NTNU – Trondheim
Norwegian University of
Science and Technology

Unified Communication and WebRTC

Xiao Chen

Submission date: May 2014
Responsible professor: Mazen Malek Shiaa, ITEM
Supervisor: Mazen Malek Shiaa, ITEM

Norwegian University of Science and Technology
Department of Telematics

Title:	Unified Communication and WebRTC
Student:	Xiao Chen

Problem description:

Web Real-Time Communication (WebRTC) offers application developers the ability to write rich, real-time multimedia application (e.g. video chat) on the web, without requiring any plugins, downloads or installations. WebRTC is also currently the only existing soon-to-be standardized technology on the market to create horizontal cross-platform communication services, encompassing smartphones, tablets, PCs, laptops and TVs, which adds value for both consumers and enterprises. WebRTC gives operators the opportunity to offer telephony services to more devices, such as PCs, tablets and TVs. This thesis considers how WebRTC can enhance the existing echo-systems for telephony and messaging services by providing the end-user rich application client.

It will also covers research about different solutions to implement WebRTC to cooperate with existing telephony services like hosted virtual Private Branch Exchange (PBX) services.

A prototype of WebRTC deployment based on different rich communication scenarios will be implemented along with this thesis. Some corresponding test and evaluation will be fulfilled in this prototype.

Research about advanced WebRTC usability in telephony and messaging services will be covered in this thesis by the feedback of the WebRTC prototype

Responsible professor:	Mazen Malek Shiaa, ITEM
Supervisor:	Mazen Malek Shiaa, ITEM

Abstract

For the development of traditional telephony echo-systems, the cost of maintenance traditional telephony network is getting higher and higher but the number of customer does not grow rapidly any more since almost every one has a phone to access the traditional telephony network. WebRTC is an Application Programming Interface (API) definition drafted by the World Wide Web Consortium (W3C) that supports browser-to-browser applications for voice calling, video chat, and Peer-To-Peer (P2P) file sharing without plugins.[Wik14d] "This technology, along with other advances in HyperText Markup Language 5 (HTML5) browsers, has the potential to revolutionize the way we all communicate, in both personal and business spheres."[JB13a]

As network operators aspect, WebRTC provides many opportunities to the future telecommunication business module. For the users already have mobile service, operator can offer WebRTC service with session-based charging to the existing service plans. Messaging APIs can augment WebRTC web application with Rich Communication Services (RCS) and other messaging services developers already know and implement. Furthermore, since WebRTC is a web based API, then the implementation of Quality of Service (QoS) for WebRTC can provide assurance to users and priority services (enterprise, emergency, law enforcement, eHealth) that a WebRTC service will work as well as they need it to. WebRTC almost provide network operator a complete new business market with a huge amount of end-users.

As an end-user aspect, WebRTC provides a much simpler way to have real-time conversation with another end-user. It is based on browser and internet which almost personal or enterprise computer already have, without any installation and plugins, end-user can have exactly the same service which previous stand-alone desktop client provides. By the system this thesis will cover, the end-user even can have the real-time rich communication service with multiple kinds of end-users.

This thesis will cover the research about how to apply WebRTC technology with existing legacy Voice over Internet Protocol (VoIP) network.

Keywords : WebRTC, AngularJs, Nodejs, SIP, WebSocket, Dialogic XMS

Preface

WebRTC is quite popular topic in the web development filed since the massive usage and development of HTML5 web application on the internet. The initial purpose of this web API is to provide the browser client the ability to create real-time conversation between each other. After many WebRTC based application come out the market, it is quite normal to think about how to integrate these kind of web application with the current legacy telephony network as the next big step for this technology. The requirement of this process is not only from the traditional telephony operator but also the normal end-users. The approach to achieve this goal is the man purpose of this thesis.

Research about current WebRTC technology usage and development of a WebRTC prototype system are the two main parts of this thesis. The prototype system is implemented by regarding to the research of WebRTC integrated with legacy telephony network.

Current status of WebRTC technology, WebRTC business use cases, analysis of different possible WebRTC implement solutions and WebRTC system architecture will be covered in this thesis. Some research regarding with the development of WebRTC prototype system will be covered in this thesis as well.

The prototype described in this thesis is implemented to cooperate with existing legacy VoIP network services through Session Initiation Protocol (SIP) server and PBX[1] service. It will provide most of essential functions which are included in the legacy telephony business, besides other communication functions used on web. Moreover, some analysis and discussion about the feedback of the prototype will be covered in this thesis.

The prototype will be implemented in programming language Javascript for both client font-end and server back-end by using the AngularJs framework and Nodejs framework mainly. The approach and reason to choose these framework and programming language will be expounded in the later chapter in this thesis.

[1] Users of the PBX share a certain number of outside lines for making telephone calls external to the PBX.[Web14]

Acknowledgment

Written by Xiao Chen in Trondheim in May 2014

Thanks for Mazen Malek Shiaa, ITEM

Frank Mbaabu Kiriinya, Gintel AS

Roman Stobnicki, Dialogic, the Network Fuel company

Special thanks for Gintel AS

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

API Application Programming Interface.

GIPS Global IP Solutions.

HTML5 HyperText Markup Language 5.

HTTP Hypertext Transfer Protocol.

HTTPS Hypertext Transfer Protocol over Secure Socket Layer.

ICE Interactive Connectivity Establishment.

IETF Internet Engineering Task Force.

NAT Network Address Translator.

P2P Peer-To-Peer.

PBX Private Branch Exchange.

PSTN Public Switched Telephone Network.

QoS Quality of Service.

RCS Rich Communication Services.

RTC Real-Time Communication.

SDP Session Description Protocol.

SIP Session Initiation Protocol.

STUN Session Traversal Utilities for NAT.

TCP Transmission Control Protocol.

TURN Traversal Using Relays around NAT.

UDP User Datagram Protocol.

VoIP Voice over Internet Protocol.

W3C World Wide Web Consortium.

WebRTC Web Real-Time Communication.

Chapter 1
Introduction

In this Chapter, introduction of WebRTC and SIP network will be covered. SIP is one of the VoIP signaling protocols widely used in current internet telephony service which is the target telephony network integrated with WebRTC application system in this thesis.

1.1 WebRTC

Gmail[1] video chat became popular in 2008, and in 2011 Google introduced Hangouts[2], which use the Google Talk service (as does Gmail). Google bought Global IP Solutions (GIPS), a company which had developed many components required for Real-Time Communication (RTC), such as codecs and echo cancellation techniques. Google open sourced the technologies developed by GIPS and engaged with relevant standards bodies at the Internet Engineering Task Force (IETF) and W3C to ensure industry consensus. In May 2011, Ericsson built the first implementation of WebRTC.

1.1.1 What is WebRTC ?

WebRTC is an industry and standards effort to put real-time communications capabilities into all browsers and make these capabilities accessible to web developers via standard HTML5 tags and JavaScript APIs. For example, consider functionality similar to that offered by Skype[3]. but without having to install any software or plug-ins. For a website or web application to work regardless of which browser is used, standards are required. Also, standards are required so that browsers can

[1] Gmail is a free , advertising-supported email service provided by Google.

[2] Google Hangouts is an instant messaging and video chat platform developed by Google, which launched on May 15, 2013 during the keynote of its I/O development conference. It replaces three messaging products that Google had implemented concurrently within its services, including Talk, Google+ Messenger, and Hangouts, a video chat system present within Google+.

[3] Skype is a freemium voice-over-IP service and instant messaging client, currently developed by the Microsoft Skype Division.[Wik14c]

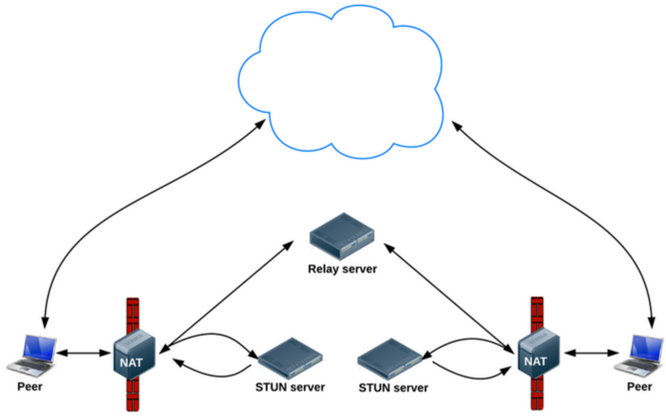

Figure 1.1: WebRTC Network: Finding connection candidates

communicate with non-browsers, including enterprise and service provider telephony and communications equipment[JB13c].

With the rapidly development of internet, more and more communication traffic is moving to web from the traditional telephony network. And in the recent decade, VoIP network services are growing to the peek of the market capacity. Solution to integrate WebRTC and existing VoIP network is the right approach the trend of the internet communication requirement.

1.1.2 WebRTC Network Structure

In the Figure1.1[Dut14] showing how the Interactive Connectivity Establishment (ICE) framework[4] to find peer candidate through Session Traversal Utilities for NAT (STUN) server and its extension Traversal Using Relays around NAT (TURN) server. The difference and usage of STUN server and TURN server will be discussed more detail in Chapter 3.

WebRTC needs server to help users discover each other and exchange 'real world' details such as names. Then WebRTC client applications (peers) exchange network information. After that, peers exchange data about media such as video format and resolution. Finally, WebRTC client applications can traverse Network Address Translator (NAT) gateways and firewalls.

Compare to the traditional telephony network which is shown in Figure1.2[Inc05], the main difference between these two communication network is that WebRTC is

[4]ICE is a framework for connecting peers, such as two video chat clients.[Wik14a]

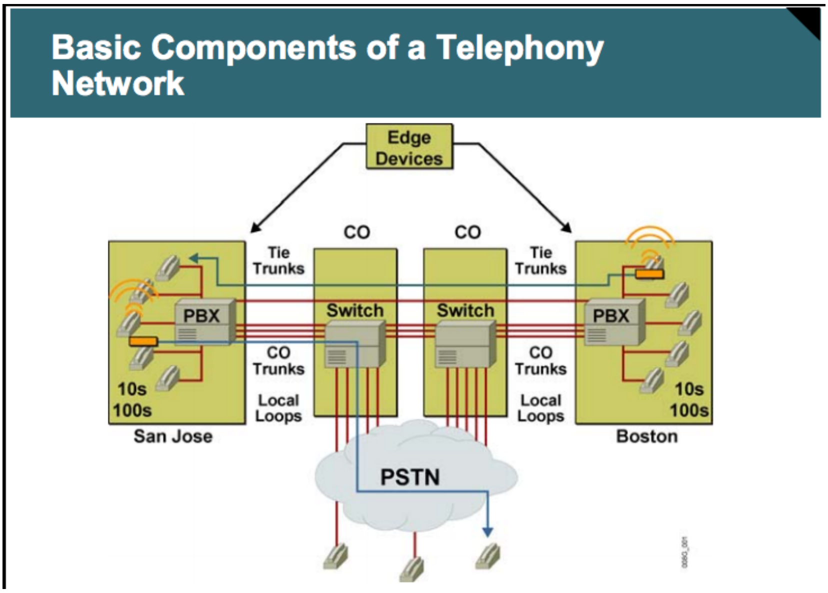

Figure 1.2: Traditional Telephony Network

P2P communication, in STUN server scenario, after the signaling between end-peers, the media data are exchanged directly between tow peers, but in the traditional telephony, all the media data are transferred to PBX and switches regarding to Public Switched Telephone Network (PSTN)[5] then reach the other side of the peer. Even in TURN server scenario, the media stream is only relaying to the TURN then directly transfer to another peer, no switches involved.

1.1.3 WebRTC Implementation APIs

There are four main steps to implement a WebRTC session. The Figure 1.3 shows that the browser client need to obtain local media first, then set up a connection between the browser and the other peer through some signaling, after that attach the media and data channels to the connection, afterwards exchange the session description from each other. Finally the media stream will automatically exchange through the real-time peer to peer media channel.

In order to obtain local media, the WebRTC APIs provide *getUserMedia()* function to get the video and audio stream from user. For privacy reasons, a web application's

[5]The PSTN consists of telephone lines, fiber optic cables, microwave transmission links, cellular networks, communications satellites, and undersea telephone cables, all interconnected by switching centers, thus allowing any telephone in the world to communicate with any other. Originally a network of fixed-line analog telephone systems, the PSTN is now almost entirely digital in its core network and includes mobile and other networks, as well as fixed telephones.[Wik14b]

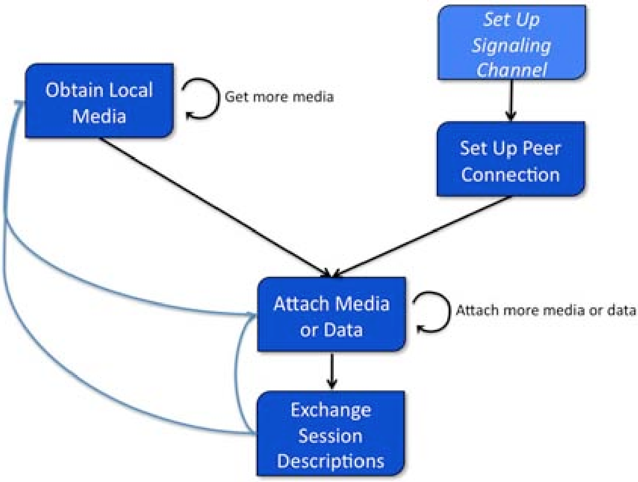

Figure 1.3: WebRTC API View with Signaling[JB13b]

request for access to a user's microphone or camera will only be granted after the browser has obtained permission from the user. *getUserMedia()* function is currently available in Chrome, Opera and Firefox. Almost all of the WebRTC APIs are slightly different in different browsers, then this part will be discussed in the Chapter 2. However, since WebRTC APIs is not standard API yet, the prototype application in this thesis will not pay too much work-load on compatibility for different browsers platform.

For setting up peer connection, the core of WebRTC is the *RTCPeerConnection* API, which sets up a connection between two peers. In this context, "peers" means two communication endpoints on the World Wide Web. Instead of requiring communication through a server, the communication is direct between the two entities. In the specific case of WebRTC, a peer connection is a direct media connection between two web browsers. This is particularly relevant when a multi-way communication such as a conference call is set up among three or more browsers. Each pair of browsers will require a single peer connection to join them, allowing for audio and video media to flow directly between the two peers. The system architecture regarding to this WebRTC factor will be discussed in later Chapter 2 since there is consideration between centralized media server network and mesh network architecture.

To establish this connection requires a new *RTCPeerConnection* object. The only input to the *RTCPeerConnection* constructor method is a configuration object containing the information that ICE, will use to "punch holes" through intervening NAT devices and firewalls. There are two APIs to handle the *IceCandidate* object

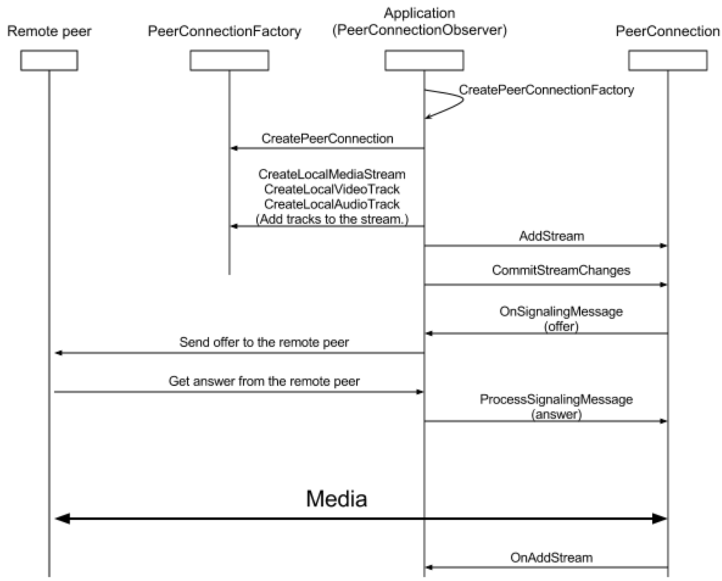

Figure 1.4: WebRTC Set up a call Process

which contains ICE information data. One is *onicecandidate* listener to trigger the function to handle the new *IceCandidate* data object. The other one is *addIceCandidate* function to add the new *IceCandidate* data object to the remote/local peer connection session description field.

ICE is a framework for connecting peers, such as two video chat clients. Initially, ICE tries to connect peers directly, with the lowest possible latency, via User Datagram Protocol (UDP). In this process, STUN servers have a single task: to enable a peer behind a NAT to find out its public address and port. If UDP fails, ICE tries Transmission Control Protocol (TCP): first Hypertext Transfer Protocol (HTTP), then Hypertext Transfer Protocol over Secure Socket Layer (HTTPS). If direct connection fails—in particular, because of enterprise NAT traversal and firewalls—ICE uses an intermediary (relay) TURN server. In other words, ICE will first use STUN with UDP to directly connect peers and, if that fails, will fall back to a TURN relay server. The expression 'finding candidates' refers to the process of finding network interfaces and ports.[Dut14]

Once the *RTCPeerConnection* is established, the client need configure where the media or data to store and display if it is necessary. In the prototype application of this thesis, media stream will be displayed in a HTML5 tag called *<video>*. It will only be shown when there is media stream in *<video>* tag source.

In order to make the WebRTC STUN server or TURN server to generate the ICE candidate for the peer client, the caller *RTCPeerConnection* need run *createOffer()* function and the callee need run *createAnswer()* function to ask the STUN/TURN server to find the path for each other peer.There is one calling process shown in Figure1.4, it is a set up call process from caller peer.

WebRTC clients (known as peers) also need to ascertain and exchange local and remote audio and video media information, such as resolution and codec capabilities. Signaling to exchange media configuration information proceeds by exchanging an offer and an answer using the SDP. The *createOffer()* function and *createAnswer()* function both have callback function to handle the SDP either to call *setLocalDescription()* by caller or call *setRemoteDescription()* by callee when callee gets the caller's SDP from WebRTC offer. The Log Snippet1.1 shown is the WebRTC answer SDP from the callee when the callee end-point decide to accept this conversion session.

1.2 SIP Network

Log Snippet 1.1 Sample WebRTC Answer SDP

```
sdp: v=0
o=xmserver 1399363527 1399363528 IN IP4 10.254.9.135
s=xmserver
c=IN IP4 10.254.9.135
t=0 0
a=ice-lite
m=audio 49152 RTP/SAVPF 0 126
a=rtpmap:0 PCMU/8000
a=sendrecv
a=rtcp:49153
a=candidate:1 1 UDP 2130706431 10.254.9.135 49152 typ host
a=candidate:1 2 UDP 2130706430 10.254.9.135 49153 typ host
...
a=acfg:1 t=1
a=rtpmap:126 telephone-event/8000
a=fmtp:126 0-15
m=video 57344 RTP/SAVPF 100
b=AS:1000
a=rtpmap:100 VP8/90000
a=fmtp:100 max-fr=30; max-fs=1200
a=sendrecv
a=rtcp:57345
a=rtcp-fb:100 ccm fir
a=rtcp-fb:100 nack
a=rtcp-fb:100 nack pli
a=rtcp-fb:100 goog-remb
a=candidate:2 1 UDP 2130706431 10.254.9.135 57344 typ host
a=candidate:2 2 UDP 2130706430 10.254.9.135 57345 typ host
...
```

Chapter 2
System Development

Chapter 3
System Deployment

References

[Dut14] Sam Dutton. Getting started with webrtc — html5rocks, 2014. [Online; accessed 2-May-2014].

[Inc05] Cisco Systems Inc. Differences between traditional telephony and voip, 2005. [Online; accessed 2-May-2014].

[JB13a] Alan B Johnston and Daniel C Burnett. *WebRTC: APIs and RTCWEB Protocols of the HTML5 Real-Time Web*, chapter Preface, page 12. Digital Codex LLC, second edition, 2013.

[JB13b] Alan B Johnston and Daniel C Burnett. *WebRTC: APIs and RTCWEB Protocols of the HTML5 Real-Time Web*, chapter How to Use WebRTC, page 33. Digital Codex LLC, second edition, 2013.

[JB13c] Alan B Johnston and Daniel C Burnett. *WebRTC: APIs and RTCWEB Protocols of the HTML5 Real-Time Web*, chapter Introduction to Web Real-Time Communications, page 15. Digital Codex LLC, second edition, 2013.

[Web14] Webopedia. Pbx - private branch exchange — webopedia, 2014. [Online; accessed 2-May-2014].

[Wik14a] Wikipedia. Interactive connectivity establishment — Wikipedia, the free encyclopedia, 2014. [Online; accessed 2-May-2014].

[Wik14b] Wikipedia. Public switched telephone network — Wikipedia, the free encyclopedia, 2014. [Online; accessed 2-May-2014].

[Wik14c] Wikipedia. Skype — Wikipedia, the free encyclopedia, 2014. [Online; accessed 2-May-2014].

[Wik14d] Wikipedia. Webrtc — Wikipedia, the free encyclopedia, 2014. [Online; accessed 29-April-2014].