Volume 5, Issue 6, June 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Research and Review of Web RTC- the Next Generation of Web-based Communication

[1]Sheetal Thakkar, [2]Kinjal Thakkar, [3]Bhavika Bhanushali, [4]Aman Arora, [5]Dishant Chawla
[1]Computer Technology Department, Shah & Anchor Kutchhi Polytechnic, Chembur, Mumbai, Maharashtra, India
[2, 4] Systems Engineer, Infosys Limited, Pune, Maharashtra, India
[3, 5] Software Engineering, Analyst Accenture Services Pvt. Ltd., Mumbai, Maharashtra, India

Abstract – Communication have always been a vital part of human's life and the means by which humans communicate have progressed drastically in the past few years with the advancement in technology. Besides, with this rapid progress in technology and with the evolvement of electronic devices like computer, mobile phones, tablets and etc. the traditional methods of communication are challenged and there is a great inclination towards their improvement. WebRTC seems to be a part of this improvement. Thus, this electronic document describes concept of WebRTC (Web Real-Time Communication), which can also be said the next generation of web based communication. The major goal of this paper is to get an overview of the future telecommunication service i.e. WebRTC and study its prominence in delivering a more flexible way of communication by enabling web browsers to provide real-time communication as part of its basic capabilities. This paper also covers the architecture, various protocols and some of its impact in the IT field of the above mentioned technology.

Keywords– Real-time communication, Browser, Peer-to-Peer, Protocols, Web-Server, W3C, HTML5, IETF, API.

I. INTRODUCTION

With the global accessibility of bandwidth, the communications over wired and wireless networks have increased. Besides, there has been drastic improvement in CPU and GPU capabilities of devices. This led to evolution in the human/computer interaction, how people interact with their devices, how they buy and consume contents and applications. Humans have quickly and incredibly adopted some new paradigms. For these reasons and others, the communication techniques have been similarly transformed to an always-on multitasking, asynchronous method of communications, easily switching between e-mail, text, voice, idea between multiple people. It often simultaneously depends on our communication needs or location or connectivity. The methods of communication have also gotten much more contextual, often adopting fine-grained preferences for how people communicate with a particular social context. [1]

Because of such advancement in technology and a great need for communication, it can be undoubtedly said that the traditional ways of communication is being challenged at every point. In the last few years the IP communication has become the main aspect of communication. The reason for this is its worldwide accessibility. Due to the improvement in usability of internet it has now provided a convenient and cost effective way of voice and video communication.

In this evolving environment of communication, WebRTC is an API definition drafted by world wide web consortium (W3C) that supports browser-to-browser applications for voice calling, video chat and P2P file sharing without plug-ins or third-party softwares i.e. by simple Java Script APIs. [2] WebRTC is an effort, initiated by Google, to build a standard based real time media Engine into all of the available browsers. With WebRTC in a browser, a web services application can now initiate the browser to provide a real time web or video connection to another WebRTC device or to a WebRTC media server using RTP. [3]

Brendon Eich, the inventor of JavaScript says that WebRTC is a new front in the long war for an open and unencumbered web. [4] It is about putting real time capabilities into a standard browser without the need for downloads, plugins or Flash. [5] The World Wide Web Consortium (W3C) and the Internet Engineering Task Force (IETF) are mutually defining the JavaScript APIs (Application Programming Interfaces), the standard HTML5 tags and the basic communication protocols for the setup and management of a reliable communication channel between any pair of next-generation web browsers. [6]

The standardization goal is to define a WebRTC API that allows a web application running on any device, through the secure access to the input peripherals (such as webcams and microphones), to exchange real-time media and data with a remote party in a peer-to-peer fashion. [6] Theoretically, WebRTC and HTML5 could enable the same transformation for real time that the original browser did for information. Similarly, WebRTC may enable to point the browser at the server where the information or applications resides. [3] The rest of this article focuses on various technology and standards for WebRTC and other related aspects.

II. NEED FOR WEBRTC

WebRTC assures easy communication between browsers for video conferencing and voice calling. Hence, the need for WebRTC is quite clear. In fact, WebRTC is a buzz word in the unified communications, collaboration and contact center market these days. [7]

The Demand for Real Time Communications is driving the need for WebRTC. Besides WebRTC requires no plug-ins, frameworks or applications, all that is needed is a WebRTC compatible browser. No Flash, no silverlight just pure video, audio and data communication on any webpage. Many people like to be in contact with friends, colleagues and family. A numerous of them use internet and mobile phones. Along with this, computers have also become the basic need of people. There is also a great escalation in the use of smartphones. According to Gartner, over 50 percent of mobile apps will be HTML5- native hybrids by 2016. [9][10] Thus, this clearly shows a need of advance communication techniques like WebRTC.

WebRTC is completely peer to peer, so one doesn't have to pay for any of the bandwidth across the wire. Additionally, because WebRTC is entirely browser to browser, one may get maximum performance and lowest latency possible. For example, say a user is trying to share a file; Without WebRTC, the user has to upload said file to a server, and the recipient user has to download that file. With WebRTC, these files are transferred directly through the WebRTC Data Channel, no servers or infrastructure required. [8]

The following are some of the needful reasons of using WebRTC:

A. Browser based: As mentioned previously, WebRTC enables browsers to provide real time communication. Thus, with WebRTC, web applications have direct API access to the microphone and camera in a device via standard JavaScript API calls, which is different from the previously used technique where applications are installed locally on a computer or server. This thereby removes the need for a plugin or download, making it device, platform, and OS independent

B. Security: Unlike traditional security applications that frequently require users to accept and install updates manually, many of the leading browsers automate that step. Hence, every time the browser is launched, it is up to date – including all the latest bug fixes – without any additional steps. Therefore, WebRTC being browser based supports only SRTP (secure real-time transport protocol) by default, which uses encryption and authentication to minimize the risk of denial of service (DOS) attacks.

C. Advancement in audio and video codecs: Though the availability of bandwidth has increased widely, the rising trend of remote workforces and the demands of high definition video and audio communication exceeds many corporations' bandwidth capabilities, causing packets to be lost and a unclear voice and video experience. In a WebRTC environment, advanced audio and video codecs are used that enables well - organized use of video and audio packets, which results in clearer communication even in circumstances rendering traditional communication environments useless. Apart from these, it supports both variable and constant bit rates and even narrowband communications.

D. Promotes stronger session authentication: Traditionally, video communication environments relied on third-party relay media servers to manage sessions in order to traverse firewalls. As WebRTC establishes a peer-to-peer reliable session through NAT's (Network Address Translation), therefore eliminating the need for third-party communications servers.[5]

III. TECHNOLOGY AND STANDARDS

In real time communication a client needs; a framework; a Graphical User Interface (GUI), and a Media Engine. These three components are shown in the figure below. The white box refers to the visual interface (GUI), the blue box is the media engine, and the rest is framework. In a typical hard client such as an IP phone, the framework consists of the processing chips and the OS. In a soft client, the framework is the device/OS the client is running in. The visual interface can be a hard interface such as a phone key pad or a screen presentation in a PC or other device. The purpose of media is to manage the real-time transmission i.e. sending and receiving of a video/audio stream.

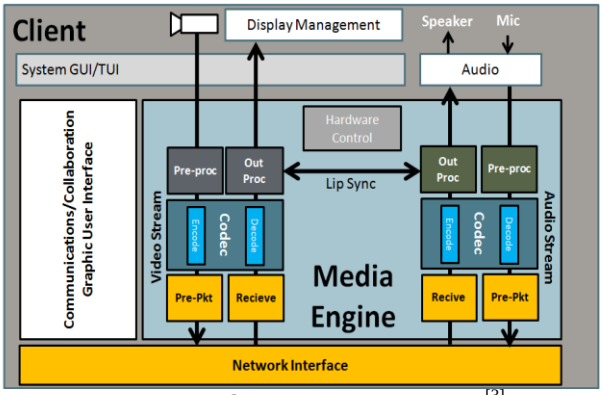

Figure 1. Media Engine Components [3]

The media Engine includes a set of functions that deliver quality voice and video:

A. Audio

- Setup and control the hardware
- RTP, compression, encryption, statistics, etc.

- Produce low-latency audio from microphone
- Conceal loss, de-jitter and play audio from the network
- Cancel echo, VAD, reduce noise, etc.
- Manage codecs

B. Video

- Render video, capture camera input
- Video processing (blue screen, gamma, etc.)
- Conceal loss, de-jitter and play video from the network
- Cancel echo, VAD, reduce noise, etc.
- Manage codecs
- Bandwidth Management

The WebRTC aims at combining the Media Engine with a set of standard APIs and thus, provide a browser based solution for real-time communication. The WebRTC Media Engine uses both a set of standard components, including codecs to minimize the issues of two WebRTC end points communicating, it also includes a set of standard APIs so a server that the browser connects to can control the WebRTC Media Engine in the client. Beyond the basic media functions, WebRTC includes an API set that enables the controlling server software to cause a direct connection between two WebRTC devices without any external signaling. By merely telling two WebRTC devices to communicate, the server can initiate an IP based voice or video communications. [3]

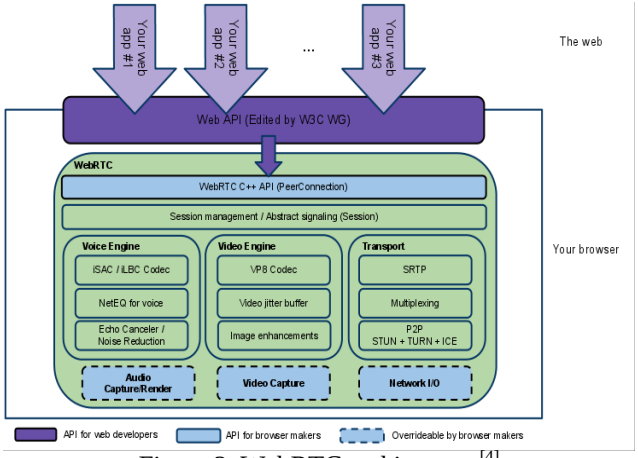

Figure 2. WebRTC architecture [4]

IV. APPLICATION MODELS

There are two types of applications WebRTC was designed to support. They are:

A. Triangular model

The primary purpose of WebRTC is to create a communication path between two browser devices. This is shown in the figure below. Here, a website has caused the WebRTC client in both the systems to connect to each other with a direct RTP/IP connection. In this case, although the media path is being delivered peer to peer, but the controller is the web server. This enables the web site owner to enable communications on the site without requiring the site or hosting to actually provide or manage the bandwidth between the devices. This level of application is appropriate to add communications to any site that could recognize two parties that might want to interact. For example, Linkedln could use this to enable its subscribers to open a voice channel. If the seller of an item was connected to eBay, a connect icon could appear on the item page, and by simply clicking it a potential buyer could be talking to the seller. The crucial thing is that this could be achieved just by a small JavaScript code in the web server. This enables the user experience to be customized, in the same way a web page provides a customized experience based on the information and purpose.

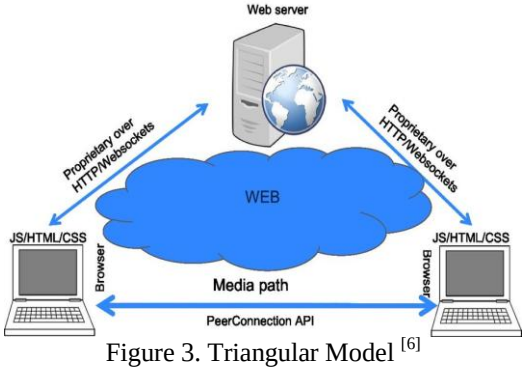

Figure 3. Triangular Model [6]

For example, a user interface for an overlay to a social networking site might look very different from one that is a sales site. The key point is that the communication is being pushed from the server, instead of being pulled by the client as in traditional telephony. In nutshell, any communications where both peers are controlled from a single server (or server system) is referred to as a triangular model. [3] This means, the triangular model defines the case where clients that want to establish a real-time channel between each other, talk to a common central server that co-ordinates passing IP address contact information between each client so they can establish a media channel directly between them. [1]

B. Trapezoid model

Different vendors can deliver direct connections between them on different servers using WebRTC standards if they decide to deliver communications services to end points using WebRTC. This opens the possibility of interconnected systems using WebRTC as the standard. The WebRTC standard does not define the mechanism for communications between the servers, merely the control of the client (or media servers). So two different vendors could have their UC control systems interoperate using SIP or any other means and then each tell their respective end point device to connect to the IP address of the end user node on the other system using WebRTC. As the end point devices are merely connecting to each other, this implementation may enable easy inter-enterprise communications. This is because each device is just following the API instructions from the server, the fact they are actually talking to two separate servers does not prevent them from connecting to each other directly. [3]

In short, the trapezoidal model defines the model where a client talks to a server associated with his particular application server, a remote client talks to his/her server, and there is an agreed upon signaling protocol and IP channel between the servers that passes the IP address contact information.[1]

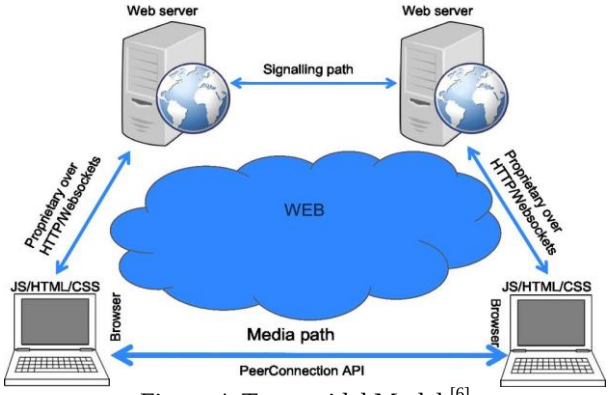

Figure 4. Trapezoidal Model [6]

V. REAL-TIME NETWORK TRANSPORTS

Real-time communication is time-sensitive. As a result, audio and video streaming applications are designed to tolerate intermittent packet loss: the audio and video codecs can fill in small data gaps, frequently with negligible impact on the output quality. Similarly, applications must implement their own logic to recover from lost or delayed packets carrying other types of application data. Timeliness and low latency can be more important than reliability.

UDP is the foundation for real-time communication in the browser, but to meet all the requirements of WebRTC, the browser also needs a large supporting cast of protocols and services above it. [11]

Various protocols used for WebRTC consist of the following:

A. Application layer protocols

- HTTP – Hypertext Transfer Protocol
- WebSocket - JavaScript interface
- SRTP - Secure Real-time Transport Protocol
- SDP – Session Description Protocol
- ICE – Interactive Connectivity Establishment
- STUN – Session Traversal Utilities for NAT
- TURN – Traversal Using Relay NAT
- SIP – Session Initiation Protocol(optional)
- Jingle – Peer-to-peer signaling is optional

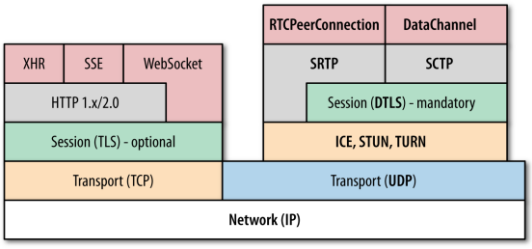

Figure 5. WebRTC Protocol Stack [11]

B. Transport layer protocols

- TLS – Transport Layer Security
- TCP – Transmission Control Protocol
- DTLS – Datagram Transport Layer Security
- UDP – User Datagram Protocol
- SCTP – Stream Control Transmission Protocol

All of these protocols will be operating over the IP network layer. [12]

VI. FUTURE SCOPE

Often, WebRTC is referred to as Peer to Peer communications. This should not be confused with Browser to Browser communications. While WebRTC can be delivered in a browser, it can also be in any other end point device. As many new endpoints are in fact becoming browsers, the capability to use WebRTC in a variety of devices will be significant. WebRTC can be incorporated in IoTs (Internet of Things).For example, WebRTC could be in television, car, a toaster, or a clock radio.

In addition to plethora of potential end points, a peer can also be values add point. For example, a Media Server could be a peer, or a gateway to the PSTN. This capability to incorporate peer services in the media stream will enable advanced capabilities far beyond simple point to point connections. [3]

The enterprise has numerous uses for WebRTC for general solutions that are not vertically specific. It can become the direct point of communication between the enterprise and external world enabling BYOD (Bring Your Own Device).

It can also be used as audio delivery path in multiplayer games. In conjunction with face recognition software and capture, the characters in the game could now have actual user faces overlaid. Think of the Barney (purple children's figure) site where children can now interact with each other and with a computerized Barney by clicking the "Barney Walkie-Talkie" that invokes WebRTC. [3]

VII. CONCLUSION

The web has revolutionized communication, and WebRTC promises to take this revolution a step further. The free, open-source project enables compliant web browsers to communicate in real-time using simple JavaScript APIs. Hence, by this it can be concluded that WebRTC aims at evolving the real – time communication service by using simple JavaScript APIs and HTML5. It can thereby improve the quality of communication, which will be delivered via browsers.

WebRTC is unleashing the power of real-time communication to any web developer or application. An important thing about WebRTC is the manner in which it's baked into browsers. WebRTC is generally thought of as enabling communication between users on browsers, but it will likely also be used to generate new user interfaces for websites. WebRTC however promises and sets a vision of the future communication technique that already exists in the present, to some extent. Ultimately, the effect of VoIP on masses seems to be evolving and with some additional efforts WebRTC can prove to be a boon in the communication field.

Hence, this paper focused on the concept, architecture and protocols used by WebRTC thereby summarizing the topic.

REFERENCES

[1] Chris Wendt, "Evolving the telecom stack and how WebRTC plays a role" url - https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CB0QFjAA&url=http%3A%2F%2Fwww.nctatechnicalpapers.com%2FPaper%2F2014%2F2014-evolving-the-telecom-stack-and-how-webrtc-plays-a-role%2Fdownload&ei=pDhgVY7rBdGUuAS01oP4Cw&usg=AFQjCNF6wfBOejgss9wrGjMpjMyzNUHnLA&sig2=Zz9NLITr3HWUCnr--xph9A&bvm=bv.93990622,d.c2E

[2] http://en.wikipedia.org/wiki/WebRTC

[3] http://www.pkeconsulting.com/pkewebrtc.pdf

[4] http://www.html5rocks.com/en/tutorials/webrtc/basics/

[5] http://www.yorktel.com/wp-content/uploads/2012/06/3-Things-You-Need-To-Know-About-WebRTC_w_Links_Final.pdf

[6] https://www.safaribooksonline.com/library/view/real-time-communication-with/9781449371869/ch01.html

[7] http://searchunifiedcommunications.techtarget.com/tip/Four-main-uses-illustrate-need-for-WebRTC-gateways

[8] http://www.marketwired.com/press-release/nudg-media-inc-announces-completion-of-webrtc-otcqb-nddg-1956462.htm

[9] http://www.gartner.com/newsroom/id/2324917

[10] http://www.tmcnet.com/tmc/whitepapers/documents/whitepapers/2013/9187-webrtc-reach-web-with-new-conversation-experience.pdf

[11] http://chimera.labs.oreilly.com/books/1230000000545/ch18.html#_real_time_network_transports

[12] http://www.webtorials.com/content/2013/05/eleven-answers-webrtc-explained.html

[13] Hitendra Patil, Amar Buchade, "Review and Study of Real Time Video Collaboration Framework WEBRTC " url - http://www.ijcsit.com/docs/Volume%205/vol5issue01/ijcsit2014050123.pdf

[14] http://webrtcbook.com/presentations/WebRTCIEEE04-02-13.pdf
[15] http://cdn.oreillystatic.com/oreilly/booksamplers/9781449371876_sampler.pdf
[16] https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API/Introduction
[17] http://www.webrtcworld.com/
[18] http://www.w3.org/TR/webrtc/
[19] http://www.webrtc.org/

