

# A NETWORK PERFORMANCE COMPARISON OF WEBRTC AND SIP AUDIO AND VIDEO COMMUNICATIONS

Sulthan Kharisma Akmal<sup>1</sup>, Aji Gautama Putrada<sup>2</sup>, Febri Dawani<sup>3</sup>

<sup>1,2,3</sup> School of Computing, Telkom University, Bandung, Indonesia  
sulthankakmal@student.telkomuniversity.ac.id

**Abstract**— Multimedia services are one of the internet needs with high data traffic network count since 2019. Two of the multimedia services, Web Real-Time Communications (WebRTC) and Session Initiation Protocol (SIP), have been widely used in applications for conducting video conferencing. The main objective of this research is to analyze network performance by an application with a client-server for audio and video communications developed with WebRTC and SIP protocols. The SIP system uses the FreePBX server, and the softphone application uses Bria. Whereas WebRTC uses JavaScript with servers on Ubuntu using Node.js. The analysis application uses a star topology and runs on a local network using Wi-Fi. After testing, the results show that the throughput, jitter, and packet loss of WebRTC are better than SIP. This result is caused by several factors including the type of codec, the type of platform, and the way of signaling and routing of each protocol.

**Keywords:** WebRTC; SIP; Video Communication; Codec; Network Performance.

## I. INTRODUCTION

Multimedia applications and services are one of the highest needs of the internet with data from calculations in network traffic since 2019 [1]. Of the various existing multimedia applications and services, Web Real-Time Communication (WebRTC) and the Session Initiation Protocol (SIP) protocol are two protocols that are interesting to discuss. WebRTC is a technology that exists in the field of communication using an Application Programming Interface (API), with JavaScript in the browser application. With this technology, WebRTC allows users to communicate using video and audio directly from the web without the need for extensions or additional tools [2] [3].

Currently, WebRTC still uses an API which is designed to communicate from browser to browser

only, because of this architecture, WebRTC is not recommended for communicating with the conference model because it can cause a load on the existing connection in the media API, it can affect user comfort when doing communication.

The SIP protocol is communication in a network between multimedia devices using two protocols, which can be called the Real-time Transport Protocol (RTP) or the Real-time Transport Control Protocol (RTCP) and the Session Description Protocol (SDP). This SIP protocol will be run by the GUI (Graphical User Interface) on the FreePBX server that manages Asterisk (PBX) [4].

This study aims to conduct a network performance comparison of WebRTC and SIP audio and video communications. This study will use WebRTC based on Ubuntu 18.04 LTS and node.js as a server, and for the SIP protocol using FreePBX as a server. This test is conducted to determine the network quality of these 2 applications and has various parameters to be tested as throughput, jitter, and packet loss.

## II. METHOD

### 1. WebRTC Design

For the system on WebRTC to run, the system design for WebRTC is as seen in Figure 1.

As the flowchart in Figure 1 explains, the first step of the system is to ask for permission to access the camera and microphone. This is so that the system can access the camera and microphone during communication. The user can allow or deny the system. If the system has allowed the program to access the camera and

microphone, the peer will initialize it. This will be obtained when the system accesses the peer server. After initialization if another peer wants to connect, the peer must send another ID to connect, the server will connect the two peers via STUN Server. If the connection is allowed between peers, then communication can take place. If one of the users wants to stop, they can reload one of the peers, and then the communication will stop.

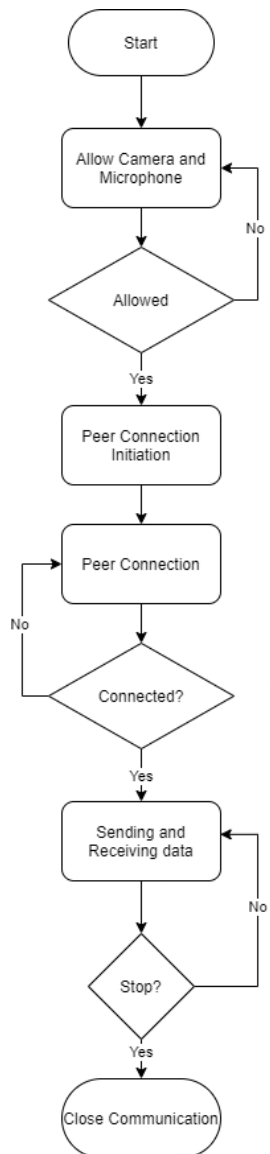


Figure 1. WebRTC System Design

The design and development of WebRTC in this study use several software tools for assistance. The following are the software specifications:

- a) VirtualBox 6.1.18, as a virtual machine;

- b) Ubuntu 18.04 LTS Operating System as a server;
- c) Node.js as Web Server.
- d) Chrome Web Browser version 88.0.4324.190 64 bit for the WebRTC GUI;
- e) The virtual machine specification uses 2 processor cores, 2 GB RAM, and a virtual hard drive of 10 GB.

## 2. SIP Design

For the system on WebRTC to run, the system design for WebRTC is as seen in Figure 2.

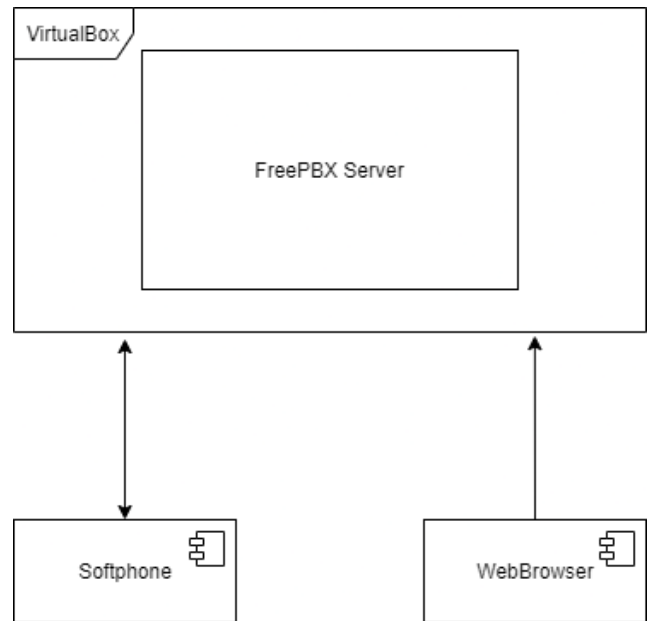


Figure 2. SIP System Design

In Figure 2, first is a virtual machine that is used for the telephone system container, then the telephone system functions to make communication arrangements. After that, PBX functions to display the admin interface that performs telephone configurations, and finally, the softphone is used for users to communicate.

The design and development of SIP in this study use several software tools for assistance. The following are the software specifications:

- a) VirtualBox 6.1.18, as a virtual machine;
- b) FreePBX Operating System as a SIP server;
- c) Bria Softphone as a communication intermediary application and GUI SIP;

- d) Chrome Web Browser version 88.0.4324.190 64 bit for FreePBX configuration GUI;
- e) The virtual machine specification uses 2 processor cores, 2 GB RAM, and a virtual hard drive of 10 GB.

### 3. QoS Parameter

The test method for WebRTC and SIP on a local network uses analysis with Quality of Service (QoS) analysis parameters [11]. Testing was carried out in a limited scope with a peer-to-peer method on 2 clients, 4 clients, and 6 clients [15]. There are 3 QoS parameters, first, there is throughput, which is the data transfer rate measured by the total number of packets received and divided by the length of time for data sampling [13]. The following is the formula:

$$\text{Throughput} = \frac{\text{Packages received}}{\text{Length of Sampling}}$$

Then the second is Jitter, or variation of delay time is the difference in packet arrival to the destination user, this is caused by congestion and lack of capacity on the network, and packets that are not arranged at the transmission [13]. The following is the formula:

$$\text{Jitter} = \frac{\text{Total delay variation}}{\text{Total packet received}}$$

And the last is Packet loss or network loss is the loss of packets sent when transmitting data between service providers to destination users, packet loss can be caused by damaged packets, full network paths (router buffer overflow), and congestion failure on the network [13]. The following formula is used:

$$\text{Packet Loss} = \frac{\text{Data packets sent} - \text{Data packets received}}{\text{data packets sent}} \times 100\%$$

## III. RESULTS AND DISCUSSION

The topology used to perform the analysis is a star topology which consists of 6 devices. This topology is applied to both WebRTC and SIP

testing. The picture of the star topology in question can be seen in Figure 3.

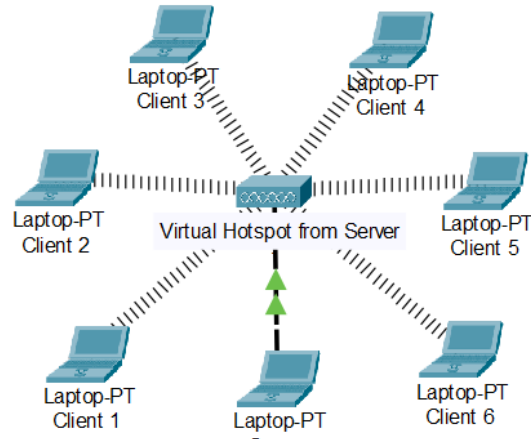


Figure 3. System Network Topology

In Figure 3, there is a virtual hotspot from the laptop server which is used as a communication bridge for all analyzed users.

Measuring network performance in this paper uses the Wireshark application. This testing tool will sniff packets on networks that have traffic [18]. In this experiment, packet sniffing is done on a virtual hotspot device that is used as a bridge between other users.

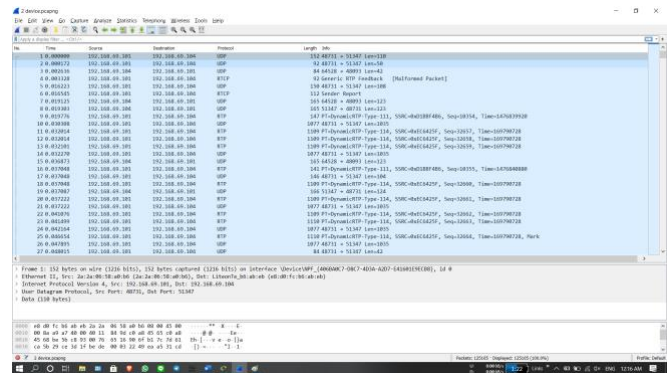


Figure 4. Wireshark capture packet

In Figure 4, some packets have been captured by Wireshark, packets that have been captured and then analyzed are UDP (User Datagram Protocol), in UDP there are RTP and RTCP packets that will be analyzed for network performance. This test lasts for 60 seconds for each test on 2 clients, 4 clients, and 6 clients [18].



Figure 5. Throughput BoxPlot Comparison Between SIP and WebRTC

In figure 5, SIP with 1-2 client testing has an average throughput of 138 KBps, then SIP with 3-4 client tests has an average throughput of 931 KBps, and the last SIP trial is 5-6 clients has an average throughput of 1755 KBps. For WebRTC 1-2 client trials have an average throughput of 524 KBps, for 3-4 client trials the throughput increases up to 1469 KBps, and trials with 5-6 WebRTC clients use a throughput of 2300 KBps.

The Box Plot analysis for SIP shows a slow increase but can use large throughput when client testing is added, this is because the H.264 codec will compress video when the connection is bad and will increase the video quality when the connection is good.

In the case of this local network, the bandwidth on the network is not limited, so this codec does not have a compression effect when it is only used by 1-2 clients, in 3-4 client trials there was also a significant increase in throughput due to the large target bitrate by the H.264 codec. The next trial is to connect 5-6 clients with a significant increase in throughput, which is an increase of 46.9% from 5-6 clients, it's just that when the network traffic increases there will be a decrease in quality by the H.264 codec and the video is compressed by lowering the video bitrate. [10].

Whereas for WebRTC, the increase is seen more significantly because the codec used is VP8, this codec will use all the bandwidth on the network to maintain video quality without compression, so that the video sent will be the same quality as that of the user's camera, this results in The throughput used is greater because there is no compression from this VP8 codec [10].



Figure 6. Jitter BoxPlot Comparison Between SIP and WebRTC

In figure 6, SIP with 1-2 client testing has an average Jitter of 0.82ms, then for SIP with testing 3-4 clients have an average Jitter of 3.45ms, and the last SIP trial is 5-6 clients has an average Jitter of 7.18ms. For WebRTC 1-2 client trials have an average Jitter of 0.55ms, for 3-4 client trials the Jitter increases to 2.02ms, and trials with 5-6 WebRTC clients have a Jitter of 6.14ms.

In the analysis of the jitter test for SIP on the use of 1-2 clients, there is no significant jitter because video compression is not carried out by the H.264 codec, this jitter is due to changes in frame rate because the H.264 codec detects a good connection.

In the SIP 3-4 client trial, there was an increase in jitter with a difference of 2.63ms from the 1-2 client trial, this was caused by changes in frame rate by compressing video by the H.264 codec because the codec detects the connection is increasing the load is quite heavy on network traffic.

In the SIP 5-6 trial, the client jitter increased to 51.9% from the 3-4 client trials, in this test case the H.264 codec will process a frame rate decrease which causes the jitter to increase and be unstable.

In the WebRTC trial, the difference in jitter was not too far from SIP because there was an increase in throughput so that network traffic was getting bigger and caused some packet delay that had to be received by the client because the VP8 codec would continue to ensure that it had to send video quality that matched both the sender and the receiver.

In figure 7, SIP by testing 1-2 clients has an average packet loss of 0%, then for SIP with testing 3-4 clients have an average Packet Loss of

3.65%, and the last SIP trial is 5-6 clients have an average Packet Loss of 6.56%. For WebRTC trials 1-2 clients have an average Packet Loss of 0%, for 3-4 client trials Packet Loss increases to 2.9%, and trials with 5-6 WebRTC clients have Packet Loss of 6.3%.

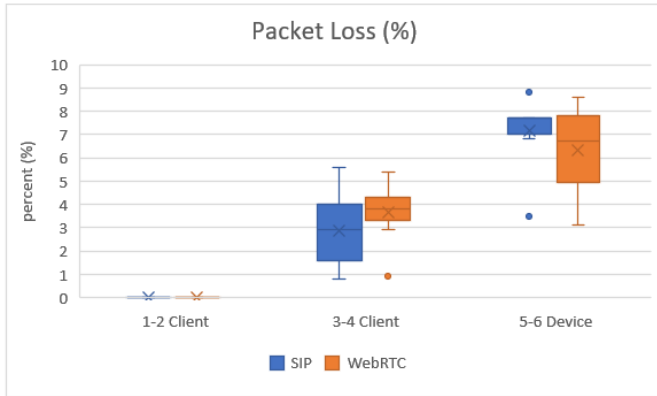


Figure 7. Packet Loss BoxPlot Comparison Between SIP and WebRTC

This packet loss occurs because of the existing obstacles from the codec, so that data transmission will be slightly hampered, according to data from the boxplot, packet loss from SIP and WebRTC does not increase so much because the network used does not have a bandwidth limit, this packet loss occurs due to frame changes.

The rate on the H.264 codec (for SIP) tends to compress due to increased load on the connection and the server, while for VP8 (WebRTC) there is packet loss due to the use of large throughput by the codec so that the server has a transmission load to provide data to the client others [12].

Apart from codec differences, several things cause differences in performance between WebRTC and SIP. One of the most important factors is signaling, in this case, WebRTC uses RTCDATAChannel which is a simple signaling API, it can reduce latency for signaling because messages are sent directly and help reduce server bandwidth for signaling. In addition, signaling in WebRTC is carried out directly from user to user without using proxy routing on the server. While SIP uses Signaling Protocol (Augmented BNF) which uses a proxy so that the routing is more complex. SIP also uses Signaling System No. 7 which has not been updated since 1993, where this method cannot communicate much signaling information.

#### IV. CONCLUSION

In this research, a performance comparison simulation between WebRTC and SIP has been successfully carried out. The test results show that the Throughput, Jitter, and Packet Loss of WebRTC are better than SIP. The average throughput of SIP 2 clients is 138 KBps, SIP 4 clients are 931 KBps, and SIP 6 clients are 1755 KBps. For WebRTC 2 clients the average throughput is 524 KBps, WebRTC 4 clients 1469 KBps, and WebRTC 6 clients 2300 KBps. Then Jitter with the results of 0.82ms SIP 2 clients, 3.45ms SIP 4 clients, and 7.18ms SIP 6 clients. For WebRTC 2 clients 0.55ms, WebRTC 4 clients 2.02ms, and WebRTC 6 clients 6.14ms. Packet loss with an average yield of 0% for SIP 2 clients, 3.65% for SIP 4 clients, and 6.56% for SIP 6 clients. 0% for WebRTC 2 clients, WebRTC 4 clients 2.9%. and WebRTC 6 clients 6.3%. This result is caused by several factors including the type of codec, the type of platform, and the way of signaling and routing of each protocol.

#### V. ACKNOWLEDGMENT

This research was supported by the School of Computing Telkom University. We thank our colleagues from Telkom University who provided insight and expertise that greatly assisted the research, although they may not agree with all of the interpretations/conclusions of this paper.

#### VI. REFERENCES

- [1] R. K. Panta, "Mobile Video Delivery: Challenges and Opportunities," IEEE Internet Computing, vol. 19, no. 3, pp. 64–67, 2015.
- [2] Nayyef, Zinah & Amer, Sarah & Hussain, (2019). Peer to Peer Multimedia Real-Time Communication System based on WebRTC Technology. International Journal for the History of Engineering & Technology. 2.9. 125-130.
- [3] Suci G., Anwar M., Virtualized Video conferencing for eLearning, 14th International Scientific Conference eLearning and Software for Education Bucharest, April 19-20, 2018.
- [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. Suci, 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] Gaus 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.
- [8] 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.
- [9] 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.
- [10] Seeling, Patrick & Fitzek, Frank & Ertli, Gergö & Reisslein, Martin & Pulipaka, Akshay. (2010). Video network traffic and quality comparison of VP8 and H.264 SVC. 10.1145/1878022.1878031.
- [11] Wulandari, Rika. "Analisis Qos (Quality of Service) Pada Jaringan Internet (Studi Kasus : Upt Loka Uji Teknik Penambangan Jampang Kulon – Lipi)." *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 2, no. 2, 2016, doi:10.28932/jutisi.v2i2.454.
- [12] Yousef Sharrab and Nabil Sarhan. Detailed comparative analysis of vp8 and h.264. pages 133–140, 12 2012.
- [13] Cahyadi, Seto Ayom et al. "Analisis Quality Of Service (QoS) Pada Jaringan Lokal Session Initiation Protocol (SIP) Menggunakan GNS3." (2013).
- [14] Pavel Segec, 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.
- [15] G. Suci, S. Stefanescu, C. Beceanu and M. Ceaparu, "WebRTC role in real-time communication and video conferencing," 2020 Global Internet of Things Summit (GIoTS), Dublin, Ireland, 2020, pp. 1-6, doi: 10.1109/GIOTS49054.2020.9119656.
- [16] P. J. Braun, P. Ekler and F. H. P. Fitzek, "Demonstration of a P2P assisted video streaming with WebRTC and network coding," 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, 2017, pp. 576-577, doi: 10.1109/CCNC.2017.7983173.
- [17] Tiamiyu O.A & Garuba A.O. (2019). On The Design And Implementation Of Peer-to-Peer Communication Using WebRTC. Vol. 44, No 1, Aug. 2019.
- [18] Y. dkk, "Metoda Real Time Flow Measurement (RTFM) untuk Monitoring QoS di Jaringan NGN," dalam *Prosiding 14 Konferensi Nasional Teknologi Informasi & Komunikasi untuk Indonesia 3 - 6 Mei 2006 Institut Teknologi Bandung, Bandung, 2006.*
- [19] A dauto 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] 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