Real Time Information and Communication Center based on WebRTC

Himanshu Bhardwaj[1], Adarsh Lunthi[2], Hitesh Bhat[3], Karan Singh Rawat[4], Aashita Chabbra5

[1-4]Student, Department of Information Technology ADGITM, Delhi, India
[5]Assistant Professor, Department of Information Technology ADGITM, Delhi, India

---***---

Abstract - *WebRTC is a free, open-source project that provides web browsers and mobile applications with real-time communication via simple application programming interfaces. WebRTC technology that offers high eminence RTC applications being established for web and mobile platforms and permit them to interconnect via API's and similarly with a set of practices. WebRTC deliberations to be focused between at least two end points by means of program based versatile/work area applications or gadget local portable applications. WebRTC consists of several interrelated APIs and protocols which work together to achieve this. The documentation you'll find here will help you understand the fundamentals of WebRTC, how to set up and use both data and media connections, and more. This paper explains about enabling online students to have live audio/video chats with the other friend and teacher so that they communicate well while explaining the problem to the other friend. The paper also proves such competence by leveraging evolving expertise like WebRTC and WSC.*

Key Words: WebRTC, RTC, WSC, Real Time, API

1. INTRODUCTION

Real Time communication (RTC) is a mode of communication, where user can exchange their information without transmission delay. RTC generally uses peer to peer transmission except multicasting and broadcasting. WebRTC is one of the communication standards that is developed by the World Wide Web Consortium in close collaboration with the WebRTC standard, further developed by the Internet Engineering Task Force (IETF) . WebRTC performs their entire task in the lower protocol layer, and it enables the task entrenching and its functions in different websites and applications. The standard of WebRTC explains the real time communication incompatibilities, a very common and general phenomenon or say problem. In present scenario, for making any type of audio or video chats with the help of a computer system, users are required to have a useful software or they need to create and maintain their account on various available sites. WebRTC influences the recent development in which the application of web browser facilitates the communication within browser-to browser in which there is no requirement of downloading the software or registering oneself. All the potentials, essential for providing support for the WebRTC standards are readily available with the browsers. WebRTC standardize the communications among different browsers, which enables the audio or video calls. WebRTC, even having its obvious implications in peer-to peer communication, is considered as an ideal solution for customer care, giving them permission of direct access to the contact center. For example, through a mobile customer care application, users can make a direct call to the agent by clicking only one button, without leaving any application. Similarly, customers (or prospective customers) having access to browse a system or website, either through mobile or at a personal computer, could easily initiate a direct or straight chat with an agent. The overall outcome is a seamless practice that removes the "context gap" – now the customers don't have to explain the matter or find out contacts or wait for a call back from a contact center agent.

1.1 Real-Time Communication with WebRTC

Peer-to-Peer in the Browser WebRTC is a free, open project, which enables the web browsers with RTC capacity through simple JavaScript APIs (Application Program Interface). The components of WebRTC are designed in a way to best serve the purpose. Until recently, web browsers were used for common purposes like web surfing, email, watching videos, except of conferencing. However, the biggest associated disadvantage of the web browser was that it was miserable at two-way voice and video calls [3]. The voice and video compression-decompression algorithms (called codecs) were really expensive, that is why real-time communication proved to be one of the challenging tasks for many companies [4]. At that time only a few companies owned them, and for this they charged more fees for pricey licensing. For a better understanding purpose, Plugins were discovered in the mid-late '90s, that allowed developers to play the videos using flash, which helps in facilitating a new shift towards the beginning of video calling or communication [5]. However, (RTC) still was a challenge due to lacking in one of the browsers method of sending as well as receiving data in real time, and often the use of expensive codecs to infer the communications between users.

1.2 Real Time Data Transmission Technology Based on WebRTC

The data transmission of the traditional B/S systems is carry out between the browser and the server. The browser sends a request to the server, and then the server respond corresponding data according to the request 375 parameters. WebRTC achieves peer-to-peer real-time data transmission between browsers. In the delivery of real-time data, timeliness and low latency can be more important than reliability. Therefore, WebRTC uses UDP at the transport layer. In the both peers of session or data transmission, one of the fundamental requirement is the ability to locate and identify each other on the network. In the trivial case, where both peers are located on the same internal network without any firewalls or NATs between them. To establish the connection, each peer can query its operating system for its IP address. But usually, there are many firewalls or NATs between peers, so ICE agent which colligates STUN, TURN, etc. is used to penetrate firewalls and NATs in WebRTC [4, 5]. Each WebRTC connections object contains an ICE agent. ICE agent is responsible for gathering local IP, port tuples and queries an external STUN server to retrieve the public IP and port tuple of the peer. Furthermore, it is responsible for performing connectivity checks between peers and sending connection keep alives to STUN server. If configured, ICE agent appends the TURN server. If the peer-to-peer connection fails, the data will be relayed through the specified intermediary.

1.3 Benefits of WebRTC

The benefits of WebRTC will affect the communications environment. WebRTC helps in enabling any Web server to convey a unique real-time communications experience, with ease and consistency, without relying on service providers or other services. Users participate in WebRTC, to experience communication, as delivered by any website without downloading, registering or incurring any cost. Following are the benefits:

- WebRTC enables any Web server to deliver a unique real-time communications experience
- Simplicity and reliability

WebRTC enables users to participate in a communications experience as delivered by any website

2. WebRTC Architecture

WebRTC follows a structured client server semantics with peer-to-peer communication concept between the browsers. Browsers run web applications in the case of the WebRTC Trapezoid model, which gets downloaded from a diverse Web Server. Without any intervention on the server, Connection organizes the media path to enable a direct flow between browsers. Network signalling goes over HTTP or WebSocket's, through the Web Servers, which help in altering, translating or managing the signals, as and when the need is felt or created. It is worth noting that the signalling between browser and server is assumed to be a part of the application, so it is not uniform in WebRTC. With the help of a standardized signalling protocol like Jingle or SIP (session initiation protocol), a proper communication may occur between two web servers . Or else, a proprietary signalling protocol can be used for such purpose. It is depicted in the above figure that the most common WebRTC scenario is likely to be the one, which enables both browsers to run through the similar web application, and also downloaded from the identical web page. In this context, the Trapezoid shape converts into a Triangle.

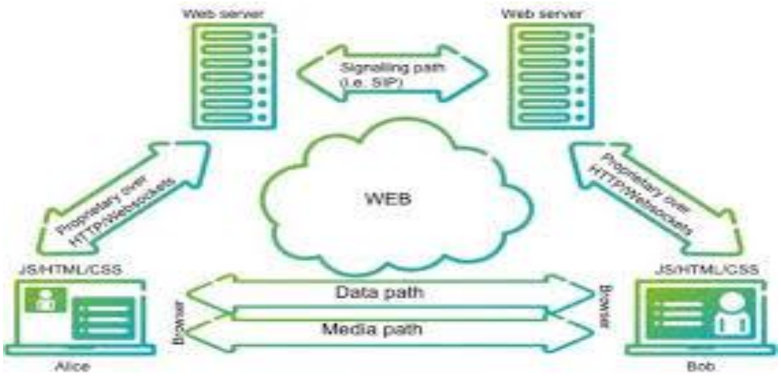

Fig-1 Architecture of WebRtc

2.1 WebRTC Signaling Mechanism

On the basis of reliable data channel, what is required is session negotiation before establishment a connection between browsers. This part of work is done by the WebRTC signalling mechanism. Before session negotiation, it must determine whether the data can be successfully transmitted to the other peer and whether the other peer ready to establish a connection. Thus, the initial peer of session need to send an Offer signal, the other peer need to respond an Answer signal. WebRTC does not define the standard of transmission channel and protocol, this allow interoperability with a variety of other signalling protocols powering existing communications infrastructure, such as SIP, Jingle, ISUP and so on. In this paper, the transmission channel is implemented using WebSocket. After transmission channel be implemented, next, it is supposed to exchange the session description information. WebRTC uses Session Description Protocol (SDP) to describe the parameters of the peer-to-peer connection. SDP describe the session profile, which represents a list of properties of the connection: types of media to be exchanged (audio, video, and application data), network transports, used codecs and their settings, bandwidth information, and other metadata. The process of SDP exchange between peers is as follow. The initiator (User A) creates an offer, and set it as his local description of the session. Then, he 376 sends the generated session offer to the other peer (User B) Once the offer is received by User B, he sets User A's description as the remote description of the session, generates the answer SDP description, and sets it as the local description of the session. Then he User B sends the generated session answer back to User A. Once User B's SDP answer is received by User A, User A sets User B's answer as the remote description of his original session.

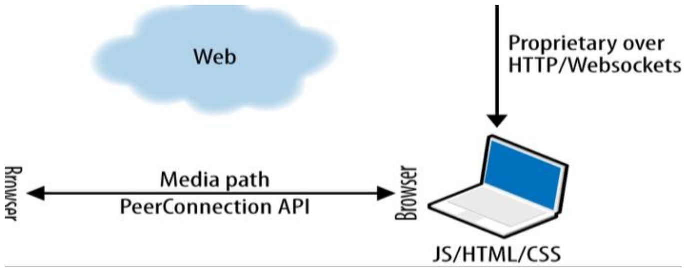

Fig-2 WebRTC Signaling Mechanism

2.1.2 WebRTC Security and Authentication

Unlike the most customary constant frameworks (for instance, SIP-based delicate telephones), WebRTC correspondences are specifically controlled by web servers. Notwithstanding the confirmation strategies WebLogic Server gives (HTTP fundamental, shape based, and customer declaration), WebRTC Session Controller bolsters an OAuth Identity Asserter, an HTTP validation supplier, and two-way SSL Security for WebRTC interchanges requires that the imparting endpoints have the capacity to verify each other. While these endpoints are making calls through the flagging administrations, their characters are verified utilizing Identity Provider (IdP), which underpins OAuth, Facebook Connect et cetera. WebRTC Session Controller can approve the guest's personality that is in the demand to get to the WebRTC-empowered customer application. The personality might be a bland URI or an email address. WebRTC Session Controller maps the personality to the frame character (telco or IMS character) of the outbound call. WebRTC Session Controller likewise offers denial of service (DoS) security (against message surges, contorted solicitations, and that's only the tip of the iceberg) at the flagging and media levels.

2.1.3 Impacts of WebRTC

On Telecom Equipment Industry: WebRTC is considered as the main element for potential changes. It permits the new users an immediate entry to the web conferencing spaces. Through the changes occurred, WebRTC drives the organization's contact center parts, which helps in opening the way for market share gains. However, WebRTC will be followed by contact centers to assure that existing contact center will not be displaced. Service Provider Industry: WebRTC could be treated as one of the larger destroyers (disturbance) in the service space. The enterprises that are following WebRTC application as customer care services and guest portals may come up with dramatically condensed need for PSTN trunk access.

3. CONCLUSIONS

This system, which is running on the normal HTTP server, has low-cost hardware costs. In the whole communication process, the server's function is negotiated the signalling before communication, the multimedia data transmit from one peer to another peer directly, server load is reduced greatly. WebSocket which suppose the bidirectional data communication used for signalling transmission channel, it is also well suited to WebRTC usage scenario Conclusion First, this paper researched WebRTC architecture and analysed the core technology. Next, for the part of signalling management that is not been defined in WebRTC, we proposed a solution that used WebSocket as signalling transmission channel and built a signalling server to forwarding signalling information. Finally, we built a real-time multimedia communication system based on the mechanism. The system is running well. The research provides an academic and practical foundation for WebRTC signalling work. What is most important is the system can running in mobile internet, the mobile smart devices can communicate with each other. In the next step of the work, multiple browsers communication framework will be researched due to the low communication efficiency because of too many connection.

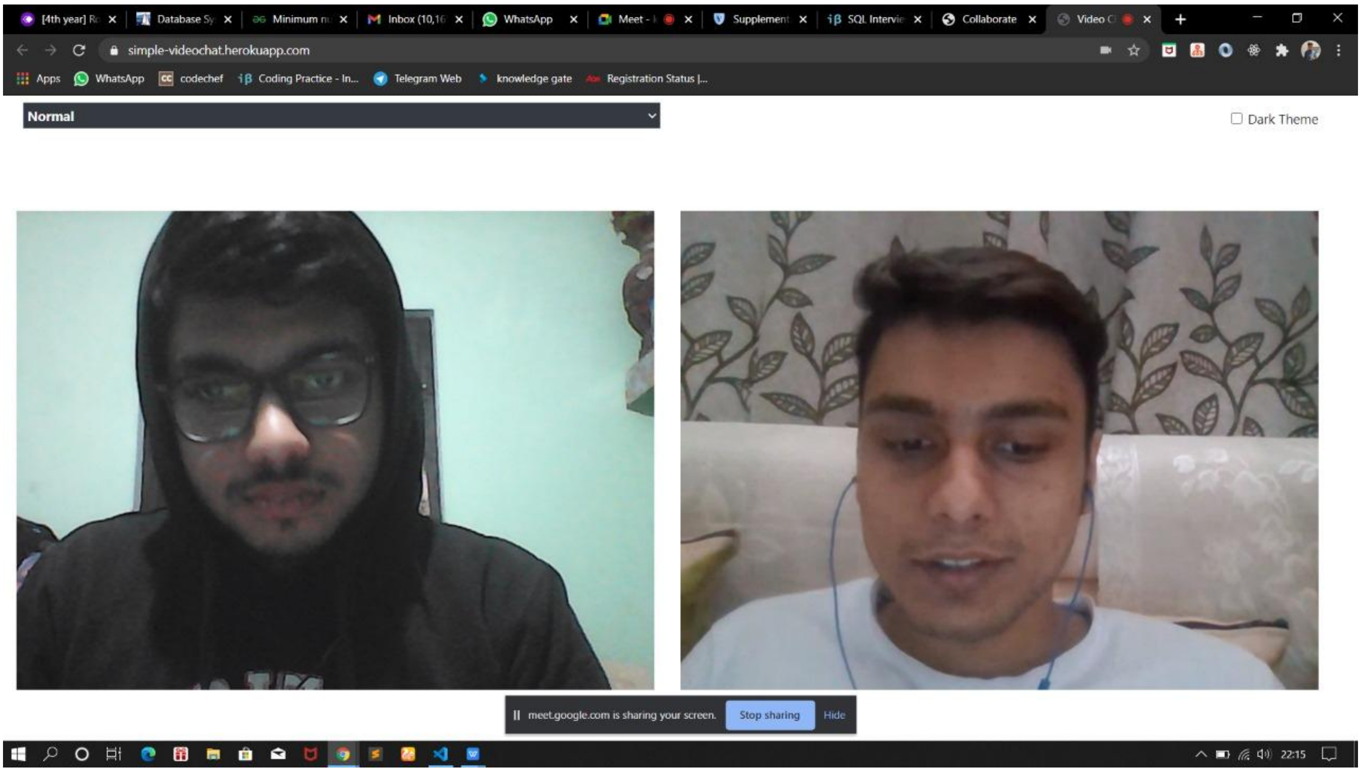

Fig. Final Result

REFERENCES

[1] Jennings, Cullen, Ted Hardie, and Magnus Westerlund. "Real-time communications for the web." Communications Magazine, IEEE 51.4 (2013): 20-26.

[2] Fette, Ian, and Alexey Melnikov. "The WebSocket protocol." (2011).

[3] Bergqvist, Adam, D. Burnett, and Cullen Jennings. "A. Narayanan," WebRTC 1.0: Real-time Communication between Browsers." World Wide Web Consortium WD WD-webrtc-20120821 (2012).

[4] S. Soltani, K. Misra, H. Radha, "Delay Constraint Error Control Protocol for Real-Time Video Communication," Multimedia, IEEE Transactions on, vol.11, no.4, pp.742, 751, 2009.

[5] C. Togay, "WebRTC technology for mobile devices," Signal Processing and Communications.