OPEN ACCESS
Indo-JC Indonesia Journal of Computing
ISSN 2460-9056
socj.telkomuniversity.ac.id/indojc

Ind. Journal on Computing
Vol. 6, Issue. 1, April 2021. pp. 73-84
doi:10.34818/indojc.2021.6.1.5491

The Quality Comparison of WebRTC and SIP Audio and Video Communications with PSNR

Muhhamad Affan Hasby [#1], Aji Gautama Putrada [#2], Febri Dawani [#3]

[#] *School of Computing, Telkom University*
Bandung, Indonesia

[1] affanhasby@student.telkomuniversity.ac.id

[2] ajigps@telkomuniversity.ac.id

[3] febridawani@telkomuniversity.ac.id

Abstract

Video and audio communications have become part of all areas of work. Two real-time communication protocols commonly used for IP-based video and audio communications are Session Initiation Protocol (SIP) and real-time web communications (WebRTC). Both protocols have been widely used in softphone and video conferencing applications. The main objective of this research is to make an analysis of the performance of a client server application for video and audio communications developed by SIP and WebRTC. The SIP system consists of a softphone on the client side using Bria and a FreePBX server, for WebRTC applications, using JavaScript and a server at Node.js. The results showed that the WebRTC audio and video communication provided better quality in terms of PSNR. This is due to the different codecs used between WebRTC and SIP. WebRTC uses VP8 as video codec, SIP uses H.246 as video codec, WebRTC uses G.711 as audio codec, and implemented SIP uses G.729 as audio codec.

Keywords: Video Communication, WebRTC, SIP, PSNR, FreePBX, NodeJs, Video, Audio.

Abstrak

Komunikasi video dan audio telah menjadi bagian dari semua bidang pekerjaan, Dua protokol komunikasi real-time yang umum digunakan untuk komunikasi video dan audio berbasis IP adalah *Session Initiation Protocol* (SIP) dan *web real-time communications* (WebRTC). Kedua protokol tersebut telah banyak digunakan dalam aplikasi softphone dan konferensi video. Tujuan utama dari penelitian ini adalah untuk membuat analisis dari kinerja aplikasi server klien untuk video dan komunikasi audio yang dikembangkan oleh SIP dan WebRTC. Sistem SIP terdiri dari softphone pada sisi klien menggunakan Bria dan server FreePBX, Untuk aplikasi WebRTC, menggunakan JavaScript dan server di Node.js. Hasil penelitian menunjukkan bahwa komunikasi audio dan video WebRTC memberikan kualitas yang lebih baik dalam hal PSNR, Ini karena perbedaan codec yang digunakan antara WebRTC dan SIP, WebRTC menggunakan VP8 sebagai codec video, SIP menggunakan H.246 sebagai codec video, WebRTC menggunakan G.711 sebagai codec audio, dan SIP yang diimplementasikan menggunakan G.729 sebagai codec audio.

Kata Kunci: WebRTC, SIP, PSNR, FreePBX, NodeJs, Video, Audio.

I. Introduction

Network and communication technology is currently developing very rapidly, one of the technological developments is shown by the number of real time communication applications. Real time communication

Received on January, 2021. Accepted on Mei, 2021

contained in applications usually uses SIP or WebRTC, both of which are very popular for applications that have real time communication features. With WebRTC, it is possible to communicate video and audio between browsers in real-time without the use of plugin. WebRTC has a component as mentioned in [1], namely MediaStream which functions to access the user's camera and microphone, RTCPeerConnection which functions to send data when browsers communicate with each other. WebRTC is supported by several browsers such as Google Chrome, Opera, and Mozilla Firefox [2]. WebRTC technology is one of the most extraordinary innovations, because with the WebRTC technology, real-time communication can be done via a browser without having to use plugins, it also makes it easier for developers to create real-time communication features using the web, WebRTC technology makes it easy from the user's side and developers because they both make things easier on both sides. WebRTC has been recognized by international organizations, namely the World Wide Web Consortium (W3C) and the Internet Engineering Task Force (IETF) [3]. SIP is a communication protocol that was often used before the existence of WebRTC, with the SIP protocol we can communicate in real-time using video, audio and text. There are two protocols for SIP, Session Description Protocol (SDP) and Transport Protocol (RTP) or Real-time Transport Control Protocol (RTCP) [4].

In this study the implementation of webRTC uses nodejs as a [5] server, while for SIP it uses FreePBX [6], which is an open source distribution based on web GUI (graphical user interface) that is able to control and manage Asterisk (PBX), an open source communication server. The test will be based on the calculation of the PSNR value to determine the video and audio quality.

II. Literature Review

In 2020, T.P. Fowdur, N. Ramkorun and P.K. Chiniah conducted a research entitled "Performance Analysis of WebRTC and SIP-based Audio and Video Communication Systems". This study discusses the comparison of the performance of WebRTC and SIP from video to audio quality. The result of this research is that the video quality of WebRTC is superior to SIP, while for audio SIP is superior to WebRTC. So it can be concluded that each WebRTC and SIP have their respective advantages [7].

In 2019, Navrattan Parmar and Virender Ranga conducted a research entitled "Performance Analysis of WebRTC and SIP for Video Conferencing". This study discusses WebRTC and SIP communication, researchers see the behavior when each WebRTC and SIP communicate in real-time, the parameters determined by the author are 4 indicators, namely bandwidth, sent packets, VO-Width and jitter. Call quality is better in WebRTC protocol on the same network conditions. RTT is independent on Bitrate. [8].

In 2018, Li Yan conducted a research entitled "Research and Design of Rich Media Communication System Based on WebRTC and SIP". This research discusses the creation of SIP and WebRTC, the experiment carried out is to enter both the WebRTC and SIP protocols into the system that has been created by the author, the result is that the system realizes communication between the WebRTC application and between the WebRTC application and SIP clients. This means that WebRTC can adapt to other communication protocols. [9].

In 2014, Pavel Segeč, Peter Palúch, Jozef Papán and Milan Kubina conducted a research entitled "The integration of WebRTC and SIP: way of enhancing real-time, interactive multimedia communication". This research discusses the technology used to communicate today, and describes the two if they are connected to each other. In this study, WebRTC and SIP are used in real-time communication applications for online learning and for collaboration between users. WebRTC has the advantage of varying support in browser platforms. [10].

III. Research Method

A. WebRTC Design

WebRTC (Web Real-Time Communication) [3] is a real-time browser communication standard using a peer-to-peer architecture. This concerns audio/video peer-to-peer communication between HTML5 [11] browsers. This is an evolution in the world of web applications, because web developers can build multimedia applications without plug-ins. WebRTC requires a STUN (Session Traversal Utilities for NAT) server [12] so that each user of the application can connect to one another. The TURN (The Traversal Using Relay around NAT) server [13] is a feature of the STUN server that places the TURN server as if it were in the middle of two users taking care of sending multimedia such as video and audio. WebRTC has several components so that they can communicate in real-time, each component has its own function, here are the components and functions :

- *getUserMedia*, Used to access the camera and microphone of a user's device.
- *RTCPeerConnection* To connect between browsers, so real-time communication can be done.
- *RTCDataChannel* To allow data transmission to be carried out while communicating.

In order to carry out its functions, the workflow from The unity of this application is designed as in Figure 1.

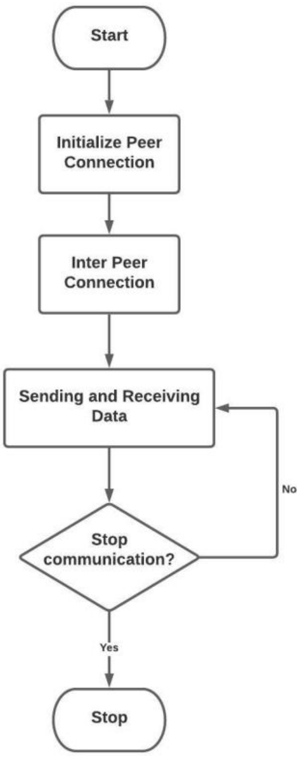

Figure 1: WebRTC Application System Overview

Based on the application workflow in Figure 1, the first process of the system is the user initializes the peer connection, which is obtained when accessing the peer server. After that, when one peer wants to connect with another peer, the peer must send an ID to the other peer who wants to be connected. The server will connect the two peers via STUN Server. each peer will send and receive data when the peer is connected. Lastly, If one of the users wants to stop communication, it can be done by reloading the browser page so that peer communication will be cut off. The data sent is the video and audio captured from the multimedia device.

The development of WebRTC in this research uses assistance software as follows:

- Visual Studio Code 1.54.1 as an IDE for application development.
- Peer.js as a module that supports peer-to-peer connections on the web.
- Node.js as a web server service provider.
- Web Browser Chrome version 88.0.4324.190 64 bit
- VirtualBox 6.1.18, as a virtual machine.
- Operating System 18.04.5 LTS

The following is a virtual machine specification of ubuntu 18.04.5 LTS:

- Processor 2 core.
- RAM 1024 MB / 1 GB.
- hard drive 16 GB.

Server implementation is done by installing some software to support the implementation process. The steps for server implementation are the installation of Node.js which functions as a place for installing the peer module and after that install the peer module. At each stage there is a configuration so that the software can support the running of the peer server.

Algorithm 1 shows the implementation pseudocode of initialize peer connection.

Algorithm 1 Initialize Peer Connection

```
Load peer.js client
if ( then Did not get ID)
    Could not initialize
else
    Peer Connection Occurs
end if
```

When the client first opens the page, the user will access the server IP. When accessing the server IP, the client initialization process occurs, if the initialization is successful the web page will display the client ID, but if the initialization fails, the client ID does not appear.

Algorithm 2 shows the implementation pseudocode to get the user's media stream. This process is carried out after initializing the peer connection.

Algorithm 2 Get Media Stream

```
getUserMedia()
if ( then Success)
    Show it on the web
else
    Show error
end if
```

The media stream obtained is then placed on a web page. The goal is to display the media stream captured by the camera. The pseudocode explanation for getting the media stream is as follows. The first step is to call the getUserMadia function provided by the client peer.js. The second step is to display the stream on the web page.

Algorithm 3 indicates a peer making a call to another peer implementation pseudocode.

Algorithm 3 Calls To Other Peers

```
Call()
if ( then Receive Calls)
    Show other user streams
else
    Call denied!
end if
```

This process is carried out to make calls between one peer to another peer. Peer will call function call to call to other peer, if that peer accepts it will display the stream from other peer. otherwise the call is unsuccessful.

In Figure 2 is the Capture Media Stream interface that has been successfully connected and got the ID from the server.

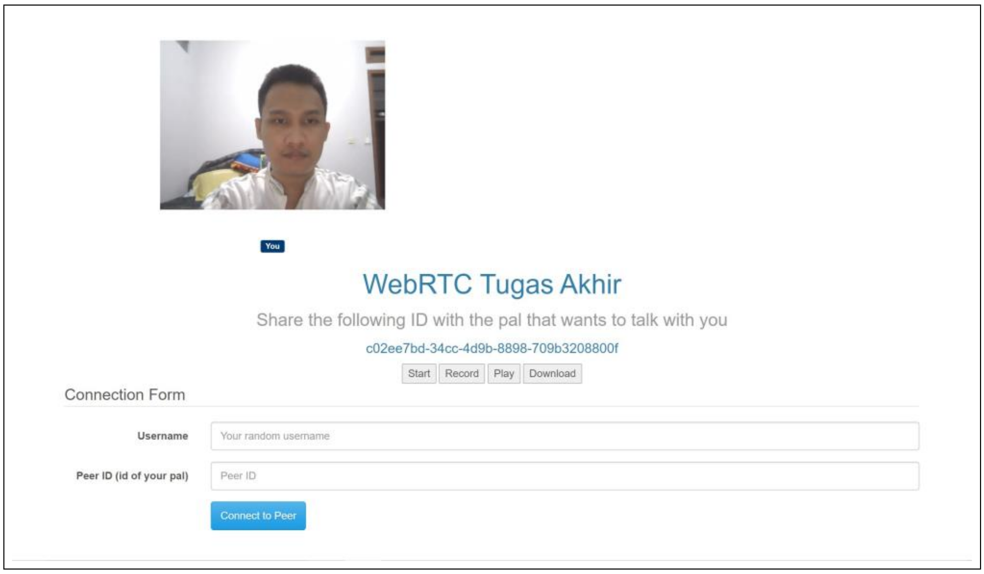

Figure 2: Capture Media Stream interface

Media devices that have permission to access media devices will display the captured media results on a web page.

In Figure 3 is the Inter Peer Connection interface. Before connecting between peers, the ID of the peer to be contacted is sent to the peer, then entered into the peer ID column.

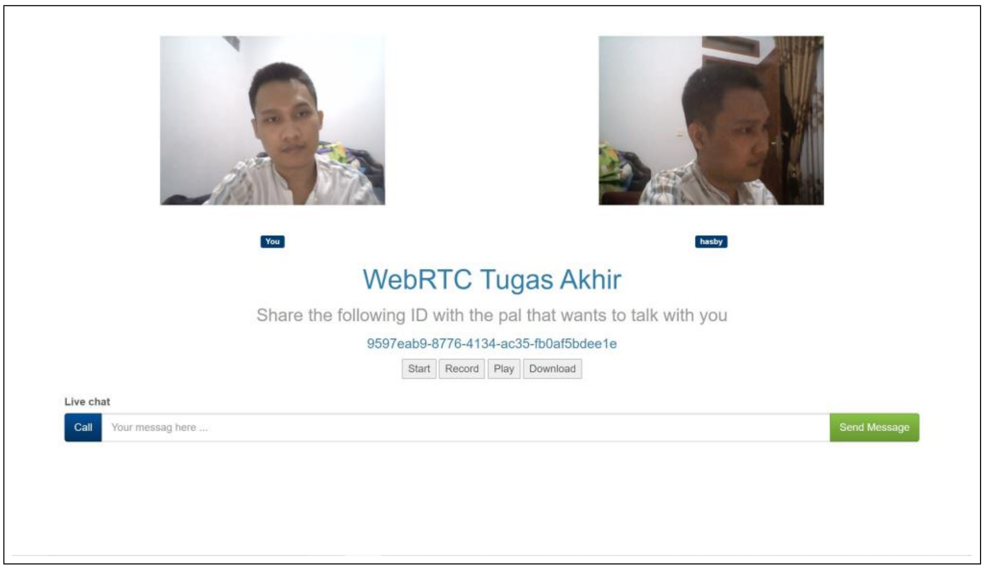

Figure 3: Peer Interface Connection

When the peer is connected, the two peers will send and receive data in the form of video and audio. The media streams from both peers will be displayed on the web page.

B. SIP Design

SIP works with other application layer protocols whose protocols are identified and carry session media, SIP incorporates many elements of Hypertext Transfer Protocol (HTTP) and Simple Mail Transfer Protocol (SMTP) [14]. SIP can make calls with several media, such as making calls using video, audio and text media.

SIP can be used by using several other protocols that are used for call management, some of the protocols used by SIP are the Session Description Protocol (SDP), this protocol determines the codec, media format and communication media during communication, then there is a real time transport protocol (RTP). or Secure real-time transport protocol (SRTP) is used for streaming voice and video media [14] [15].

In order to carry out its functions, the workflow of this SIP unit is designed as shown in Figure 4.

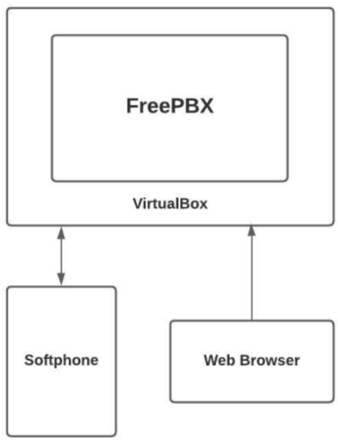

Figure 4: Overview of the SIP Application System

To meet system requirements, the authors divides the system into components. The components are:

1) Virtual Machine
   Serves as a place where the telephone system is installed.
2) Telephone System
   Serves to perform communication settings and provides a communication connection between the connected softphones.
3) GUI PBX
   GUI PBX serves to provide a user interface for the author to perform the necessary configuration of the telephone system, for example, registering a softphone ID number.
4) Softphone
   Serves as an application used to communicate between users. This application will connect to the phone system using the IP server and store the ID number as a unique number that can be contacted by other applications.

The development of SIP in this study uses the following software assistance:

- VirtualBox 6.1.18, as a virtual machine.
- Operating System FreePBX as a communication server.
- *softphone* Bria as communication between users.
- Web Browser Chrome versi 88.0.4324.190 64 bit as a GUI PBX configuration.

In the telephone system that was built, the author uses the FreePBX operating system as a SIP communication server. The author implemented it with virtualBox as a virtual machine. The following is a virtual machine specification for implementing a freePBX communication server:

- Processor 2 core.
- RAM 1024 MB / 1 GB.
- hard drive 16 GB.

With the configuration of internet protocol addresses that have been adapted to the local network.

Before the trial, it is necessary to make an extension to FreePBX as a tool for accounts on the softphone later. This account is useful for identifying each softphone whether it is connected to the server. When the softphone is connected, communication is possible.

Table I: Extension Account Softphone

Extension	Name
111	affan
222	hasby
333	muhammad
444	444
500	500
600	600

In Table I, there are extensions that have been made by the author. There are 6 extensions made, because the test will be carried out by 6 users. Creating an account on the softphone is useful so that the softphone can connect to the server with certain information and the server can contact the softphone when there is a call.

Table II: Softphone configuration

SIP Username	SIP Password	Domain
111	111	192.168.8.114:5160
222	222	192.168.8.114:5160
333	333	192.168.8.114:5160
444	444	192.168.8.114:5160
500	500	192.168.8.114:5160
600	600	192.168.8.114:5160

In Table II there is a softphone configuration, with adjustments to the domain configuration, SIP username and SIP password according to the FreePBX server.

C. PSNR (Peak Signal to Noise Ratio)

PSNR (Peak Signal to Noise Ratio) [16] is a measure that is the ratio between the maximum signal strength and the noise strength. In general, the signal has a fairly wide range of values, so that for simplicity the PSNR is written in decibels (dB). How to calculate PSNR by first determining the Mean Square Error (MSE) value as written in equation 1.

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \tag{1}$$

PSNR is defined as follow:

$$PSNR = 10Log_{10}\left(\frac{MAX^2}{MSE}\right) \tag{2}$$

$$PSNR = 20Log_{10}\left(\frac{MAX}{\sqrt{MSE}}\right) \tag{3}$$

$$PSNR = 20Log_{10}MAX - 10Log_{10}MSE \tag{4}$$

i and j are the coordinates of the image, m and n are the dimensions of the image, I(i,j) represent the stego-image and K(i,j) represent the cover image. MAX the maximum value of the pixel value is 255.

IV. Results and Discussion

This test is carried out to obtain video quality and audio quality results by looking at the calculated PSNR results, when testing the number of client pairs will be added, in order to see the difference in quality from WebRTC and SIP servers when providing services to users. Figure 5 shows the Topology of the system.

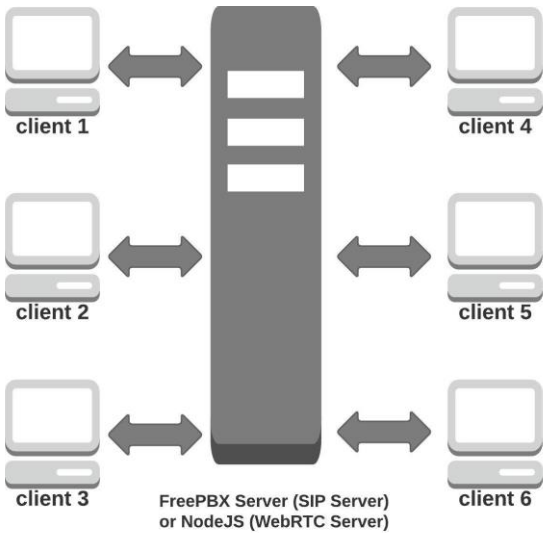

Figure 5: Testing Setup

Overall, 6 laptops were used for testing this time, bringing a total of 3 pairs of clients. In detail, the pairs are Client 1 with Client 2, Client 3 with Client 4, and Client 5 with Client 6. Experiments and tests are carried out on the same LAN.

The first step is to perform a PSNR test for video quality. This test is done by collecting video data on the sending client and the receiving client. The data is then compared with the PSNR calculation. The PSNR value will determine the decline in video quality due to the communication process. This is done for both WebRTC and SIP and is done for 3 different scenarios, 2 clients, 4 clients, and 6 clients. PSNR test results for video quality, can be seen in Fig 6.

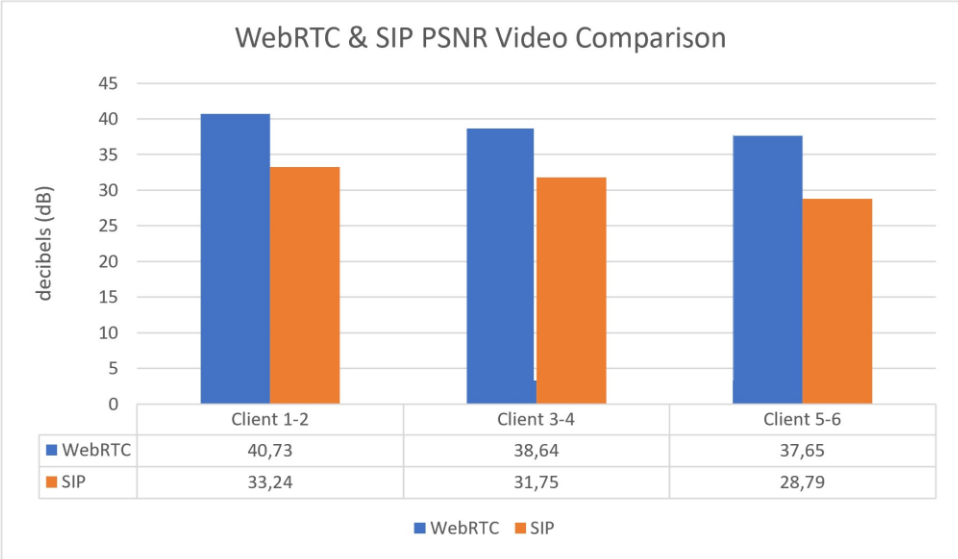

Figure 6: PSNR Video test results

Fig 6 shows that the WebRTC PSNR value for all scenarios is better than SIP. On 1-2 clients WebRTC is higher than SIP video quality. Then when there were 3-4 client additions, the PSNR value on WebRTC decreased by 1.82, and the SIP PSNR value decreased by 1.49. And when 5-6 clients are connected, the SIP and WebRTC PSNR values decrease with a higher slope, namely 28.79 and 33.79, respectively.

WebRTC uses the VP8 codec for video, the VP8 codec has a resolution of 16,384 × 16,384 pixels and has a bitrate of 14 bits. This makes the VP8 codec have no restrictions on the resolution and size of data when sending between users. SIP uses the H.246 codec, the codec is often compared to the VP8 codec, many studies have made comparisons of the two codecs [17]. The above comparison shows the WebRTC codec quality is higher than SIP.

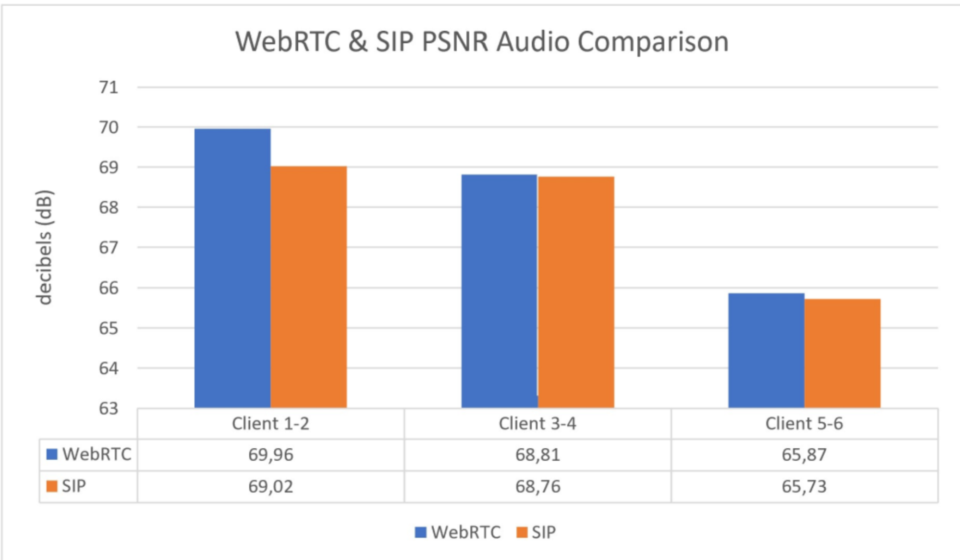

Figure 7: PSNR Audio test results

In Fig 7, It can be seen that WebRTC audio quality is better than SIP audio. WebRTC uses G.711 as the [18] audio codec, while freePBX uses G.729 [19]. G.711 delivers high quality, uncompressed sound, but uses a lot of bandwidth. G.729 is compressed so that it uses less bandwidth at the expense of [20] sound quality. This proves the quality of WebRTC is better than SIP. As more users use the server, the quality of WebRTC and SIP decreases simultaneously. There was no significant PSNR audio gap due to two things, namely the test was carried out locally and the two codecs used. The codec used by SIP is G.729 which performs voice compression to optimize bandwidth, because the test is carried out without any bandwidth limitation on the local network so that there is minimal compression. This results in no significant gaps in the sound test results for Codec G.711 and G.729.

The PSNR for one-two clients is higher than for others Because only two clients are connected to the server and only two clients are communicating. After testing two clients, the third and fourth clients will connect to the server and communicate so that there are four clients connected to the server, after which another test is carried out for four clients, after completion, the fifth and sixth clients will connect to the server. server and communicate so that there are six clients connected to the server and communicating. This is what makes server performance decrease due to the increasing number of clients connecting and communicating.

There are differences in the PSNR results from reference [7] with this study because, for reference there is the Ozeki VoIP SDK, the Ozeki VoIP SDK embeds advanced digital sound in its software such as noise cancellation, adaptive jitter buffering and echo cancellation, while this paper does not use additional features such as Ozeki. That is what causes SIP in this paper to have lower audio quality than WebRTC.

V. Conclusion

This research succeeded in implementing the WebRTC and SIP based video communication design. The comparison test of the quality of the two protocols was carried out using the PSNR metric and 3 different scenarios, namely 2 clients, 4 clients, and 6 clients. The test results show that the video and audio quality of WebRTC is better than the video and audio quality of SIP. After analyzing and benchmarking with other research, it turns out that the type of video and audio codec used affects the communication quality of a protocol. For the future work, there are still many aspects that can be compared, such as network performance, Quality of Service (QoS), and Quality of Experience (QoE). In future studies, the authors suggest that there be studies that compare these aspects so that the advantages of each protocol can be observed.

Acknowledgment

The author would like to thank colleagues who have supported this research and the authors thank God, because for the blessings this research can be completed.

References

[1] Branislav Sredojev, D. Samardzija, and Dragan Posarac. Webrtc technology overview and signaling solution design and implementation. *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1006–1009, 2015.

[2] Boris Magnusson, Thomas Sandholm, and Björn Johnsson. An on-demand webrtc and iot device tunneling service for hospitals. 08 2014.

[3] Salvatore Loreto and Simon Pietro Romano. *Real-time communication with WebRTC: peer-to-peer in the browser*. " O'Reilly Media, Inc.", 2014.

[4] K. K. Tam and H. L. Goh. Session initiation protocol. In *2002 IEEE International Conference on Industrial Technology, 2002. IEEE ICIT '02.*, volume 2, pages 1310–1314 vol.2, 2002.

[5] G. Suciu, S. Stefanescu, C. Beceanu, and M. Ceaparu. Webrtc role in real-time communication and video conferencing. In *2020 Global Internet of Things Summit (GIoTS)*, pages 1–6, 2020.

[6] Hardik Tandel and Dr. Parag Rughani. Forensic analysis of asterisk-freepbx based voip server. *International Journal of Emerging Research in Management and Technology*, 6:166, 06 2018.

[7] TP Fowdur, N Ramkorun, and PK Chiniah. Performance analysis of webrtc and sip-based audio and video communication systems. *SN Computer Science*, 1(6):1–22, 2020.

[8] Navrattan Parmar and Virender Ranga. Performance analysis of webrtc and sip for video conferencing. *Int. J. Innov. Technol. Explor. Eng.(IJITEE)*, 8(9S):679–686, 2019.

[9] Li Yan. Research and design of rich media communication system based on webrtc and sip. Master's thesis, Xi'an University of Science and Technology, 2018.
[10] Pavel Segeč, Peter Palúch, Jozef Papán, and Milan Kubina. The integration of webrtc and sip: way of enhancing real-time, interactive multimedia communication. In *2014 IEEE 12th IEEE International Conference on Emerging eLearning Technologies and Applications (ICETA)*, pages 437–442. IEEE, 2014.
[11] Matthew Hoy. Html5: A new standard for the web. *Medical reference services quarterly*, 30:50–5, 01 2011.
[12] R. Mahy and P. Matthews. Session traversal utilities for nat (stun). 5389, 01 2008.
[13] J. Rosenberg. Traversal using relays around nat (turn): Relay extensions to session traversal utilities for nat (stun). 01 2010.
[14] Alan B Johnston. *SIP: understanding the session initiation protocol*. Artech House, 2015.
[15] Eric Coll and M Eng. *Telecom 101 Telecommunications Reference Book: 2016 Fourth Edition. High-Quality Reference Book and Study Guide Covering All Major Telecommunications Topics... in Plain English.* Teracom Training Institute, 2016.
[16] Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics letters*, 44(13):800–801, 2008.
[17] Yousef Sharrab and Nabil Sarhan. Detailed comparative analysis of vp8 and h.264. pages 133–140, 12 2012.
[18] Daniel C Burnett. Webrtc: Handling media on the web. In *Multimodal Interaction with W3C Standards*, pages 155–169. Springer, 2017.
[19] Adauto Cavalcante Menezes, Toniclay Andrade Nogueira, Edward David Moreno Ordonez, and Admilson de Ribamar Lima Ribeiro. An approach to the performance and efficiency power analysis on embedded devices using asterisk. 2018.
[20] Gaous Afrizal et al. Impact of random and burst packet loss on voice codec g. 711, g. 722, g. 729, amr-nb, amr-wb. In *2018 4th International Conference on Wireless and Telematics (ICWT)*, pages 1–4. IEEE, 2018.