Albert Abelló Lozano

Performance analysis of WebRTC

School of Electrical Engineering

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 20.3.2012

Thesis supervisor:

Prof. Jörg Ott

Thesis advisor:

M.Sc. (Tech.) Varun Singh

Aalto University
School of Electrical
Engineering

AALTO UNIVERSITY
SCHOOL OF ELECTRICAL ENGINEERING

ABSTRACT OF THE
MASTER'S THESIS

Author: Albert Abelló Lozano		
Title: Performance analysis of WebRTC		
Date: 20.3.2012	Language: English	Number of pages:4+6
Department of Communication and Networking		
Professorship: Networking Technology		Code: S-55
Supervisor: Prof. Jörg Ott		
Advisor: M.Sc. (Tech.) Varun Singh		

Your abstract in English. Try to keep the abstract short, approximately 100 words should be enough. Abstract explains your research topic, the methods you have used, and the results you obtained.

Keywords: Resistor, Resistance,
Temperature

Preface

Thank you everybody.

Otaniemi, 9.3.2012

Albert Abelló Lozano

Contents

List of Tables

List of Figures

Definitions and abreviations

1 Introduction

The need of a new way to communicate between two points of the planet has been arising as a problem to be solved by our new media distributors. Old systems such as Skype or traditional video calls are not able to cope the needs of the new generations of developers and users that everyday require a more integrated way of communication.

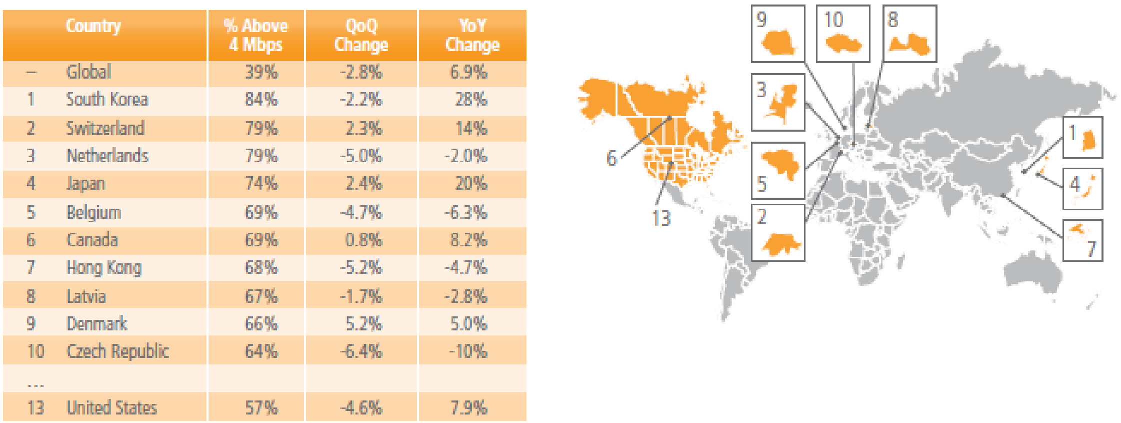
Besides this, the amount of data being transferred during the last years and the prevision for the future allocates a new scenario where non-centralized systems such as P2P are required as data bandwidth grows and systems need to become more scalable. Nowadays networks are still manly content-centric, meaning that data is provided from a source to a client in a triangle scheme, clients upload data to central servers and this data is transferred to the endpoint. This architecture has been provided since long time as reliable and scalable, but with the appearance of powerful applications and media transfer it has been proven to become a real problem.

Those circumstances lead to a whole new world of real-time browser based applications which require also a totally new framework to be developed into. Starting from the online videoconferencing to real-time data applications, for this purpose few attempts where made in the past being highly reliable on specific hardware and custom-built no-compatible system. Taking these decisions made those proposals not be accessible by the massive amount of normal users that could not afford to adapt the requirements.

All the previous concepts are now possible thanks to the increase of performance related to the hardware components available in every average computer nowadays, this increase has helped to build more complex browsers that are able to perform many tasks not just related to web browsing. Having a browser handling OpenGL style of applications is now possible thank to the increase of performance. Besides this multimedia abilities have also been able to reproduce on those browsers and handling webcam media as html is now a reality. Even dough there are still some issues to be considered before being able to freely communicate between two browsers: there is no common standardized protocol that allows developers to do this. WebRTC goal is to approach this problem to build a simple and standard solution for peer-to-peer browser communication [1].

Internet bandwidth has helped to take the decision to start integrating peer-to-peer solutions in browsed based applications, this is due the year-by-year increase of user bandwidth connectivity during the last 10 years. As before the latency was too high being unable to maintain real-time applications working resiliently. But recently the amount of users being able to transfer at high speed has been rapidly increasing as you can see in Figure 1, about 39% of users are now able to download at speeds greater than 4Mbps being this a very good average speed for media content [2].

Regarding the specs on the client side, recent surveys and statistics taken by the game manufacturer Steam [3] has proven that nowadays more than 61% of machines are carrying 1 to 4 gigabytes of RAM and nearly 90% of computers are handling 2 to



	Country	% Above 4 Mbps	QoQ Change	YoY Change
–	Global	39%	-2.8%	6.9%
1	South Korea	84%	-2.2%	28%
2	Switzerland	79%	2.3%	14%
3	Netherlands	79%	-5.0%	-2.0%
4	Japan	74%	2.4%	20%
5	Belgium	69%	-4.7%	-6.3%
6	Canada	69%	0.8%	8.2%
7	Hong Kong	68%	-5.2%	-4.7%
8	Latvia	67%	-1.7%	-2.8%
9	Denmark	66%	5.2%	5.0%
10	Czech Republic	64%	-6.4%	-10%
…				
13	United States	57%	-4.6%	7.9%

9
10
8
3
6
13
5
2
1
4
7

Figure 1: Broadband connectivity statistics about the speeds over 4Mbps around the globe.

4 core CPU with a 64 bit OS, being this environment optimistic for media enhanced applications which require high performance for video encoding and similar. Due to the layering needed to standardize a system like WebRTC running in top of an underlying application such as the browser that handles many processes is very important to rely on a powerful machine. Usually this was not able to be done in the past and one of the challenges has always been performance. They key to success is always to optimize the performance of a process so it does not affect the user experience, one of the challenges of WebRTC.

Traditionally, this concept of performance was approached by the usage of plug-ins or other separate software components which made the system run smoother by avoiding one layer of processing (browser) but being non-standard and not cross-compatible, one of the most import ant concepts when designing applications nowadays. Now the traditional approach has become outdated with the arrival of the new HTML5 where WebRTC is integrated as one of the new APIs available alongside other many different interesting approaches.

1.1 History

Web Real-Time Communication is an API definition with the aim to start an interoperability standard that will help to build P2P applications in the developer layer. The first announcement concept went public in a working group of the World Wide Web Consortium (W3C) in May 2011 [4] and starting the mailing list in April 2011 [5]. During the first stage of the working group the main goal was to define a public draft for the API implementation and a route timeline with the goal to standardize the protocol by ends of 2012. The first public draft of the W3C came public the 27th of October 2011 written by Adam Bergkvist (Ericsson), Daniel C. Burnett (Voxeo), Cullen Jennings (Cisco) and Anant Narayanan (Mozilla) [6]. During this first W3C draft only media (audio and video) where considered as candidates to be sent over the network to other peers, manly focusing in the way browsers will be able to access the media devices without using any plugin or external software.

Parallely to the W3C working group the WebRTC concept also joined the IETF with a working group in May 2011 [7] with the first public announcement charter done the 3th of May 2011 by Magnus Westerlund (Ericsson), Cullen Jennings (Cisco) and Ted Hardie (Ericsson). The milestones of the working group charter initially marked December of 2011 to provide the information and elements required to the W3C for the API design input. On the other side, the main goals of the working group covered the definition of the communication model, session management, security, NAT traversal solution, media formats, codec agreement and transport of the data [8]. Those goals have been evolving during the standardzation process and the work done along with the W3C working group.

An important point during the process of standardization came the 1st of June 2011 when Google publicly released the source code of their API implementation [9].

During all this period both working groups have been working alongside to provide a reliable solution to enable applications to perform media and data transfer in a plugin-free environment.

1.2 Support

The following companies have publicly supported and are actively working in the development of WebRTC standards in the W3C: Google, Mozilla and Opera [10]. Other companies such as Microsoft have been publicly supporting a browser-to-browser solution but have provided their own proposal which differs with the one published in the WebRTC group [11].

1.3 Milestones

2 Conclusion

The end.

References

[1] H. Alvestrand. Real time protocols for brower-based applications. Technical report, IETF, 2012. Also available in http://tools.ietf.org/html/draft-ietf-rtcweb-overview-04.

[2] Akamai. The state of the internet, 2nd quarter, 2012 report, 2012.

[3] Steam. Steam hardware and software survey, October 2012. Also available in http://store.steampowered.com/hwsurvey.

[4] Web real-time communications working group, May 2011. Also available in http://www.w3.org/2011/04/webrtc/.

[5] Harald Alvestrand. Welcome to the list!, April 2011. Also available in http://lists.w3.org/Archives/Public/public-webrtc/2011Apr/0001.html.

[6] Cullen Jennings Adam Bergkvist, Daniel C. Burnett and Anant Narayanan. Webrtc 1.0: Real-time communication between browsers. Technical report, W3C, October 2011. Also available in http://www.w3.org/TR/2011/WD-webrtc-20111027/.

[7] Real-time communication in web-browsers, May 2011. Also available in http://tools.ietf.org/wg/rtcweb/.

[8] Cullen Jennings Magnus Westerlund and Ted Hardie. Real-time communication in web-browsers charter. Technical report, IETF, May 2011. Also available in http://tools.ietf.org/wg/rtcweb/charters?item=charter-rtcweb-2011-05-03.txt.

[9] Google release of webrtc source code, June 2011. Also available in http://lists.w3.org/Archives/Public/public-webrtc/2011May/0022.html.

[10] Rian Liebenberg and Jan Linden. Introducing webrtc an open real-time communications project, April 2011. Also available in http://lists.w3.org/Archives/Public/public-webrtc/2011Apr/0001.html.

[11] Bernard Adoba and Martin Thomson. Customizable, ubiquitous real time communication over the web (cu-rtc-web). Technical report, Microsoft, August 2012. Also available in http://html5labs.interoperabilitybridges.com/cu-rtc-web/cu-rtc-web.htm.