Network Working Group Internet-Draft Intended status: Informational Expires: September 24, 2015

# DASH and HTTP2 draft-ruellan-httpbis-dash-http2-00

### Abstract

This document describes possibilities for DASH to take advantage of HTTP/2 features, in particular of server push.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of <u>BCP 78</u> and <u>BCP 79</u>.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <u>http://datatracker.ietf.org/drafts/current/</u>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 24, 2015.

### Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to <u>BCP 78</u> and the IETF Trust's Legal Provisions Relating to IETF Documents (<u>http://trustee.ietf.org/license-info</u>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

| $\underline{1}$ . Introduction           | • |  | • | • |  | · | • | • | • | ÷ | ÷ |  |  | • | <u>2</u> |
|--|---|--|---|---|--|---|---|---|---|---|---|--|--|---|----------|
| $\underline{2}.$ Using Push with DASH .  |   |  |   |   |  |   |   |   |   |   |   |  |  |   | <u>2</u> |
| <u>2.1</u> . Live Video Latency          |   |  |   |   |  |   |   |   |   |   |   |  |  |   | <u>2</u> |
| <u>2.2</u> . FDH Proposal                |   |  |   |   |  |   |   |   |   |   |   |  |  |   | <u>3</u> |
| <u>2.3</u> . Other Proposals             |   |  |   |   |  |   |   |   |   |   |   |  |  |   | <u>3</u> |
| $\underline{3}$ . Informative References |   |  |   |   |  |   |   |   |   |   |   |  |  |   | <u>5</u> |
| Author's Address                         |   |  |   |   |  |   |   |   |   |   |   |  |  |   | <u>5</u> |

### **<u>1</u>**. Introduction

MPEG Dynamic Adaptive Streaming over HTTP (DASH [DASH]) is an adaptive bit-rate HTTP-based streaming solution standardized under MPEG. In DASH, a media is split into segments. Each segment of the media is encoded with different bit-rates to provide adaptation possibilities.

The client controls the streaming and the adaptation. To play the media, the client requests the segments one after the other, when needing them. For each segment, it can select the most appropriate bit-rate.

DASH defines a format for describing a media, the Media Presentation Description (MPD). The MPD contains information about the media, in particular the list of segments making up the media, and the different available bit-rates. DASH also defines formats for the media segments themselves.

The Full-Duplex HTTP based Protocols (FDH) is a Core Experiment inside MPEG for studying bi-directional communication protocols, such as HTTP/2 [<u>I-D.ietf-httpbis-http2</u>] and WebSocket [<u>RFC6455</u>], in the context of DASH. Inside this Core Experiments, the usage of the new features of HTTP/2 for DASH is investigated. Among these features, server push is attracting much interest.

## 2. Using Push with DASH

## **<u>2.1</u>**. Live Video Latency

When using DASH for streaming a live video, there are some trade-off to make for choosing the duration of a segment. Shorter segments enable a lower latency, which is of importance for live video. However shorter segments also induce a larger number of requests, which can negatively impact the network performances.

In [LowLatencyStreaming], a solution to this problem is proposed: the live video is split into short segments. However, for each segment requested by the client, the server not only sends the corresponding segment, but also pushes some of the following segments. This

enables to use a low latency while still keeping a low number of requests, therefore not incurring any negative impact on the network.

To let the client keep the control of the streaming, the client selects the number  $_k$  of segments the server should push and transmit it to the server in its request. This strategy of segment transmission is therefore called  $_k$ -push\_.

## 2.2. FDH Proposal

The FDH Core Experiment has built upon this \_k-push\_ strategy for taking advantage of the push feature of HTTP/2, while letting the client control the streaming.

This Core Experiment is defining a new HTTP header field, named "DASH-PUSH", that enables the client to indicate what it would like the server to push.

The "DASH-PUSH" header field contains parameters defining the strategy proposed by the client for indicating which segments it would like the server to push. Currently two strategies are defined.

o Pushing next \_K\_ segments:

The  $_K$  segments following the requested segment are pushed by the server. For example, for having the server push the next 5 segments, the following header field is used:

"DASH-PUSH: type=push-next; K=5"

o Pushing for \_T\_ seconds:

The segments corresponding to the given duration following the requested segment are pushed by the server. For example for having the server push the segments corresponding to the next 10 seconds of the media, the following header field is used:

"DASH-PUSH: type=type=push-time; T=10"

#### **<u>2.3</u>**. Other Proposals

The "DASH-PUSH" header field can be extended for increasing its expressiveness.

o Using a list of URIs:

DASH and HTTP2

All the resources listed are pushed by the server. For example, for having the server push the media segments "seg-1.mp4", "seg-2.mp4", and "seg-3.mp4", the following header field is used:

"DASH-PUSH: type=uri-list; seg-1.mp4; seg-2.mp4; seg-3.mp4"

o Using a URI template:

Instead of listing all the URIs of the resources to push, a URI Template [<u>RFC6570</u>] can be used. The URI Template can use numbered variables whose values are defined as parameters of the "DASH-PUSH" header. These variable can have a single value, a list of values, or a range of values. For a variable with several values, the URI Template is expanded for each value of the variable.

For example, for having the server push the same media segments as in the previous example, using a URI Template with a variable defined as a list of values, the following header field is used:

"DASH-PUSH: type=uri-template; seg-{1}.mp4; 1 : 2 : 3"

Note that the values in the list are separated with colons.

The same example can also use a range of values instead of a list, with the following header field:

"DASH-PUSH: type=uri-template; seg-{1}.mp4; { 1 - 3 }"

Several variables can also be used inside the URI Template, in this case, the URI Template is expanded for each combination of values for the defined variables. For example, to have the server push several media segments, numbered 1 to 3, each with two different representation levels, R1 and R2, the following header field is used:

"DASH-PUSH: type=uri-template; seg-rep-{1}-{2}.mp4; R1 : R2; { 1 3 }"

o Using a URI wildcard:

Instead of specifying a list of URIs, a wildcard can be used, allowing to have the server push all the resources corresponding to a given media type. For example, for having the server push all the "text/css" resources linked to the requested resource, the following header field is used:

"DASH-PUSH: type=uri-wildcard; text/css"

DASH and HTTP2

o Header field value:

The "DASH-PUSH" header field is not restricted to target only URIS, but can also be used to target other header fields, in defining the resources the client would like the server to push. For example, for having the server push different ranges of the requested resource, the following header field is used:

"DASH-PUSH: type=header; Range; 501-1000; 1001-1500"

This usage is similar to the list of URIs, but targeting specific header fields. It could be extended to support templates.

## **<u>3</u>**. Informative References

- [DASH] , "Dynamic adaptive streaming over HTTP (DASH)", ISO/IEC 23009-1:2014, 2014.
- [I-D.ietf-httpbis-http2] Belshe, M., Peon, R., and M. Thomson, "Hypertext Transfer Protocol version 2", <u>draft-ietf-httpbis-http2-17</u> (work in progress), February 2015.

[LowLatencyStreaming]

Wei, S. and V. Swaminathan, "Low Latency Live Video Streaming over HTTP 2.0", NOSSDAV 2014, 2014.

- [RFC6455] Fette, I. and A. Melnikov, "The WebSocket Protocol", <u>RFC</u> 6455, December 2011.
- [RFC6570] Gregorio, J., Fielding, R., Hadley, M., Nottingham, M., and D. Orchard, "URI Template", <u>RFC 6570</u>, March 2012.

Author's Address

Herve Ruellan

Email: herve.ruellan@crf.canon.fr