Towards the Application of WebRTC Peer-to-Peer to Scale Live Video Streaming over the Internet

Flávio Ribeiro Nogueira Barbosa[1], Luiz Fernando Gomes Soares[2]

[1]WebMedia – Globo.com
Rio de Janeiro – RJ – Brazil

[2]Department of Informatics – PUC-Rio
Rio de Janeiro – RJ – Brazil

flavio.ribeiro@corp.globo.com, lfgs@inf.puc-rio.br

Abstract. *Given the growth on the number of Internet users and the quality of their connections, building large Internet live broadcasts has become increasingly challenging. This paper introduces the use of WebRTC peer-to-peer technology to analyze a hybrid CDN-P2P structure in order to decrease the number of requests to CDN servers, reducing the cost of transmission and enhancing system's scalability.*

1. Introduction

Following audience growth on the Internet and the users' inherent preference for multimedia over text-based content consumption, it is widely believed that video distribution will dominate the traffic over the Internet. Despite the audience records and the desire of large broadcast companies to increase the amount of online broadcasts, challenges related to scalability, economic costs and the quality of consumer's experience of these transmissions are still an open issue.

Assuming that the absolute majority of the audience watches online videos using web browsers, this paper makes use of WebRTC peer-to-peer technology to propose a simple hybrid P2P-CDN model that tries to relieve the amount of requests to video chunks on content servers, reducing transmission cost and taking advantage of peer-to-peer distributed computing to improve user experience and system scalability.

2. Motivation

The last big events broadcasted worldwide have received a huge number of online viewers and this phenomenon is expected to continue in the coming years. As examples, Red Bull Stratos 2012 was broadcasted by Google's YouTube Live Streaming platform and attracted more than 8 million concurrent users [Katz 2012]. FIFA's Confederations Cup 2013 was broadcasted only to Brazilian residents by Globo.com and reached almost half a million users at its peak.

Not only audience is growing but also the quality of videos produced. As users get better connections, they become more demanding, and content producers are trying to meet their expectations. Sochi 2014 Winter Olympics was streamed in Brazil in High Definition (at 720p), with 3.5Mbps as its maximum bitrate. Netflix is expecting to be delivering 4K videos within a year or two [Sandoval 2013]. Analyzing the prediction of transmission quality of FIFA's next World Cup and comparing it with the last two World Cup online broadcasts, we can observe an exponential growth in the quality of videos delivered (Figure 1).

The high desire to consume high-quality video brings several technical challenges. Usually, big companies that broadcast videos to hundreds of thousands, or even millions of people, use Content Delivery Networks (CDN's) in order to meet all the audience's demand.

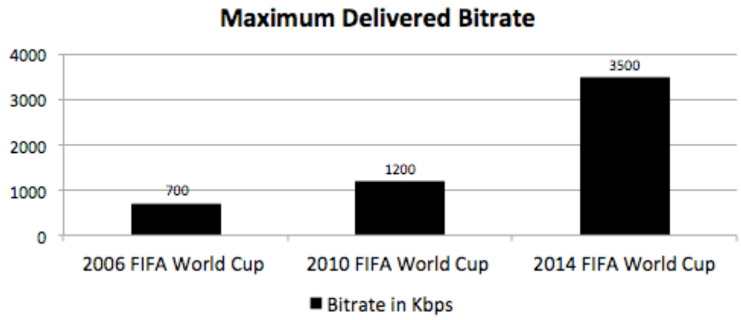

Figure 1. Maximum Delivered Video Bitrate by Globo.com at FIFA World Cup

The main purpose of a CDN is to distribute contents over a set of agglomerated web servers highly distributed around the world (also known as Points of Presence or just PoP's), so as to guarantee a reliable, scalable and efficient delivery of the contents to end users [Bronzino *et al* 2012]. However, this approach has some downsides:

*Scalability***.** CDN's serve users through PoP's. When someone is far from the PoP, he depends on some communication links to reach the requested content. This way, CDN's scale adding PoP's wherever they can. In countries with poor telecommunications and slow Internet exchange points (IXP), the necessity for more PoP's is greater and this problem is further aggravated.

Cost. CDN providers are, in fact, expensive. According to [Spangler 2009], Google YouTube would spend 1 million dollars per day on bandwidth accounts in 2009.

Quality of Experience. It was noticed that last large online transmissions based entirely on CDN's had several network issues, and audience's experience was severely degraded [Zimmerman 2014] [Nordyke 2014]. CDN-based approaches have shown that the crucial dependency of certain PoP's makes the system relatively fragile, and the overload of one PoP can lead to a domino effect.

3. Background

Before explaining the proposed model and our implementation, an introduction regarding the techniques and protocols used in this study is required.

3.1 Current Video Distribution Techniques

Since the beginning of multimedia online broadcasts, stateful protocols such as Real-Time Transport Protocol (RTP) and Real-Time Messaging Protocol (RTMP) were preferred instead of stateless ones. These protocols have the capability of maintaining one connection between server and client while sending streams of video, audio and data packages [Parmar and Thornburgh 2012].

The mentioned protocols require specialized media servers to generate the streaming and handle users' connection. They are mostly implemented on top of User Datagram Protocol (UDP) [Yuste and Melvin 2012], and several network providers, and even some firewalls, block or penalize UDP traffic.

These limitations gave space to the creation of some stateless HTTP-based streaming protocols on top of TCP, like Microsoft Smooth Streaming, Adobe HTTP Dynamic Streaming (HDS), Apple HTTP Live Streaming (HLS) and MPEG DASH. HTTP is the foundation of data communication for the World Wide Web (WWW), which eliminates the issues related to traffic penalization. All the content is served by ordinary web servers like Apache HTTP Server or Nginx and favors the use of CDN's. The downside of HTTP-based protocols is the insertion of delay, since the act of slice the video and playlist creation or update can take a few seconds.

This work focuses on HTTP Live Streaming that was developed by Apple and documented as an Internet Draft, the first stage in the process of submitting it to the IETF as an Informational Request for Comments (RFC).

3.2 WebRTC

Since October 2011, The World Wide Web Consortium (W3C) is developing a Working Draft to add Real-Time Communications (WebRTC) capabilities between web browsers [Berkvist *et al* 2011]. These capabilities include the direct sharing of video, audio and data. Although being a draft, WebRTC capabilities are already integrated in 2 of the 3 most used browsers in Brazil (Google Chrome and Mozilla Firefox) and at the time this document is being written, almost 73% of the users who visit Globo.com already have WebRTC in their browsers. Before WebRTC Working Draft, direct communication between browsers was possible only with third-party plugin software and significant proprietary server infrastructure [Naylor 2013].

3.3 Peer-to-Peer

Peer-to-Peer (P2P) is a distributed network architecture in which nodes share a part of their resources to contribute to the service and content offered by the network. With the advent of WebRTC on browsers, P2P is presented as a promising technique for the scalability problem in video streaming over the Internet.

4. Proposed Solution

Using the browsers' ability to connect to others through WebRTC, we propose a hybrid peer-to-peer network to assist video chunks delivery. The proposal is called hybrid due to the fact that one node can exchange messages and request chunks to other peers or request it directly to the CDN. CDN-P2P models are widely studied [Huang *et al* 2008] [Bronzino *et al* 2012] and the application of these models in our scenario can provide many benefits.

An inherent characteristic of P2P services is that network performance does not deteriorate (and usually improve) as network size increases and when resource relevance is high, cooperation in a P2P solution evolves naturally [Roussopolous *et al* 2004]. This way, huge broadcasts can take advantage of peer's upload bandwidth to exchange video chunks. Peer-assisted data sharing can decrease the upload bandwidth of content servers up to about 96% [Cho *et al* 2010], decreasing the cost of transmission and reducing bottlenecks, resulting on the improvement of the audience's experience.

The implementation of this approach was released under Apache 2.0 License and can be obtained in http://github.com/flavioribeiro/bemtv.

4.1 Peer entrance and Signaling

In our work, we use ISP-location and Geolocation Awareness concepts [Kovacevic 2009] to build clusters of nodes that can exchange messages between then. These clusters are called peer swarms. When a node wants to watch a live streaming, it first reaches a URL that will estimate peer's location and Internet provider, returning a swarm name for the node to connect to their common.

The act of connecting to their common is basically to announce the node and swarm name, captured on the earlier step, to a central signaling server, which will propagate the announcement to all nodes in the swarm. Every node that receives the announce will try to connect directly to the node throughout STUN servers [Rosenberg *et al* 2008] and, if it succeeds, both earn a directly peer-to-peer channel communication (WebRTC Data Channel). Nodes store all communication channels in a hash table to use for chunk negotiation on next steps.

4.1 Video Chunks Exchange Protocol

In this section we describe a protocol used to exchange chunks between peers on the same swarm.

As in every HTTP-Based Streaming Protocol, when the user presses play, the video player requests a playlist from the CDN. On HLS, when a playlist doesn't have the EXT-X-ENDLIST tag at playlist's end, it means that it is a live streaming and the player will need to hit the same playlist periodically looking for new video chunks.

After the player has received and parsed the playlist, it starts to request video chunks. Instead of the common HTTP request for the chunk from the CDN, the node sends a DESIRE to every node of the peer swarm and each node that receives the DESIRE searches for the chunk in its cache. Our implementation stores the last 10 chunks watched, and if the chunk requested is cached, the node sends back a DESACK. The desiring peer looks for the best node from the pool of nodes that sent DESACK and sends a REQ to the chosen one. Every peer that receives a REQ is ensured by the previous step that the chunk is in its cache and then it sends an OFFER with the chunk to the desiring peer.

When the desiring peer sends the DESIRE to swarm, it waits for a timeout of 0.7 seconds and if nobody answers, it requests directly to the CDN using the traditional server-client HTTP schema. The same occurs if the node chosen to send the chunk takes more than 1 second to send the file itself. On Figure 2, Peer A presents the desiring peer.

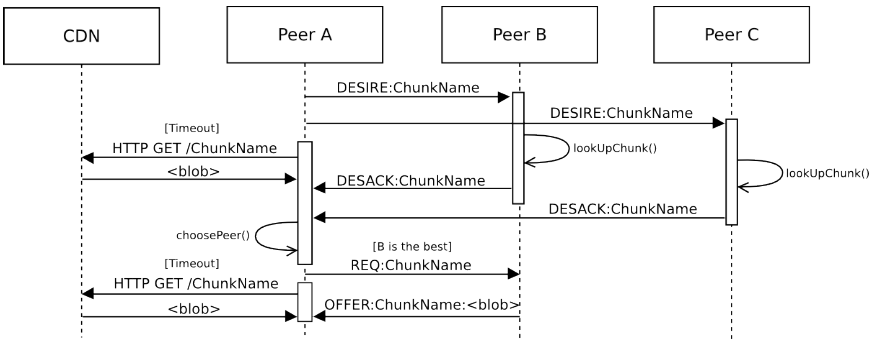

Figure 2. Chunks Exchange Protocol

4.2 Early Experiments

We submitted a live stream to the model proposed in this paper looking for the reduction in the number of requests to the CDN.

We used a total of ten Apple MacBook White with 2 CPU cores and 2GB SDRAM running a Mozilla Firefox 27.1 browser, fully compatible with WebRTC's last draft specification, and Apple Safari 6.0.5 Browser which is capable of playing HLS streams natively. All computers were in the same wireless 10/100 Mbps hotspot, which means that they were in the same geolocation and Internet provider and, consequently, peers in the same swarm. The streaming was split in chunks with 5 seconds of duration and 600Kbps of bitrate quality using HTTP Live Streaming Protocol. The CDN was represented by one server with 1 logical core and 512MB of SDRAM. Swarm name discoverer and signaling server were also running on this instance.

All computers were subjected to one hour of streaming consumption using the native Apple Safari player and then compared with one hour of streaming using the player implementation described on this paper.

4.2.1 Results

The results have shown that the native player made 7457 requests to the CDN while our player made 4482. Since none of the players had playback issues, it suggests that 2975 chunks were exchanged throughout P2P, meaning a reduction of 39.89% on the total of direct requests to the CDN.

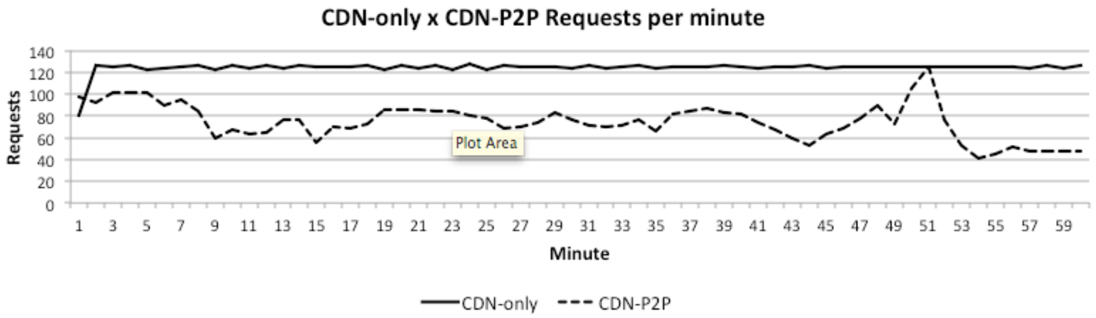

Figure 3. Number of Requests using CDN-only and CDN-P2P

5. Conclusion

Based on the experiment described and aware that web browsers are evolving, we can assume that the application of WebRTC in a CDN-P2P architecture to support HTTP-based live streams are quite promising. We believe that improvements on the data exchange protocol can increase much more the percentage of chunks trade around peers, reducing the cost of transmissions, increasing the scalability of the system and enabling a better experience for the consumer.

5.1 Future Work

A lot of challenges still remain. The proposed protocol demonstrated to be quite unstable, as shown by the fluctuation of requests per minute in Figure 3. This section describes some improvements that need to be applied to the protocol in order to be considered a truly robust and production-ready system.

*Peers Convergence and Over swarming***.** The approach used on this paper has shown effective for a small amount of peers. However, build swarms to support all users on a given Geolocation and ISP can accumulate hundreds or even thousands of users leading to an excessive exchange of DESIRE and DESACK messages. The use of reputation [Xiong and Liu 2004], partnership [Li *et al* 2008] or leader election [Kutten 2013] should be investigated.

*Video Chunks Exchange Protocol***.** With the improvement on swarm formations, we believe that the proposed protocol's stability should be improved and the number of chunks transferred over the P2P overlay should increase. However, researches regarding the ideal chunk size, video bitrate and node's cache size are required. The protocol must also ensure that chunks negotiation and transfer cannot interfere in the user experience. All these scenarios must be explored.

Content Security. The protocol described in this paper does not guarantee that the content exchanged between peers is indeed the same as in the CDN. Algorithms that detect poisoned chunks and DoS starvation such [Medina-López *et al* 2013] needs to be applied.

References

Berkvist, A., Burnett, D., Jennings, C. Narayanan, A. (2011) "WebRTC 1.0: Real-Time Communication Between Browsers". Working Draft.

Bronzino, F. Gaeta, R. Grangetto, M. Pau, G. (2012) "An Adaptive Hybrid CDN/P2P Solution for Content Delivery Networks". VCIP, page 1-6, IEEE.

Cho, S., Cho, J., Shin, S. (2010) "Playback Latency Reduction for Internet Live Video Services in CDN-P2P Hybrid Architecture". 2013 IEEE International Conference on Communications.

Huang, C., Wang, A., Li, J., Ross, K. (2008) "Understanding hybrid CDN-P2P: why limelight needs its own Red Swoosh". Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video.

Katz, T. (2012) "Mission Complete: Red Bull Stratos lands safely back on Earth". http://goo.gl/bX31hl, accessed in March 2014.

Kovacevic, A., Graffi, K., Pussep, K., Steinmetz, R. (2009) "Underlay awareness in P2P systems: Techniques and challenges". IEEE Symposium on Parallel & Distributed Processing.

Kutten, S., Panduragan, G., Peleg, D., Robinson, P., Trehan, A. (2013) "On the complexity of universal leader election". Proceedings of the ACM symposium on Principles of distributed computing.

Li, B., Xie, S., Qu, Y., Keung, G.Y. (2008) "Inside the New CoolStreaming: Principles, Measurements and Performance Implications". IEEE 27th Conference on Computer Communications.

Medina-López, C., Naranjo, J.A.M., García-Ortiz, J. P., Casado, L. G., González-Ruiz, V. (2013) "Execution of the P2PSP protocol in parallel environments". XXIV Jornadas de Paralelismo. Madrid, Spain.

Naylor, A. (2013) "WebRTC is almost here, and it will change the web". http://goo.gl/IgxF33, accessed in March 2014.

Nordyke, K, (2014) "HBO Go Crashes During 'True Detective' Finale". http://goo.gl/zixUr7, accessed in March 2014.

Parmar, H. and Thornburg, M. (2012) "Adobe's Real Time Messaging Protocol". http://goo.gl/cGMWPI, accessed in March 2014.

Rosenberg. J, Mahy, R., Matthews, P. Wing, D. (2008) "Session Traversal Utilities for NAT (STUN)". IEFT Proposed Standard.

Roussopolous, M., Baker, M., Rosenthal D. S. H., Giuli T. J., Maniatis, P., Mogul, P. (2004) "2 P2P or Not 2 P2P?". Third International Workshop, IPTPS 2004.

Sandoval, G. (2013) ". Netflix Chief Product Officer: expect 4k Streaming within a year or two". http://goo.gl/F4S133, accessed in March 2014.

Spangler, T. (2009) "YouTube May Lose $470 Million in 2009: Analysts". http://goo.gl/oNgAzY, accessed in March 2014.

Xiong, L. and Liu, L. (2004) "PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities". IEEE Transactions on Knowledge and Data Engineering.

Yuste, L. and Melvin, H. (2012) "A Protocol Review for IPTV and WebTV Multimedia Delivery Systems". Scientific Letters of the University of Zilina, vol 14.

Zimmerman, A., (2014) "ABC Promised to Livestream the Oscars and Totally Failed". http://goo.gl/sTNb7d, accessed in March 2014.