Webrtc implementations and ortc

Science, Computer Science

WebRTC Implementations and ORTC

WebRTC (Web Real-Time Communication) is a cutting-edged browser-based real-time multimedia communication technology. The technology makes the Web browser a universal platform for real-time audio, video and data communication between all user devices, such as mobile terminals and personal computers without relying on any third-party plug-ins, and it is an API that supports web browsers for real-time voice conversations or video conversations, which has played an increasingly significant role in recent years. So this technology is developed to give Web browsers real-time communication capabilities and at the same time, web application developers can quickly develop them through HTML tags and JavaScript APIs by encapsulating these capabilities and opening them to Web application developers in the form of JavaScript APIs[1].

It has been open source since June 1, 2011 and included in the W3C Recommendations of the World Wide Web Consortium with support from Google, Mozilla, and Opera. Prior to the advent of WebRTC technology, Web browser-based multimedia communication services are built on specific browser plug-ins such as Adobe Flash Player since the HTTP protocol that the browser relies on is a connectionless state[2]. That is, the browser establishes a connection with the server, then downloads the file, disconnects and displays the received file. Therefore, the browser cannot dynamically exchange data with the server in real time. The connectionless HTTP protocol causes services that require real-time communication such as online chat, which needs to be implemented only by the means of periodic access to the server. However, as the number of visiting users increases, this approach surprisingly consumes server-side resources, which in turn seriously affects server performance.

ORTC(Object Real-Time Communication). ORTC is a real-time object-based messaging solution for web and mobile applications. ORTC delivers messages, content and data to a variety of platforms and devices as well as across applications. As Avram[3] states that ORTC has now completed enough content to achieve experience so that more people can be called to implement it. More exactly, there are several benefits which will be brought by ORTC as below. First of all, the ORTC API is well-suited for the “ mobile-first, cloud-first” world because it supports advanced video features like scalable video encoding and simulcast. These advanced video technologies have been proved difficult to support in an interoperable manner in SDP in WebRTC 1. 0.

In contrast, using these advanced video technologies in the JavaScript Object API is much simpler. In addition, the issues on the ORTC also lead to the discovery of the need to create new web communication specifications. With JavaScript, ORTC gives us more access to more controls, which provides more power and flexibility when web developers build real-time communication applications and features such as layered video encoding, encoding format based on each track and so on. Services of WebRTC 1. 0 The rapid spread of WebRTC technology will have a huge impact and provide many services. For example, web application like IM(Instant Message) application services are provided and gradually form an ecosystem through a large number of users they own and through derivative applications built on these IM applications[4]. These IM applications and the ecosystem they generate are trying to replace browsers as Internet portals.

In fact, this phenomenon is much more prominent in mobile terminals. However, the deployment and use of WebRTC technology will show explosive growth, which will lead to the transition of IM applications to IM Web applications and the migration of users to IM Web applications and beyond the WebRTC 1. 0. Therefore, the ecosystem built on existing IM applications could probably be broken so the IM applications and their ecosystems will be adapted to the new technology environment. At the same time, the popularity of WebRTC will bring regulatory and security issues to enterprises. At present, some companies have blocked the ports of IM applications like Skype within the company to prevent employees from working in the process due to the safety problems caused by the inadvertent use of audio and video tools[5].

The new IM Web application based on WebRTC is not different from a normal web page, so it is difficult to detect such an IM Web application since they probably do not follow the WebRTC 1. 0. This has caused huge hidden dangers to the supervision and security of enterprises. Secondly, the service of smart TV will be greatly provided and enhanced. Currently, smart TVs mainly include intelligent operating systems and display devices. Inside of them, session management for WebRTC is also known as abstract signaling, which is also offered and of course following the WebRTC 1. 0. Further, the role of abstract signaling is to separate the specific services from the signaling control in order to adapt to the signaling control in the majority of Web applications.

Finally, the services of both audio and video are provided by WebRTC 1. 0. More precisely, the audio engine is responsible for audio processing from the microphone to the network side, the network side to the speakers. It contains functions mainly in audio codec and sound processing. In audio codec, WebRTC mainly uses two voice coding formats: iSAC (internet Speech Audio Codec) and iLBC (internet Low Bitrate Codec) to encode the voice of broadband and narrowband environments respectively[6]. In sound processing, WebRTC mainly includes functions such as echo cancellation, error concealment and noise reduction processing to reduce the impact on network quality caused by network jitter and packet loss, and to reduce the sound delay as much as possible. What is more, the video processing engine is responsible for video processing from the camera to the network side and from the network side to the screen display. It mainly includes video codec and image processing.

In video codec, WebRTC currently uses VP8 technology, enabling WebRTC to provide higher quality video in a lower bitrate environment. In image processing, WebRTC mainly includes functions such as jitter buffering and image enhancement to reduce the noise of images captured from the camera. Domains based on WebTRC 1. 0First, it offers some browser-based Web applications for the common users in relevant existing VoIP business field. More specifically, we are currently experiencing an era of hardware diversification, such as the rise of wearable devices, which will facilitate the emergence of operating systems that adapt to new devices and increase the differentiation of existing operating systems. As a result, the heterogeneity and complexity of operating systems has led to a dramatic increase in the complexity and cost of developing, updating, and maintaining VoIP applications[7].

With the integration of WebRTC into the browser, Web-based real-time communication applications can be quickly developed through HTML tags and JavaScript APIs, and avoid duplicate development due to heterogeneity of the operating system and maintenance due to version upgrades. Fees and inconvenience will lead existing application developers to move from existing browser-based plug-ins or native VoIP application models to WebRTC for Web application development. At present, the customer service and online education fields involving VoIP services benefit a lot from the application of WebRTC technology in their own fields.

The main business on smart TV includes also the live broadcast and on-demand of multimedia sources. With the popularity of WebRTC and the emergence of a large number of IM Web applications, smart TVs will be more likely to include camera devices natively, which provides the user more entertainment. In the meanwhile, the video conferencing service will become the main business of smart TVs, just previous two sources. Then, due to the limitations of the audio and video formats supported by WebRTC, and the limitations of the optional mesh structure are used by WebRTC in multiplayer video sessions. A Web browser that integrates with WebRTC will further change the landscape of traditional applications, Web browsers and operating systems in the internet portal field[8]. To be more specific, web browsers will become a platform between operating systems and web applications. Browsers that include WebRTC will further strengthen their Internet portal status. In recent years, popular applications such as WeChat have attempted to become Internet portals through the large number of users they own and the derivative applications developed on them[9].

Web browsers and their web applications will become an ecosystem. The development of high-performance browsers will become the next competitive hotspot. Browser security field will also be more prominent. Sometimes, the developers or designers will not follow WebRTC 1. 0 because there are several main disadvantages. More specifically, it lacks design and deployment of server solutions. In addition, the transmission quality is difficult to guarantee as the transmission design of WebRTC is based on P2P, which is difficult to guarantee the transmission quality, and the optimization methods are limited[10]. By what it means, only end-to-end optimization can be done, and it is hard to cope with the complex Internet environment. For example, the transmission quality in scenarios such as cross-regional, inter-operator, low-bandwidth, etc. Besides, WebRTC is more suitable for one-to-one single chat, although it can be extended to achieve group chat, there is no optimization for group chat, especially for large group chat. As for device-side adaptation, such as echo, recording failure and other issues emerge one after another.

Although it can also be used for native development, it has more complex domain design and API granularity because of the domain knowledge like audio and video acquisition, processing, codec, real-time transmission, so the compilation of projects is not an easy task to implement. Purpose of ORTCObject Real-Time Communication (ORTC) is a free, open-ended project which allows mobile devices to communicate with real-time communication and enabled servers and web browsers via native and simple Javascript APIs[11]. The Object RTC component is being optimized to best accomplish the functions, based on WebRTC. ORTC-related APIs and features will continue to be added to the WebRTC browser a little bit. This means that the competition between WebRTC and ORTC has been alleviated, especially Microsoft plans to support WebRTC 1. 0 in the Edge browser.

This also means that new features and functionality may not be added to the SDP layer of WebRTC, but will be exposed through the object model defined in ORTC. Relationship between ORTC and WebRTCSometimes, people tend to regard ORTC as WebRTC easily but actually ORTC is not a substitute for WebRTC 1. 0 since both of them are real-time communication technologies which can transfer the audio and video files via mobiles or computers. As a matter of fact, some ORTC concepts have been incorporated into the WebRTC 1. 0 specification like RtpSender and RtpReceiver so some people consider that ORTC is the next generation of WebRTC 1. 0[12]. In the meanwhile, Google is also trying to add ORTC API into the Chrome source code. Indeed, there is actually no such thing as ORTC and WebRTC being mutually exclusive.

There are already many ORTC objects in WebRTC 1. 0 and then in the next version of WebRTC NV in 1. 0, WebRTC will continue to strengthen its integration with ORTC objects while maintaining backward compatibility. Microsoft only supports ORTC in its original version of Edge, but has now added support for WebRTC 1. 0. Moreover, the main APIs of ORTC are implemented by JavaScript so if we want to modify the same controls in WebRTC 1. 0, we just focus on the source code changes, which provides the convenience for web developers since they prefer to have more controls in their applications, rather than waiting for updates from browsers or other software vendors. ConclusionIn this paper, we first introduce the specification and analyze the significance of the WebRTC and ORTC technology. Then we illustrate the various services of the WebRTC technology architecture and the encapsulated JavaScript API and the correspondence.

Next, we describe the impact of WebRTC on the existing real-time communications industry and the smart TV industry, and explain that the introduction of WebRTC will further change the landscape of applications, Web browsers and operating systems. In addition, we figure out the domains based on WebRTC 1. 0 and the downsides and reasons why some designers or developers do not tend to choose WebRTC 1. 0. Also, we make the judgment that the deployment and use of WebRTC technology will show explosive growth in mobile terminals and personal PCs and the support of these carriers to WebRTC. Next, we demonstrate the purpose of the ORTC and finally, we compare these two technologies and draw upon the conclusions.

References:

1. L. Wyse,” The Viability of the Web Browser as a Computer Music Platform,” Apr. 9, 2013.[Online]. Available: https://www. mitpressjournals. org/doi/pdf/10. 1162/COMJ_a_00213.[Accessed Sept. 29, 2018].
2. Remus,” Real-Time Communication Without Plugins,” Nov. 12, 2015. [Online]. Available: http://avchat. net/blog/what-is-the-state-of-webrtc/. [Accessed Sept. 29, 2018].
3. A. Avram,” ORTC and the Future of WebRTC, ” Aug. 21, 2014.[Online]. Available: https://www. infoq. com/news/2014/08/ortc-webrtc. [Accessed Sept. 29, 2018].
4. Moxie,” Amazon threatens to suspend Signal’s AWS account over censorship circumvention,” May. 01, 2018.[Online]. Available: https://signal. org/blog/.[Accessed Sept. 29, 2018].
5. C. Hava,” Tech companies are betting the future of work is messaging,” Jan. 08, 2018.[Online]. Available: http://www. hrmonline. com. au/section/featured/tech-companies-future-work-messaging/. [Accessed Sept. 29, 2018].
6. H. Astrom,” Internet Low Bit Rate Codec (iLBC),” Jul. 01, 2018.[Online]. Available: https://www. researchgate. net/publication/277293811_Internet_Low_Bit_Rate_Codec_iLBC. [Accessed Sept. 29, 2018].
7. E. Ogheneovo,” On the Relationship between Software Complexity and Maintenance Costs,” Nov. 05, 2014.[Online]. Available: https://file. scirp. org/pdf/JCC_2014112117243300. pdf.[Accessed Sept. 29, 2018].

8. S. Wahid,” Rich Internet Application (RIA) New Dimension,” Apr. 30, 2015.[Online]. Available: https://pdfs. semanticscholar. org/b1b3/6498de7580c4aa9d6a716ccb3e37cc2237be. pdf.[Accessed Sept. 30, 2018].
9. H. Zhou,” Making Sense of Web 3,” May. 27, 2018.[Online]. Available: https://tampub. uta. fi/bitstream/handle/10024/103861/1530187275. pdf? sequence= 1&isAllowed= y.[Accessed Sept. 30, 2018].
10. P. Vílchez,” Real-time Communication Network Architecture Design,” Aug. 6, 2017.[Online]. Available: https://repositori. upf. edu/bitstream/handle/10230/25522/VilchezBlanco_2015. pdf? sequence= 1. [Accessed Sept. 30, 2018].
11. X. Chen, “ Unified Communication and WebRTC,” Mar . 11, 2009. [Online]. Available: https://brage. bibsys. no/xmlui/bitstream/handle/11250/263027/749349_FULLTEXT01. pdf? sequence= 1. [Accessed Sept. 30, 2018].
12. S. Loreto,” Object RTC (ORTC) API for WebRTC,” Aug. 20, 2016. [Online]. Available: http://publications. ortc. org/2016/20160820/. [Accessed Sept. 30, 2018].
13. R. Sheppy,” WebRTC API, ” Apr. 03, 2018.[Online]. Available: https://developer. mozilla. org/en-US/docs/Web/API/WebRTC_API. [Accessed Sept. 30, 2018].