

WebRTC Enabled Video Communication

¹Naman Avasthi ²Palakh Mignonne Jude ³Rhea Thomas

^{1,2,3}Student (Computer Engineering)
Fr. C. R. Institute of Technology, Vashi,
Navi Mumbai, India

Abstract- Since the invention of the telephone, the modes of communication have been improving at a steady and gradual pace. From the older bulky phones to the modern day smart phones, mankind has been enhancing the techniques of communication. With the ability to communicate over large distances, people from across the globe have been able to contact each other without having to be physically present. With this, the concept of video communication came into existence. Here, people from different ends of the earth can not only talk to each other (voice communication) but also see each other (video communication/ video-telephony).

We plan to develop a web application using WebRTC which is a set of open source APIs. The application will permit voice as well as video communication among multiple users simultaneously, providing additional features such as recording of the conversation and filtering a conversation by allowing only a select number of people to listen in on a conversation. Since our application is based on web interface it is platform-independent and will ensure security to protect the users. Additionally, we will provide a broadcast feature.

Keywords- video communication; WebRTC; videoconference; videotelephony

I. INTRODUCTION

We plan to create a web application that would help users to conduct a video conference with one another over the internet. This application will be completely browser based and would not require any plugins or software/application installation, thus, making it platform independent. This web application will require a user to create an account. It would allow for multiple or one-on-one video, voice/video communication. It would also include features such as Join, Disconnect and Mute. This web application unlike other conference applications will provide security, thus, making the voice and video conferences completely secure. Our web application will use Web Real Time Communication (WebRTC) for establishing and carrying out all the communications.

Video communication was initially implemented in the form of videotelephony. Videotelephony includes a set of technologies that are used for the reception and transmission of both audio as well as video signals sent by different users located at different places. Using videotelephony, data can be sent in real-time. It is a full-duplex technology. While voice communication enables us to communicate with people residing in distant locations, video communication provides a

more realistic method of communication. Video communication can be made much simpler with the use of WebRTC. Web Real-Time Communication or WebRTC is an open source API by Google which was drafted by the World Wide Web Consortium (W3C) that allows browser-to-browser applications for video calling, sending files without the use of plug-ins.

The work of standardization of WebRTC is an ongoing process. Advanced implementations have already been carried out mainly in Chrome and Firefox browsers. Currently WebRTC is supported on desktop browsers like Google Chrome 23, Mozilla Firefox 22 and Opera 18 and for Mobile browsers like Google Chrome 28, Mozilla Firefox 24 and Opera 12 and also for Google Chrome OS. The other browsers currently don't offer this facility but are working on the same.

In our daily lives, communication plays a very important role. For a while, people at different geographical locations had to communicate using mails and telephone calls but with the advent of videoconferencing, individuals can now have real-time two-way communication with text, audio or video [2]. Videoconferencing is a technology that comes under telecommunication; it enables individuals to communicate through bidirectional, audio and video transmissions at once[3].

Digital compression of audio and video streams in real time is the primary technology used in a videoconferencing system, and this is performed by codecs.

The biggest technical issue of videoconferencing is signal latency time is required for the information transport of digital signals in many steps. Latency larger than 150-300 ms becomes noticeable and is distracting. In order to avoid a larger latency, a small round-trip time is desired [7].

II. WEBRTC

To establish a connection between the sender(s) and receiver(s), a number of protocols and techniques can be used. Among these, VoIP is one of the more popular and well-known methodologies. Another such method is the use of WebRTC. WebRTC (Web Real-Time Communication) is an open source API. It was developed by Google, drafted by the World Wide Web Consortium (W3C) and enables peer-to-peer applications for video communication without plug-ins using web browsers[4]. WebRTC was introduced in 2011

and comprises of a set of JavaScript APIs which enables the users to communicate via web browsers [6].

One of the last major challenges for the web is to enable human communication via voice and video: Real Time Communication, RTC for short. RTC should be as natural in a web application as entering text in a text input. Without it, the ability to innovate and develop new ways for people to interact is limited.

The advantages of WebRTC include the following; it simplifies communication by moving dependency from operating system and clients into the browser environment, no external plugins are required to be installed. It allows on-click browser embedded calls (voice or video) for enterprise or customer care [1]. It has secure voice and video calls using Secure RTP protocol (SRTP) for encryption and authentication of both voice and video, preventing eavesdropping and recording of voice and video as well as advanced voice and video quality using Opus audio codec for voice communication and VP8 for video.

The main difficulty about WebRTC is that this standard is still under development. Thus, the code we use now can undergo serious changes in the foreseeable future. There's an existing WebRTC API version 1.0 but it is mentioned by W3C team that it's only a working draft and no one can tell when this status will be changed to "stable". Many specialists though think that this version is final and stable, and draft status can be kept by W3C for a long time without any actual changes to the document [5].

III. SCOPE

In our project, we plan to create a web application that enables video communication between two and more than two users. We are going to establish a connection between the users using WebRTC (via web browsers such as Google Chrome, Firefox and Opera). Our web application is going to be efficient (in terms of bandwidth) and reliable.

IV. PROPOSED SYSTEM

We plan to develop a web application that allows video communication. The users will be allowed to create accounts on our web application through which they will communicate with other users.

One of the drawbacks of the existing systems is the lack of security. Our web application will provide security. WebRTC makes use of the AES algorithm to encrypt the data being sent, thus, making the system secure. Additionally, the #key that will be sent to the peers will be encrypted by implementing the DES algorithm. Another major drawback of the existing systems is the high bandwidth requirements. Our web application will adjust the resolution of the video based on the bandwidth available, which will thereby optimize the bandwidth.

V. DESIGN

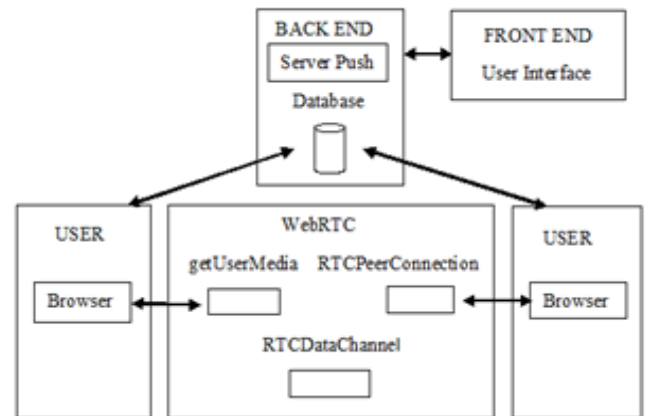


Fig. 1. Block Diagram

The block diagram of our system consists of the back-end, front end, users and the WebRTC module. The back-end consists of Server Push and a database (which will store information about all the registered users). This forms the link between the front-end and the users. The front-end consists of the user interface which will be simple and easy-to-use. The users will be connected to each other through the WebRTC module via web browsers (like Google Chrome, Firefox, and Opera). This is made possible through getUserMedia, RTCPeerConnection and RTCDaChannel APIs.

The connection is established by means of the getUserMedia API. This allows access to the camera and microphone. The connection channels make use of Secure Real-time Transport Protocol (SRTP) and Datagram Transport Layer Security (DTLS) which is used as SRTP key for association management. Additionally, DTLS is used to multiplex traffic in a single RTP session and thereby avoid opening a new port for each stream. SDP Security Description for Media Streams (SDS) is used as an alternate key. Synchronization source identifier (SSRC) is used to identify the source of the stream.

The video/audio call is set up using the RTCPeerConnection API. Session Traversal Utilities for NAT (STUN) and Traversal Using Relays around NAT (TURN) are used to let the data streams traverse the NAT boxes and firewalls. Initially, the STUN servers are used and if this fails, then the TURN servers are used.

Browsers can share data via peer-to-peer standards using RTCDaChannels. Stream Control Transmission Protocol (SCTP) is used because it supports reliable, unreliable and partially reliable delivery modes. For security purposes encapsulation is done of SCTP over DTLS over ICE over UDP. This provides a NAT traversal solution, confidentiality, source authentication, integrity and allows sharing same transport layer port number.

The security is maintained in the following manner. The packets are encrypted using DTLS which is an end-to-end encryption between peers. The STUN/TURN servers only parse the UDP layer of a WebRTC packet; it cannot understand or modify the application data layer which ensures integrity.

The following table describes the bandwidth requirements of our system considering the various scenarios.

TABLE I. BANDWIDTH REQUIREMENTS

Scenario	No. of RTP Ports	Bandwidth
Peer-to-Peer audio/video	2 for outgoing video and audio, 2 for incoming video and audio	4 MB (for each peer)
Broadcaster in One-Way Scenario	1 for outgoing video and 1 for outgoing audio	2 MB
Participant in One-Way Scenario	1 for incoming video and 1 for incoming audio	2 MB
Many-to-Many Scenario	Depends upon number of peer connections multiplied by 4	Variable based upon number of connections

The following table gives a comparison between our system and the existing systems.

TABLE II. COMPARISON BETWEEN PROPOSED SYSTEM AND EXISTING SYSTEMS

Parameter	Skype	Hangouts	Proposed System
Protocol used	Proprietary VoIP-Skype Protocol	XMPP proprietary protocol	WebRTC APIs
Features	Skype-to-Skype calls, Calls to mobiles and landlines, Group calls, Voice messages, Screen Sharing	Call phones from computer, works on Android and Apple devices and computers, video calling	Video/voice calls to-and-from any device that is browser-enabled. Allows video conferencing among multiple users
Maximum users in a conference	Video: up to 25 users (free)	Video: up to 10 users	Video: up to 15 users (proposed value)
Security	Not very secure. Reason for security concern.	More secure than Skype	Will make use of encryption algorithms to ensure security.

The flowchart of the system describes the flow of all the functions that the system can perform. Initially, the user will be required to login into the web application. After this

step, the user can choose to either make an audio call or a video call. Once this has been selected, the user will be asked to select the number of users which can be either one-to-one or multiple.

In case of one-to-one communication, the user who initiates the call will send a request to the other user to accept the call and will then wait for the other user to accept the call. Once the call has been accepted, a call will be setup using WebRTC. At the end of the call, the session will be disconnected. At this point, the user has the option to either place another call or to logout.

In case of multiple user communication, a counter 'k' will be initialized. The user who initiates the call will send a request to all the other users. If none of the other users are online, the user will be allowed to place another call. If only a few users are online, then the user who initiates the call will have to wait till all the users are online. If all the users are online and all the users accept the request, the call will be setup using WebRTC. If all the users are online, but only a few accept the request, then the counter 'k' will be incremented. If the value of 'k' is greater than 2, then the call will be setup with the existing users. Here, as in the case of one-to-one communication, the session will be terminated once all the users disconnect the call.

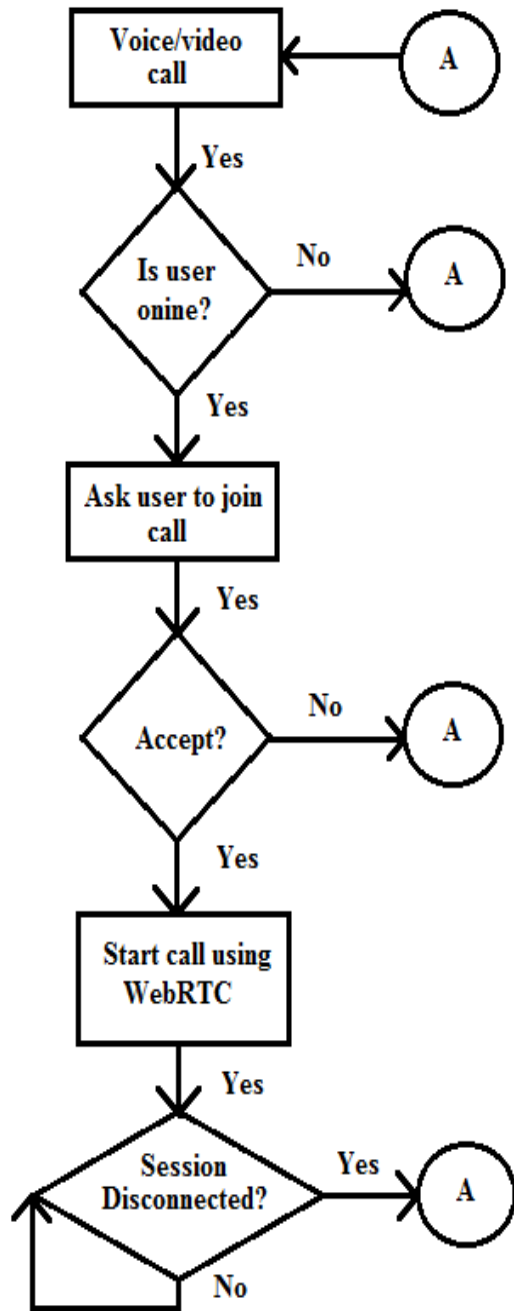


Fig. 2. Flowchart

VI. CONCLUSION

We have proposed to develop a web application that will allow video communication with the use of WebRTC. This web application will not require any additional plug-ins or softwares and will be platform independent, thus, giving it an edge over the existing systems. Additionally, it will also be secure. This is an advantage over existing systems such as Skype.

REFERENCES

- [1]. Alan Johnston, John Yoakum, and Kundan Singh, Avaya Inc, "Taking on WebRTC in an Enterprise", IEEE Communications Magazine, April 2013.
- [2]. Chong Luo, Wei Wang, Jian Tang, Jun Sun and Jiang Li, "Multiparty videoconferencing system over an application-level multicast protocol," IEEE Transactions on Multimedia, Vol. 9, No. 8, December 2007.
- [3]. Sang-Beom Lee, In-Yong Shin and Yo-Sung Ho, "Gaze-corrected view generation using stereo camera system for immersive videoconferencing," IEEE Transactions on Consumer Electronics, Vol. 57, No. 3, August 2011.
- [4]. <http://en.wikipedia.org/wiki/WebRTC>
- [5]. <http://thmsglchrst.hubpages.com/hub/Advantages-and-disadvantages-of-WebRTC>.
- [6]. <http://www.html5rocks.com/en/tutorials/webrtc/basics/>
- [7]. http://www.telephonyworld.com/training/brooktrout/iptel_latency_wp.html