Volume: 9 Issue: XII Month of publication: December 2021

DOI: https://doi.org/10.22214/ijraset.2021.39277

Video Conferencing WebApp

Sanjay Majhi[1], Ratul Hans[2], Naman Ahuja[3], Sudha Narang[4]
[1, 2, 3, 4]CSE, MAIT

Abstract: During the last few years, video conferencing has become very popular and very reliable as a tool to bridge the gap where travel is not an option. And the COVID-19 epidemic has also led to lockdown orders that have led to dramatic changes in the way people work. The number of people working in the home (WFH) had increased significantly during the pandemic. The need for distance learning has also increased and has become a compulsory education system in the midst of this current situation. The Companies are also adopting an innovative recruitment process at such time. So to address this issue, our project aims to build a conference app that helps to provide communication between people through audio conferencing, video conferencing, screen sharing and messaging in real time.

In this, we have created group video chat with the help of WebRTC technology and socket programming. Also we have added real-time chat feature and screen share feature. We had created the web app using Jquery for front end and node.js express.js for signaling server and real time database of Firebase for storing chats and user information. WebRTC helped us to create peer to peer connection and with the help of sockets we have done transfer of sdp packets and ice candidates. We have discussed extensively about them in our paper.

Keywords: Video conferencing app, sockets, webrtc, peer-to-peer, realtime data transfer

I. INTRODUCTION

Video Conferencing is a web application which we have developed as part of our B-Tech 4th Year Minor Project. It provides users with lag free video calling experience. It would be very helpful in connecting with people who live far from each other by helping them to communicate via Video Chat or using real-time instant messaging. It is also very helpful in connecting people who live far from each other by helping them to communicatevia Video Chat. Usually, the users having poor network connection will find the mobile app to be very useful since it has been built to be consume less data and work even in low bandwidth.

Video Conferencing is a way of communication between two or more users in which audio, video and data signals are being transmitted electronically to enable interaction bwtween the users. In comparison to audio conference, everyone involved can see the facial expressions and body language that are so important in the way we communicate. Video conferencing can be implemented through a number of technologies. Few of them are hardware based while others are software based.

Video conferencing is the ability to transfer a live video stream got from a computer camera to the all of the individuals. The video conference can be between two users, or the conference can link multiple users. Using this, individuals in different locations can see and hear each other progressively.

And in recent times, the scope and usage of Video Conferencing app has extremely widened as well. Primarily, it is used in business and multi-national companies. It easily provides a great way to get mobile staff involved, reduce travel-related costs, and help in better communication between businesses, partners, or clients. Video conference also made a name of itself in the world of education. Teachers are now able to use this technology to spread knowledge across a large range of fields. Online courses, interactive visual classes, and other uses of these technologies helped the education sector and actually improve how we can disseminate knowledge and promote higher education. Moreover, its demand has also been increased in the health care industry. Whether remote physicians communicating with each other to determine treatment strategies, or physicians treating patients remotely via video communication, a video conference has expanded health care options around the world.

Thus, due to its growing popularity and demand in the recent times, we have decided to go with developing our video conferencing web application which primarily focuses on providing a platform where people can communicate in real-time over video and audio and can also chat with other in real-time

II. LITERATURE SURVEY

The most important factor was to understand the basics of WebRTC in detail and to gain a technical understanding of how WebRTC is built. We got this information by reading the official WebRTC documentsand examples of code provided on WebRTC.org.

To learn how WebRTC application development happens, we explored various forums and blog posts where software developers shared their knowledge about WebRTC development.

Our conclusion after all this is that there are three main design patterns used in the development of WebRTC applications.

1) Full mesh topology
2) Star topology with help of MCU in the root.
3) Star topology with help of SFU in the root.

Appear.in, one popular WebRTC-based conference application, uses the full mesh. Discord, one VOIP app dedicated to gamers, uses a custom SFU solution and is free from peer to peer technology. There is also a lot of research done about these design patterns and experimental hybrids among these.

A. Full mesh

Full mesh is the most used topology in WebRTC application. In this case, all peers make a connection with every other peer in the network, thus have a limit on count of peers.

B. Multipoint Control Unit

The MCU represents the Multipoint Control Unit and is based on "star" topology. The MCU is a central unit and operates as a gateway for all peers. As a result, all streams sent between peers have to first pass through the MCU unit before reaching other peers.

C. Selective Forwarding Unit

The Selective Forwarding Unit, also acts as a central server that transmits individual peer streams. An important comparison is that SFU does not do mixing of streams, it only passes them. Therefore, the individuality of each stream is maintained, and the delay is reduced compared to the MCU. Since there are both pros and cons of both design patterns, we are proposing a solution where we use combination of them, to get all benefits in the same application and satisfy our research.

III.PROPOSED ARCHITECTURE

To make the video conferencing webapp, we decided to work on below technologies :

A. WebRTC

WebRTC is a very popular and open source software which helps in providing web browsers and mobile applications with real-time communication and easy-to-use editing tools. With WebRTC, we can add real-time communication ability to our application . It supports vision, audio, and data to be shared among peers, allowing developers to create powerful voice and video based communication solutions.

B. Socket.io

Socket.IO is a popular JavaScript library which is used for providing real-time data transmission capability to web applications. It also enables bidirectional data transfer between web clients and servers. It will help us in making connection between client and signalling server.

C. JQuery

JQuery is a popular JavaScript library which we will be using as it simplifies HTML DOM access by providing a somple t use syntax.

D. Node.js

Node.js is a runtime environment for javascript that uses V8 engine and helps to run JavaScript code without a web browser that is directly on server operating system. We will use this to make our signalling server.

E. Express.js

Express.js is a framework for nodejs backend development. It gives us feature like middlewares, add most of the boiler plate code for making the server and gives us a simple way to add routes and controllers to serve the requests from clients.

F. Realtime Database Firebase

Firebase Realtime Database is a cloud-based database. Data is stored in the form of JSON and synced in real time to all connected clients. We are using this to store the data of users and the rooms created by users. It helps in fast access of data from database to server.

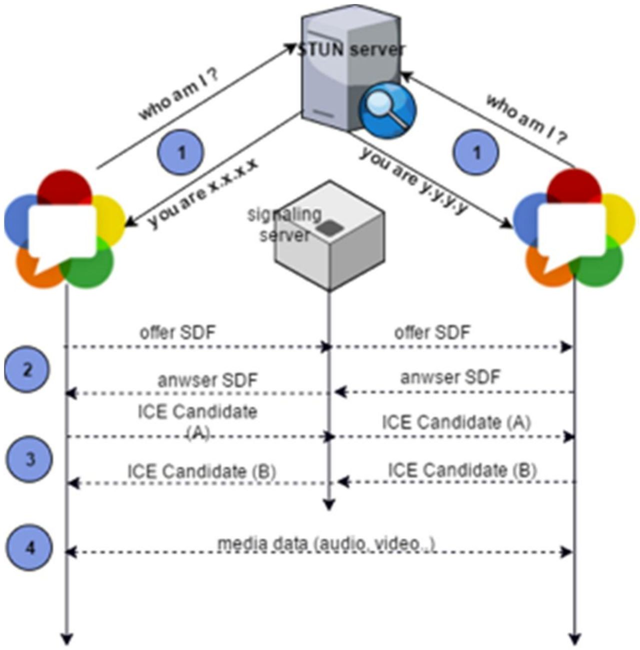

Fig. 1 WebRTC Architecture

G. Signaling server

It is used for establishing a WebRTC connection between two devices, as it acts as a way to transfer the SDP offer and SDP answer between the clients and will also help in transfer of ICE Candidates from one client to another. And it will also help us to get the user informations.

H. STUN server

The STUN server allows clients to find their public address, type of NAT behind which they are present and network node associatedwith a local NAT port. The STUN server is connected to the UDP 3478 port, however, the server will prompt clients to perform tests on another IP and port number as well.

I. SDP

SDP stands for Session Description Protocol, is a very important and widely used standard, which is used for defining multimedia content for connections such as resolution, formats, encryption, etc. so that both peers can communicate to each other when data is transferred. This is,in fact, the metadata that describes the content and not the content of the media itself.

J. ICE Candidate

It is a framework for allowing web browsers to interact with peers browsers. There are many reasons why direct communication from Peer A to Peer B does not work. It needs to bypass firewalls that can block connectivity, provide us with a unique address if as in most cases our device does not have a public IP address, and transmit data via a server if our router does not allow us to communicate directly with peers. ICE uses STUN and TURN servers to accomplish this.

K. NAT

NAT is used to provide our device with a public IP address. The router will have a public IP address and every device connected to it will have a private IP address. Requests will be translated from the device's private IP to the public router IP with a unique port. That way you do not need a separate public IP for each device but can still be found online.

L. TURN Server

Some NAT operators use a restriction named ‘Symetmetric NAT’. This means that the router will only accept communications from our pre-connected peers. The Traversal Using Relays around NAT (TURN) is intended to bypass the Symmetric NAT limit by opening a connection to the TURN server and transmitting all information about that server.

M. Process flow

1) First, peer A calls Google’s free STUN server to know it’s public IP address and type of NAT it is present in.
2) In the same way, peer B also request any of Google’s free STUN server and in response it gets it’s IP address and type of NAT it is in.
3) Now, peer A want to call peer B, so it need to send its SDP to peer B that can be done with the help of signallingserver to which both are connected. This signalling server is made with the help of nodeJs and sockets.
4) So there will be transfer of SDP from peer A to peer B via server, then peer B accepts it and make an answer SDP and send to peer A via the same signalling server.
5) After SDP transfer, ICE candidates generated from STUN server for both peers are exchanged by them through signalling server.
6) After that, both peers can make direct connection between them without the need of signalling server. The rest of the communication will be done as continuous streams of data from one peer to another directly without anyone in between achieved with the help of webRTC technology.
7) Whenever a new peer wants to connect to this conference, SDP’s will be exchanged between all the connected peers and this new peer, then all the ice candidates will be shared between all connected peers and this new peer. Thus direct communication will be established between all of them.
8) After a limit of 10 users, this mesh topology will be converted to a star topology with the help of SFU in a smooth manner.
9) In this way, connection is established for direct communication with the help of WebRTC.

IV. CONCLUSIONS

Video conferencing can lead to a dual approach, giving students more responsibility for their learning, working in groups, and doing teaching activities, all of which can benefit traditional teaching, but the video conference offers the opportunity to implement them. It does not replace the use of print or other methods used in the thought process. It can be used to encourage construction and its real use lies in encouraging dialogue and increasing the breadth of dialogue. And with the use of WebRTC and the SFU combination, video conferencing can be made much smoother and more comfortable.

V. ACKNOWLEDGMENT

It gives us great pleasure to express our deepest gratitude and sincere gratitude to our esteemed guide Ms. Sudha Narang, Assistant Prof., MAIT, for their valuable guidance, encouragement and assistance in completing the project. Their helpful suggestions for all this work and co-operative behavior were warmly welcomed. We also wish to express our gratitude to our parents and family members whose blessings and support have always helped us to face the challenges ahead.

REFERENCES

[1] WebRTC at https://webrtc.org/getting-started/overview
[2] Socket.io at https://socket.io/docs/v4/
[3] Stackoverflow at https://stackoverflow.com
[4] w3schools at https://www.w3schools.com
[5] FreeCode Camp at https://www.freecodecamp.org/
[6] “Signaling and video calling,” MDN Web Docs. at https://developer.mozilla.org/enUS/docs/Web/API/WebRTC_API/Signaling_and_video_calling.
[7] “What are STUN, TURN, and ICE?” Twilio at https://www.twilio.com/docs/stun-turn/faq
[8] “The design and architecture of a WebRTC application” at https://www.diva-portal.org/smash/get/diva2:1480111/FULLTEXT01.pdf