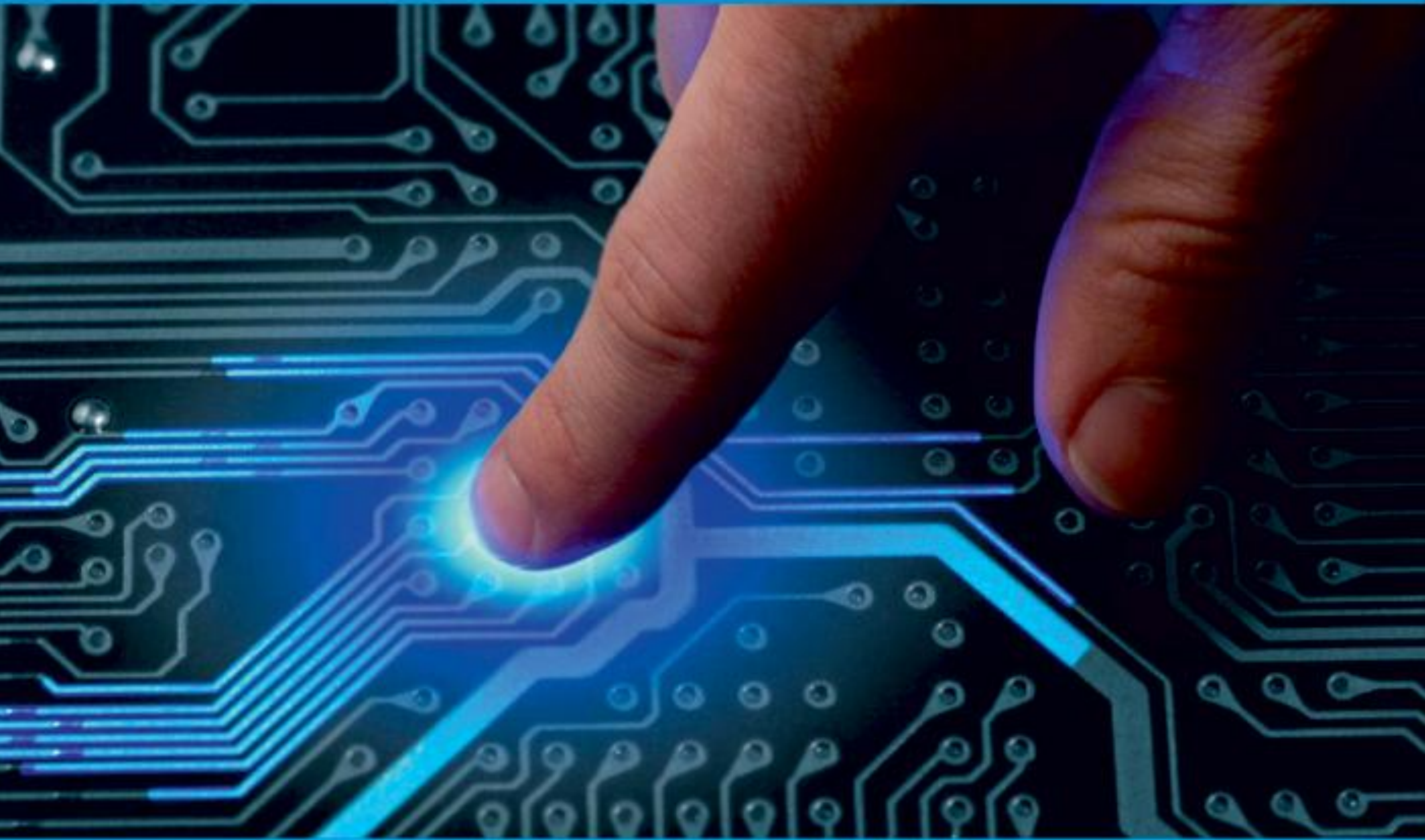
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 10, Issue 5, May 2022

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.165

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005116|

Studying the Performance of WebRTC for Video Conferencing

Rahul Kumar Mohata, Amita Goel, Vasudha Bahl, Nidhi Sengar

Student, Dept of IT, Maharaja Agrasen Institute of Technology, Delhi, India

Professor, Dept of IT, Maharaja Agrasen Institute of Technology, Delhi, Indiain

Assistant Professor, Dept of IT, Maharaja Agrasen Institute of Technology, Delhi, India

Assistant Professor, Dept of IT, Maharaja Agrasen Institute of Technology, Delhi, India

ABSTRACT: With the development of technologies like Web Real-time Communication (WebRTC), we can communicate via video conferencing from a web browser without using any plugins or installing any software. It is an open-source technology and was developed by World Wide Web Consortium. It is a quite popular technology for building web-based video conferencing applications as most of the browsers has support for it. Despite being the handiest tool for developing web-based video calling applications, there are some questions floating around its performance. With this research paper, we will analyze the performance of WebRTC during video conferencing in different browsers and conditions. We will examine the different parameters like bandwidth used, packets loss, latency during a video call to evaluate the performance. We will also analyze the performance of various video and audio codecs available.

KEYWORDS: Real-time Communication; WebRTC; Peer to Peer; Video Conferencing; Web Socket

I. INTRODUCTION

WebRTC is the most popular technology for developing a web-based video conferencing application. Before WebRTC was launched, two browsers communicated with each other in real-time via a web server. With WebRTC, browsers can communicate with each other in peer-to-peer fashion without need of any connection broker in between. It uses the User Datagram Protocol (UDP) with Google Congestion Control (GCC) algorithm to achieve congestion control during real-time communication. It is now widely used in the industry, with an increasing number of businesses switching from native video calling apps to web-based video calling apps. However, there is still a lot of uncertainty about the performance of WebRTC and the bottlenecks it possesses. The primary goal of this analysis is to examine the performance of WebRTC technology under various network conditions like increased latency, packet loss and limited available bandwidth.

WebRTC is a web API developed using a set of libraries and frameworks and it comes built-in with most of the modern browsers. The architecture of WebRTC is fairly simple. It is used for peer-to-peer real-time communication where a peer is a user's browser.

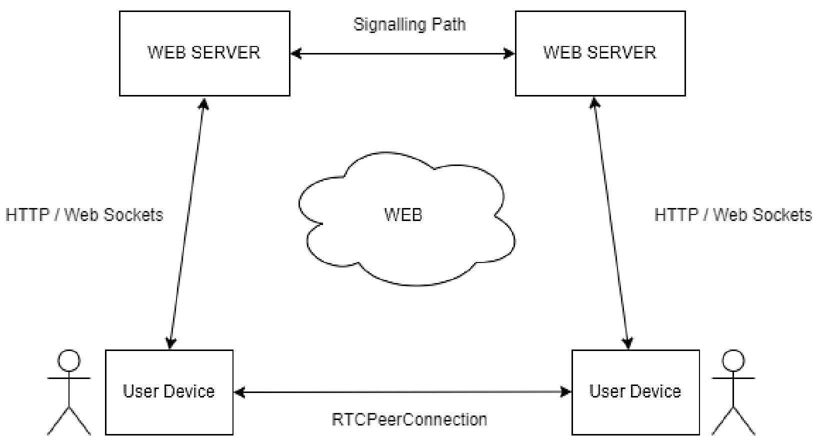


Fig1. WebRTC Architecture

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005116|

There are three primary APIs which are responsible for most of the work required: MediaStream, RTCPeerConnectionand RTCDataChannel.

- MediaStream: It uses the getUserMedia API to get the media stream and adds it to the RTCPeerConnection. It first establishes a connection with the signalling server before configuring for direct communication. It manages all the cameras and microphones present on a device.
- RTCPeerConnection: It uses the onAddStream event to connect to peers and attach audio/video streams. At lower level, it used JavaScript Session Establishment Protocol for making connections between peers. The Secure Real-Time Transport Protocol is used to ensure the security and safety of audio and video delivery. Secure Real-Time Control Transport Protocol is also included for added security.
- RTCDataChannel: It's used to convey data with video and audio streams. It's employed to create a two-way bidirectional channel between two peers. This channel is used to send and receive further data. For data transmission security, it employs the Stream Control Transmission Protocol (SCTP). Multiplexing, flow management, and reliability are among the TCP features added.

II. Related Work

In the past few years, WebRTC has been an active topic of research. There have been researches conducted around using WebRTC for peer-to-peer real-time communication. There have also been studies performed around designing and architecting scalable and performant video conferencing applications based on WebRTC. In the post pandemic world, online teaching has been adopted globally but not everyone has access to high-speed internet connectivity. To mitigate this problem, researches have been done around building WebRTC based platforms that uses extremely low network bandwidth. Studies have also been done around the security and encryption of audio and video streams with WebRTC and comparing it with most other VoIP services.

III. Methodology

A. *Experimental Setup*

Hardware:

A Laptop with Intel Core i7 processor operating at 4.4GHz with 16GB RAM and 512GB SSD and an Intel Core i5 laptop running at 3.4GHz with 8GB RAM make up the basic configuration. Both computers run a WebRTC-based video calling application written using Node and React.

Browser setup:

The current version of Google Chrome browser with the OPUS audio codec and VP8 video codec is used for most experiments since it is leading with its google cloud congestion implementation and has the option to insert a custom media feed.

Network limiter:

Dummynet is a network simulator that can be used to simulate various network properties like latency, packet loss , limiting bandwidth.

B. *Performance Analysis*

Impact of increasing Latency:

In general, latency refers to the delay between when a device captures a video frame and when that frame is shown on the end user's display. We try to add some latency in both uplink and downlink directions. We tried adding 100ms, 200ms, and 300ms latency. The results of the test shows that increased delay has no effect on the data rate or quality of the stream.

Impact of increasing Packet Loss:

When one or more packets of data travelling across the network fail to reach their destination, packet loss occurs. For both transmitted and received packets, we tested by dropping a particular percentage of all packets. The results of the test reveal that when packet loss occurs, the system reduces the packet sending rate.

Impact of limiting Bandwidth:

The quantity of data carried from one point to another inside a network in a certain amount of time is referred to as bandwidth. We ran tests with bandwidth limits of 1200kbps, 900kbps, and 600kbps on both the uplink and downlink. When bandwidth is constrained, WebRTC consumes 75% of the available bandwidth and maintains a steady transmission rate.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005116|

Impact of using different Video Codecs:
A video codec is a piece of software that allows you to compress and decompress digital video. For analyzing the performance of different codecs, we tested the VP8, VP9, and H.264 video codecs which are the most popularly used codecs. The findings reveal that VP9 functions similarly to VP8, however H.264 struggles to maintain a steady data rate.

IV. RESULTS& DISCUSSION

After performing various experiments, it is clearly observed that WebRTC adjusts the data rate based on different network effects as expected. If we limit the bandwidth, WebRTC used 75% of the bandwidth available. If there is packet loss, it reduces the sending data rate depending upon how many packets are lost. If we increase the latency in sending the data, there is no effect on the data rate and stream quality. When packet loss happens at the same decreased capacity as when the bandwidth is restricted, the framerate is more strongly affected than the video stream's resolution.

Fig2. Impact on data rate with increasing Latency

Latency (in ms)	Average Data rate (in kbps)
No Latency	1500
100	1500
200	1500
300	1500

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005116|

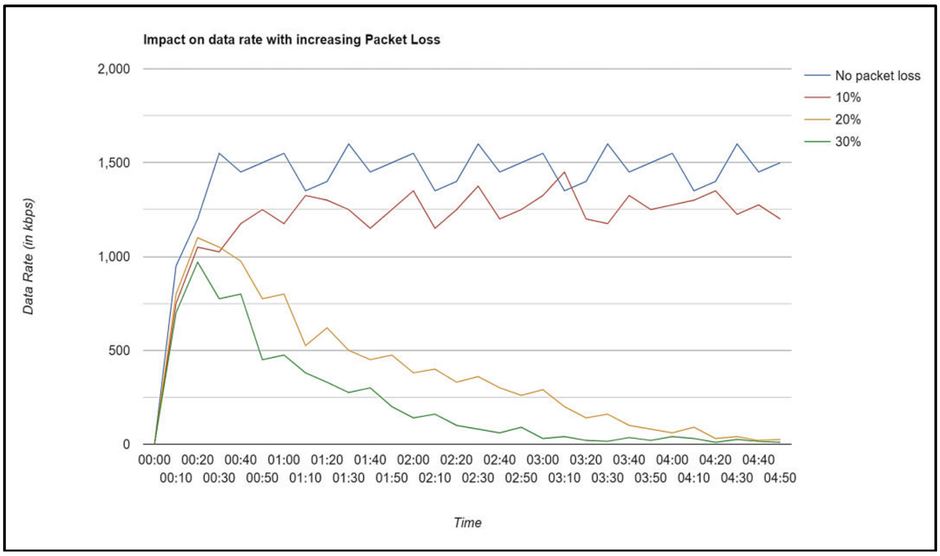


Fig3. Impact on data rate with increasing Packet loss

Packet Loss (in %)	Average Data rate (in kbps)
No packet loss	1500
10%	1250
20%	250
30%	50

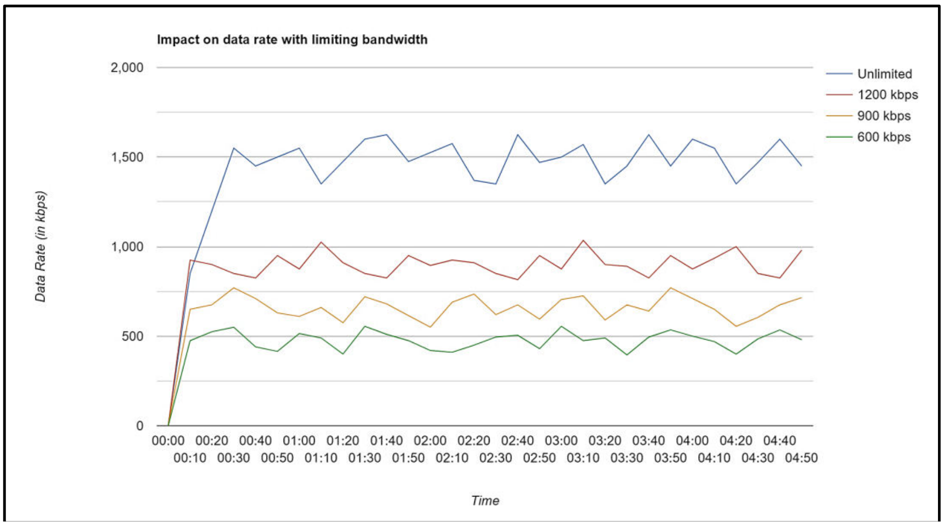


Fig4. Impact on data rate with limiting bandwidth

Bandwidth limit (in kbps)	Average Data rate (in kbps)
Unlimited	1500
1200	900
900	675
600	450

International Journal of Innovative Research in Computer and Communication Engineering
| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |
|| Volume 10, Issue 5, May 2022 ||
| DOI: 10.15680/IJIRCCE.2022.1005116|

Fig5. Impact on data rate with different video codecs

Video codec	Average Data rate (in kbps)
VP8	1500
VP9	1500
H.264	1300

V. CONCLUSION AND FUTURE WORK

WebRTC was launched in 2012 and it has evolved a lot in the last decade. Initially in 2012, only 2% of the browsers supported WebRTC but today it is supported by 90% of web browsers. WebRTC under the hood uses User Datagram Protocol (UDP) which is a connection less protocol used at the transport layer of OSI model. Although, UDP is faster than TCP, it is unreliable. WebRTC uses Google Congestion Control (GCC) algorithm to control the data rate when congestion occurs in the network. We were successful in evaluating WebRTC behavior in a variety of network scenarios, and the results demonstrate that increasing packet loss has a significant influence on data throughput. We hope that our research will aid developers in creating WebRTC-based apps that are both reliable and performant. Our findings will also aid researchers in the development of algorithms to minimize performance deterioration when packet loss increases. Although we attempted to investigate a few network circumstances, there are more scenarios that may occur in the actual world and could be the subject of a future research.

REFERENCES

1. Mohata, Rahul & Goel, Amita&Bahl, Vasudha &Sengar, Nidhi. (2021). Peer To Peer Real-Time Communication Using WebRTC. International Journal of Scientific Research in Computer Science, Engineering and Information Technology. 178-183. 10.32628/CSEIT217647.
2. Nayyef, Zinah Amer, Sarah Hussain, Zena. (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. Ranga, Virender & Parmar, Navrattan. (2019). Performance Analysis of WebRTC and SIP for Video Conferencing. International Journal of Innovative Technology and Exploring Engineering. 8. 10.35940/ijitee.I1109.0789S19.
4. G. Suciu, S. Stefanescu, C. Beceanu and M. Ceaparu, "WebRTC role in real-time communication and video conferencing," 2020 Global Internet of Things Summit (GIoTS), 2020, pp. 1-6, doi: 10.1109/GIOTS49054.2020.9119656.
5. Azom, Edim&Dunka, Bakwa. (2017). A Peer-To-Peer Architecture For Real-Time Communication Using Webrtc. Volume 4. 2394-4404.
6. Bhagatkar, Nikita &Dolas, Kapil & Ghosh, Ratan & Das, Sajal. (2020). An integrated P2P framework for E-learning. Peer-to-Peer Networking and Applications. 13. 10.1007/s12083-020-00919-0.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 5, May 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1005116|

7. https://eytanmanor.medium.com/an-architecturaloverview-for-web-rtc-a-protocol-for-implementing-videoconferencing-e2a914628d0e
8. https://webrtc.github.io/webrtc-org/architecture
9. https://developer.mozilla.org/enUS/docs/Web/API/WebRTC_API
10. https://www.w3.org/TR/webrtc

INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 8.165

doi crossref

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

Scan to save the contact details