TURN Servers Impacts Over WebRTC QoE in 4G Network

Antonin MARECHAL, Ewa JANCZUKOWICZ,
IMT/OLN
Orange Labs
Lannion, France
Email: {antonin.marechal, ewa.janczukowicz}@orange.com

Abstract— **WebRTC is a new technology allowing web players to offer communication services to their customers. It optimizes the communications by privileging peer-to-peer connections. However, in restrictive networks, media relays (TURN servers) are mandatory for establishing the call. The location and behavior of these media relays can have an undesirable influence on the quality of real-time communication. The goal of this study is to compare the impact on the QoE depending on the TURN server used. The comparison is focused on audio calls over 4G networks. The paper describes a test environment and suggests a measurement methodology based on the Mouth to Ear delay criteria. It gives the assessment of the first results focusing on TURN server location.**

- Keywords—WebRTC; 4G network; TURN servers; QoE

I. Introduction

Communication services and web real-time technologies have evolved. More and more real-time communication services are offered by web-related companies, notably Apple (FaceTime), Google (Hangouts) and Microsoft (Skype). The emerging WebRTC (Web Real Time Communication) [*1*] technology is currently under standardization and enforces the trivialization of web communication services. Therefore these services are challenging Telco solutions. Furthermore, network operators and web players do not have the same constraints and objectives. Actually, Telco and web ecosystems differ in various aspects such as service types, service distribution models, regulatory and contractual aspects [*2*]. Telco solutions are more reliable but are usually limited to a given territory and paying customers whereas web solutions are global, users do not pay for them directly. However, web applications are vulnerable to poor Internet quality [*3*].

The recent evolutions in real-time communication domain lead to the delamination of application and network layers [*2*]. Web actors have to adapt to existing best-effort network quality even if it is not sufficient, but network providers cannot improve the quality for these services since they are usually lacking visibility over this delaminated traffic.

Only few options allow network providers to participate in the quality improvements of WebRTC calls. Providing media relays to web actors is thus a relevant solution, since it allows flow identification that can be afterwards used to provide different services, including those related to QoS and QoE. This paper gives the first results of a comparison of the achieved quality using third-party and network operator provided media relays. The comparison is focused on the quality of WebRTC audio calls over 4G networks. The paper is structured as follows. Section 2 gives an overview of WebRTC technology along with current effort in improving communication quality. Section 3 introduces the approach taken during the study and the test environment. Section 4 focuses on results assessment. In Section 5 conclusion and future research works are discussed.

II. Motivations

A. *WebRTC technology overview*

WebRTC makes developing real-time applications easier by allowing browser to browser real-time communication (audio, video and data transfer) without the need of any plugins and by using web technologies [*4*]. It reduces implementation costs and allows avoiding interoperability issues.

In WebRTC the media plane is under standardization in IETF and W3C, but the signaling can be chosen freely by a communication service provider. There are different connection modes possible. ICE (Interactive Connectivity Establishment) is used to establish the connection by discovering different possible IP paths [*5*]. Ideally the media should be sent directly between devices. If any of the devices is behind a NAT (Network Address Translator), a STUN server is used to learn its public address [*6*]. However it is not always possible to avoid the intermediaries, like in case of certain implementations that block peer to peer traffic [*4*]. In this case a TURN server needs to be used [*7*]. TURN server is a media relay meaning that it forwards the traffic from one endpoint to another. Usually TURN server is placed in the media path throughout the communication, but it can be also used for a fast call set up, before switching to a standard peer to peer connection [*8*].

B. *WebRTC technology limits*

Since there is a separation of application and network layers, WebRTC uses application built-in adaptive mechanisms to improve the users' experience. As a result the adaptation to best-effort network quality maybe insufficient for instance in case of bottleneck congestion. Receive-Side Real-Time Congestion Control for WebRTC is an example of browser mechanisms that generally adapts well to losses and delays but still faces some problems in the presence of high latencies or certain concurrent traffic [*9*].

There are also ongoing works to improve the ICE protocol in order to offer dynamic selection of the network interface/path to use based on the RTT (Round Trip Time) and packet-loss measured during the communication [*10*]. However the collected information may be incomplete since it lacks network operator assistance. Continuous measurements on otherwise inactive interfaces can also negatively influence the battery usage.

As it was mentioned in section II.A, there are cases where a TURN server needs to be used. But this type of server may impact the communication, mostly because of two reasons. Firstly, the TURN server could be overloaded. For instance, its configuration could not handle WebRTC communications due to the usage of its CPU, its memory, the network interface, etc. Secondly, its location could also impact WebRTC calls. TURN servers are often provided by web companies, so they are situated in datacenters outside internet access provider networks. Thus, two communicating users situated in France, may end up using a TURN server placed in the USA. If we consider that the approximate delay is 5µs per kilometer, the flow between two users in France using a TURN server in Texas, USA (about 8000km one-way) takes much longer to be sent, i.e. about 80ms. That may have an undesirable influence on the real-time communication quality.

As the number of applications and smartphone users relying on web communications grow, the challenge of overcoming the limits of best-effort networks with the assistance of network providers becomes more important [*2*].

C. Network QoS for WebRTC

There are some ongoing works on providing QoS (Quality of Service) for WebRTC technology by using network resources.

The first solution is based on managed VoIP principles and uses TURN servers to differentiate the flows and provide specialized network services to them [*3*]. The second solution uses WebRTC session information to dynamically provide EPS (Evolved Packet System) QoS mechanisms [*11*]. Offering specialized network services is also in the scope of reThink European project [*12*].

III. APPROACH

WebRTC aims at supporting real-time communications. ITU recommends meeting some quality criteria concerning communication characteristics, i.e. delay, packet loss, etc. [*13*]. Among these criteria, we have chosen to analyse Mouth to Ear delay, because this value objectively describes the user's perception of the communication regardless of the technologies used (2G/3G, VoLTE, etc.).

The Mouth to Ear delay takes into account the propagation of the IP packets on the network, the sizes of the buffers and the amount of time to render the audio to the listener. Thus, this value describes the interactivity of a communication. If it is too high, it will negatively influence the interactivity and as a result the users will not make the difference between natural pauses in a conversation and delays introduced by some equipment [*14*]. ITU explains in a study [*13*] that users are very satisfied if the Mouth to Ear delay is lower than 200ms. Under 300ms, the communication is still correct. However, over 400ms, many users are dissatisfied. For instance, a national call in 2G, or 3G circuit switched, offers a Mouth to Ear delay around 190ms [*15*]. In VoLTE, this value is between 160 and 240ms [*16*].

Furthermore, datacenter traffic is bursty and may cause jitter or packet loss [*2*]. Routers and network equipment in the IP path impact the delay of the packets. Each packet, before being forwarded, has to spend a certain amount of time in their buffers. So if a WebRTC call needs to reach a TURN server hosted far from the users, it is likely that some buffers will retain packets and consequently impact the jitter, thus the jitter buffer and the Mouth to Ear delay. Moreover, network links between autonomous systems (AS) (Transit or Peering) can be seen as bottlenecks. Hence if a lot of WebRTC calls, typically with data and video, need to reach a TURN server through those links, it may also affect the cost for the network operators. To decrease the usage of links between AS, the WebRTC users could connect to a TURN hosted in the NSP (Network Service Provider) network. NSPs have assets that third-party operators do not have, i.e. the access and core network.

This study shows the impact of a TURN server located near the users (hosted in the core network of an ISP for instance) and compares it to the impact of TURN servers hosted in datacenters by third-party operators. It focuses on the audio calls. The video calls are out of scope of this paper but a similar study could be also interesting.

A. TURN Services providers

In this study, different actors who offer or rent TURN services have been identified:

- Third-party operator 1 (TPO-1): Situated all around the world, this actor provides TURN servers for any client who connects to their WebRTC website demo. In our case, each time a client tried to retrieve an ICE configuration (STUN and TURN), it indicated that the relay server was hosted in TPO-1's closest datacenter from the client.
- Third-party operator 2 (TPO-2): Based in the USA only, this company offers TURN services for any WebRTC service provider. The developer only has to create an account in the TPO-2's service. Then the WebRTC application has to retrieve the configuration of the ICE Servers, it will finally give all of those information to the client.
- The Network Service Provider (NSP): a TURN server has been deployed near the exit of the 4G mobile network, that is to say after the PDN-Gateway of the network service provider from the end-users' perspective. The service was handled by the Coturn server[1] on a Linux Ubuntu 14.04. This server was

[1] https://github.com/coturn/coturn

only dedicated to this function and was equipped with 6GB RAM, an Intel Xeon CPU (2.8 GHz, 4 cores) and a 1 Gb/s interface connected to the LTE experimental network.

We were not able to manage TPO-1 and TPO-2 servers. Therefore, the performances and the load (CPU usage, RAM, network interface, etc.) are unknown.

B. *Tools and test platform*

Two clients were used for this study. The first one was a Samsung Galaxy S4 (Android 4.4.2), and the second one was a Samsung Galaxy S5 (Android 5). Both were using Google Chrome (v43). A rendez-vous server used for WebRTC connection establishment has been developed and deployed on a classic web server on the Internet. It was used to initiate WebRTC audio calls and also for retrieving and storing the KPIs (Key Performance Indicators). Figure 1 shows this architecture.

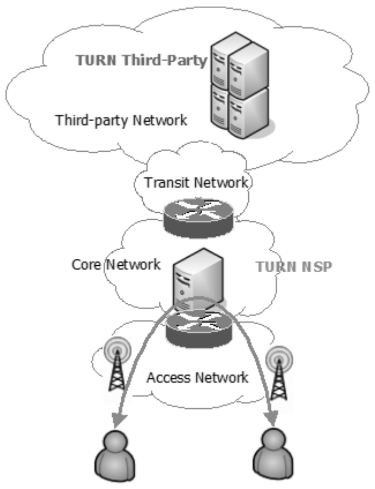

Figure 1: TURN server architectures (third-party and NSP)

The getStats() [*17*] function, available in Chrome, was used for collecting the statistics. Several KPIs were saved during the call, by using an internal JavaScript code calling the getStats function every second. The duration of each call was more or less around 2 minutes. All the calls took place in Lannion (France) over 4G LTE networks.

Two LTE networks have been used during the test:

- The first one is an experimental network which represents the architecture of a production LTE network. Devices were connected to the eNodeB inside a Faraday cage.
- The second one is the LTE network in production managed by an operator. The results obtained by this network show performances that customers could observe.

We assessed the retrieved data and analyzed it according to the ITU recommendations [*13*].

C. *Values analyzed*

Once stored and then combined, the statistics highlight the performances perceived by the user for each communication. For example, we can approximate the minimum value of the Mouth to Ear delay by doing the following sum

$$\mathit{CurrentDelayMs} + \mathit{RTT}/2 \approx \mathit{Min(Mouth\ to\ Ear\ Delay)} \quad (1)$$

The authors believe based on [*18*] that the CurrentDelayMs value retrieved by our tool describes the amount of time that a packet received by the user has to wait before being rendered. It includes the duration that a packet has to wait in the different buffers of the receiver.

In the case of this study, the Mouth to Ear delay is the minimum value that could be reached. Some other values (rendering, codec, etc.) that have not been retrieved during the calls should be added to calculate the effective delay. Currently the getStats function does not provide enough KPI. For instance, the approximation of the mouth to ear delay is not provided natively by the function. Moreover, Web applications do not have access to any information about source of an issue. For example, a web application cannot determine in case of congestion if the bottleneck is in the radio access network, in the core network or in the datacenter hosting the service. KPI of the quality of the radio performances could help to provide some information.

IV. Mesurement and interpretation

A. *Boxplots*

A convenient way to display the statistics of each call is to use a boxplot[2]. This kind of graphic shows the distribution of all the measurements stored during each call. The black line inside each box represents the median. Half of the points retrieved during a communication are in the box. Relevant values are mainly present inside the vertical lines. Finally, the outliners are displayed as points above and under those vertical lines.

The below boxplots show behaviors of the calls depending on the TURN server used.

Figure 2 shows the approximation of the Mouth to Ear delay between callee's mouth and caller's ear. The X-axis indicates the ID of each call. The Y-axis shows the Mouth to Ear delay by using boxplots. Calls from 1 to 7 have been performed in P2P. Calls from 8 to 22 have been relayed by a TURN server hosted at the exit of the PDN-GW. All the measurements are collected on Samsung S4.

[2] http://web.pdx.edu/~stipakb/download/PA551/boxplot.html

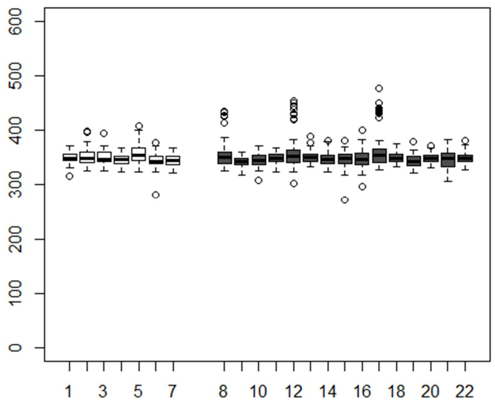

Figure 2: Mouth to Ear delay [ms] in WebRTC calls (P2P and through NSPO) measured on Samsung S4

Figure 3 shows the same measurements as the above figure but performed on Samsung S5.

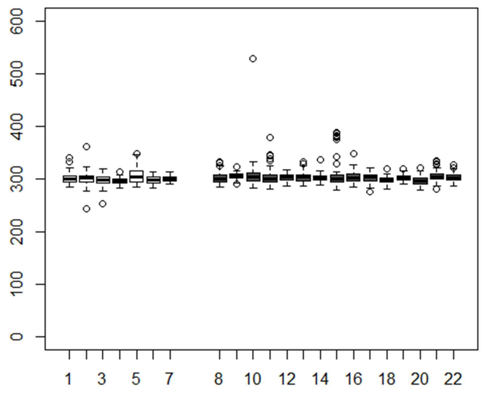

Figure 3: Mouth to Ear delay [ms] in WebRTC calls (P2P and through NSPO) measured on Samsung S5

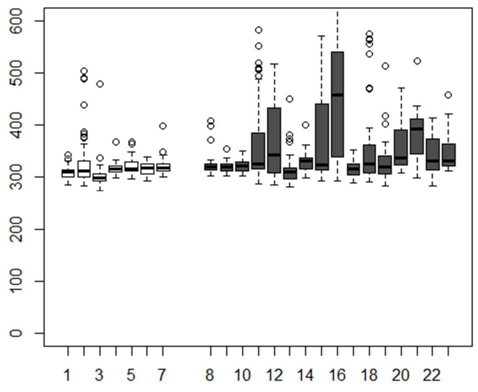

Figure 4: Mouth to Ear delay [ms] in WebRTC calls (P2P and through TPO-1) measured on Samsung S5

Figure 4 also shows the approximation of the Mouth to Ear delay between caller's mouth and callee's ear. As before calls from 1 to 7 have been performed in P2P, but this time calls from 8 to 23 have been relayed by a TURN server hosted in the datacenter of TPO-1.

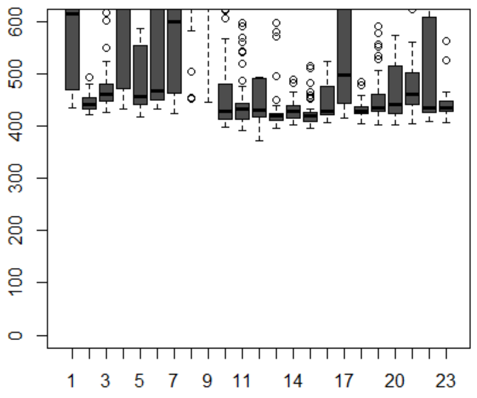

Figure 5: Mouth to Ear [ms] delay in WebRTC calls (TPO-2) measured Samsung S5

Figure 5 shows the approximation of the Mouth to Ear delay between caller's mouth and callee's ear. Here all the calls have been relayed by a TURN server hosted in the datacenter of TPO-2.

B. *Interpretation*

In the first place the study highlights the lowest Mouth to Ear delay for WebRTC calls over the LTE network. The Mouth to Ear delays were never lower than 300ms. This value was compared with Telco technologies (VoLTE [*16*], 2G/3G circuit switched network [*15*]). Therefore it can be stated that WebRTC audio calls on 4G network do not offer performances as good as calls with VoLTE. The Mouth to Ear delay is at least 300ms with WebRTC, but in VoLTE this value is more or less 200ms [*16*]. This behavior can be caused by type of network used since WebRTC calls have been made over Best Effort networks without any network QoS. Additionally, we could observe that in order to face the variation in delays, the user equipment had to increase the jitter buffer, thus store a certain number of packets.

WebRTC audio calls that went through TURN servers hosted near the exit of the PDN Gateway perform as good as calls in P2P. As it can be seen on Figures 2 & 3, the Mouth to Ear delays are in the same order of magnitude. The figures also show that different measurements do not vary, thus the quality stays stable.

Unlike Figure 2 & 3, the measurements on Figure 4 & 5 are more scattered. When WebRTC audio calls go through TURN servers hosted in a datacenter on the same continent for calls 8, 9, 10, 13, 14, 17, 18 and 19 of figure 4, the performances are surprisingly as good as P2P calls, meaning that the distance of a TURN server does not necessarily make the quality worse. However in certain cases the Mouth to Ear delay was less stable and higher delays could be observed (calls 11, 12, 15, 16, 20 to 24). This can be explained by the fact that packets go through more routers and network equipment. As a result the risk of congestion, burst, etc. which could impact the communication, increases.

We can see on figure 5 that the performances are the worst when the TURN is not on the same continent, because the

Mouth to Ear delay is at least 450ms with TPO-2. This is linked to the fact that the packets had to travel the longest way.

Moreover, TURN servers are not the only source of high Mouth to Ear delay. The study also shows that the user equipment could have a significant impact on this delay. Indeed we noticed that Samsung Galaxy S4 (Android 4.4.2) is up to 50ms slower to render the voice than the Galaxy S5 (Android 5). We cannot determine precisely where this behavior comes from, but we have two suggestions:

- Hardware: The Galaxy S5 is equipped with a faster CPU (Qualcomm Snapdragon 801 – 2.5GHz Quadcore) than the S4 one (Qualcomm Snapdragon 600 – 1890MHz Quadcore)
- Software (OS): Enhancements on the audio management have been offer with the 5.0 version of Android[3]

V. Conclusion and Future Work

Firstly, this study proves that calls using a TURN server located close to the original path of media perform as well as communications performed in P2P. Furthermore, calls using a TURN server hosted further in the network in certain cases perform as well as P2P calls. This could be observed in our study for half of the calls. However, the other half of communications did not offer good performances. Finally, unsurprisingly, a TURN server far from the client does not perform well at all.

Secondly, it could be observed that current calls, even in P2P, using Chrome implementation of WebRTC do not offer performances as good as VoLTE or 2G/3G switched circuit calls over 4G networks.

To generalize the outcome and to obtain more significant results, another study should be done with more calls (taking place all around the country, at different times throughout the day, etc.). Different browsers could also be tested as it is known that performances differ depending on the browser used [*19*]. This new study should emphasize the fact that TURN servers located far from the user impact WebRTC communications due to the bigger number of routers and other network equipment in the path.

Thirdly, the study discusses that Mouth to Ear delay is interesting to observe since it objectively describes the user's perception of the communication regardless the underneath technology. However Chrome's current implementation of the getStats function, normalized by the W3C, does not provide directly information about the Mouth to Ear delay. Thus, it could be an interesting enhancement to allow getStats function to retrieve this KPI. Moreover, the radio quality information (SNR, retransmission ratio, etc.) could also be interesting for web applications. This would allow getting the whole picture, i.e. how different factors impact the quality. However, some discussions have to be done before offering these functionalities, for example user's privacy has to be taken into account before proposing any values to web developers.

[3]https://developer.android.com/about/versions/lollipop.html#Audio

Finally, the study shows that when using resources provided by a network service provider, good performances can be obtained. As a result, a specialized network services could improve the quality of WebRTC communications. Hence further studies will be done in this field.

Acknowledgement

The authors would like to thank Stéphane Tuffin, Arnaud Braud, Gaël Fromentoux and Jean-Yves Le Saout for their help and their feedback.

References

[1] WebRTC. [Online]. http://www.webrtc.org/

[2] S. Becot, E. Bertin, J.M. Crom, V. Frey, and S. Tuffin, "Communication Services in the Web Era. How can Telco join the OTT hangout?," *ICIN Proceedings*, pp. 208-215, February 2015.

[3] E. Janczukowicz et al., "Specialized network services for WebRTC: TURN-based architecture proposal," *AWeS*, April 2015.

[4] E. Janczukowicz et al., "Approaches for Offering QoS and Specialized Traffic Treatment for WebRTC," *Advances in Communication Networking*, pp. 59-69, September 2014.

[5] J. Rosenberg. (2010, April) Interactive Connectivity Establishment (ICE). [Online]. https://tools.ietf.org/html/rfc5245

[6] J. Rosenberg, R. Mahy, P. Matthews, and D. Wing. (2008, October) Session Traversal Utilities for NAT (STUN). [Online]. https://tools.ietf.org/html/rfc5389

[7] R. Mahy, P. Matthews, and J. Rosenberg. (2010, April) Traversal Using Relays around NAT (TURN). [Online]. https://tools.ietf.org/html/rfc5766

[8] P. Hancke. (2015, April) webrtcH4cKS: ~ What's up with WhatsApp and WebRTC? [Online]. http://techupdates.com/go/1140476

[9] V. Singh, J. Ott, and A.A. Lozano, "Performance Analysis of Receive-Side Real-Time Congestion Control for WebRTC," *Packet Video Workshop (PV), 2013 20th International*, pp. 1-8, Decembre 2013.

[10] J. Uberti and J. Lennox. (2015, March) Improvements to ICE Candidate Nomination. [Online]. https://tools.ietf.org/html/draft-uberti-mmusic-nombis-00

[11] K. Haensge and M. Maruschke, "QoS-based WebRTC Access to an EPS Network Infrastructure," *Intelligence in Next Generation Networks (ICIN)*, pp. 9-15, February 2015.

[12] reThink. [Online]. https://rethink-project.eu/

[13] ITU-T, Series G: Transmission systems and media, digital systems and networks, May 2003, One-way transmission time G.114 E 24058.

[14] John Evans and Clarence Filsfils, *Deploying IP and MPLS QoS for multiservice networks*. San Francisco, USA: Morgan Kaufmann Publishers, 2007.

[15] 3GPP. (2014) TR 26.975 Specification detail. [Online]. http://www.etsi.org/deliver/etsi_tr/126900_126999/126975/12.00.00_60/tr_126975v120000p.pdf

[16] Michael Anehill, Magnus Larsson, Göran Strömberg, and Eric Parsons, "Validating voice over LTE end-to-end," *Ericsson Review*, pp. 1-10, February 2012.

[17] W3C. (2015) Identifiers for WebRTC's Statistics API. [Online]. http://www.w3.org/TR/webrtc-stats/

[18] Chromium Browser sources. (2015, June) appspot.com. [Online]. https://webrtc-codereview.appspot.com/51149004

[19] Arto Heikkinen, Timo Koskela, and Mika Ylianttila, "Performance evaluation of distributed data delivery on mobile devices using WebRTC," *Wireless Communications and Mobile Computing Conference (IWCMC)*, 2015.