A Survey on Real-Time Communication for Web

Md. Habibur Rahaman
Department Of Computer Science and Technology
University of Bedfordshire
Luton, United Kingdom
h.r.sozib@gmail.com

Abstract- **Web Real time communication (webRTC) is a new standard and industry effort where real time media can be exchanged using web browser in a peer to peer transmission. In this survey paper reader will be able to get an idea about the features of WebRTC and will know how the users gain a communication experience with this. It helps in providing a knowledge about various security issues related to WebRTC. This survey will focus on the Voice over IP (VoIP) technology and the Web Real-Time Communications (WebRTC) protocols. It has explained specific advantages and disadvantages for the users. How WebRTC enables the servers to communicate inreliable way. Further this survey paper will enable different insights related to the topic. In today's era, internet servers are threatened by various attacks, this paper has also highlighted some of the risks for WebRTC.**

Index Terms— **RTC, WebRTC, Signalling. protocols**

I. 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 [1].

WebRTC is one of the communication standards that is developed by the World Wide Web Consortium in close collaboration with the RTCWeb standard, further developed by the Internet Engineering Task Force (IETF) [2]. RTCWeb 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-tobrowser 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 [2]. 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.

II. 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.

III. WEBRTC ARCHITECTURE

WebRTC follows a structured client server semantics with peer-to-peer communication concept between the browsers (1).

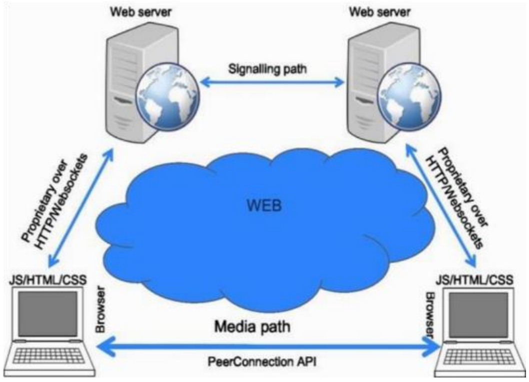


Figure. 1:The WebRTC Trapezoid [6]

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, Peer

Connection organizes the media path to enable a direct flow between browsers. Network signaling goes over HTTP or WebSockets, 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 signaling 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 [7]. Or else, a proprietary signaling protocol can be used for such purpose.

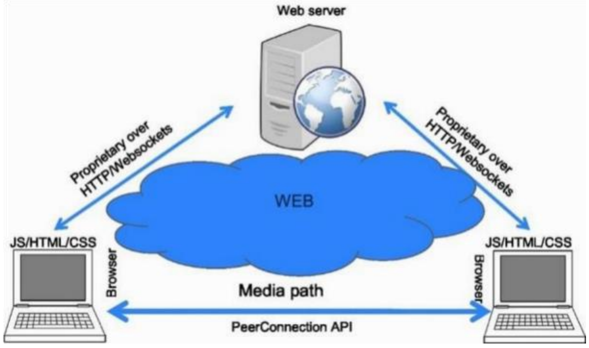


Figure. 2: WebRTC Triangle [6]

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.

IV. WebRTC in the Browser

A web application of WebRTC (typically written as a mix of HTML and JavaScript) interrelates with the several web browsers through the standardized WebRTC API, which permits them to make proper utilization and manage or handle the functions of Real-Time browser (see Figure 1-3) [8]. An interaction between the WebRTC web application and the browser takes place through WebRTC and other standardized APIs, in both a proactive (e.g., to query browser competencies) and a reactive (e.g., to receive browser-generated notifications) way.

The WebRTC API, therefore, must provide varied types of functions, including connection management (in a peer-to-peer fashion), negotiation of encoding/decoding capabilities, selection and media control, firewall and NAT (Network Address Translation) element traversal, and many others like these [9].

For illustration, consider a real-time audio and video chat or call between two different browsers. Communication, in such a scenario, might involve or go for selecting direct media streams between the two browsers, with the media path negotiated and instantiated through a multifaceted series of interactions, which involve the following entities:

- The application of caller browser and the caller JavaScript (e.g., through the mentioned JavaScript API) [10].
- The caller JavaScript application and the application source

(typically, a web server)

- The application source and the callee JavaScript application
- The callee JavaScript application and the callee browser (again via application-browser JavaScript API)

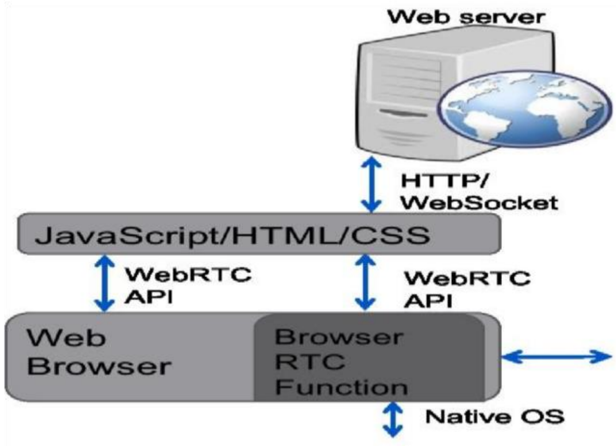


Figure. 3: Real Time Communication In the Browser [6]

V. Signaling

The initiation of general idea related to design of WebRTC, specifies the methods as how to control

the media plane, while leaving the signaling plane totally on the application layer. The basis is that different applications may have a preference to use diverse standardized signaling protocols (e.g., SIP or eXtensible Messaging and Presence Protocol [XMPP]) or even something habitual.

Session description characterizes the most significant source of information that is required to be ex_changed. It indicates the transport (and Interactive Connectivity Establishment [ICE] information, as well as the type of media to be used, format, and all related media configurations

factor needed to set up the pathway for media. Since the original idea of exchanging the session description information to a newer form, Session Description Protocol (SDP), "blobs" created various types of shortcomings in this process, some of which proves to very challenging while addressing. The IETF is now standardizing the JavaScript Session Establishment Protocol (JSEP) [11]. JSEP provides the application, an interface that is required to deal with the negotiated local and remote session descriptions (with the negotiation carried out through whatever signaling mechanism might be desired), together with a proper way of interacting with the ICE state machine [12]. The JSEP approach entrusts the entire responsibility for motivating the signaling state machine to the application. The application must identify the correct APIs at the available times, and must be able to convert the session descriptions and related ICE information into the definite messages of its selected signaling protocol, rather than simply forwarding the message to the remote side.

VI. Protocol Used

There a several types of protocols available that will be used at the WebRTC application layer:

- HTTP - Hypertext Transfer Protocol
- WebSocket - JavaScript interface
- SDP - Session Description Protocol
- ICE - Interactive Connectivity Establishment
- SRTP - Secure Real-time Transport Protocol

- TURN - Traversal Using Relay NAT
- STUN - Session Traversal Utilities for NAT
- SIP - Session Initiation Protocol is elective
- Jingle - Peer-to-peer signaling is also optional

The transport layer protocols consist the following:
- TLS- Transport Layer Security
- DTLS- Datagram Transport Layer Security
- TCP- Transmission Control Protocol
- UDP- User Datagram Protocol
- SCTP- Stream Control Transmission Protocol

All of the above mentioned protocols will be operating over the Internet Protocol (IP) network layer.

VII. The Security Mechanisms Within WebRTC

The security mechanism of WebRTC is pretty scary to think upon it. It reminds of a situation of the complicated attack at the Iranian nuclear facilities where the hackers had hacked the microphones and video cameras, to see the functions performed within these type of facilities. How can the security mechanisms of WebRTC's will be able to prevent unauthorized access from taking over our devices?

Security depends on trust factor, and in WebRTC, any security or trust factor needed by the individual would need a guarantee by the web browser [13]. In reality, however, in an operational system, the browser needs to depend on some other faithful sources. There are an ample number of third-parties available as an identity providers such as BrowserID, Federated Google Login, Facebook Connect, LinkedIn, OpenID, and WebFinger [14]. WebRTC can also take the help of these trusted third parties to detect a user's identity.

VIII. WebRTC Security Architecture

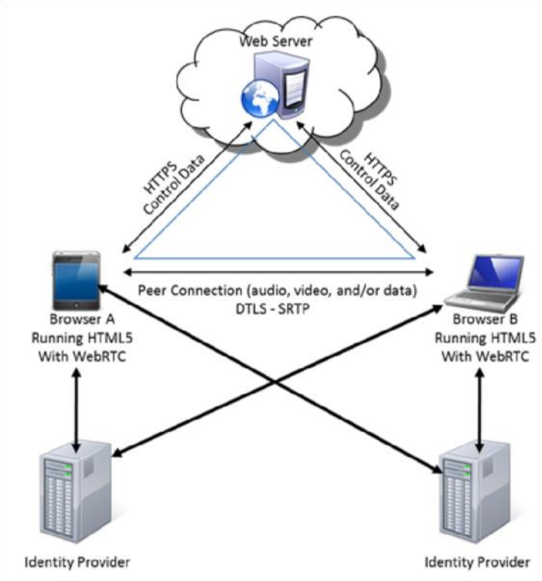


Figure. 4: The WebRTC security architecture [15]

User A and User B, both the servers are connected to the same website via HTTPS. The users have also validated their identity by using their documentation with either an external characteristic provider or through the website itself.

User A decides to make a call to user B. This will likely be done through clicking the call button, which is next to B's name. When the call button is clicked by user A, a message is sent to the JavaScript in the web server, running already in A's browser that helps in producing two peer connections: one for audio and the other one for video (considering this is an audio/video call--a peer connection is necessary for both media types). At this point, no security has been called upon and any website can continue with WebRTC to this point. Next thing needs to be done is, the calling application requires the audio and video from that of a microphone and camera [15]. User A is presented through a pop-up window, that asks if the website is capable of accessing with user A's camera and microphone.

IX. WebRTC Security Concerns

The design of WebRTC was based on peer-to-peer communication. It is likely to make WebRTC calls interoperable with other Internet Protocol or legacy networks, by using WebRTC to SIP gateways (15), that are released by Session Border Controllers and Media Gateway manufacturers, like the Oracle WSC.

The WebRTC environment opens up great opportunities for innovative services for residential (OTT services, vertical applications for e-health, online banking, etc.) and corporate users (BYOD, teleworking, etc.). From the new bunch of services and devices, it means that an open opportunity or invitation for the intruders and attackers that should be prevented.

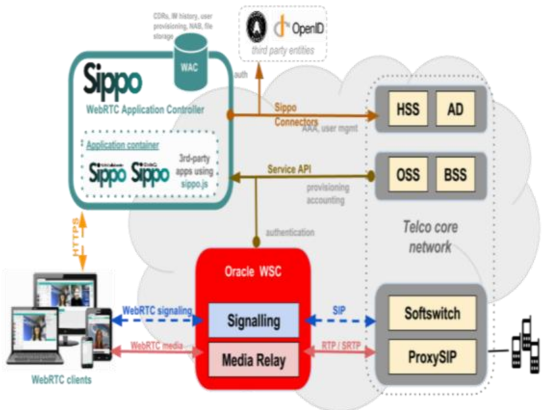


Figure. 5: WebRTC implementation [16]

The above figure shows a representative WebRTC implementation, that is required to introduce within the webserver (like the Sippo WebRTC Application Controller) that can be hijacked. A new access can be introduced like the Oracle WSC (that could be attacked via DoS attacks or illegally accessed) and, at last a smart endpoint (laptop, smartphone) that can be risky as they may be full of viruses or Trojans. In addition, proper care of ID of the users should be involved in the process, which is further included in the WebRTC session.

A. TRADITIONAL VOIP ATTACKS

By using the WebRTC applications, certain potential threats are going to appear and part of the traditional VoIP attacks will be inherited, especially in those connections where linkage is made between servers. WebRTC gateway enables

VoIP operators and various enterprises to expand their span of services from Session Initiation Protocol and PSTN towards browser based application in a flawless way. The VoIP investments are protected through WebRTC gateway by enabling the enterprises to immediately enlarge their offerings reach, which will include voice, video, IM and presence [17]. As traditional VoIP SIP traffic is unlocked, this is more dangerous in Wi-Fi networks where traffic is not ciphered. Denial of Service attacks (DoS attacks) [19] is based on signalling traffic, which is responsible for collapsing CPU, bandwidth or memory of the attacked device. The general objective is to obtain access to the logins, password or just extension numbers, through using a brute force kind of attacks. Fraud practices or illegal interception can be adopted through making use of this source of information, as well as this attack can also affect the SIP proxy server.

When an attacker is aware of the extension of a valid user or with the password or identifies a publicly accessible device that is unsafe, it becomes risky for the servers and the telephony services are also prone to more attacks as well as the wrong use of those services. In this case, generally calls are made to international destinations or on premium numbers, in order to get important revenues within a limited time framework. A stolen WebRTC account with an associated PSTN number, which can create vital legal problems that are beyond financial losses.

- **B. IdP Authentication Mechanism**

Security of third party IdPs- The IdP it is essential to keep a less number of IdPs. Most authentication mechanism, for its security purpose depends upon the IdP and peer connection. The IdP of every third party signifies a new universal trust point, hence of the IdPs can be recast as reliable IdPs, even if they issue unqualified identities, such as Facebook [17]. However, in some of the cases, implication of the user interface are not fully desirable. One transitional approach is to have unusual (potential user configurable) UI for huge, authoritative IdPs, hence permitting the user to directly grasp that the call is being confirmed by Google or Facebook.

C. AD-HOC ATTACKS IN WebRTC WebRTC is a browser-centric technology, and web browsers are supposed to be designed in such a manner in order to provide protection from different attacks. Traditional VoIP attacks include network attackers, who are aware with the weaknesses of the use of web-based services. WebRTC services must be provided a structured design, so that it can be used to prevent both possibilities [18].

The other concern of WebRTC is how to provide entrance to physical input devices to the system, namely microphone and webcam (or screen and file systems, in some scenarios). Through figure 6, how a user access a fake web address, and a malicious script is downloaded with the same link. When it gets an easy access to the camera or microphone, it starts a background call in the recording server where the voice calls are stored, which later on can be misused.

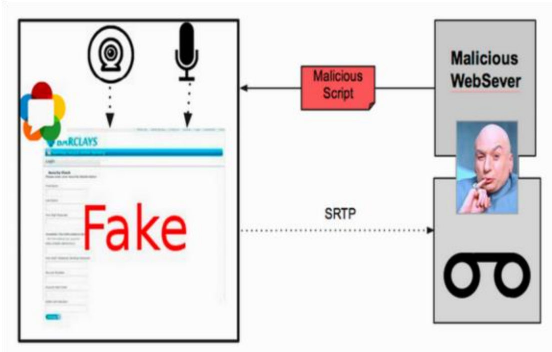
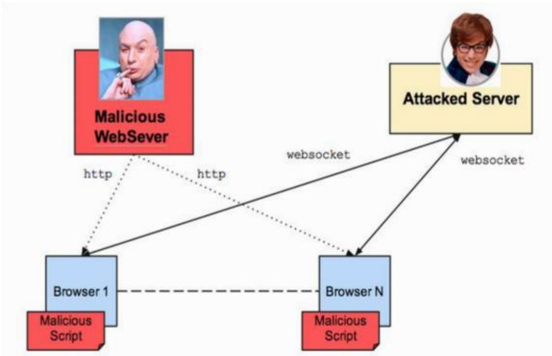


Figure. 6: WebRTC attack [16]

It is very necessary to properly detect and notify the calls to users. This threat is common for both events ringing tones and ringback respectively, reproducing the actions of any other physical telephone. A challenging issue for WebRTC is to provide call notification in smartphones when the browser is functioning in the background, a push mechanism must be used.

Websockets support cross-domain. It means that two distinct server domains of system can be connected [19]. It assists in providing necessary flexibility to make the Websockets useful, but it can also cause disruption by allowing distributed DoS.



- **Figure. 7: WebRTC attack [19]**

Suppose a web server is controlled by Doctor Evil, who hosts a forged web server, which offers free premium on an annual basis. A malicious server will be downloaded after getting access to the web. As soon as it is executed in the web browser, it will then try to make a connection to the attacked server. The execution in web browser can be controlled by implementing vigorous DoS protection policies.

Another appealing point regarding websockets, since they unlock a TCP socket through which all kinds of traffics are sent by using scripts [19]. If this traffic looks similar to the HTTP request and response, agents can potentially infer the traffic as valid and, for instance, involve the fake pages in their cache. In the case of DoS attacks, the services of WebRTC must be prevented from any other internet attacks [16]. Servers should be capable to block the IPs from where they are coming in contacts of various types of attacks.

In implementation of WebRTC, access of physical devices and DoS or DDoS are the two illustrations of potential threats. Besides this, network-level protections as well as the server authentication is treated as another key security element that should also be implemented.

X. Comparison between WebRTC and VoIP

	Pros	Cons
VOIP	Stable technology that supported by major telecommunication companies. It has extensive internet literature forums.	Specialize equipment needed with different variation. High cost of ownership needed
WebRTC	A free project based on browser communication system and easy suitable for client management system and low maintenance cost	Connectivity issues for client behind symmetric firewall. It has limited in just several browser.

Figure. 8: VoIP vs. WebRTC [20]

XI. 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:

1) **WebRTC enables any Web server to deliver a unique real-time communications experience:** The crucial point with WebRTC is that any website can develop into a control and delivery point for real-time communications and that the website, with HTML use, manages the experience, which a user receives. Among millions of operating websites, WebRTC delivers communication process in a major unique way. Further, communication experience does not have to include or reflect any traditional method of telecommunications values. For example, an icon of a walkie-talkie could signify the beginning of communications, rather than using a more traditional telephone. And real time can now be added to each website. For example, if the user wants to tag an image or other thing like status, using Pinterist, that pin is now available in easy representation. Thus, the beginning of communications is now through Pinterist, not through a phone call.

2) **Simplicity and reliability:**

WebRTC conveys simplicity, and through that reliability and availability of servers in two ways. First, the communication elements are put into a browser in an open standard. It removes the complexity associated with networks of building separate soft clients for each device [21]. With many devices, operating systems, and even in Android skins, a big challenge is to deploy a policy today is the complexity of network communication support. Every time an OS change, the client must be notified. With WebRTC, that challenge moves into the browser area, where there is a low number of touch points and the eco-system of interoperability has been developed [22].

The second key element of simplicity is that the client of WebRTC is stateless and take the help of the stimulus effort through the graphic side of the browser to the server to set off a change in status. During the initial period, when VoIP was started to make deliveries in the mid 90s, the model for real time on the IP communications was H323 operating between Personal Computers. As the initial VoIP systems were developed, H323 was rejected as associated with huge complexity and unreliability. All of the initial developers of VoIP followed the model, not the earlier H323 model [23]. With the development of SIP, the concept of intelligent, independent end point came into emergence. In SIP, the end points are both stateful and capable of self state change. And it led to enhanced complexity and the general lack of interoperability that exists in todays SIP servers [22]. WebRTC is an arrival to a stateless implementation where the stimulus effort is through the visual browser edge and the WebRTC media engine is under the management of web servers. This dramatically simplifies the process of implementing.

From the user perspective, the benefits are similar.

1) **WebRTC enables users to participate in a communications experience as delivered by any website:** With WebRTC a user can use any website and immediately can access to communication experience that is distinct from other websites. Now the user can literally select a website, rather than depending on other service providers for a unique experience. In this way, communication is not considered a separate event, but is a part of the overall experience.

2) **No extra downloading:**

Now a day, televisions, cars, appliance, kiosks all are becoming web-enabled, having a separate client or plug-in is virtually very difficult to maintain. For each service, maintaining a separate plug-in would prove to be very difficult. Most of the internet sites provide their own communications client [21]. With WebRTC, whether a site is visited on a regular basis or once in whole life, the user does not require to undertake any different activity to enable the communication process on real-time.

XII. 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 organisation'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.

XIII. Challenges

Mobility is one such area, where the WebRTC requires some improvement. WebRTC is still lacking in mobile device applications, so it is the area where performance is needed to increase [24].

Another specific lacuna is that, WebRTC is not getting proper support from Apple or Microsoft companies. Therefore, this reason make the other people also hesitant in providing support to the WebRTC application services.

XIV. FUTURE TRENDS

The WebRTC standards are not in last approved appearance, including the use of SDP and its interpretation. The manufacturers of browser have to incorporate WebRTC into their products. Discussions are going on as what type of codecs should be incorporated [4]. The area of internet browsing also requires an improvement with SIP and Jingle. Some vendors have produced a short term invention that uses Flash to achieve experience in the application of WebRTC capabilities.

XV. CONCLUSION

From the above paragraphs included in different section, it is concluded that WebRTC helps in enabling the network server to make the communication process easy as well as to provide it in a simple way. It is analysed from the above paragraphs that is the use of HTTP in JAVA application is also helpful for security purpose (the browser will provide notification or give an alert call that whether the page has both secure and nonsecure data, so that the user can make decisions regarding WebRTC continuance). Furthermore, the survey reveals that if there is no identity server involved, then the level of security can become better, but it will not be as much good as if there were an independent identity server.

Through the research paper, it is analysed that WebRTC requires a communication application to make an inquiry from the user if it can access the camera and microphone. Various useful applications of WebRTC has been discussed in the above sections. The research paper will be of great help for the readers.

REFERENCES

[1] L. L. Fernandez, M. P. Diaz, M. R. Benitez, F. J. Lopez, J. A. Santos, J.A. "Catalysing the success of WebRTC for the provision of advanced multimedia real-time communication services," Intelligence in Next Generation Networks (ICIN), 2013 17th International Conference on , pp.23,30, 2013.

[2] L. L. Fernandez, M. P. Diaz, R. B. Mejias, F. J. Lopez, J. A. Santos, "Kurento: a media server technology for convergent WWW/mobile real-time multimedia communications supporting WebRTC," World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a , pp.1,6, 2013.

[3] S. Loreto, Romano, S. Pietro, "Real-Time Communications in the Web: Issues, Achievements, and Ongoing Standardization Efforts," Internet Computing, IEEE , vol.16, no.5, pp.68,73, 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 Applications Conference (SIU), 2014 22nd , pp.256,259, 2014

[6] WebRTC Architecture. n.d. [Online]. Available at: https://www.safaribooksonline.com/library/view/real-time-communicationwith/9781449371869/ch01.html [Accessed on: 8 December 2014].

[7] L. Wenfeng; C. Guojun; H. Fei, "Notice of Retraction Short-Distance Wireless Voice Communication Technology Based on ZigBee," Management and Service Science, 2009. MASS '09. International Conference on , pp.1,4, 2009

[8] M. J. Werner, C. Vogt, T. C. Schmidt, "Let Our Browsers Socialize: Building User-Centric Content Communities on WebRTC," Distributed Computing Systems Workshops (ICDCSW), 2014 IEEE 34th International Conference on , pp.37,44, 2014.

[9] W. Elleuch, "Models for multimedia conference between browsers based on WebRTC," Wireless and Mobile Computing, Networking and Communications (WiMob), 2013 IEEE 9th International Conference on , pp.279,284, 2013

[10] C. Vogt, M. J. Werner, T. C. Schmidt, "Leveraging WebRTC for P2P content distribution in web browsers," Network Protocols (ICNP), 2013 21st IEEE International Conference on , pp.1,2, 2013

[11] S. Loreto, S. P. Romano, "Real-Time Communications in the Web: Issues, Achievements, and Ongoing Standardization Efforts," Internet Computing, IEEE , vol.16, no.5, pp.68,73, 2012.

[12] R. L. Barnes, M. Thomson, "Browser-to-Browser Security Assurances for WebRTC," Internet Computing, IEEE , vol.18, no.6, pp.11,17, 2014.

[13] V. Beltran, E. Bertin, N. Crespi, "User Identity for WebRTC Services: A Matter of Trust," Internet Computing, IEEE , vol.18, no.6, pp.18,25, 2014.

[14] D. Mohney, 2014. Establishing Trust: Real Time Communications Identity Management. [Online]. Available at: http://www.realtimecommunicationsworld.com/topic s/realtimecommunicationsworld/articles/385397establishing-trust-real-time-communicationsidentity-management.htm [Accessed on: 10 December 2014].

[15] A. Zeidan, A. Lehmann, U. Trick, "WebRTC enabled multimedia conferencing and collaboration solution," WTC 2014; World Telecommunications Congress 2014; Proceedings of , pp.1,6, 2014.

[16] L. Soto. 2014. WebRTC Security Concerns. [Online]. Available at: http://blog.webrtcsolutions.com/webrtc-security-concerns/ [Accessed on: 11 December 2014].

[17] E. Rescorla. 2013. Security Considerations. [Online]. Available at: https://tools.ietf.org/html/draft-ietfrtcweb-security-arch-07 [Accessed on: 11 December 2014].

[18] L. Cai, R. R. Cessa, "Containing sybil attacks on trust management schemes for peer-to-peer networks," Communications (ICC), 2014 IEEE International Conference on, pp.841,846, 2014.

[19] M. Adeyeye, I. Makitla, T. Fogwill, "Determining the signalling overhead of two common WebRTC methods: JSON via XMLHttpRequest and SIP over WebSocket," AFRICON, pp.1,5, 2013.

[20] I. Maglogiannis, V. Pigadas, and V.P. Plagianakos. CLOUD BASED AUDIO AND VIDEO COMMUNICATION IN PERSONAL HEALTH SYSTEMS: VOIP AND WEBRTC. European, Mediterranean & Middle Eastern Conference on Information Systems. 2014.

[21] P. Edholm. The Benefits of WebRTC. [Online]. Available at: http://www.webrtcworld.com/topics/from-theexperts/articles/319037-benefits-webrtc.htm [Accessed on: 11 December 2014].

[22] A. Johnston, J. Yoakum, K. Singh, "Taking on webRTC in an enterprise," Communications Magazine, IEEE , vol.51, no.4, pp.48,54, 2013

[23] J. Sun, G. Zhu, D. Xu, "Possibilities of Voice Resource DoS Attacks Based on H.323 Protocol in Softswitch Network,"

Multimedia Technology (ICMT), 2010 International Conference on, pp.1,4, 2010.

[24] C. S. Park, J. W. Jeon, "Application of Mobile Communication System to Improve the Accessibility of the Visually Disabled," New Trends in Information and Service Science, 2009. NISS '09. International Conference on , pp.792,796, 2009.