INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC1/SC29/WG11 MPEG2015/M37254
October 2015, Geneva, Switzerland

Source **Telecom ParisTech**
Status **For consideration at the 113th MPEG Meeting**
Title **Guidelines for DASH Fast Start**
Author Cyril Concolato, Jean Le Feuvre, Nassima Bouzakaria

1 Introduction

This contribution presents the results of an experimentation made to reduce the delay to start a DASH session. It proposes to update the DASH Guidelines based on the results of this experimentation and to amend the DASH standard. These results are extracted from a research paper presented at MMSP 2015 at the same time as the 113th MPEG meeting [1].

N. Bouzakaria, C. Concolato and J. Le Feuvre, *Fast DASH Bootstrap*, MMSP, Xiamen, China, October 2015.

2 Experimentation

We propose to reduce the startup delay of a DASH session by removing the need for the multiple HTTP requests required for the MPD and the associated initialization segments (IS).

We tested two approaches that require only one request to retrieve the MPD and its associated IS. The approaches are as follows:

a) the IS are embedded in the MPD URLs using Base 64 encoding and the "data:" scheme
b) the MPD is extended with an XML element that provides the necessary information to reconstruct the initialization segment from the MPD. This extension element called ISOBMFFMoov contains a Track element which gives the track id, the edit list information (if any) and a Base64 encoded version of the 'stsd' box.

We applied the above approaches on the DASH-IF Test vectors, only the non-encrypted ISOBMFF Live profile ones, taking only the first period into consideration.

We then deployed 2 DASH clients (GPAC MP4Client and the DASH-IF DASH.js player) and a HTTP/1.1 server in a network with an RTT of 50ms and a bandwidth of 2 Mbps. The server was using a TCP init congestion window of 3 or 10 TCP segments.

We then measured:
- the total download size (HTTP/1.1 headers, MPD and IS)
- the delay between the initial MPD request and the reception end of the last IS

2.1 Download size comparison

We compared the total download size as downloaded by GPAC's MP4Client, DASHIF DASH.js and a simple wget, when using HTTP/1.1, with or without our approaches.

The table below summarizes the results.

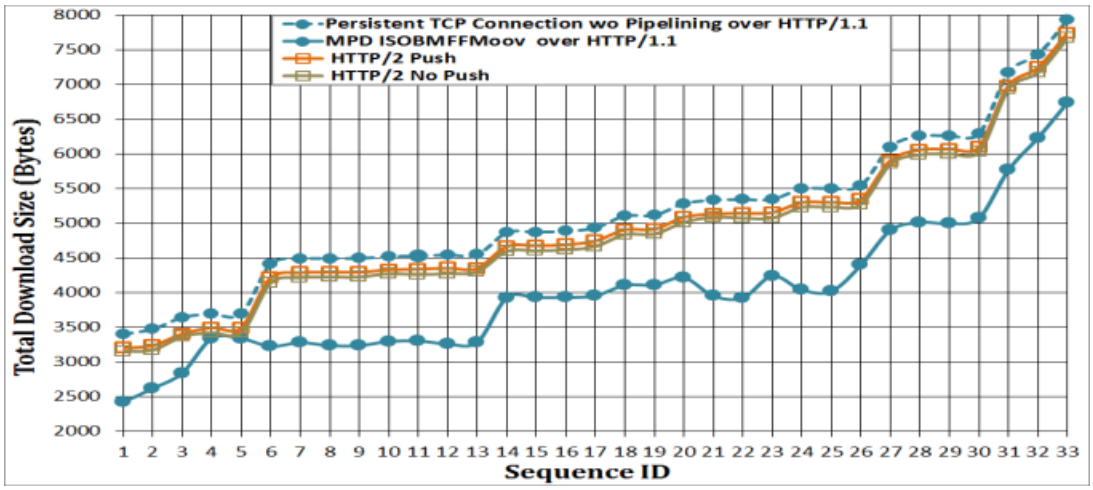
Downloader client	**GPAC**	**DASH.js**	**wget**	
Download method	**all IS**	**A/V IS**	**MPD + Base64 All IS**	**MPD + ISOBMFFMoov all IS**
Average	7524	5221	7627	4075
Min	3972	3641	3829	2451
Max	10364	8168	11211	6844

We note that:

- Download sizes are small in all approaches, at most 11 KB. This is small compared to media segment sizes and this is independent of the resolution, bitrate … The sizes only depend on the size of the MPD, the configuration of the HTTP server, the number of IS and the configuration of those IS (number of tracks, use of avc3 or avc1, use of edit lists …).
- GPAC downloads more bytes compared to Dash-Js. This is due to the fact that GPAC downloads all IS to prepare for future switching. During a switch GPAC will not fetch the new IS while DASH.js will.
- The approach using Base64 is bigger by 46% compared to the DASH.js download, but very similar to the GPAC download. In any case, the size is still small compared to media segment size.
- The ISOBMFFMoov approach reduces the download size by an average of 25% compared to Dash-JS approach on HTTP/1.1.

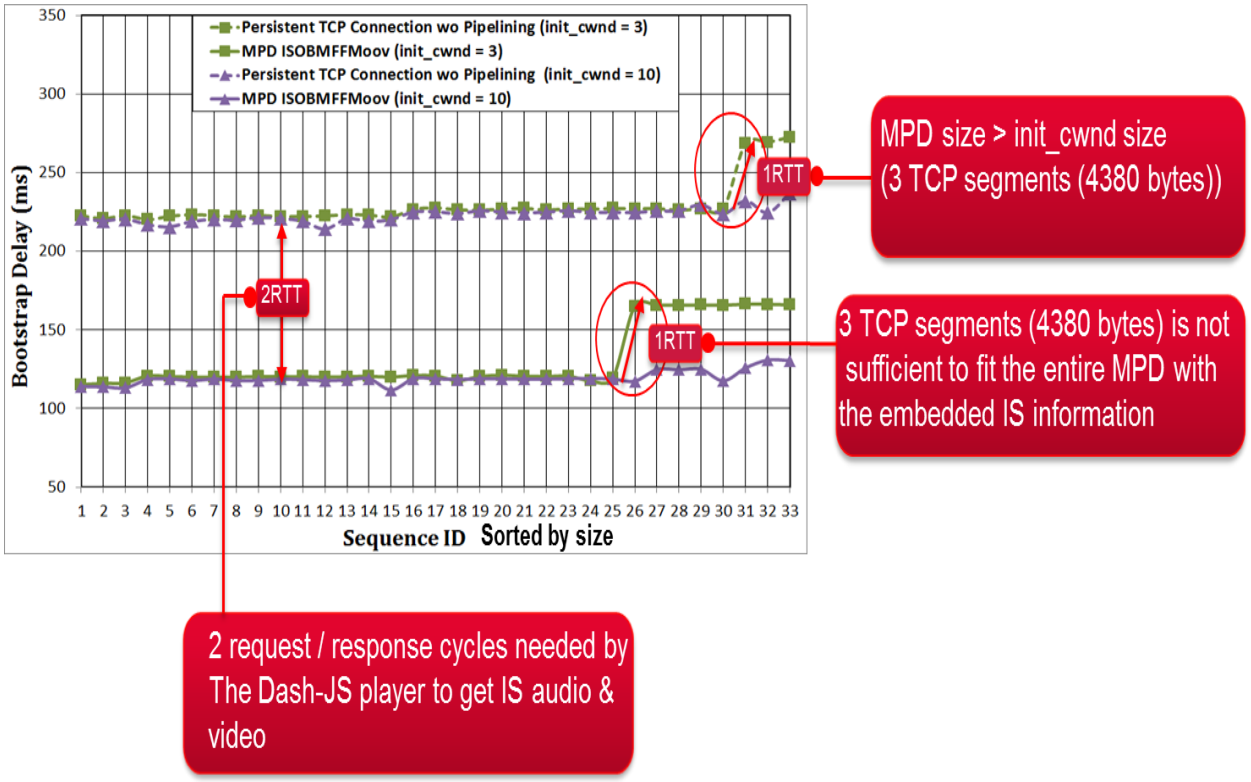
It should be noted that the CORS header have been removed for the HTTP header sizes because they were large (+250 bytes per request) and would provide even worse results for the first 2 columns above (i.e. not using our optimization).

The figure below shows the detailed download sizes for each DASH IF sequence, where the sequences are sorted by download size using DASH.js over HTTP/1.1. We see that our approach is always beneficial (even compared to HTTP/2).



2.2 *Download delay comparison*

We then compared the time difference between when the client makes the request for the MPD and when it receives the last IS. The "Persistent TCP Connection wo Pipelining" approach corresponds to DASH.js, while the MPD ISOBMFFMoov approach corresponds to wget.



Whatever the sequence, using an init congestion window of 10, the download delay is almost constant. Using the base64 approach or the ISOBMFFMoov approach, the download delay is reduced by 2 RTT, because the player downloads 2 IS (Audio+Video).

3 Conclusion

The experimental results reported in this contribution indicate that delivering some (even all) IS at the same time as the MPD can be beneficial to reduce the start-up delay in DASH sessions, on HTTP/1.1. The results also show that the total download size can be reduced by embedding the

IS in the MPD (even in some cases using Base 64). The exact IS to be embedded could be determined by the server using the existing HTTP "Accept" headers or the Client-hints.

We recommend MPEG to describe as a guideline to DASH to embed initialization segments in the MPD to reduce the start-up delay.

We finally recommend MPEG to consider defining in DASH the ability to reconstruct initialization segments from MPD information to reduce the total download size.