MPEG-DASH for low latency and hybrid streaming services

Jean Le Feuvre, Cyril Concolato, Nassima Bouzakaria, Viet-Thanh-Trung Nguyen
Telecom ParisTech; Institut Mines-Telecom; CNRS LTCI
46, rue Barrault
75013 PARIS, France

{lefeuvre, concolato, bouzakaria, vnguyen}@telecom-parsitech.fr

ABSTRACT
While over-the-top video distribution is now widely deployed, it still suffers from much higher latencies than traditional broadcast, typically from a few seconds up to half a minute. In this paper, we demonstrate a novel DASH system with latency close to broadcast channels, and show how such a system can be used to enable combined broadcast and broadband services while keeping the client buffering requirements on the broadcast link low.

Categories and Subject Descriptors
C.2.4 [**Computer-Communication Networks**]: Distributed Systems – *client/server.*

General Terms
Algorithms, Performance, Experimentation, Standardization.

Keywords
Adaptive Streaming, Hybrid Broadcast Broadband, Low Latency.

1. INTRODUCTION
Video streaming today represents a large fraction of the Internet traffic. Among the streaming methods, HTTP adaptive streaming solutions are the most popular. In particular, the MPEG-DASH (Dynamic Adaptive Streaming over HTTP) standard [1] is foreseen as the future interoperable solution for various types of services, from on demand to live. One of the drawbacks of HTTP adaptive streaming solutions, including of MPEG-DASH, is the associated latency, which can be in the order of tens of seconds when deployed with typical parameters in some systems. Such latency may be acceptable for on demand services but is not acceptable for live events, especially when compared to existing broadcast solutions. This paper describes first the demonstration of open-source software used in the realization of an end-to-end audio/video delivery system using MPEG-DASH. This system is capable of low-latency, in the order of a frame duration (excluding the encoding time), similar to conventional RTP-based streaming. This is unique to the best of the authors' knowledge. Additionally, with such a system, it becomes possible to efficiently combine broadcast services, which can deliver the same content to a large audience with a reasonable latency, together with HTTP adaptive streaming services, which can deliver personalized content to individual users. This demonstration also shows the use of the MPEG-TEMI (Timeline and External Media Information) standard [2] to achieve the combination of MPEG-2 TS-based broadcast services and DASH-based services to offer hybrid broadcast/broadband services, using scalable HEVC videos. Finally, MPEG-DASH live services rely on accurate UTC timing configuration of both server and clients to locate the live edge, using e.g. common NTP servers. The demonstration shows how MPEG-DASH services can be accurately synchronized to broadcast events even if NTP servers are absent.

MM'15, October 26-30, 2015, Brisbane, Australia
ACM 978-1-4503-3459-4/15/10.
http://dx.doi.org/10.1145/2733373.2807977

2. SYSTEM ARCHITECTURE

2.1 Low-latency live DASH streaming

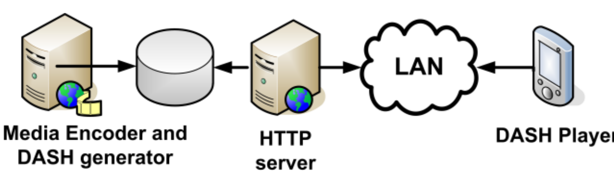

Figure 1 Architecture of the low-latency streaming system

In previous work [3], an analysis of the latency of live streaming services using MPEG-DASH was presented, indicating that the major component inducing latency was the segmentation process. This paper presented a method capable of achieving low latency, as low as frame duration, based on HTTP/1.1 chunked-transfer encoding and on a specific packaging using the ISO Base Media File Format (ISOBMFF) movie fragments. The system, demonstrating such results, is depicted in Figure 1 and composed of the following open-source tools, developed within the GPAC[1] project [4]:

- A live MPEG-DASH encoder, called DashCast. This tool encodes, using FFmpeg[2], raw audio and video frames, captured from a web cam or from screen captures, into multiple MPEG-4 AVC|H.264 and AAC streams of different qualities, and packages them in ISOBMFF compliant media segments. It also produces MPEG-DASH Media Presentation Descriptions (MPD) describing the service. To achieve low latency, the

[1] http://www.gpac.io

[2] http://www.ffmpeg.org

video encoder uses a low-latency coding profile without bidirectional prediction; and segments are flushed to disk and published as small fragments as soon as ready.

- An HTTP/1.1 compliant server[3], written as a NodeJS[4] module, is used to deliver both the MPD and media segments. To achieve low latency, the server detects when new movie fragments are flushed on disk and pushes them immediately using HTTP/1.1 chunked-transfer encoding.
- An MPEG-DASH player, called MP4Client. This tool reads the DASH MPD, selects an appropriate stream quality and fetches the media segments. To achieve low latency, the de-jittering network buffer is configured to a small value. Movie fragments are read progressively and frames from these fragments are dispatched to the decoder immediately.

Different settings are demonstrated for the encoder (group of pictures size), for the client buffer (de-jittering buffer size), and for the packager (movie fragment duration).

2.2 Hybrid Broadcast/Broadband Delivery

Hybrid broadcast/broadband services allow providing additional, personalized content on the broadband channel, synchronized to the mass content delivered over the broadcast channel. One challenge in such hybrid scenario is that typical broadcast channels feature a constant latency, usually lower than HTTP streaming services. Extending broadcast services with HTTP streaming services requires accurately synchronizing data from both channels, possibly at the frame level (e.g. for scalable enhancement) without introducing additional buffers and latency. In our test bed, we observed a delay of 1 to 2 s on a DVB-T2 link fed by IP multicast. Given such delay, with the use of low latency DASH as described previously, i.e. with fragments pushed in typical CDN early, the broadband data can arrive at the receiver before the corresponding broadcast data. This can guarantee a proper synchronization.

When the broadband content to be synchronized with the broadcast is live content, i.e. not entirely available at the beginning of the session, MPEG-DASH requires the use of 'dynamic' MPD, which implies accurate UTC clock sampling at both server and client sides in order to locate the live edge. This requirement may not be always satisfied: different, not accurately synchronized NTP servers may be used by client and servers; or no NTP may be available at the client side. The latest DASH standard specifies that a specific time server may be indicated in the MPD. This approach requires the client to fetch the time information before processing segments, therefore introducing an additional delay. In order to avoid this, in this demonstration, we inject NTP used by the DASH origin server in the broadcast, thereby helping the client find the live edge unambiguously.

In this demonstration, the system depicted in Figure 2 is as follows:

- A low latency MPEG-DASH broadband channel is produced with the system described in Section 2.1, with the exception that the live encoder is replaced by a live simulator. This simulator uses as input an HEVC enhancement layer stored in an MP4 file.
- A broadcast channel is simulated by the use of an IP multicast delivery of an MPEG-2 Transport Stream (TS), generated from the base HEVC layer. The MPEG-2 TS PCR information is randomly initialized at startup to demonstrate synchronization aspects, and enriched with location, i.e. the HTTP URL of the DASH session, and timing information, i.e. UTC and PCR-to-DASH-time mapping, compatible with the TEMI standard.
- A hybrid broadcast/broadband client runs on a separate machine, with inaccurate UTC settings. It uses NTP information from the broadcast stream to locate the live edge despite its clock configuration and plays the video reconstructed from the base and enhancement layers.

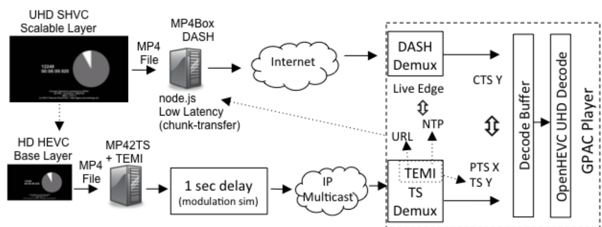

Figure 2 Demonstration of low-latency hybrid broadcast broadband service using MPEG-DASH and MPEG-TEMI

Different settings are demonstrated for the low-latency packager (movie fragment duration), for the multicast (propagation delay) and for the client (de-jittering buffer size).

3. CONCLUSION

We demonstrate a unique low latency DASH system for both broadband and hybrid broadcast/broadband delivery chains, based on open source tools and standards. In future works, we plan to extend these tools by enabling live encoding in the hybrid chain.

4. ACKNOWLEDGMENTS

This work is supported by the European Celtic-Plus H2B2VS[5] and 4Ever 2[6] projects funded by the French government.

5. REFERENCES

[1] Stockhammer, T. 2011. Dynamic adaptive streaming over HTTP --: standards and design principles. In Proc. of ACM MMSys '11. pp. 133-144. DOI=10.1145/1943552.1943572

[2] ISO/IEC 13818-1:2015:Amd.2 Timeline and External Media Information

[3] Bouzakaria, N., Concolato, C., & Le Feuvre, J. 2014. Overhead and performance of low latency live streaming using MPEG-DASH. In Proc. of IEEE IISA 2014. pp. 92-97.

[4] Le Feuvre, J, Concolato, C, Dufourd, J-C, Bouqueau, R, and Moissinac, J-C. 2011. Experimenting with multimedia advances using GPAC. In Proc. of ACM MM '11. pp. 715-718. DOI=10.1145/2072298.2072427

---

[3] https://github.com/gpac/node-gpac-dash

[4] https://nodejs.org/

[5] http://h2b2vs.epfl.ch/

[6] http://www.4ever2-project.com/