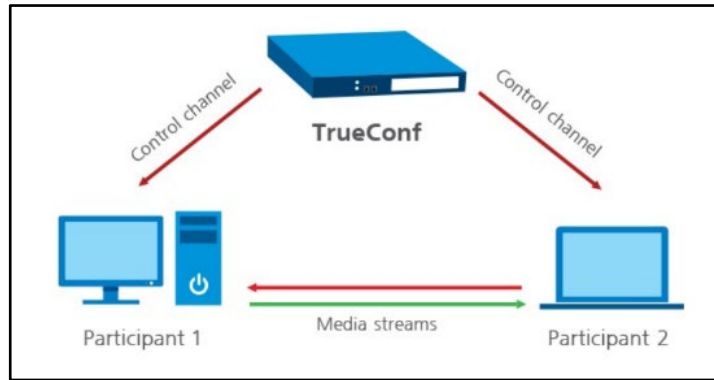


Figure 2.3 Point to Point Desktop Video Conference Topology
(Source: <https://trueconf.com/what-is-video-conferencing.html>)

Figure 2.3 shows the point to point desktop video conference topology. Point-to-point video calling is a video session involving two participants who can simultaneously see and hear each other. There are various collaborative tools for both video call and video conference users, such as text messaging, file transfer and slide show.

2.2 Network Topologies of Multiparty Video Conferencing

There are two main network topologies for multiparty video conference, which are mesh network topology and star topology.

2.2.1 Mesh Network Topologies

Figure 2.4 shows a topology with more than two participants in a video conference session. Individual direct link to every other link must be made in order to and maintain connection and session. It burdens the network equipment and significantly affects network bandwidth and costs. However, this topology provides excellent selectivity that can be provided to each user for ad-hoc connections. Furthermore, it eliminates a single point of failure problem, whereas if a link between two users breaks, it has several other links that act as backups. Meanwhile in Wi-Fi Mesh Network environment which equipped with mesh routers, it has the advantage of transmitting high volumes of data by mesh routers are equipped with multiple network interface cards (NIC) and each NIC is assigned to an orthogonal channel (Wu, Li, Xu & Tian, 2018).

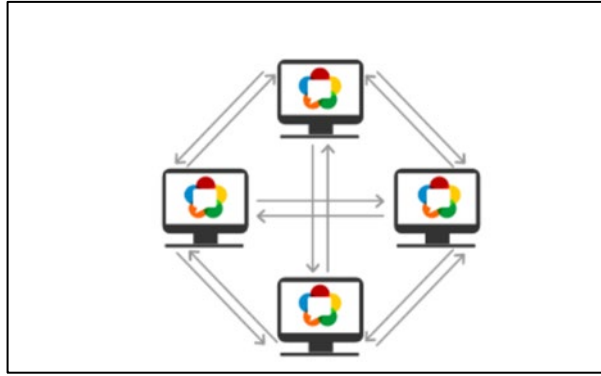


Figure 2.4 Mesh Network Topology

(Source: www.webrtc.ventures)

2.2.2 Star Network Topologies

Figure 2.5 shows a star network topology. In video conference, it uses a Multipoint Control Unit (MCU) to perform a bridge function, interconnecting participants from different sources. The MCU will attempt the connection to all the participants. Alternatively, the participants may even call the MCU. An MCU can be separated into two main functions: A Multipoint Controller and a Multipoint Processor. The controller is responsible for the signalling plane, and it is responsible for handling the conference session, creation, closing and negotiates with every unit in the conference and controls resources. While the mixing and handling of media from each terminal are done by the Media Processor which resides in the Media Plane, it allows the data stream to each terminal and redirects it to the destination endpoint. With the help of a central manager, it enables controlling the bandwidth used upon each link.

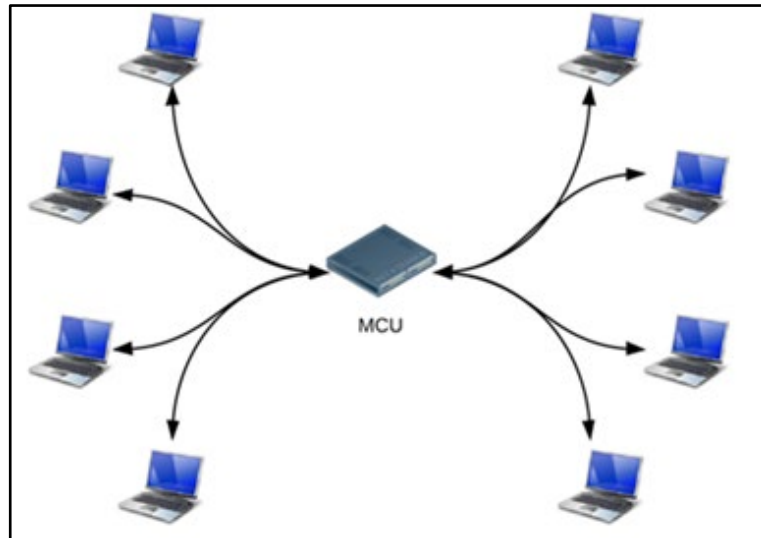


Figure 2.5 Star Network Topology

(Source: www.html5rocks.com)

On the other hand, mesh is better than star topology in terms of different input signal power. Furthermore, mesh topology supports many users with acceptable quality. Mesh topology also can manage more users than star network topology (Singh & Dewra, 2015). While video conferencing connection that is being bridged by an MCU can cause delay and communication bottleneck for the system (Yang, Zhang, Yao & Yang, 2016).

2.3 Video Conference Protocols

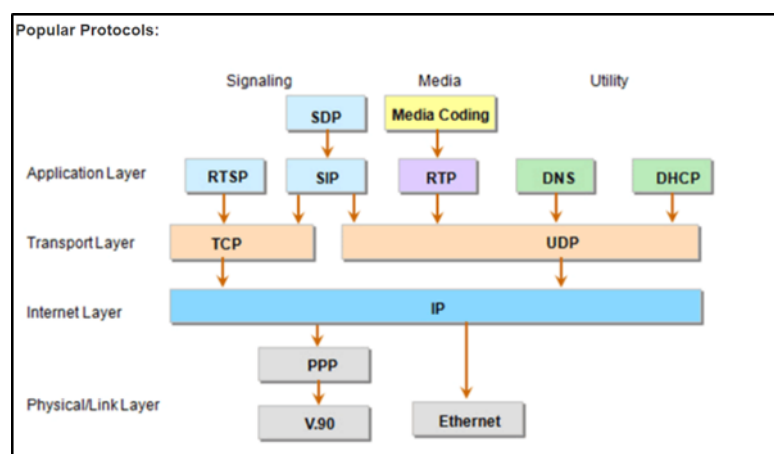


Figure 2.6 Popular Video Conference Protocol

(Source: www.engineersgarage.com)

Figure 2.6 shows some of the protocols involve in video conference at TCP/IP network layer model. At the Internet Layer point of view, the Internet Protocol

is supported, and the most well-known transport protocols are TCP and UDP (Sambath, Abdurahman & Suryani, 2016). Session Initiation Protocol (SIP) supports either TCP or UDP.

2.3.1 Real-Time Transport Protocol (RTP)

RTP resides at the application layer of the TCP/IP network layer model. RTP is used by WebRTC to standardize an interoperable and efficient framework for real-time communication using Web browsers over the RTP (Carlucci, Cicco, Holmer, & Mascolo, 2017). The issue was real-time video applications employ UDP sockets. The drawback is different applications cannot interoperate which hinders mass adoption of Real Time Communication (RTC) applications. Hence, the main international standards organization for the World Wide Web (W3C) and Internet Engineering Task Force (IETF) address this issue by having WebRTC over the web browsers.

2.3.2 Session Initiation Protocol (SIP)

Most live video communications currently follow the SIP paradigm (Rong, Sun & Kadoch, 2016). The SIP protocol is a widely known signalling protocol for establishing, changing and terminating sessions between two end points of IP sessions. Communication in a real-time system starts with the initiation process of the session. In the case of a multicast situation, the communicating parties exchange Session Initiation Protocol packets and Session Description Protocol and then build the sessions for each peer (Ramakrishna & Karunakar, 2016). IETF has standardized the SIP protocol instead of ITU (International Telecommunication Union), as most other VoIP signalling protocols are standardized by ITU (Nalawade, Nema & Yalampati, 2017).

2.3.3 Real Time Streaming Protocol (RTSP)

RTSP is a standard application layer protocol used to develop a video conference system. It is also an application layer protocol that is not connection-oriented and uses a session associated with an identifier. Normally, RTSP uses the UDP protocol to transmit video and audio data and TCP for the control. TCP is used only when needed (Gonzalez, Garcia, Barroso, Gil & Gil, 2016). There are two versions of RTSP, which are version 1.0 and 2.0. The latest version is not backwards compatible with the previous version. One of

the reasons is the change of syntax for some headers (Schulzrinne, Rao & Lanphier, 2016).

2.3.4 Real Time Messaging Protocol (RTMP)

Macromedia develops RTMP to enable its Flash Player software to stream audio / video over the Internet. For session control, data link, and quality control, RTMP uses only one TCP connection. Additionally, RTMP could also be tunnelled through HTTP or HTTPS, making it look more like web traffic in terms of traffic signature, allowing it to penetrate firewalls that block RTMP protocol through its port. YouTube is the largest provider of video streaming using the RTMP protocol (Shi & Biswas, 2016).

Figure 2.7 shows the comparison between RTSP and RTMP conducted by (Nurrohman & Abdurrohman, 2018), in the aspect of delay, the difference is significant. RTSP that use H263 codec has higher delay(millisecond) in both high- and low-resolution video than RTMP that use H264 video codec.

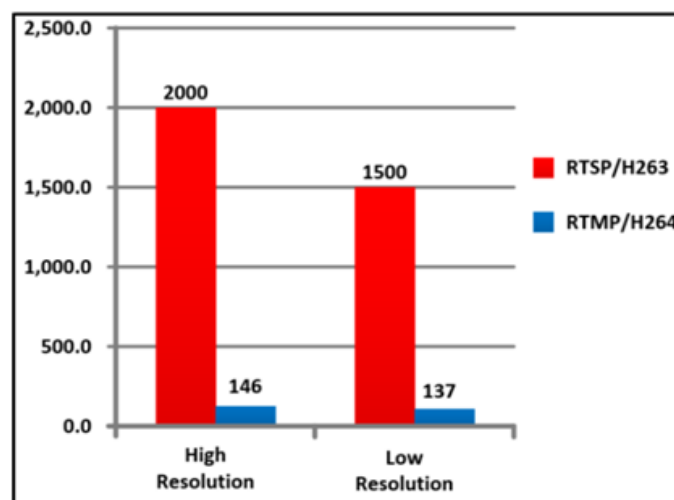


Figure 2.7 Comparison of delay(ms) between RTSP/H263 and RTMP/H264

(Source: High performance streaming based on H264 and RTMP)

2.4 STUN, TURN, and Signalling Servers

To transmit the messages to the peer, a NAT traversal approach is needed. This is because most people are behind the firewall or the home router configured with private subnet when connecting to the Internet. As a result, the IP Address

of the computer is not the wan IP address so the computers cannot directly connect (Li, Ding, Xu & Li, 2019).

Session Transversal Utilities for NAT (STUN) is a protocol where it standardized the method of transversal of Network Address Translator (NAT) gateways of a real time applications. STUN is used by an endpoint to determine the IP address and port allocated to it by a NAT (Garcia, Gortazar, Fernandez, Gallego & Paris, 2017). Despite this, the method is very troublesome for some users and requires administration privileges.

STUN protocol can solve the problem of going through the general home router (NAT) environment, but for most business network environment, it is not very good. Then there is a need for a new solution, which TURN (Traversal Using Relay NAT, allowing TCP or UDP connections across NAT or firewall. TURN servers also act as relays in the presence of firewall (Kirmizioglu & Tekalp, 2019).

2.5 WebRTC

WebRTC has always been one of the hot topics of discussion, mainly because it does not require additional user-side programs to be installed, and it needs only a web browser to immediately perform messaging on the web. To make this technology work well for many users, cloud service is the one of the methods. The cloud architecture is ideal when the system requires scalable service capacity (Lu, Chen, Kuo, Tseng & Chou, 2019). WebRTC is not inherently decentralized, it relies on a signalling mechanism for initiating communications, whether by exchanging messages manually or using a server (Romano, 2019).

2.6 WebRTC-based Video Conferencing API

WebRTC-based Video Conferencing API is an API that is based on WebRTC. There are many WebRTC free or paid services provided to ease developers to develop web-based real-time communication. Some of them are OpenTok and Jitsi-Videobridge.

2.7 OpenTok

OpenTok is a public WebRTC servers (Balan, Robu & Sandu, 2017). OpenTok uses WebRTC for audio-video communication purposes. All OpenTok platform applications require two main components, which are client and server.

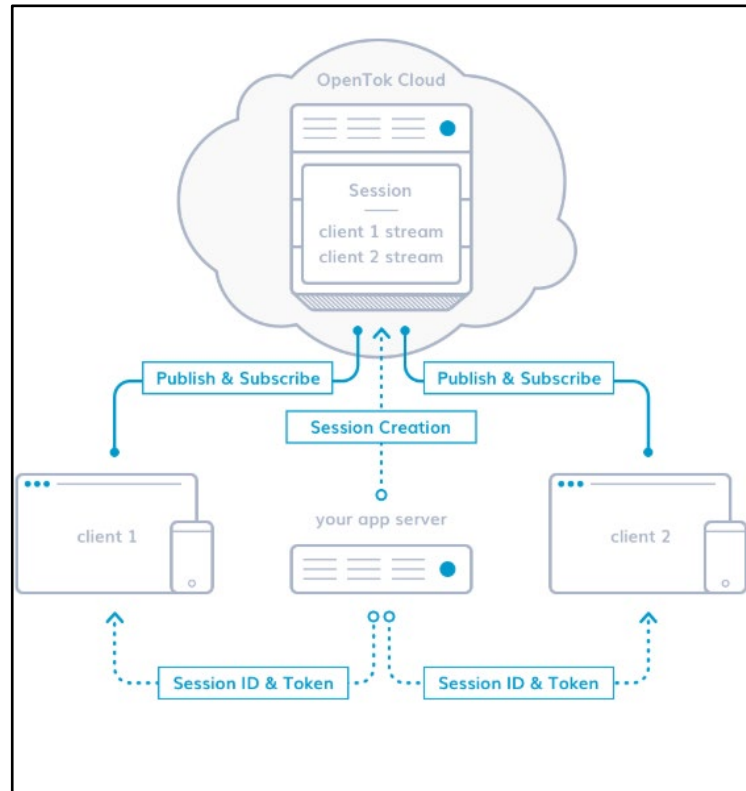


Figure 2.8 OpenTok Client and Server Flow

(Source: <https://tokbox.com/developer/guides/basics/>)

Figure 2.8 shows the OpenTok client and server flow. The app server is responsible to create a session where clients can interact with each other. Session is hosted on the OpenTok cloud and manage user connections, audio-video streams, and user events. Then, the Client 1 load the app server making request to get Session ID (session identifier) and Token (authentication key). The app server will send the Session ID and Token. When a new user loads the client-side application in a separate web page or mobile device (Client 2), the new client receives the session ID and a unique token from app server. The client uses that info to establish a connection to the session.

Now that it's connected to the session, Client 2 can subscribe to the stream of Client 1. Client 2 then publishes its own stream of video to the session, and Client 1 subscribes to it. Both clients are now subscribed to each other's stream.

2.8 Jitsi

Jitsi platform is used as a controller for the conference. One of the key features is that the dynamic stream forwarding, and bitrate recomputing functionality can be extended. The Jitsi-Videobridge's default behaviour is to relay a subset of WebRTC streams to all participants in the conference. The software automatically detects the stream of the participant currently speaking, the so-called dominant speaker, and is always included in these streams. This logic overrides the dynamic stream forwarding, so that a different dominant speaker can be configured manually per receiver. In addition, there is only one limit to the number of streams that can be sent to a single receiver. The Jitsi-Videobridge uses this mechanism to dynamically assign a sub-sender per receiver, so the encoding bit rate is lower than the estimated bandwidth of the receiver (Petrangeli, Pauwels, Hooft, Wauters, Turck & Slowack, 2018).

2.9 Video Codec

Videos have been used in many applications such as traffic monitoring, spacecraft operations, robotic applications, machine tools operations and security surveillance. Video codec is a technique of video compression. It converts raw digital video into a stream of bytes. Some of the applications mentioned compressed video data are sent in packets over a communication channel, which can be wired or wireless. In many non-real-time applications, retransmission of the lost packets is handled by TCP/IP protocol. However, in real-time video applications such as video broadcasting, video conferencing, video chatting and video streaming, packet retransmission will introduce excessive delay and inefficient bandwidth usage (Kwan, Shi & Um, 2018).

2.9.1 VP8

VP8 is an open, royalty-free video compression format owned by Google and developed as a successor to VP7 by On2 Technologies. VP8 encoding format can achieve a significant speed-up with respect to the mostly optimized software encoder (up to $\times 6$), with minimum degradation of the visual quality

and low processing latency (Grossi, Paglierani, Pedersini & Petrini, 2018). The frames of the video are decomposed into square blocks of pixels (called macroblocks) in the VP8 encoding scheme as in any other standard video compression method. Based on previously coded macroblocks, intra-frame and inter-frame prediction is performed and then the image discrepancy between the real and the expected macroblock is transformed using the discrete cosine (DCT) and Walsh-Hadamard transforms (WHT).

2.9.2 VP9

VP9 is Google's latest royalty-free video codec format. VP9's development began in the third quarter of 2011. VP9 lowers the bit rate by 50% compared to its predecessor VP8 while retaining the same quality of video. It is designed to improve the efficiency of compression compared to HEVC. VP9 also has the same fundamental structure as VP8. However, compared to VP8, it has many refinements as it supports the use of superblocks (64x64 pixels) with a quad tree coding structure (Sabry & Ramadan, 2019).

2.10 Gesture Recognition

Computer does not interpret images like humans. It needs a mechanism to be able to 'see' an image. Gesture recognition is one of the key components of human – computer interaction's research field. Recognition of distinct movements of the hand and arm is becoming increasingly important as it enables intelligent communication with electronic devices. In addition, gesture identification in video is a first leap towards recognition of sign language, where even subtle movement differences can play a significant role (Pigou, Oord, Dieleman, Herreweghe & Dambre, 2016).

2.11 Machine Learning

Machine learning is a subfield of Artificial Intelligence. Machine learning and intelligent systems such as search engines, recommendation platforms, and software for speech and image recognition have become an important part of modern society (Bottou, Curtis & Nocedal, 2018). Machine learning provides programs with the ability to learn and learn automatically from experience without explicit programming. Machine learning focuses on computer programs that can access data and use it to learn on their own. Machine learning

approaches learn the rules underlying a dataset by evaluating a portion of that data and building a prediction model (Butler, Davies, Cartwright, Isayev & Walsh, 2018).

2.12 Deep Learning

Deep learning is a subfield of machine learning. It has progressed rapidly since the early 2000s and is now showing state-of-the-art performance in different fields. It also has overcome previous limitations and academic interest has increased rapidly since early 2000s. One of the key elements of deep learning is that it is built on a foundation of significant algorithmic details. Some of the groups of the deep learning architecture are Convolutional Neural Network (CNN) and K-Nearest Neighbour.

2.12.1 Convolutional Neural Network (CNN)

It has been shown that CNN is effective deep learning model that can extract high-level features directly from raw data. CNN can be used as an effective and high-performance classification method in the field of image processing and has achieved exceptional performance (Leng, Li, Bai, Dong & Dong, 2016).

2.12.2 K-Nearest Neighbour (KNN) Algorithm

The algorithm is used for classification problem. It is also called a lazy learning approach. According to (Manjusha & Harikumar, 2016), the algorithm is called a lazy learning method because it does not actually learn from the data, instead it only stores all the samples in training data.

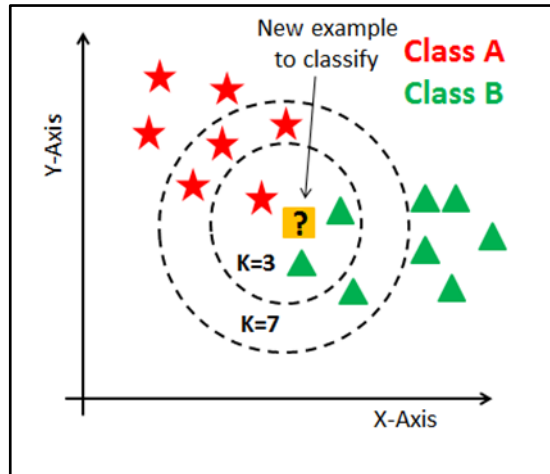


Figure 2.9 KNN Graph Visualization

(Source: <https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn>, 2018)

Figure 2.9 shows the data representation in a graph. When there is unclassified data, it uses the nearest neighbour which is K to determine the class of the data by finding most of the class. When $K = 3$, the class of the data is Class B as Class B is the majority count in the K range. When $K = 7$, the class of the data falls into Class A because the majority count is Class A within range K .

2.13 Summary

In conclusion, the best reasonable topology to implement is Star Network Topology. The reason is a central hub is needed to manage the session between server and clients. Next, WebRTC API by Google is a decent choice since it provides the platform to develop for server and client side. For video encoding and translating, even though VP9 encoding is the best quality video codec to use, it requires more computing power over VP8. In this way, VP8 is the most reasonable approach in this project because of the computing power and resource limitations. Meanwhile for the hand motion gesture recognition, KNN classifier is the most appropriate as it is easy to implement and provide better results than CNN.

CHAPTER 3

METHODOLOGY

This section discusses important information and explanations of the project's methodology and flows to achieve the project goals. It primarily explains how to develop and implement the web-based system. This chapter also helps to improve the project in a proper way and in timely manners.

3.1 Methodology of the Project

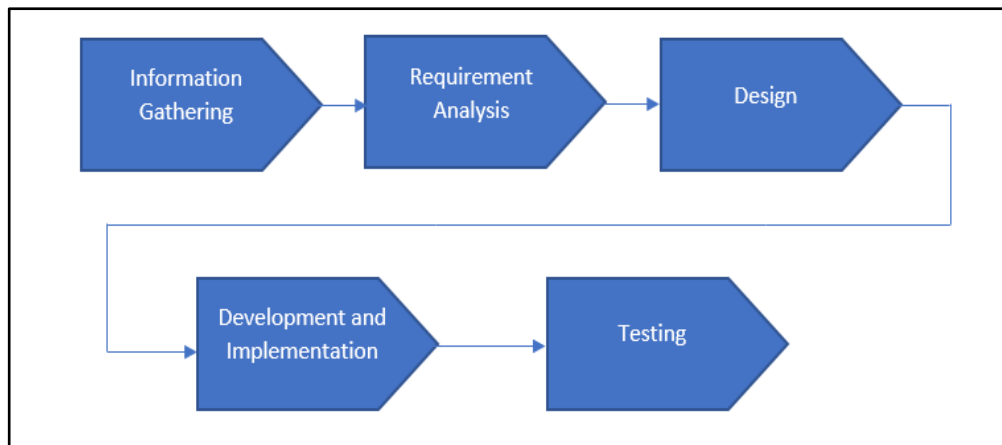


Figure 3.1 Project Methodology

Figure 3.1 illustrates the methodology of the project. Each of the phases describes the process need to be done to complete the project. The methodology model used for this project is Waterfall methodology. The project development is linear and sequential. Each phase has its goals and it must be completed before moving into next phase.

Table 3.1 Descriptions of the Phases

Project Phase	Activities	Expected Outcome
<ul style="list-style-type: none">• Information Gathering	<ul style="list-style-type: none">• Writing Literature Review• Journals and articles research	<ul style="list-style-type: none">• Problem statement• Project objectives• Project scopes• Project significant• Literature Review
<ul style="list-style-type: none">• Requirement Analysis	<ul style="list-style-type: none">• Hardware requirement• Software requirement• Choosing the best techniques	<ul style="list-style-type: none">• Client and server hardware requirement• Client and server software requirement• Suitable techniques that will be used
<ul style="list-style-type: none">• Design	<ul style="list-style-type: none">• Designing system flow of the project• Choose suitable network topology	<ul style="list-style-type: none">• Network topology• System flow
<ul style="list-style-type: none">• Development	<ul style="list-style-type: none">• Connecting to a signalling server	<ul style="list-style-type: none">• Connection with a signalling server• Source code
<ul style="list-style-type: none">• Testing	<ul style="list-style-type: none">• Test the network performance of latency, bitrate and packet loss.• Testing environmental setup	<ul style="list-style-type: none">• Results of the testing

Table 3.1 shows planned project's phase in developing the web-based system. The activities column indicates the tasks that be expected to be done in the phase. While expected outcome column indicates the expected result after tasks executions.

3.2 Information Gathering

The first phase in project development is Information Gathering. This phase is necessary and important because it provides the project initial overviews by collecting information from different and reliable sources such as surveys, interviews and journals. Furthermore, it helps to identify project's problem

statement, objectives, scope and significant which is also important in indicating the relevance of the project.

In developing the project system, it is important to identify the current problem faced by the deaf community to communicate with other people by using video conference systems. Several related journals and articles have been reviewed as a guide to identify the current video conference systems. Hence, it helps to identify the improvement that can be done in this project so the project's objectives can be achieved.

3.3 Requirement Analysis

In this phase, requirements to develop the project is identified. Tools and API that will be used, protocols, hardware and software requirements and limitations are identified.

3.3.1 Hardware Requirements

The hardware requirements for clients, server and signalling server of this project is different. This is because, in order to capture the user's sign language motion gesture and to translate them into text, the server needs more computing power and resources than the client. Meanwhile, the signalling server that control the video conference session is a public server, so the hardware specifications is vary. For the clients, they require low computing power than both server and signalling server because they only transmit video image and receive text.

Table 3.2 Hardware Requirements

Hardware	Specification and Function
Server <ul style="list-style-type: none"> • Laptop: Asus ROG G551JW 	<ul style="list-style-type: none"> • Processor: Intel i7 4th Generation • RAM: 12 GB ddr3 • Webcam: 720p resolution • Server of the video conference
Signalling Server <ul style="list-style-type: none"> • Scaledrone public server 	<ul style="list-style-type: none"> • Signalling public server • Handle session between server and client
Client 1 <ul style="list-style-type: none"> • Laptop: Asus Client 2 <ul style="list-style-type: none"> • Laptop: Asus Client 3 <ul style="list-style-type: none"> • Laptop: HP 	<ul style="list-style-type: none"> • Processor: Intel i5 5th Generation • RAM: 8GB ddr3 • Webcam: 720p resolution • Processor: Intel i5 5th Generation • RAM: 4GB ddr3 • Webcam: 720p resolution • Processor: Intel i5 5th Generation • RAM: 8GB ddr3 • Webcam: 720p resolution
<ul style="list-style-type: none"> • Broadband with internet connection 	<ul style="list-style-type: none"> • 1Mbps internet connection

3.3.2 Software Requirements

The software requirements for clients and servers also is different. For the server, it needs additional API which is Tensorflow.js. The API helps to translate user sign language motion into text.

Table 3.3 Software Requirements

Software and API	Function and Description
Server	
<ul style="list-style-type: none"> • Node.js 	<ul style="list-style-type: none"> • JavaScript based asynchronous web server hosting • Version 8.0
<ul style="list-style-type: none"> • API: WebRTC 	<ul style="list-style-type: none"> • A JavaScript API for developing the video conference • Version 1.0
<ul style="list-style-type: none"> • API: Tensorflow.js 	<ul style="list-style-type: none"> • A JavaScript API for capturing user's gestures and translate them using machine learning into text • Version r1.13
<ul style="list-style-type: none"> • Internet browser 	<ul style="list-style-type: none"> • Platform for video conference • Desktop Chrome Version 74.0.3729.131
Signalling Server <ul style="list-style-type: none"> • Scaledrone 	<ul style="list-style-type: none"> • Server for signalling protocol • Public Server
Clients <ul style="list-style-type: none"> • Internet Browser 	<ul style="list-style-type: none"> • Platform for video conference • Desktop Chrome Version 74.0.3729.131

3.4 System Design

In this subchapter, the design phase helps to develop the system in a proper manner. Flowchart is important because it helps to provide project's flow and logic. While the Server Flow Diagram helps to provide better understanding of the flow of the WebRTC API.

3.4.1 User Flowchart

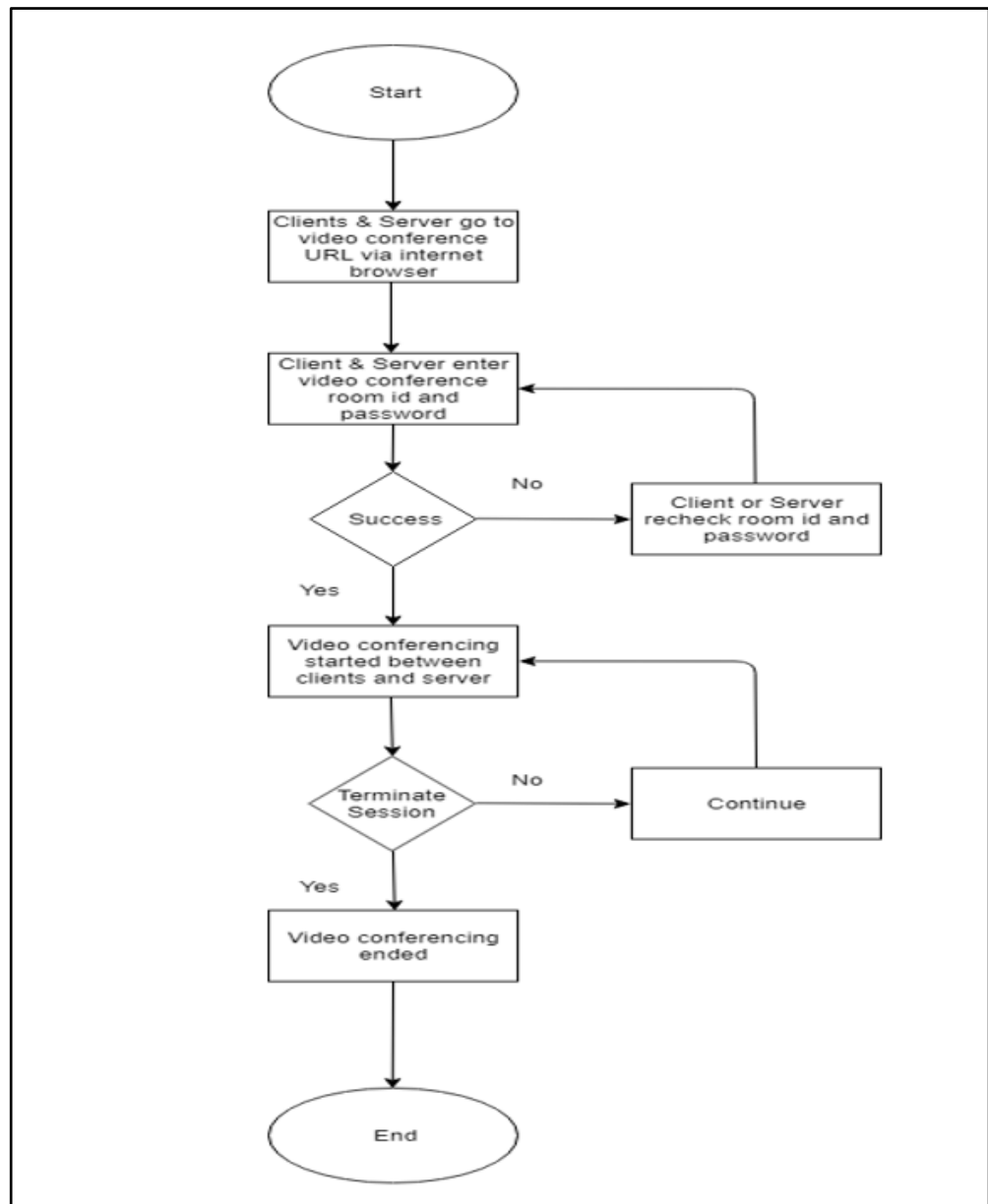


Figure 3.2 User Flowchart

Figure 3.2 shows the proposed users' flowchart. The flowchart helps to supply the logic of the system based on users view.

3.4.2 Topology

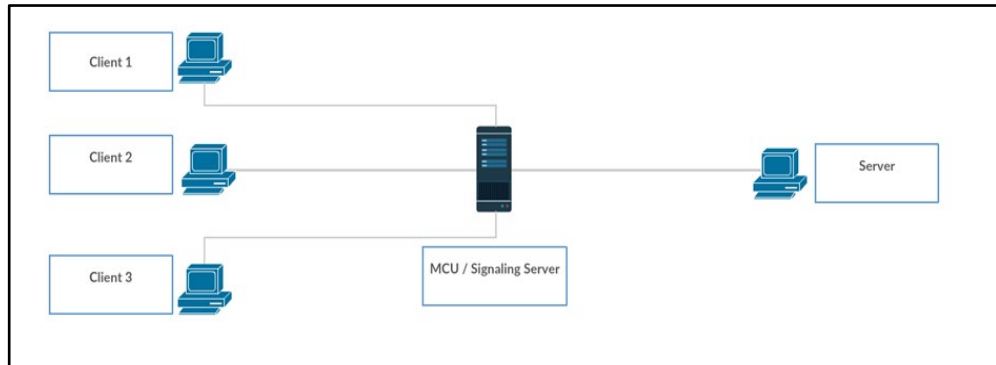


Figure 3.3 Network Topology

Figure 3.3 shows the purposed network topology of the projects. The server, which is the deaf person host, will act as publisher. The host publish its streams to all the clients. While the MCU or Signalling server will manage all the connections and streams.

3.5 Development and Implementation

In this subchapter, Development and Implementation discusses about on how the system will be implemented from a technical view initially. It covers on the implementation of connecting the local device to a public signalling server using OpenTok script.

3.5.1 Database Setup

In this project, Firebase Cloud Firestore is used as a database platform in developing the web application. Firebase is a No SQL database structure that has a lot of optimizations and features compared to relational database (Guerrero, 2016). Firebase also provides some other services such as Firebase Auth and Firebase Hosting.

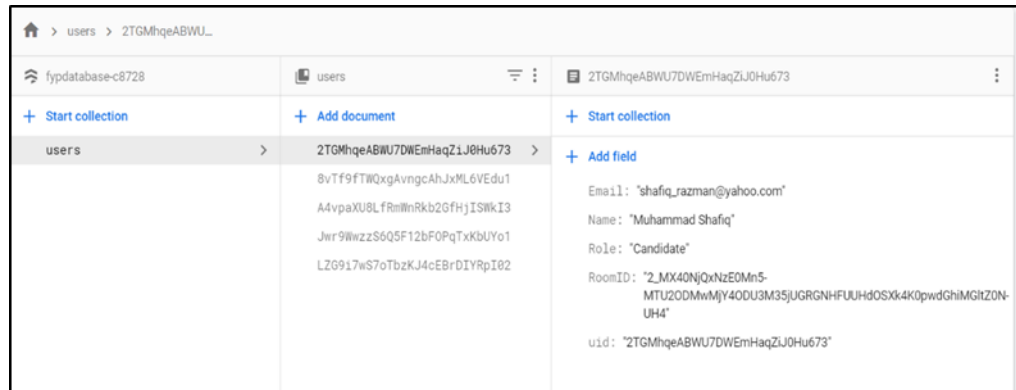


Figure 3.4 Firebase Cloud Firestore Structure

Figure 3.4 shows the structure of the database. The unit of the storage is called documents in Cloud Firestore. Each document is uniquely identified by its name. In this project, the name of the document is the user ID. The documents contain fields that map with values. For example, the document highlighted from the figure 3.4 contains fields of Email, Name, Role, RoomID and uid that map to their values respectively. While collection acts like a container for the documents.

3.5.2 Code Segment of Firebase Implementation

```

154 function login() {
155     var userPass = document.getElementById("password_field").value;
156     var userEmail = document.getElementById("email_field").value;
157     var resetButton = document.getElementById("resetBtn").value;
158
159     firebase.auth().signInWithEmailAndPassword(userEmail, userPass).catch(function(error) {
160         // Handle Errors here.
161         var errorCode = error.code;
162         var errorMessage = error.message;
163
164         window.alert("Error : " + errorMessage);
165     });

```

Figure 3.5 Firebase Implementation

Figure 3.5 shows the code of Firebase implementation in JavaScript. The function login gets the user email and password to a JavaScript variable from HTML element. The `firebase.auth().signInWithEmailAndPassword()` is a function to login the authorised user. It takes two parameters which are *userEmail* and *userPassword*. If there are any errors, alert the error message to the user.

```

106 ▼ function getUserRole(uid) {
107
108     var db = firebase.firestore();
109     var userRef = db.collection("users").doc(uid);
110     userRef.get().then(function(doc) {
111 ▼         if (doc.exists) {
112             sessionStorage.setItem("uname",doc.data().Name);
113             if(doc.data().Role == "Interviewer") {
114                 window.location.href = "Interviewer.html";
115 ▼             } else {
116                 document.getElementById("room_id").value = doc.data().RoomID;
117                 window.location.href = "Interviewee.html";
118             }
119 ▼         } else {
120             // doc.data() will be undefined in this case
121             console.log("No such document!");
122         }
123     }).catch(function(error) {
124         console.log("Error getting document:", error);
125     });
126 }

```

Figure 3.6 Getting User Role

Figure 3.6 shows the function of getting user role after user is signed in. Firebase database works like a file structure. Variable *userRef* is used to point a collection of users that is uniquely identified by user id. If the document exists, if the assigned role of the user is interviewer, the browser will redirect to the interviewer interface, else it will redirect to the interviewee interface.

3.5.3 Code segment of OpenTok.js Implementation

```

26     // Subscribe to a newly created stream
27     session.on('streamCreated', function(event) {
28 ▼         session.subscribe(event.stream, 'subscriber', {
29             insertMode: 'append',
30             width: '100%',
31             height: '100%'
32         }, handleError);
33     });
34
35     // Create a publisher
36     var publisher = OT.initPublisher('publisher', {
37         insertMode: 'append',
38         width: '100%',
39         height: '100%',
40         name: intervieweeUserName
41     }, handleError);

```

Figure 3.7 Code Segment of OpenTok

Figure 3.7 shows the code segment of openTok implementation. *OT.initPublisher* initialize a stream of publisher with append video mode. Function *handleError* is passed to manage all the errors if errors occur. When session is created, a subscriber element is created to subscribe the stream created by the publisher with same parameters.

3.5.4 Code segment of Tensorflow.js KNN Implementation

```

399
400 ▼ train(gestureIndex) {
401     console.log(this.videoPlaying);
402 ▼     if (this.videoPlaying) {
403         console.log("entered training");
404         // Get image data from video element
405         const image = dl.fromPixels(this.video);
406
407         // Add current image to classifier
408         this.knn.addImage(image, gestureIndex);
409
410         // Get example count
411         const exampleCount = this.knn.getClassExampleCount()[gestureIndex];
412
413 ▼         if (exampleCount > 0) {
414             //if example count for this particular gesture is more than 0, update it
415             this.exampleCountDisplay[gestureIndex].innerText = ' ' + exampleCount + ' examples';
416
417             //if example count for this particular gesture is 1, add a capture of the gesture to
418 ▼             if (exampleCount == 1 && this.gestureCards[gestureIndex].childNodes[1] == null) {
419                 var gestureImg = document.createElement("canvas");
420                 gestureImg.className = "trained_image";
421                 gestureImg.getContext('2d').drawImage(video, 0, 0, 400, 180);
422                 this.gestureCards[gestureIndex].appendChild(gestureImg);
423             }

```

Figure 3.8 Training Image Function

Figure 3.7 shows the function to train an image. It takes an integer *gestureIndex* as function parameter. The *gestureIndex* is used to keep track of the number of the image had been trained. If the webcam of the user is running, get the current image of the webcam feed. Then, add the image using *addImage* function to the KNN model.

3.6 Summary

As a conclusion, this chapter generally discusses about the project methodology. In other words, how the project is being developed and implemented. It also discusses about the flow and logic of the system through flowchart and the flow diagram. In addition, by using Waterfall methodology model in developing the project, it helps to develop the project efficiently and effectively. A process or phase must be completed first before moving into the next phase. There are 5 phases total which are Information Gathering, Requirement Analysis, Design, Development and Implementation and Testing.

CHAPTER 4

EVALUATION

This chapter will explain about the processes involved when the system is being tested.

4.1 Evaluation Scope

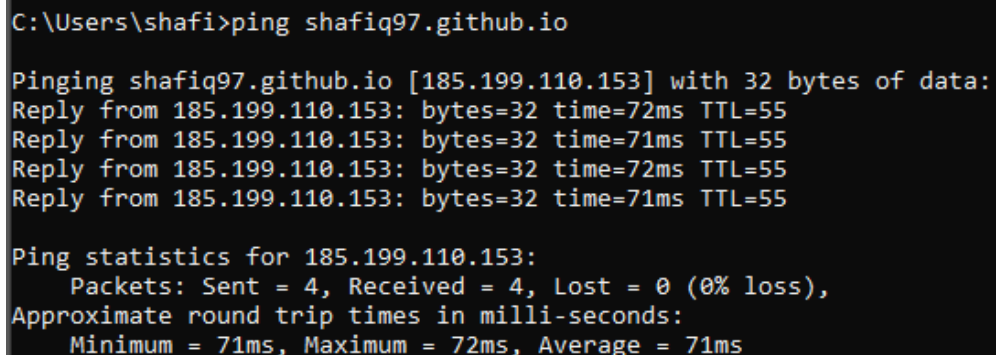
It covers network performance, functionality and accuracy of predictions evaluation. Three parameters of network performance are latency, packet loss and bitrate.

4.2 Network Evaluation

In this phase, network performance is being tested. The network test consists of latency, routed hops, packet loss and bitrate. OpenTok does not provide its user the IP addresses of the servers. This makes the network evaluation process difficult. However, OpenTok provides a tool for network and user monitoring called Inspector.

4.2.1 Hosting Server

The web application is hosted on “https://shafiq97.github.io”. To verify the web-based system, ping tool can be used. If the web server replies with ICMP packets, it indicates that server is running and hosting the web system.



```
C:\Users\shafi>ping shafiq97.github.io

Pinging shafiq97.github.io [185.199.110.153] with 32 bytes of data:
Reply from 185.199.110.153: bytes=32 time=72ms TTL=55
Reply from 185.199.110.153: bytes=32 time=71ms TTL=55
Reply from 185.199.110.153: bytes=32 time=72ms TTL=55
Reply from 185.199.110.153: bytes=32 time=71ms TTL=55

Ping statistics for 185.199.110.153:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 71ms, Maximum = 72ms, Average = 71ms
```

Figure 4.1 Result of Ping Command

Figure 4.1 shows the result of command “ping shafiq97.github.io”. The current IP address is 185.199.110.153. Packets with 32 bytes of data are sent. The server responded with the average of 71ms with TTL of 55. TTL indicates how long to use or hold the packet before the router discards the packets.

4.3 Network Evaluation Environment

The evaluation is based on the four users joining the video conference session. They joined the video session from different geographical area. The interviewee (User 1) joining the session from Malacca while the other 3 interviewers are from Tokyo, Johor and Malacca as well. The session duration is 37 minutes.





USER	LOCATION	SYSTEM	SDK	ERRORS	GUID
 User 4	Chiyoda-ku, Tokyo, Japan	Opera# Win10	js-2.16.3	0	61876d45-d374-4e45-9064-7fbdfced3440
 User 3	Malacca, Melaka, Malaysia	Chrome# Win10	js-2.16.3	0	e45ecd25-81b5-4e76-8954-abcd03d80f92
 User 2	Johor Bahru, Johor, Malaysia	Chrome# Win10	js-2.16.3	0	a99d6071-e2ce-4330-aa5a-48c1686d9c73
 User 1	Malacca, Melaka, Malaysia	Chrome# Win10	js-2.16.3	0	7b0969dc-baa1-4844-9f3f-0f7a7d1e0cd7

Figure 4.2 User Information for the Session

Figure 4.2 shows the information of the users that joined the session. Server Development Kit (SDK) version for all the user is JavaScript 2.16.3 release. Global Unique Identifier (GUID) is the identifier used to identify unique users.

4.3.1 Network Latency Evaluation

Latency is the amount of time a packet takes to get one from endpoint to the other. Hardware variability can be a cause of network latency variation, for example, if a CPU disables certain cores to retain power when not in use. Factors that affect network devices such as routers and switches can be a source of contention in which buffers used during routing of packets. Network stalls, where a packet is dropped due to a busy receiver or full buffer, have been evidentially identified as an important predictor in parallel application

performance (Underwood, Anderson & Apon, 2018). The x-axis is the date and time of the session. While the y-axis is the round-trip delay time (ms). The straight red horizontal line indicates a 300ms of round-trip delay time threshold. Any latency higher than this may result in a bad experience (OpenTok.com).

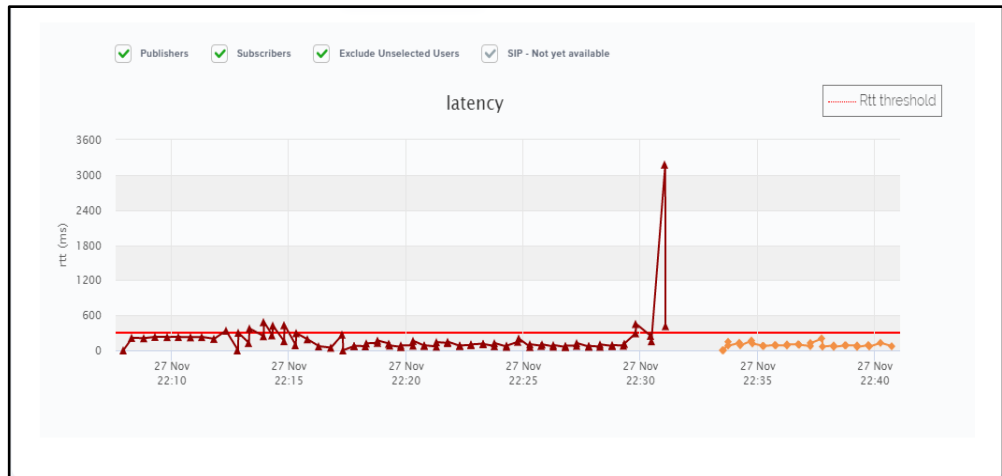


Figure 4.3 Latency (ms) for User 1

Figure 4.3 shows the latency of User 1 (Interviewee) located in Malacca. The latency is measured from the web server to the OpenTok Server. The average round-trip-time (ms) is below the Rtt threshold. However, at 22:30 the latency exceeds the threshold with 3000ms.

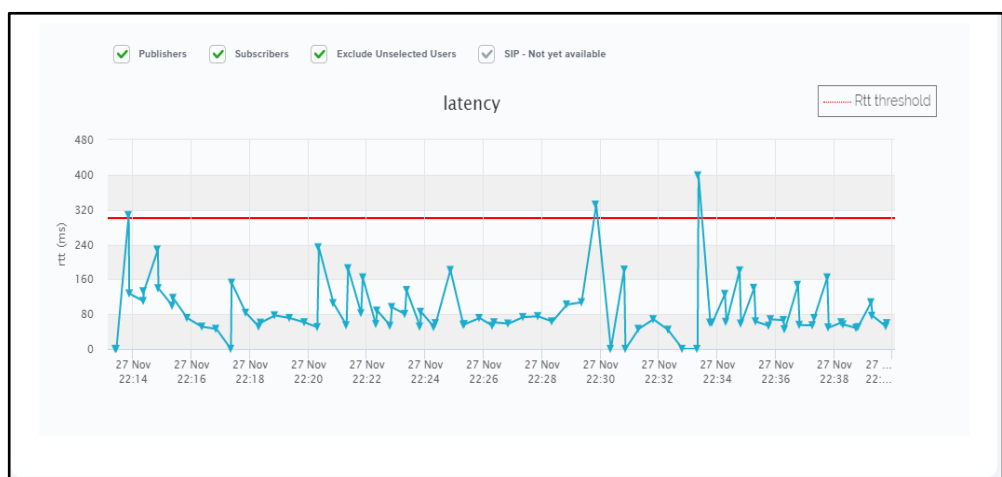


Figure 4.4 Latency (ms) for User 2

Figure 4.4 shows the latency of User 2 (Interviewer) located in Johor Bahru. The average round-trip-time (ms) is below the Rtt threshold. At 22:30 and 22:39 the latency exceeds the threshold with 400ms.

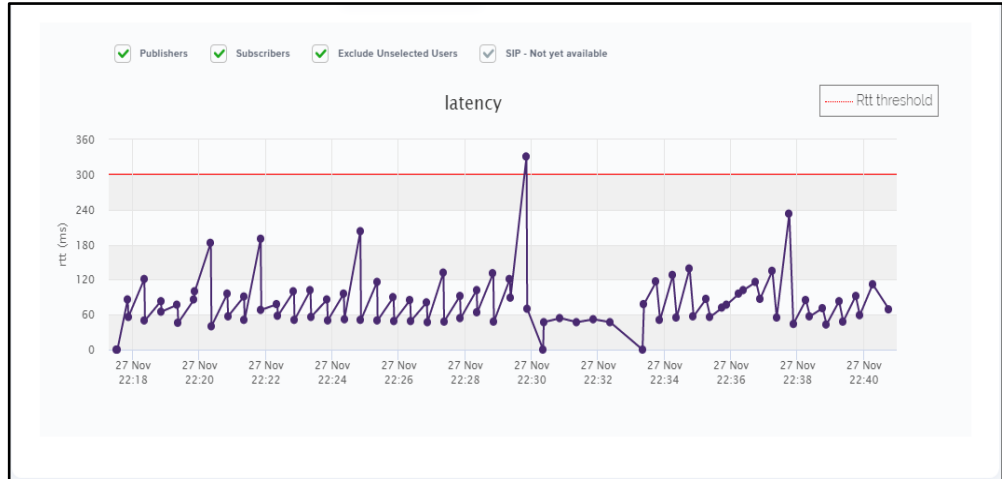


Figure 4.5 Latency (ms) for User 3

Figure 4.5 shows the latency of User 3 (Interviewer) located in Malacca. The average round-trip-time (ms) is below the Rtt threshold. At 22:30 the latency exceeds the threshold with 350ms.

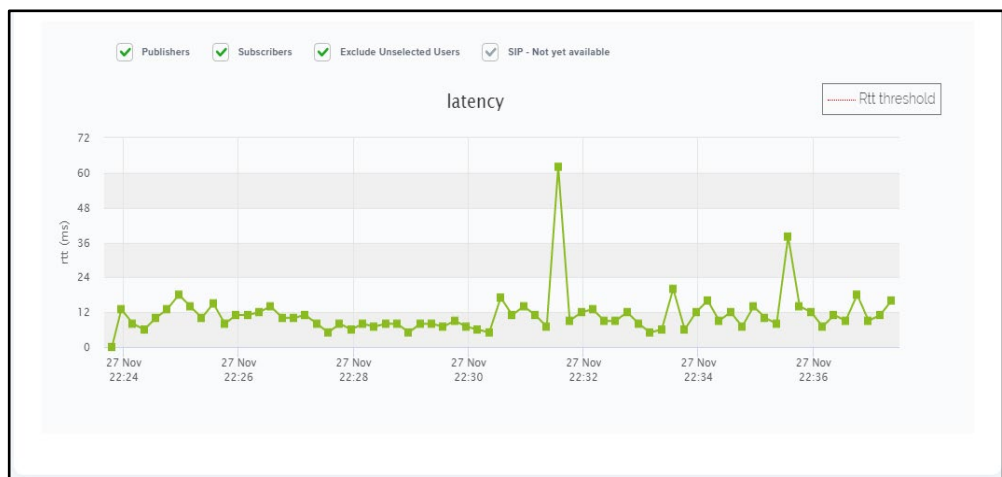


Figure 4.6 Latency (ms) for User 4

Figure 4.6 shows the latency of User 4 (Interviewer) located in Tokyo, Japan. The average round-trip-time (ms) is below the Rtt threshold. The maximum latency never exceeded the Rtt threshold. The highest latency is 60ms.

4.3.2 Packet Loss Evaluation

There are various factors that can affect network packet loss. Packet loss happens when one or more data packets are not reaching their destination via a computer network. Some of the reasons of packet loss are router failures, fiber link down and software errors and it can be both random and burst (Wuttidittachotti & Daengsi, 2016).

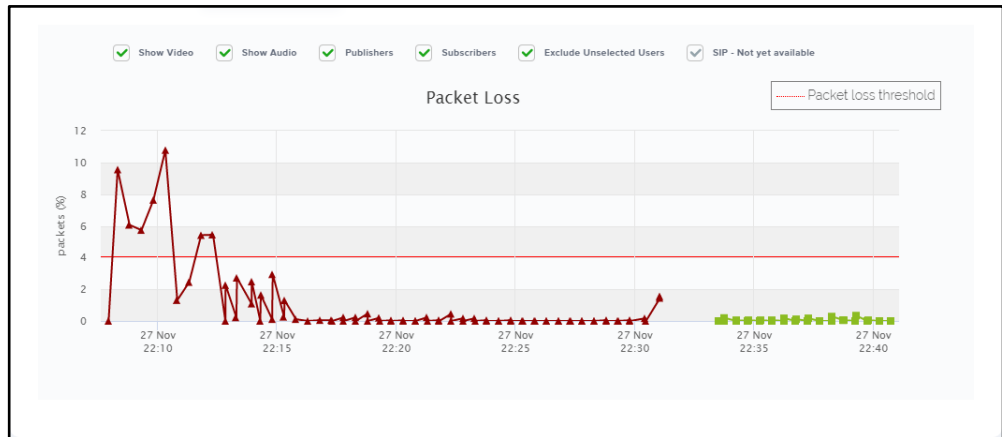


Figure 4.7 Packet Loss (%) for User 1

Figure 4.7 shows the packet loss of User 1 (interviewee). The first 10 minutes of the session, packet loss percentage exceeded the threshold with maximum of 11%. Any packet loss that is higher than 4% may result in a bad experience (openTok.com).

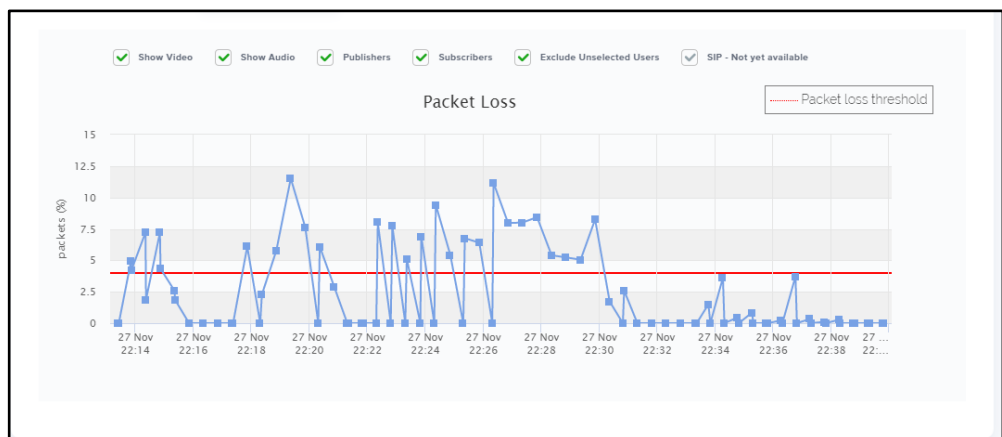


Figure 4.8 Packet Loss (%) for User 2

Figure 4.9 shows the packet loss of User 1 (interviewee). The first 14 minutes of the session, packet loss percentage exceeded the threshold with 12.5%.

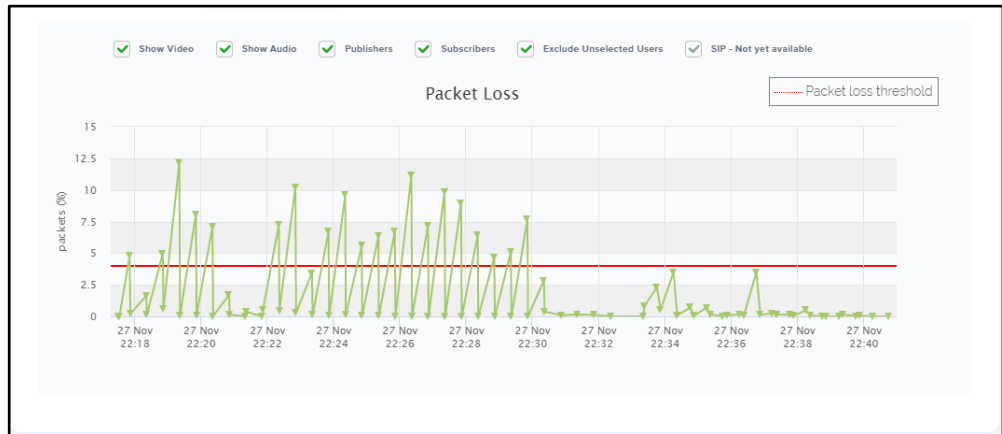


Figure 4.9 Packet Loss (%) for User 3

Figure 4.9 shows the packet loss of User 1 (interviewee). The first 12 minutes of the session, packet loss percentage exceeded the threshold with 12.5%.

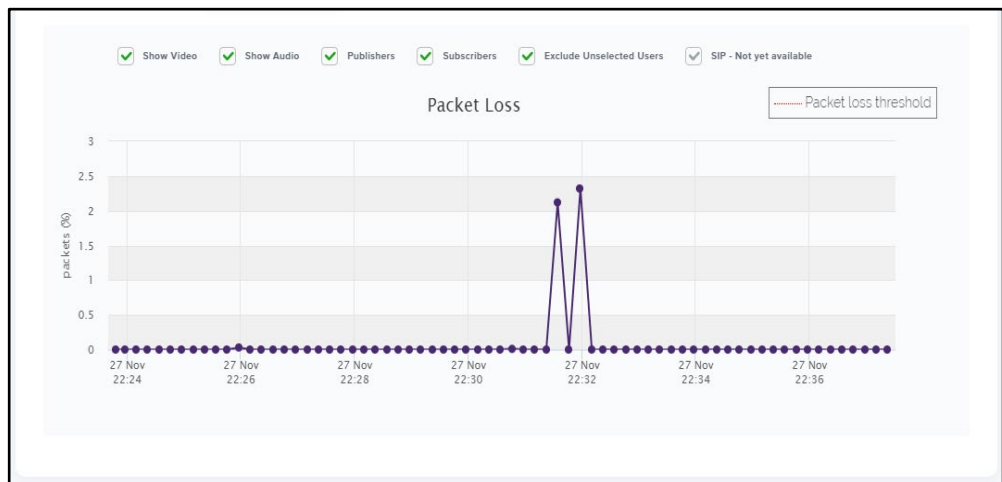


Figure 4.10 Packet Loss (%) for User 4

Figure 4.10 shows the packet loss of User 1 (interviewee). The maximum percentage of packet loss is never exceeded the threshold with 2.5%.

4.3.3 Bitrate Evaluation

An excessively high bitrate results in frequent video freezes such as video rebuffering. While a too low bitrate leads to poor quality of the video (Spiteri, Urgaonkar & Sitaraman). Bitrate also is a feature that is correlated with codec (Pal & Vanijja, 2017).

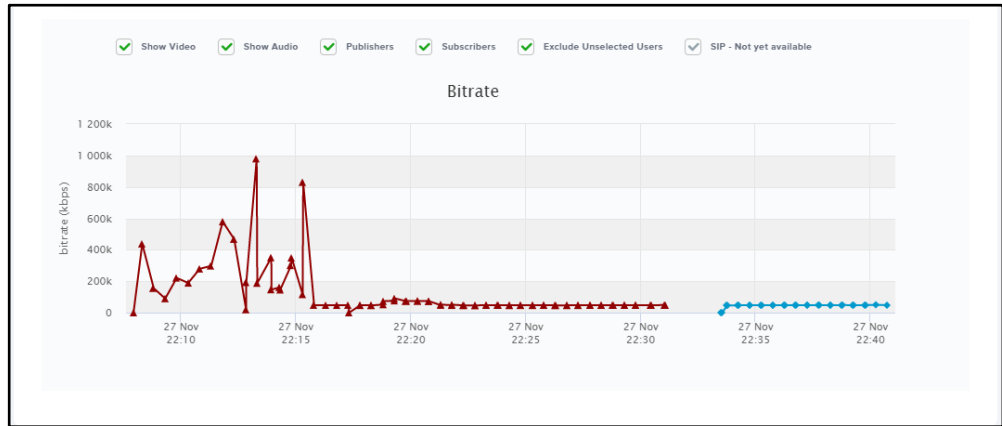


Figure 4.11 Bitrate (kbps) for User 1

Figure 4.11 shows the bitrate for User 1. Maximum bitrate is 1000kbps

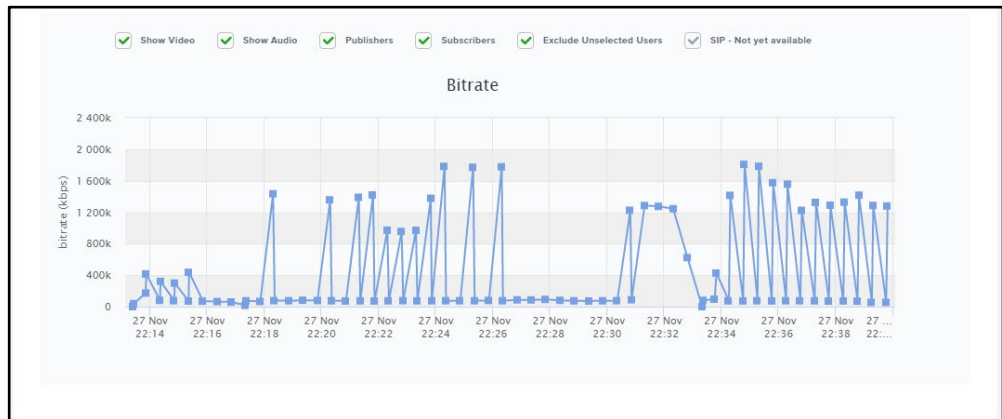


Figure 4.12 Bitrate (kbps) for User 2

Figure 4.12 shows the bitrate for User 2. Maximum bitrate is 1800kbps



Figure 4.13 Bitrate (kbps) for User 3

Figure 4.13 shows the bitrate for User 1. Maximum bitrate is 2300kbps

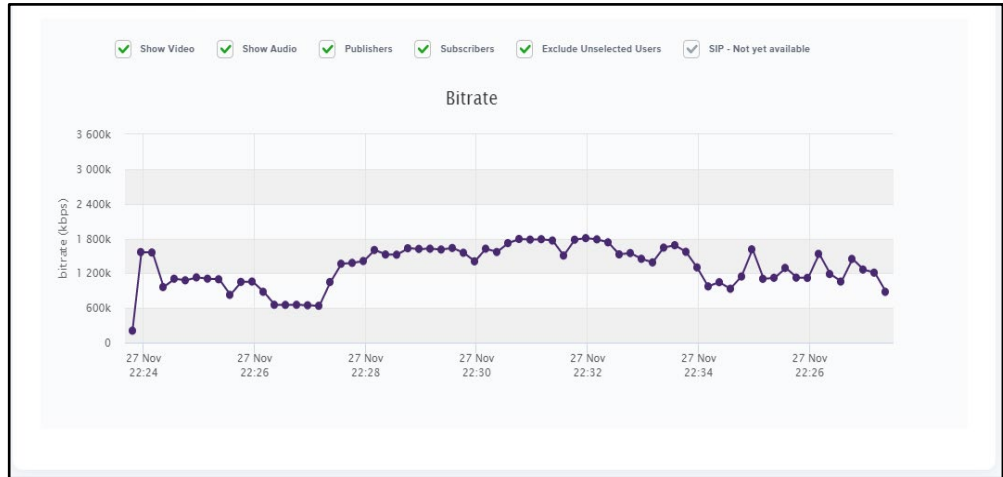


Figure 4.14 Bitrate (kbps) for User 4

Figure 4.14 shows the bitrate for User 1. Maximum bitrate is 1800kbps

4.3.4 Summary of Findings and Analysis of Network Evaluation

Table 4.1 shows the maximum latency, packet loss and bitrate of each users. User 1 has the highest maximum latency, User 2 and 3 have highest maximum packet loss and user 3 has highest maximum bitrate among all the users.





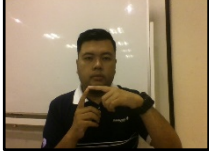
Table 4.1 Maximum Value of Each Network Performance Parameter







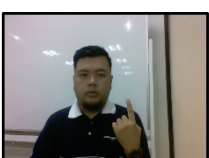
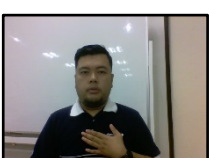
User / Network Parameter	Latency (ms)	Packet Loss (%)	Bitrate (kbps)
User 1	3000	11.0	1000
User 2	400	12.5	1800
User 3	350	12.5	2300
User 4	60	2.5	1800








4.4 KNN Model Accuracy Evaluation

The accuracy evaluation covers both KNN model. For the KNN, 20 images have been trained. Each image is trained 50 times. The version of sign language used is American Sign Language. Note that the trained images have static image background. The KNN model will return the word if only the confidence (threshold) exceeded 90%.

Table 4.2 KNN Model Accuracy Test

No.	Images	Word Assigned	Confidence (%)	Word Predicted
1)		Hello	98	Hello
2)		Opposite	92	Opposite
3)		See	90	Opposite
4)		Who	96	Who
5)		When	95	When

6)		Say	98	Say
7)		Six	95	Six
8)		Five	84	No result
9)		Four	80	No result
10)		Three	94	Three
11)		Two	93	Two
12)		One	94	One
13)		Please	97	Please

14)		Yes	96	Yes
15)		Forget	95	Forget
16)		Hello	97	Hello
17)		Which	96	Which
18)		What	92	What
19)		Where	96	Where
20)		Love	96	Love

4.5 Speech Synthesis Accuracy Evaluation

From interviewee perspective, Speech Synthesis is used to “speak up” the words. After KNN model translates the images into words, the words are then

translated into audio. From interviewer perspective, the spoken words (audio) are translated into text format.

Table 4.3 Speech Synthesis Testing

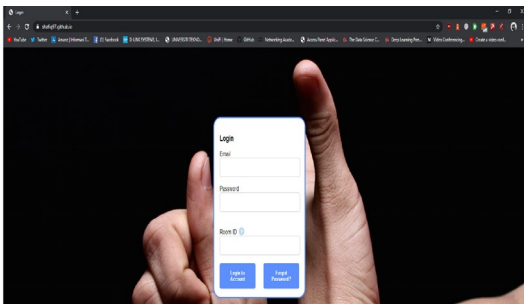
User	Spoken Words (audio)	Speech Synthesis Output (text)
2	Hi, please introduce yourself.	Hi please introduce yourself.
	How was your past employment?	How was your past achievement?
3	Since when you are deaf?	Since when you are dead?
	What do you expect from this company?	What do you accept from this company?
4	How do you see yourself in the next 5 years?	How do you see yourself in the next five years?

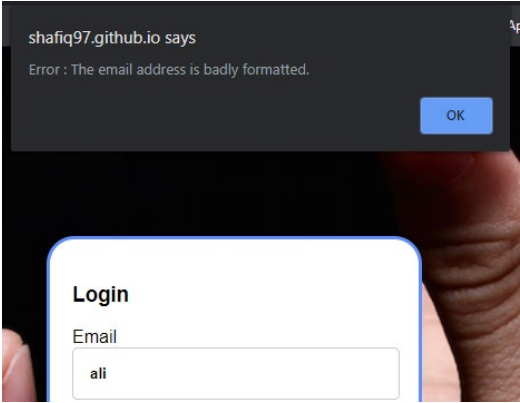
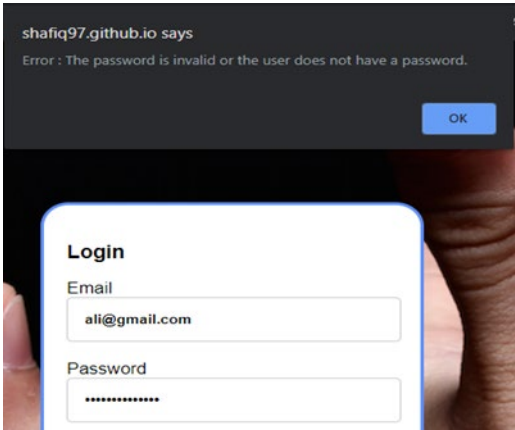
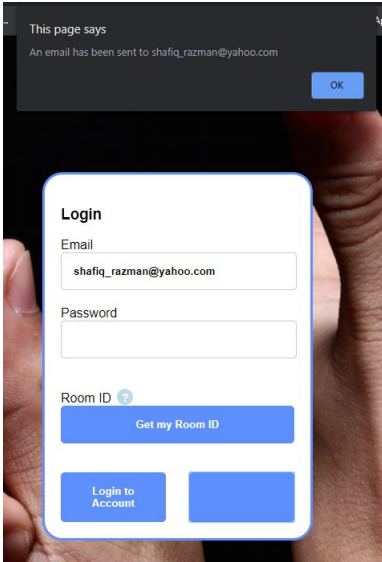
Table 4.3 shows the results of each User 2, User 3 and User 4. Some of the intended spoken words are wrongly translated.

4.6 Functionality Test

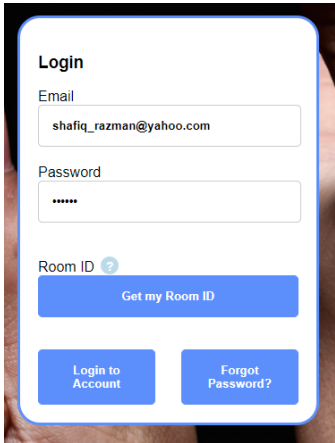
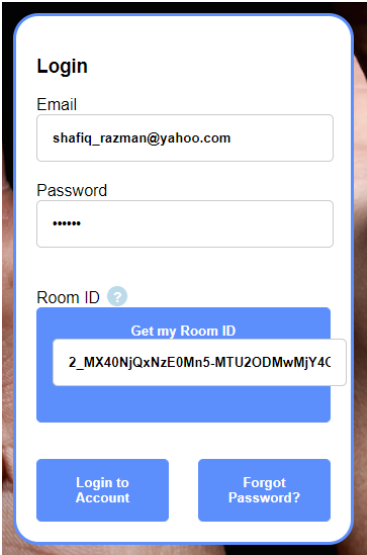
All functionalities of the system are tested. The function then is marked as passed if it succeeds to do the tasks.

Table 4.4 Functionality Test

Task and Function	Output	Remarks and Status
<ul style="list-style-type: none"> • Web hosting • Provide accessibility for users 		<ul style="list-style-type: none"> • The website is accessible. • Passed

<ul style="list-style-type: none"> • Email validation • Remove invalid email 		<ul style="list-style-type: none"> • The email entered is not in email format. • Passed
<ul style="list-style-type: none"> • Password check - Users Authentication 		<ul style="list-style-type: none"> • Password entered for the user is wrong. • Passed
<ul style="list-style-type: none"> • Forgot Password Button is clicked 		<ul style="list-style-type: none"> • Alert Message confirming an email. • Passed

<ul style="list-style-type: none"> • Verifying email sent for password reset 		<ul style="list-style-type: none"> • An email has been sent to reset password. • Passed
<ul style="list-style-type: none"> • Logging in as interviewer. Get room id by entering interviewee's email - Get room id from Firebase, users do not need to enter room manually. 		<ul style="list-style-type: none"> • Result can be examined when Login to Account Button is clicked (all form is entered correctly). • Passed
<ul style="list-style-type: none"> • “Login to Account” is clicked. 		<ul style="list-style-type: none"> • Interviewer interface is loaded with no other user logged in into the session. • Passed
<ul style="list-style-type: none"> • Get user media - camera 		<ul style="list-style-type: none"> • Must be allowed, if not, the webcam will display black screen. • Passed
<ul style="list-style-type: none"> • Get user - mic 		<ul style="list-style-type: none"> • Must be allowed for interviewer so that microphone on the computer can be used. • Passed

<ul style="list-style-type: none"> • Logging in as interviewee 	 <p>The screenshot shows a mobile app interface for logging in. It has a title 'Login' and three input fields: 'Email' (containing 'shafiq_razman@yahoo.com'), 'Password' (masked with dots), and 'Room ID' (with a question mark icon). Below the 'Room ID' field is a blue button labeled 'Get my Room ID'. At the bottom are two more blue buttons: 'Login to Account' and 'Forgot Password?'.</p>	<ul style="list-style-type: none"> • Interviewee email and password is filled in. • Passed
<ul style="list-style-type: none"> • “Get my Room ID” button is clicked 	 <p>This screenshot is identical to the previous one, but the 'Room ID' field now displays a generated alphanumeric string: '2_MX40NjQxNzE0Mn5-MTU2ODMwMjY4C'. The 'Get my Room ID' button is still present below the field.</p>	<ul style="list-style-type: none"> • Room id is displayed based on email. User does not require to enter manually the room id. • Passed

CHAPTER 5

LIMITATIONS, RECOMMENDATIONS AND CONCLUSION

This chapter states the known limitations and suggested recommendations for the future use. It also concludes the project after previous chapters had been made.

5.1 Limitations and Recommendations

Table 5.1 Limitations and Recommendations

No.	Limitation	Recommendation
1.	As the number of words to be trained increase, the number of images had to be trained increase. Requires a huge amount of CPU resources for predicting the word of the image.	The predictions can be made in cloud computing that has higher processing power, thus only return the predicted word. The images could be uploaded to the cloud computer instead added locally to the browser.
2.	The background of gestures during training the image of into word must be static. The KNN Algorithm converts the entire image pixels into RGB format data.	Track only user image instead of training and predicting the whole image.
3.	The system only supports up to 4 users in a session.	The system can support higher number of users for higher scalability.

Table 5.1 shows the limitations and recommendations for future work.

5.2 Conclusion

In conclusion, this project enables deaf people to attend a video conferencing interview. With proper images trainings, KNN image classifier can predict well with confidence of above 90%. After training phase, the deaf (interviewee) makes the sign language gesture and the predictions is broadcasted to the interviewers. The interviewers can ask questions by using the chat box in the system or using computer microphone to speak. The spoken words are then translated into words using Speech Synthesis. The words are also broadcasted

to the other users. The deaf then read the questions and response back by using sign language.

In relative to the objectives, all 3 objectives are achieved. The first objective is to design and develop a web-based system that can host a group video conference for an interview session. The indicator of this objective can be seen in functionality test where 4 people are attending an online interview session.

The second objective is to implement a sign language interpreter algorithm that converts sign language into text and audio using Artificial Intelligence (K-Nearest Neighbour) algorithm and Speech Synthesis. As the result of the KNN algorithm and Speech Synthesis implementation, the KNN model succeed to translate the images to words and the translate words into audio.

The third objective is to evaluate the network performance of the video conference latency, bitrate and packet loss. All the evaluations is completed in Chapter 4: Evaluation and the data is recorded.

References

- Ahmed, M., Idrees, M., Ul Abideen, Z., Mumtaz, R., & Khalique, S. (2016). Deaf talk using 3D animated sign language: A sign language interpreter using Microsoft's kinect v2. *Proceedings of 2016 SAI Computing Conference, SAI 2016*, 330–335. <https://doi.org/10.1109/SAI.2016.7556002>
- Alimudin, A., & Muhammad, A. F. (2019). Online video conference system using WebRTC technology for distance learning support. *International Electronics Symposium on Knowledge Creation and Intelligent Computing, IES-KCIC 2018 - Proceedings*, 384–387. <https://doi.org/10.1109/KCIC.2018.8628568>
- Balan, T., Robu, D., & Sandu, F. (2017). Multihoming for Mobile Internet of Multimedia Things. *Mobile Information Systems, 2017(i)*. <https://doi.org/10.1155/2017/6965028>
- Bottou, L., Curtis, F. E., & Nocedal, J. (2018). Optimization methods for large-scale machine learning. *SIAM Review*, 60(2), 223–311. <https://doi.org/10.1137/16M1080173>
- Bruno, L. (2019). 濟無No Title No Title. *Journal of Chemical Information and Modeling*, 53(9), 1689–1699. <https://doi.org/10.1017/CBO9781107415324.004>
- Butler, K. T., Davies, D. W., Cartwright, H., Isayev, O., & Walsh, A. (2018). Machine learning for molecular and materials science. *Nature*, 559(7715), 547–555. <https://doi.org/10.1038/s41586-018-0337-2>
- Carlucci, G., De Cicco, L., Holmer, S., & Mascolo, S. (2017). Congestion Control for Web Real-Time Communication. *IEEE/ACM Transactions on Networking*, 25(5), 2629–2642. <https://doi.org/10.1109/TNET.2017.2703615>
- Fulfillment, I. P. (2016). *Technology Case Study on Web Real-Time Communications (WebRTC)*. (May).
- Garcia, B., Gortazar, F., Lopez-Fernandez, L., Gallego, M., & Paris, M. (2017). WebRTC Testing: Challenges and Practical Solutions. *IEEE Communications Standards Magazine*, 1(2), 36–42. <https://doi.org/10.1109/MCOMSTD.2017.1700005>
- Garichev, S., & Vedyakhin, A. (2019). Conception and Development Program of National Technology Initiative Center for Artificial Intelligence at MIPT. *Proceedings - 2018 International Conference on Artificial Intelligence: Applications and Innovations, IC-AIAI 2018*, 3–5. <https://doi.org/10.1109/IC-AIAI.2018.8674436>
- Grossi, G., Paglierani, P., Pedersini, F., & Petrini, A. (2018). Enhanced multicore–manycore interaction in high-performance video encoding. *Journal of Real-Time Image Processing*, 0(0), 0. <https://doi.org/10.1007/s11554-018-0834-4>

- Hauervig-Jørgensen, C., Jeong, C. H., Toftum, J., & Christensen, E. C. (2017). Subjective rating and objective evaluation of the acoustic and indoor climate conditions in video conferencing rooms. *24th International Congress on Sound and Vibration, ICSV 2017*, (June 2019).
- Kirmizioglu, R. A., & Tekalp, A. M. (2019). Multi-party WebRTC Services using Delay and Bandwidth Aware SDN-Assisted IP Multicasting of Scalable Video over 5G Networks. *IEEE Transactions on Multimedia, PP(c)*, 1–1. <https://doi.org/10.1109/tmm.2019.2937170>
- Kwan, C., Shi, E., & Um, Y. Bin. (2018). High performance video codec with error concealment. *Data Compression Conference Proceedings, 2018-March*(February), 417. <https://doi.org/10.1109/DCC.2018.00070>
- Lakhal, S., & Khechine, H. (2016). Student intention to use desktop web-conferencing according to course delivery modes in higher education. *International Journal of Management Education, 14*(2), 146–160. <https://doi.org/10.1016/j.ijme.2016.04.001>
- Lawrence, D. R., Palacios-González, C., & Harris, J. (2016). Artificial Intelligence: The Shylock Syndrome. *Cambridge Quarterly of Healthcare Ethics, 25*(2), 250–261. <https://doi.org/10.1017/S0963180115000559>
- Li, G., Ding, Y., Xu, B., & Li, X. (2019). Development and research based on WebRTC mobile phone video communication. *Proceedings of 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2019, (It nec)*, 2487–2490. <https://doi.org/10.1109/ITNEC.2019.8729024>
- Lu, Y.-F., Chen, H.-M., Kuo, C.-F., Tseng, B.-K., & Chou, S.-C. (2019). Container-based load balancing for WebRTC applications. 20–26. <https://doi.org/10.1145/3338840.3355655>
- Luevano, L., Lara, E. L. De, & Quintero, H. (n.d.). *We are IntechOpen , the world ' s leading publisher of Open Access books Built by scientists , for scientists TOP 1 % Professor Avatar Holographic Telepresence Model*.
- Luevano, L., Lopez de Lara, E., & Quintero, H. (2019). Professor Avatar Holographic Telepresence Model. *Holographic Materials and Applications*, 1–16. <https://doi.org/10.5772/intechopen.85528>
- Nalawade, N. R., Nema, S., & Yalampati, S. (2018). Efficient IP-based voice & video communication through session initiation protocol (SIP). *Proceedings of 2017 International Conference on Intelligent Computing and Control, I2C2 2017, 2018-January*, 1–5. <https://doi.org/10.1109/I2C2.2017.8321862>
- Onishi, Y., Tanaka, K., & Nakanishi, H. (2016). Embodiment of video-mediated communication enhances social telepresence. *HAI 2016 - Proceedings of the 4th International Conference on Human Agent Interaction*, 171–178. <https://doi.org/10.1145/2974804.2974826>

- Paglierani, P., Grossi, G., Pedersini, F., & Petrini, A. (2016). GPU-based VP8 encoding: Performance in native and virtualized environments. *2016 International Conference on Telecommunications and Multimedia, TEMU 2016*, 52–56. <https://doi.org/10.1109/TEMU.2016.7551915>
- Pal, D., & Vanijja, V. (2017). A No-Reference Modular Video Quality Prediction Model for H.265/HEVC and VP9 Codecs on a Mobile Device. *Advances in Multimedia, 2017*. <https://doi.org/10.1155/2017/8317590>
- Pigou, L., van den Oord, A., Dieleman, S., Van Herreweghe, M., & Dambre, J. (2018). Beyond Temporal Pooling: Recurrence and Temporal Convolutions for Gesture Recognition in Video. *International Journal of Computer Vision, 126*(2–4), 430–439. <https://doi.org/10.1007/s11263-016-0957-7>
- Prateek, S. G., Jagadeesh, J., Siddarth, R., Smitha, Y., Hiremath, P. G. S., & Pendari, N. T. (2018). Dynamic Tool for American Sign Language Finger Spelling Interpreter. *Proceedings - IEEE 2018 International Conference on Advances in Computing, Communication Control and Networking, ICACCCN 2018*, 596–600. <https://doi.org/10.1109/ICACCCN.2018.8748859>
- R., S., Kalhapure, S., Khatake, A., Gandhi, S., & Jain, K. (2016). Real Time Communication using Embedded System beyond Videoconferencing and towards Telepresence. *International Journal of Computer Applications, 134*(14), 28–31. <https://doi.org/10.5120/ijca2016908134>
- Ramakrishna, M., & Karunakar, A. K. (2017). SIP and SDP based content adaptation during real-time video streaming in Future Internets. *Multimedia Tools and Applications, 76*(20), 21171–21191. <https://doi.org/10.1007/s11042-016-4017-7>
- Rao, N., Maleki, A., Chen, F., Chen, W., Zhang, C., Kaur, N., & Haque, A. (2019). Analysis of the effect of QoS on video conferencing QoE. *2019 15th International Wireless Communications and Mobile Computing Conference, IWCMC 2019*, 1267–1272. <https://doi.org/10.1109/IWCMC.2019.8766591>
- Rixe, J., Carter, K., Sheng, A. Y., Spector, J., Doering, K., Chien, J., & Joshi, N. (2018). Hosting an eConference: Interactive video conference grand rounds between two institutions. *Journal of Education and Teaching in Emergency Medicine, Vol 3, Iss 1, Pp 1-7 (2018)*, (1), 1. <https://doi.org/10.21980/J88P80>
- Romano, J., & Fonseca, A. R. (2019). *WebMesh : A Browser-Based Computational Framework for Serverless Applications*.
- Sabry, E., & Ramadan, R. (2019). Implementation Of Video Codecs Over IPTV Using Opnet. *Bioscience Biotechnology Research Communications, 12*(1), 89–98. <https://doi.org/10.21786/bbrc/12.1/11>
- Shi, Y., & Biswas, S. (2016). Protocol-independent identification of encrypted video traffic sources using traffic analysis. *2016 IEEE International Conference on Communications, ICC 2016*. <https://doi.org/10.1109/ICC.2016.7511096>

- Singh, R., & Dewra, S. (2016). Performance evaluation of star, tree & mesh optical network topologies using optimized Raman-EDFA Hybrid Optical Amplifier. *International Conference on Trends in Automation, Communication and Computing Technologies, I-TACT 2015*.
<https://doi.org/10.1109/ITACT.2015.7492667>
- Sorokin, R., & Rougier, J. L. (2018). Video conference in the fog: an economical approach based on enterprise desktop grid. *Annales Des Telecommunications/Annals of Telecommunications*, 73(5–6), 305–316.
<https://doi.org/10.1007/s12243-017-0613-4>
- Spiteri, K., Urgaonkar, R., & Sitaraman, R. K. (2016). BOLA: Near-optimal bitrate adaptation for online videos. *Proceedings - IEEE INFOCOM, 2016-July*.
<https://doi.org/10.1109/INFOCOM.2016.7524428>
- Staflilov, Z. (n.d.). *Cisco Telepresence Implementation for Telekom 's Corporate Requirements*.
- Underwood, R., Anderson, J., & Apon, A. (2018). Measuring network latency variation impacts to high performance computing application performance. *ICPE 2018 - Proceedings of the 2018 ACM/SPEC International Conference on Performance Engineering, 2018-March*, 68–79.
<https://doi.org/10.1145/3184407.3184427>
- Wu, J., Xu, Y., Li, H., & Tian, J. (2018). Joint design of WiFi mesh network for video surveillance application. *Q2SWinet 2018 - Proceedings of the 14th ACM International Symposium on QoS and Security for Wireless and Mobile Networks*, 140–146. <https://doi.org/10.1145/3267129.3267130>
- Wuttidittachotti, P., & Daengsi, T. (2017). Subjective MOS model and simplified E-model enhancement for Skype associated with packet loss effects: a case using conversation-like tests with Thai users. *Multimedia Tools and Applications*, 76(15), 16163–16187. <https://doi.org/10.1007/s11042-016-3901-5>
- Yang, E. Z., Zhang, L. K., Yao, Z., & Yang, J. (2016). A video conferencing system based on SDN-enabled SVC multicast. *Frontiers of Information Technology and Electronic Engineering*, 17(7), 672–681.
<https://doi.org/10.1631/FITEE.1601087>
- Zhang, S., Donnelly, M. P., Scotney, B. W., Sanders, C., Smith, K., Norton, M. C., & Tschanz, J. (2016). *Impact of Medical History on Technology Adoption*. (June 2018), 98–103. <https://doi.org/10.1007/978-3-319-48799-1>