



**HAL**  
open science

# Increasing End-User's QoE with a Hybrid P2P/Multi-Server streaming solution based on dash.js and webRTC

Joachim Bruneau-Queyreix, Mathias Lacaud, Daniel Négru

► **To cite this version:**

Joachim Bruneau-Queyreix, Mathias Lacaud, Daniel Négru. Increasing End-User's QoE with a Hybrid P2P/Multi-Server streaming solution based on dash.js and webRTC. 2017. hal-01585219

**HAL Id: hal-01585219**

**<https://hal.science/hal-01585219>**

Preprint submitted on 11 Sep 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Increasing End-User's QoE with a Hybrid P2P/Multi-Server streaming solution based on dash.js and webRTC

Joachim Bruneau-Queyreix  
University of Bordeaux, LaBRI-Lab  
Viotech Communications, Versailles  
France  
joachim.bruneau-queyreix@labri.fr,  
jbruneauqueyreix@viotech.net

Mathias Lacaud  
Viotech Communications, Versailles  
France  
mlacaud@viotech.net

Daniel Negru  
University of Bordeaux, LaBRI-Lab  
France  
daniel.negru@labri.fr

## ABSTRACT

HTTP Adaptive Streaming (HAS) protocols have become the de-facto solutions to deliver video over the Internet, mainly due to their ability to limit video freezing and thus enhance consumers' Quality of Experience (QoE). Nevertheless, they do not have the possibility to improve the actual delivered video quality, limited by the available throughput between the delivering server and the client. Compared with this single-server approach, multiple-server streaming and P2P streaming offers the opportunity to obtain enhanced QoE by benefiting from expanded bandwidth, link diversity and reliability in distributed streaming infrastructures. We present a prototype for a hybrid P2P/multi-server quality-adaptive evolving HAS-compliant streaming solution, simultaneously using several servers and peers, and trading off the server infrastructure capacities and QoE gains.

## 1 INTRODUCTION

With the advent of technology characterized by the increasing offers of video services, end-users' Quality of Experience (QoE) has become a crucial factor for their success. According to Cisco [6], video traffic represents 75% of today's Internet traffic and will face a near two-fold increase by 2020. Such increases are usually not followed by the necessary upgrade of core networks capacity, due to the important costs it incurs. As a consequence, QoE along with scalability costs are to be considered as major issues in the design of both current and future content delivery solutions.

Content Delivery Networks (CDNs) architectures are extensively used for the large-scale delivery of live content over the Internet. Most of the time, HTTP Adaptive Streaming (HAS) techniques -such as the widely adopted DASH standard- are employed to reach the highest possible end-users' QoE by adjusting the desired content bitrate to the observed network conditions. Consequently, HAS protocols tend to avoid the video the freezing events that are mainly caused by the lack of throughput at the client or at the server side. Although CDNs solutions can handle large request volumes, they are still constrained by the scale of their infrastructures. In view of the drastic video traffic growth forecasts, the deployment and maintenance costs paid by content delivery service providers will rise and eventually make these services highly pricey for content providers and consumers seeking high QoE. As an alternative to CDNs, live P2P systems solve the scalability cost issue by leveraging on consuming clients' resources. However, they also come with limitations such as peer churn, heterogeneous capabilities, and lack

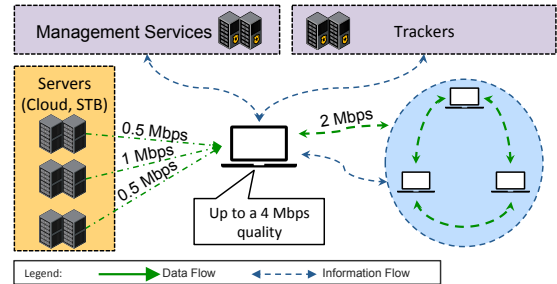


Figure 1: High level solution overview

of control that can rapidly result in video re-buffering and degraded QoE due to the tight latency constraints of live streaming.

In order to address the issues of high scalability costs and the potentially low QoE inherent to unreliable live P2P systems or due to the lack of throughput at server side, we exploit the quality adaptation advantages of DASH into a hybrid P2P/multi-server live streaming architecture. Deriving from the work [1-5], we demonstrate our solution for a cost- and quality- adaptive hybrid P2P/multi-server live-streaming trading off between the system's scalability and the end-users' QoE. We consider the use-case of a video streaming provider delivering a single live video flow in  $Q_{max}$  video qualities to a population of  $N(t)$  clients. In this system, clients can simultaneously use multiple servers with heterogeneous upload capacities so as to obtain higher QoE by exploiting expanded bandwidth and link diversity in distributed streaming infrastructures. Additionally, clients leverage on the upload capacities of other consuming clients to engage in P2P data transfer in order to achieve higher scalability and QoE. Each client adjusts the requested video quality with regards to P2P and client-server communication channels' conditions. Finally, clients locally optimize their utilization of the server infrastructure's limited upload throughput capacities in order to increase the system's scalability without sacrificing on QoE.

## 2 A HYBRID P2P/MULTI-SERVER LIVE-STREAMING SOLUTION

As shown in Fig.1, our system is composed of three major components: (1) the streaming portals including Management Services in charge of delivering MPD files to the peers. Additionally, trackers are responsible for serving peers with lists of  $K$  candidate neighbors, periodically computing global indicators on the current health of the P2P overlay based on metrics reported by the peers, and forwarding the latter indicators used in the adaptation logic; (2) Servers (cloud servers or set-top-boxes) that are dynamically provisioned with the

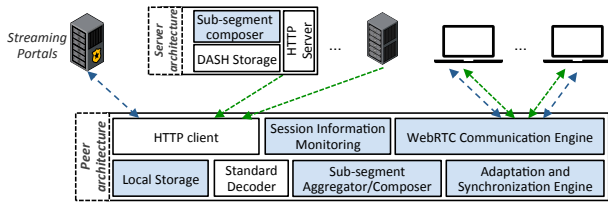


Figure 2: Functional software architectures

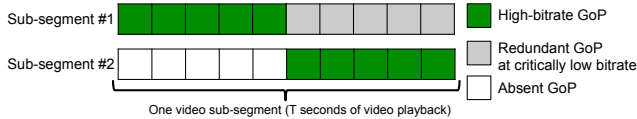


Figure 3: Example of the composition of two sub-segments for one video segment

live video flows at multiple video bitrates. (3) An application-layer mesh-pull overlay composed of  $N(t)$  peers consuming the content, exchanging their buffer maps and their estimated upload rates. In addition to the details of our system architecture, we also provide the functional software architectures of servers and peers, shown in Fig.2 (additional modules compared to a DASH architecture are highlighted in plain blue). We adopt the HTTP protocol in both the signaling and the video streaming from servers. The P2P communications are performed over the WebRTC technology that enables the contribution of peers behind NAT.

For a every new video segment, a client simultaneously requests several servers and remote peers to deliver independent codec-compliant playable and aggregatable sub-segments generated by the *sub-segment composer* from the existing *DASH storage*. The *DASH storage* contains several bitrate representations of the same content, split in several segments of a few seconds. Sub-segments are generated by interleaving the Group Of Pictures (i.e. sequence of correlated frames -GoPs-) available at two different bitrate representations of the same segment: one critically low bitrate referred as redundant bitrate, and one target bitrate. The redundant bitrate is set to such low values (e.g., 150 Kbps for a 720p spatial resolution in H.264) in order to provide video playback at the lowest possible network transfer cost. During the sub-segment requests specification step, the client specifies the composition of each sub-segments in terms of number of GoPs at the target and at the redundant bitrate. Sub-segments can also be composed of GoPs at a null bitrate, as shown in Fig.3. The latter composition, as well as the number of sub-segments for a given segment (i.e., the number of used servers/peers), are evolving parameters during the streaming session, determined by the client according to the heterogeneous characteristics of the considered paths, servers and peers, and to the targeted bitrate. This sub-segment generation scheme has features of interests, easing its adoption by streaming actors: video-codec standard compatibility, tunable redundancy (i.e., the redundant critically low bitrate GoPs copied into sub-segments), low additional complexity, and high flexibility in the number of possible sub-segments for a given sub-segment that translates into flexibility in the number of used servers/peers.

When retrieved, the requested sub-segments are used to reconstruct the original requested content quality (in the *Sub-segment Aggregator*). This operation is done by selecting the GoPs of higher size from the set of available sub-segments. In order be resilient

to network heterogeneity and to ensure uninterrupted video experience, we provide a sub-segment *synchronization engine* at client side that cancels late requests including redundant GoPs. Hence, in the event of sub-segment loss or late delivery, the content is still playable (content playback continuity is not affected) with sub-optimal visual quality, due to loss of information at the GoP level (inducing lower quality on a GoP-duration basis). This synchronization mechanism can also handover the delivery of late or missing GoPs to other peers/servers in order to benefit from available resources and provide better QoE.

Each peer implements a quality adaptation algorithm in the *Adaptation Engine* that controls the transitions from one quality to another. By relying upon local and global indicators respectively reflecting the local streaming session’s status and the status of the delivery of each quality, the adaptation logic strives to enhance the end-users’ QoE while preserving the good functioning of the delivery of each quality. Finally, for every video segment  $n$  the client decides on the percentage  $X_n$  of GoPs composing the segment that should be delivered by the servers. The value of  $X_n$  is adjusted for every new segment in order to locally minimize the server-infrastructure utilization to provide better scalability without ceding on the QoE gains.

### 3 RESULTS

Our hybrid P2P/multi-server live streaming protocol has been implemented in a pragmatic vision conforming to the H.264 and DASH [9] standards. The video player was implemented on top of the dash.js player [8] from the DASH Industry Forum [7]. We conducted evaluations over the Internet at large scale for more than 400 hours of video streaming. 50 consumers spread around Europe participated, equipped with various devices (smartphones, tablets, laptops, desktops, on Linux, Windows, MacOS X, Android). A dozen of servers and set-top-boxes, with variables capacities (in terms of processing and upload capabilities), were deployed over 7 cities located in 4 European countries. The Internet connectivity was provided by 9 different Internet Service Providers. Each client could simultaneously use from 1 to 10 servers. 15 different 4 to 7-minute videos were made available at 7 video bitrates: 200 Kbps, 1 Mbps, 1.5 Mbps, 2 Mbps, 3 Mbps, 4 Mbps and 6 Mbps.

For the evaluations, we have derived a set of criteria, each considered essential for the Quality of Experience of video streaming services, and compared MS-STREAM vs MPEG-DASH (from the dash.js player) on them. Our implementation obtained in average a 5.47Mbps mean bitrate whereas the DASH player reached 3.3Mbps only (Fig 4). Our player experienced very few rebuffering (0.4 per 5 minutes) compared to the DASH player (4 per 5 minutes) and also had a smaller start-up delay (1.1 sec vs. 2.03 secs) as shown in Fig. 5 and Fig. 6. Fig. 7 also exposes the average quality distribution displayed in time on end-users terminals that shows our solution to be more efficient than the evaluated DASH player. Thanks to the utilization of the P2P data exchange, these QoE gains also came along with 67% of bandwidth savings at the server infrastructure side. Finally, as the resiliency mechanism of our solution includes cancelling some late sub-segment requests, hence having unused GoPs transferred to the consuming clients, a bandwidth overhead of 6.5% was observed in our experiments.

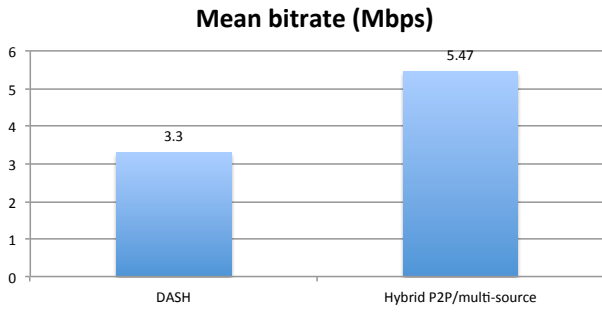


Figure 4: Mean Bitrate

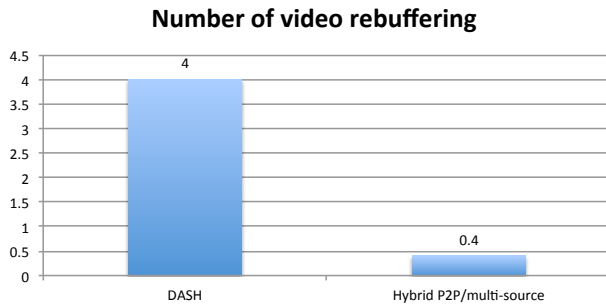


Figure 5: Average rebuffering events

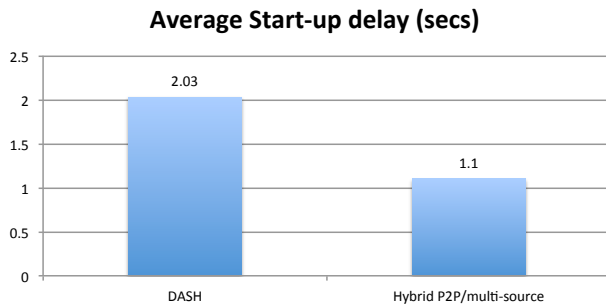


Figure 6: Average start-up delay

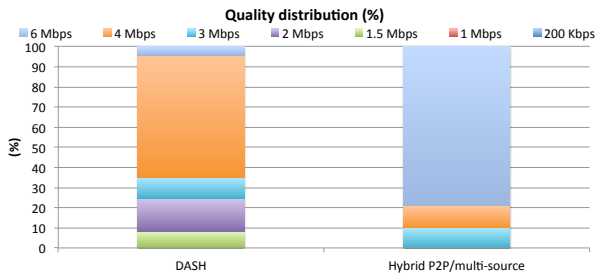


Figure 7: Quality Distribution

## 4 ONLINE DEMONSTRATION

A demonstration of this work is made available online at [10]. This platform implements individual, personalizable and interactive demonstrations of the the P2P/multi-server protocol for multiple users at the same time and in heterogeneous network environments with servers (set-top-boxes in end-users' homes) in France, Poland, Romania and Greece. On this online streaming platform, end-users are given the possibility to add/discard servers contributing to their streaming session, and can decide whether to act as contributing peers or not. Additionally, live feeds are displayed to end-users regarding the displayed bitrate, the number of video freezing events, the number of used servers/peers, the amount of data from servers and peers as well as the uploaded data. End-users can also simulate several peers on their own devices.

## REFERENCES

- [1] J. Bruneau-Queyreix, M. Lacaud, and D. Negru. 2017. A multiple-source adaptive streaming solution enhancing consumer's perceived quality. In *IEEE Consumer Communications and Networking Conference (CCNC), demonstration track*. Las Vegas, United States. <http://msstream.net>.
- [2] J. Bruneau-Queyreix, M. Lacaud, D. Negru, J. Mongay Batalla, and E. Borcoci. 2016. Multiple Description-DASH: Pragmatic streaming maximizing users Quality of Experience. *International Conference on Communications (ICC) (2016)*.
- [3] J. Bruneau-Queyreix, M. Lacaud, D. Negru, J. Mongay Batalla, and E. Borcoci. 2017. Adding a New Dimension to HTTP Adaptive Streaming through Multiple-Source Capabilities. *IEEE MultiMedia (2017)*.
- [4] J. Bruneau-Queyreix, M. Lacaud, D. Negru, J. Mongay Batalla, and E. Borcoci. 2017. MS-Stream: A multiple-source adaptive streaming solution enhancing consumer's perceived quality. *IEEE Consumer Communications and Networking Conference (CCNC) (2017)*.
- [5] J. Bruneau-Queyreix, M. Lacaud, D. Negru, J. Mongay Batalla, and E. Borcoci. 2017. QoE Enhancement Through Cost-Effective Adaptation Decision Process for Multiple-Server Streaming over HTTP. *IEEE International Conference on Multimedia and Expo (ICME) (2017)*.
- [6] Cisco. 2016. Cisco Visual Networking Index. (2016). <http://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf>
- [7] DASH Industry Forum. 2017. Catalyzing the adoption of MPEG-DASH. <http://dashif.org/>. (2017).
- [8] DASH Industry Forum. 2017. dash.js video player. <http://github.com/dash-industry-forum/dash.js/>. (2017).
- [9] ISO/IEC. 2014. 23009-1:2014 Dynamic adaptive streaming over HTTP. <https://www.iso.org/standard/65274.html/>. (2014).
- [10] MS-Stream. 2017. MS-Stream Demonstration. <http://msstream.net>. (2017).