

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: April 30, 2015

A. Begen  
Cisco  
K. Streeter  
Adobe Systems Incorporated  
I. Bouazizi  
Samsung Research  
F. Denoual  
Canon Research Centre France  
October 27, 2014

**MPEG DASH Requirements for a webpush Protocol**  
**draft-begen-webpush-dash-reqs-00**

Abstract

This draft presents two of the ongoing core experiments for the amendment of the MPEG Dynamic Adaptive Streaming over HTTP (DASH) specification and discusses some requirements for a webpush protocol in the context of these core experiments.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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 <http://datatracker.ietf.org/drafts/current/>.

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 April 30, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) 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

- [1. Introduction](#) . . . . . [2](#)
- [2. DASH Requirements in the Core Experiments](#) . . . . . [3](#)
  - [2.1. Requirements from the FDH CE](#) . . . . . [3](#)
  - [2.2. Requirements from the SAND CE](#) . . . . . [3](#)
- [3. Guidance from the IETF](#) . . . . . [4](#)
- [4. Security Considerations](#) . . . . . [5](#)
- [5. IANA Considerations](#) . . . . . [5](#)
- [6. Informative References](#) . . . . . [5](#)
- Authors' Addresses . . . . . [5](#)

**1. Introduction**

HTTP adaptive streaming (HAS) has become the technology of choice for delivering video content over the Internet. HAS will play an increasingly important role in delivering video to the primary and secondary screens.

In HAS, streaming clients are offered multiple representations of the same content. Clients independently choose a segment (i.e., a piece of the content) belonging to a representation to fetch from the content delivery network, a choice that is made at every switching boundary (usually every 2-10 seconds). The choice is based on a number of parameters, including the network throughput observed by the client.

To date they have been several different HAS implementations from different vendors. To achieve interoperability, MPEG has started working on a specification in 2010 and published the first edition of the Dynamic Adaptive Streaming over HTTP specification in 2012. Earlier this year, the second edition has been published [[DASH-Part1](#)].

To add new features and capabilities to the DASH applications, the DASH subgroup is currently running a number of core experiments (CE) [[DASH-CE](#)]. Two of these CEs are (i) DASH over Full Duplex HTTP-based Protocols (FDH) and (ii) Server and Network Assisted DASH (SAND). In these CEs, there is a need for a webpush protocol. This draft discusses some requirements for a such protocol to be used in DASH applications.



## **2. DASH Requirements in the Core Experiments**

### **2.1. Requirements from the FDH CE**

The FDH CE is investigating the problem of delivering media segments with shorter delay and/or reducing the request processing on the HTTP server. The technologies considered are HTTP/2 and WebSockets.

The main idea is that once a client is interested in streaming a particular content (e.g., a live channel), it may be beneficial not to send an individual GET request for each segment to the network. The network may keep sending segments once they become available to the client for a predetermined amount of time or until the client tells the server to stop.

We realize that a content delivery network or an operator network can consist of both HTTP/2 enabled servers and HTTP 1.1 servers as well as proxy servers, some of which could have been implemented improperly. An important requirement for us is that the push feature should not cause any caching inconsistency or reduce caching efficiency. Note that not all clients streaming the same content will necessarily use the push feature. Some may continue fetching each segment separately (the conventional way), some may ask for the push of the next k segments from the server or some may ask for push till a further command.

An important issue is to decide how the clients will be able to ask for not one but multiple segments in a single GET request, i.e., specifying a list of segments or potentially a pattern that includes wildcards or any other agreeable formats. At least two options are considered: a modified URL scheme and the use of an extension header. In both cases modifications are expected on the origin server and possibly the cache servers (We will likely need a special handler on the servers). We would like to get advice on any of these technologies taking into account among others backward-compatibility, efficiency, caching issues, processing load, extensibility and likelihood of implementation.

The DASH clients may run in browsers, mobile applications, on personal computers or other connected devices. Typically it cannot be assumed that in all applications a DASH client will have access to the HTTP stack in a given platform. Such constraints should be taken into account when looking for a broadly acceptable and highly-scalable solution.

### **2.2. Requirements from the SAND CE**



The SAND CE has the goal of establishing a bi-directional messaging plane between the clients and other so-called Dash-aware Network Elements (DANE). The DANEs may also communicate with each other. Common DANEs include proxies and caches as well as analytics servers and other network devices. On the direction from a DANE to a DASH client (or another DANE), the messages can carry any kind of operational information and/or assistance information so that the DASH client (or the DANE) can take a proper action. On the direction from a DASH client to a DANE, the messages are expected to carry metrics and status information.

A simple use case for sending an operational information to a DASH client is to ask the client to switch to a particular CDN provider. Another use case is to ask the DASH client to fetch a particular representation, simply because that representation's segments are already cached at the edge of the network. Yet other use cases may include network status information such as bitrate guarantees, delays, etc.

An obvious choice for the protocol to carry these SAND messages back and forth is HTTP. The DASH clients can send their metrics and status messages upon demand or periodically to a pre-designated server. However, sending messages in the other direction (from a DANE to a DASH client) may be more challenging since DASH clients may or may not be expecting such a message. It is important to note that we do not want to put time-sensitive client-specific feedback messages on top of the HTTP responses containing non-time-critical and non-client-specific media segments.

To enable bi-directional communication between a DASH client and a DANE, we can also use WebSockets or the HTTP/2 PUSH technologies. However, we consider a more lightweight protocol that will scale to large populations when sending a common message or individual messages.

### **3. Guidance from the IETF**

We would like to ask the webpush WG to consider media delivery use cases as part of the webpush activity, including the delivery of MPEG DASH content. We also would like to ask the webpush WG to advise us on best practices on using header and URL based signaling for push behavior, and get feedback on proper message channels for SAND.



#### **4. Security Considerations**

There are no security considerations.

#### **5. IANA Considerations**

There are no IANA considerations.

#### **6. Informative References**

[DASH-Part1]

Available at: <http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html>, , "ISO/IEC 23009-1:2014: Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats (2nd edition)", May 2014.

[DASH-CE] Available at: [http://wg11.sc29.org/doc\\_end\\_user/current\\_meeting\\_intermediate.php?id\\_meeting=162](http://wg11.sc29.org/doc_end_user/current_meeting_intermediate.php?id_meeting=162), , "Descriptions of core experiments on DASH amendment (w14858)", Oct. 2014.

#### Authors' Addresses

Ali Begen  
Cisco  
181 Bay Street  
Toronto, ON M5J 2T3  
Canada

EEmail: [abegen@cisco.com](mailto:abegen@cisco.com)

Kevin Streeter  
Adobe Systems Incorporated  
345 Park Avenue  
San Jose, CA 95110  
USA

EEmail: [kstreete@adobe.com](mailto:kstreete@adobe.com)





Imed Bouazizi  
Samsung Research  
1301 E. Lookout Dr.  
Richardson, TX 75082  
USA

E-Mail: [i.bouazizi@sta.samsung.com](mailto:i.bouazizi@sta.samsung.com)

Franck Denoual  
Canon Research Centre France  
Rue de la Touche Lambert  
Cesson-Sevigne 35517  
France

E-Mail: [franck.denoual@crf.canon.fr](mailto:franck.denoual@crf.canon.fr)