Dynamic Adaptive Streaming over HTTP (DASH) – A Flexible and Efficient Solution for Mobile Multimedia

[some authors]
Corporate R&D
Qualcomm Incorporated

QUALCOMM Incorporated
5775 Morehouse Drive
San Diego, CA 92121-1714
U.S.A.

ABSTRACT

DASH is a web-compatible international standard for describing multi-rate encoded multimedia, allowing intelligent client playback applications to dynamically choose which portions of a media presentation to select for rendering to the user based on network dynamics or available device resources. DASH also provides a framework for hosting a common encryption mechanism, usable by multiple digital rights management (DRM) systems. We describe the benefits of the DASH standard relative to other proprietary streaming solutions, along with some of the value-added algorithms and implementation techniques for DASH players provided by Qualcomm.

Keywords

3GPP, MPEG, DASH, video, mobile video, standards, adaptive streaming, web, HTML5, DRM, common encryption

1. Introduction

In the last few years, video content has grown to be a significant portion of Internet traffic. In 2011, according to Sandvine, Netflix's video streaming service alone accounted some 20% of peak Internet download traffic [1]. In parallel, broadband wireless network capacity and mobile device capabilities have improved significantly and rapidly. Using LTE, download speeds in excess of 10Mb/s to a single device are not uncommon. Furthermore, many mobile devices are able to render and display full 1080p high-definition video.

According to a recent study [2], mobile data traffic will grow by a factor of 26 between 2010 and 2015, nearly doubling every year. Figure 1 shows expected growth of video as a fraction of the total mobile data, representing over 4 Exabytes or 65% of the total amount in 2015. While the maximum network capacity and device capabilities can in principle provide a high-quality user experience, various degradations can occur, due largely to variability in available network capacity or device resource consumption. Service providers, operators, device manufacturers and other ecosystem players must coordinate their offerings in order to satisfy user demands while continuing to support viable business opportunities in the mobile multimedia marketplace. One step in this direction is to develop and deploy a common, efficient and flexible Internet-compatible distribution platform that scales to the rising demands. The Dynamic Adaptive Streaming over HTTP (DASH) standard aims to meet this goal. It is a standard for optimizing the delivery of multimedia over the open Internet. However, its scope is sufficiently flexible to address additional use cases. In addition, it allows for considerable latitude in client implementations. Qualcomm's client implementation uses a number of algorithms to optimize the user experience on mobile, SnapDragon-based devices.

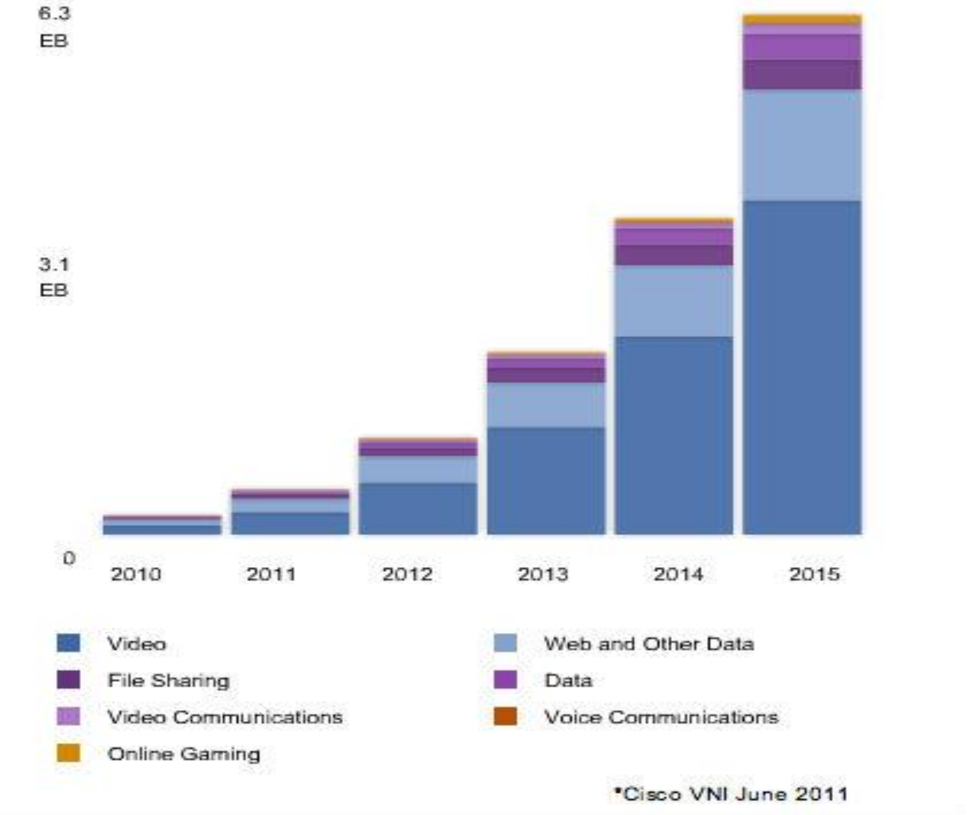


Figure 1 Video accounts for 65% percent of global mobile data traffic by 2015 [3]

2. Traditional Streaming Technologies

To date, a combination of proprietary and non-proprietary methods have been used to provide Internet-compatible video streaming. Traditional streaming generally uses a protocol requiring the server to keep per-client state (e.g., playback position). For example, in the Real-Time Streaming Protocol (RTSP) [4] after a session between the client and the server has been established, the server packages and sends the appropriate media content as a continuous stream of selected segments over either RTP/UDP/IP (the Real-Time Protocol [5]) or TCP/IP. In addition to RTSP, HTTP can be used for carrying multimedia content. The HTTP streaming approach is stateless at the server, and consequently far more scalable. If an HTTP client requests some data, each HTTP request is handled as a completely independent transaction and may be satisfied either by the server or by an HTTP caching agent, which are now in widespread use.

As an alternative to “pure” streaming approaches with RTSP and HTTP, which buffer only small amounts of data, progressive download (PD) may be used for media delivery from standard HTTP Web servers. Clients that support HTTP PD can seek to positions in the media stream by performing byte range requests to a Web server (assuming the server supports HTTP/1.1 [6]) and cache a substantial volume of content locally.

The disadvantages of PD can be important. Bandwidth may be wasted if the user decides to stop watching the content after downloading has started, and while PD is being stopped other content selection (e.g., fast forward) may not be available. In addition, PD is not typically bitrate adaptive, leading to potential degradation of the user experience or unnecessary loading of the supporting infrastructure and devices. Finally, having potentially large volumes of content stored in a user’s device may be objectionable to premium content owners or the device’s owner. DASH addresses the weaknesses of RTP/RTSP-based streaming and PD while preserving the scalability and compatibility benefits of today’s existing HTTP and web infrastructure.

3. DASH Infrastructure Compatibility

DASH streaming provides the benefits of rate-adaptive content while leveraging the significant existing investment in caching and data center web technology.

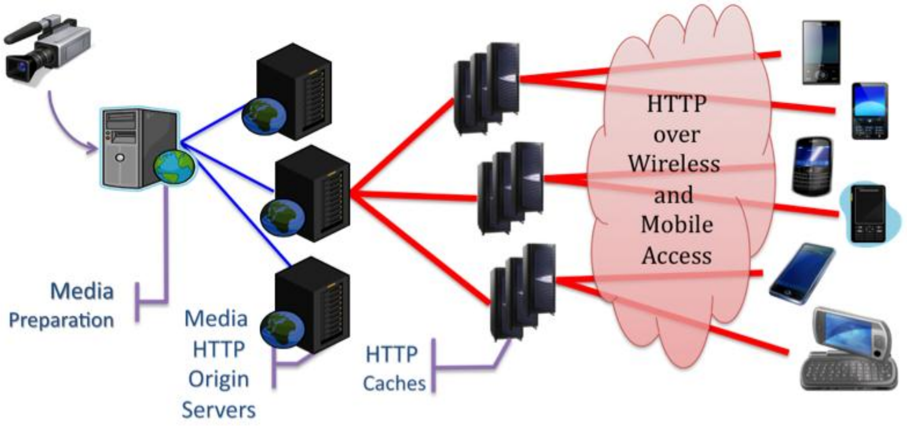

HTTP over Wireless and Mobile Access
Media Preparation
Media HTTP Origin Servers
HTTP Caches

Figure 2 Example media distribution architecture

Figure 2 shows a possible media distribution architecture for DASH. The media preparation process (also called "ingestion") typically generates encoded media segments. The segments are arranged by type into components (e.g., audio and video) and multiple distinct collections of segments ("tracks") may represent the same time period within a presentation. The purpose of this configuration is to support user selection of options such as camera angle, quality/rate, language, etc.

The ingestion process also produces a catalog or index of the timing, encoding, and type parameters for each segment. For deployment, the segments and index are then hosted on one or more media origin servers. The servers are preferably conventional HTTP servers. Any communication with the servers uses HTTP 1.1 with optional byte range requests and pipelining (indicated by a red line in the picture). Clients first retrieve the index when preparing to display a presentation. Based on meta-data in the index, clients request media segments using HTTP GET or partial GET methods.

The client fully controls the streaming session, as no state is maintained at the server. The client is responsible for making timely requests executing whatever processes are required to produce eventual smooth playback to a user. The presence of multiple quality encodings of the content allows intelligent clients to decide which encodings to display at which times, influenced by characteristics of the environment (e.g., network bandwidth or variability, device battery level, user preference).

Scalable media distribution requires the availability of infrastructure to handle the aggregate client load. HTTP-based Content Distribution Networks (CDNs) have been used very successfully to serve Web pages from a variety of locations, reducing load on origin servers and reducing download latency for users. Such systems generally consist of a distributed set of caching Web proxies and a set of request redirectors. Given the scale, coverage, and reliability of HTTP-based CDN systems, it is appealing to use them as the base to launch streaming services.

While leveraging CDN infrastructure is appealing, the use of HTTP brings other benefits:

1. HTTP streaming is already the primary method for delivering Internet multimedia, including video.

2. HTTP-based delivery avoids NAT and firewall traversal issues that arise with other protocols such as RTSP and RTP.

3. HTTP-based delivery utilizes the underlying TCP transport protocol and need not implement its own error repair or congestion control methods.

4. HTTP-based delivery provides the ability to use standard HTTP servers and standard HTTP caches (or inexpensive servers in general) to deliver the content. Cloud-based web servers can be used without change.

5. HTTP-based delivery avoids per-client state at each server. Consequently, ROI is improved by allowing less expensive infrastructure to serve an equal-sized client population.

6. Adaptive HTTP-based delivery provides the ability to change content rate at the client on the fly in reaction to changes in available bandwidth or device state. No negotiation with a streaming server is required.

Based on these considerations, ISO/IEC SC29 WG11 (aka MPEG) and 3GPP have created and adopted the DASH standards. The master specification will be available publicly as ISO/IEC 23009-1 [7]. However, unlike some other 3GPP and MPEG specifications, the standards are in no way restricted for use on 3GPP radio access networks (RANs).

The standards support features such as

- fast initial startup and seeking (including "trick modes")
- adaptive bitrate switching and content insertion
- use of CDN components (HTTP servers and caches)

- use of existing media frameworks for playback
- support for on-demand, live and time-shifted delivery
- support for a common encryption algorithm, usable by multiple different types of Digital Rights Management (DRM) systems

DRM is required by most content owners for the distribution and playback of premium content such as high-definition recent movies and sporting events.

4. DASH Design

DASH structures multimedia content in a hierarchical fashion. The top level of the hierarchy describes an entire media object (e.g., movie) and is called a Presentation. A Media Presentation Description (MPD, a form of meta-data) describes the various constituent objects that comprise a Presentation. The bottom of the hierarchy consists of Segments, each of which are named using a URI[1]. There are additional structures used to select among and within Segments to achieve various playback requirements. The selection may be based on a number of factors including user preference, device state, and network performance.

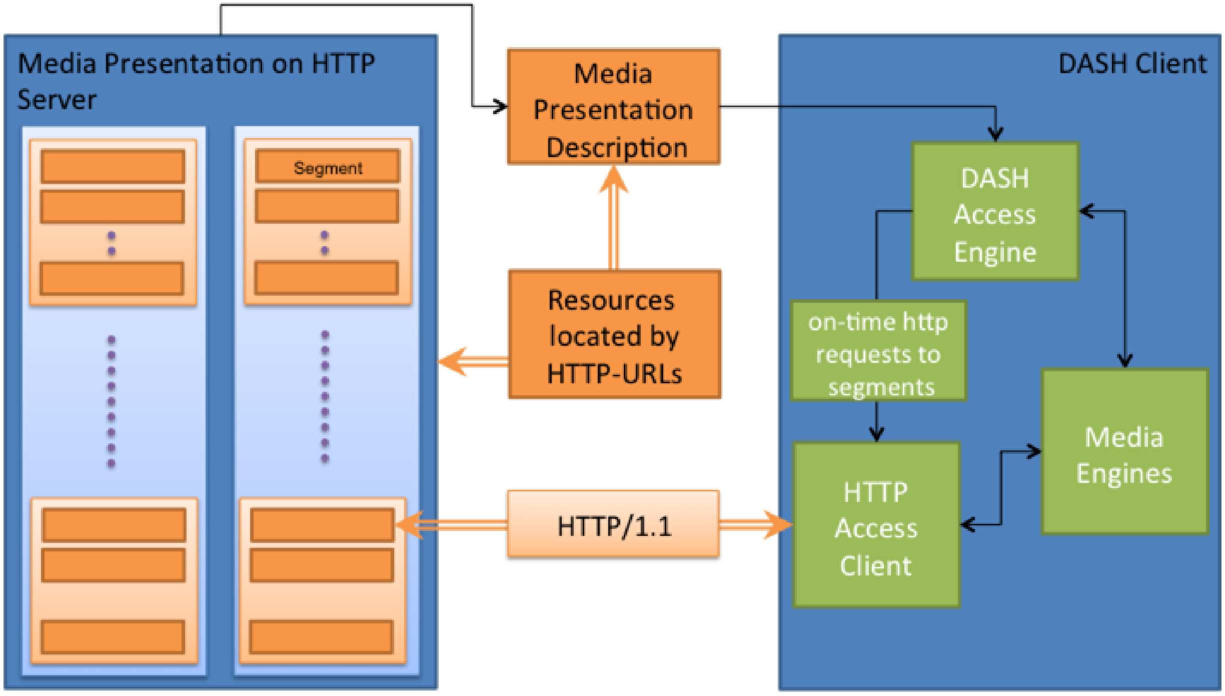

Media Presentation on HTTP Server
Segment
Media Presentation Description
Resources located by HTTP-URLs
HTTP/1.1
DASH Client
DASH Access Engine
on-time http requests to segments
Media Engines
HTTP Access Client

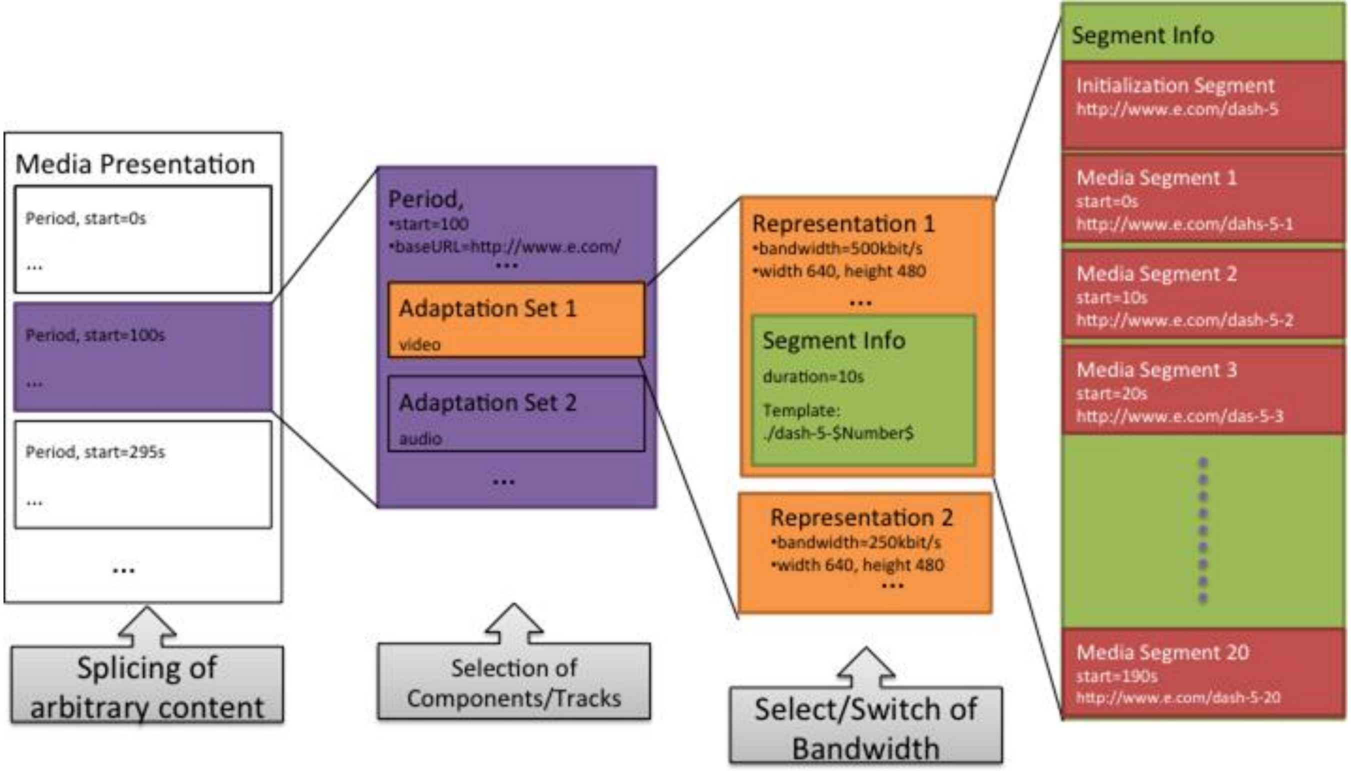
Figure 3 DASH High-level architecture

HTTP 1.1 is used for transfer of individual Segments, or portions thereof, as indicated in Figure 3. The choice of which Segments to retrieve (and when to retrieve them) is determined entirely by the client. The client passes selected Segments to the Media Player application (or framework/library) for display to a user.

[1] SubSegments are also possible, but are not identified by URIs.

4.1 Media Presentation

A Media Presentation is a structured collection of encoded data of a media object (e.g., movie, program). As shown in Figure 4, a Presentation consists of a sequence of one or more Periods. Periods are defined by an interval of time, and the Periods comprising a Presentation must be consecutive and non-overlapping. Each Period contains one or more Adaptation Sets, which in turn contain Representations of the same media content. Adaptation Sets are envisioned to correspond to a particular media component (or a set of multiplexed components) and hold a collection of alternative representations. For example, a video component may have three quality options and a corresponding (and synchronized) audio component may be available in two or six channels. It is expected only a single Representation in an Adaptation Set will be chosen for playback at any instant in time, but clients are free to switch between Representations (and possibly display more than one simultaneously).



Within an Adaptation Set, each Representation contains a sequence of Segments, which encode multimedia data considered to be perceptually equivalent. Segments can be further sub-divided into SubSegments, which are most relevant for switching among Representations or random seek operations. A common method for encryption, if used with DASH, applies to Segments or SubSegments. Multiple Digital Rights Management (DRM) systems may employ common cryptographic processing based on AES, and specified for use with DASH.

At Period boundaries, various coarse-grained operating parameters may be changed, including server location, encoding parameters, or the available Representations of the content. The Period concept has been introduced to support splicing of new content (e.g., insertion of advertisements). Each Period has a start time, relative to start of the Presentation.

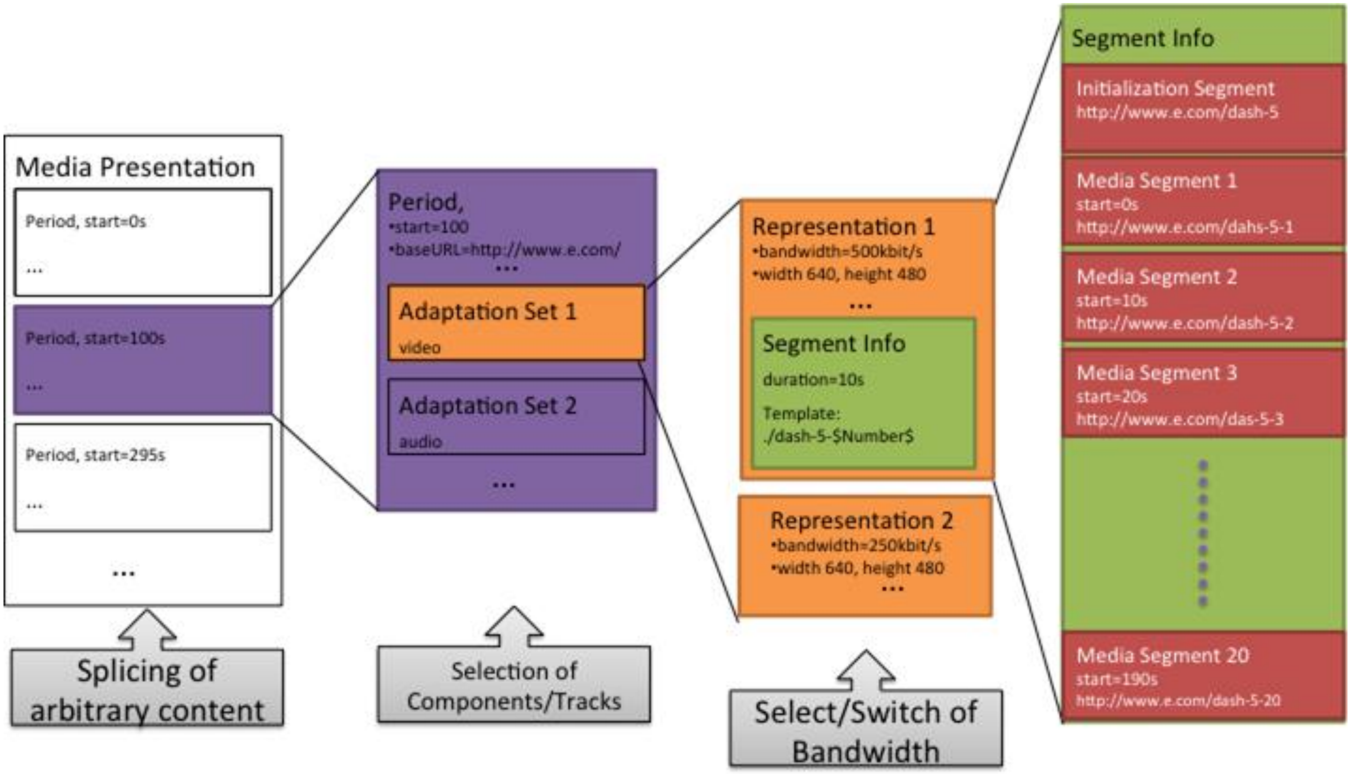


Figure 4 Media Presentation Data Model

Media Segments are typed. An Initialization Segment provides the client with the metadata required to process a number of subsequent Media Segments. Consistent with ISO BMFF File Format [8] [9] terminology, Segments contain a number of data objects called boxes (known as "atoms" in other specifications). An Initialization Segment consists of the "ftyp" (file type) box, the "moov" (movie, all meta-data) box, and optionally the "pdin" (progressive download information) box. A "moov" box, which ordinarily contains media samples, contains only meta-data for Initialization Segments. This arrangement supports a reduced start-up time because the relatively short Initialization Segment is downloaded prior to any Media Segment processing. Note that the DASH standards also provide for an MPEG Transport Stream (TS) alternative to the use of ISO BMFF. However, we focus only on the BMFF use cases, as we believe they will be more common due to the superior efficiency and flexibility of BMFF.

To support random access (e.g., seek operations) and switching, Media Segments typically contain at least one Stream Access Point (SAP). SAPs are pairs of the form (T, B) that are signaled within the MPD (or "sidx" box, as described in the following paragraph), where T is a presentation time within the Representation and B is a byte position within the Segment. The property of a SAP (T, B) is that the media of the Representation starting at time T and continuing to the end of the Period can be played back by downloading the media data from byte position B forward within the Segment (and continuing with subsequent Segments, if any, of the Representation within the Period).

Segment Index ("sidx") boxes may be recovered by downloading only a small prefix of a Media Segment, and are therefore compact and easily processed. They are used as follows:

1. To provide a mapping of each track contained in the Media Segment to the Media Presentation timeline, such that synchronized playback of components within and across Representations is possible.
2. To permit seamless switching across Representations.
3. To enable fast navigation through segments, possibly using byte range requests.
4. To minimize the downloading of media data that is never played back when seeking or switching.
5. To locate the position of SAPs within Segments without downloading unnecessary media data.

4.2 Usage of HTTP

As mentioned, DASH enables content delivery using standard HTTP servers and caches. Each Segment is individually assigned a URI and can be separately retrieved. By using byte range (partial GET) requests, portions of Segments can effectively be retrieved. Clients and servers are expected to comply with the HTTP/1.1 protocol [6].

Especially in live operation, Segments may have a well-defined URI prior to their existence. This presents a challenge, as the client should avoid requesting Segments that do not yet exist. DASH handles this situation by introducing a Segment availability time concept that indicates the UTC time the Segment addressed by a particular URI is available for access.

No further details on HTTP, caches or proxies are specified by DASH. The specifications focus primarily on the MPD and Segment formats and how associated meta-data is handled. Additional (non-normative) material included with the standard describes uses of error and status codes.

4.3 On-Demand and Live Profiles

DASH supports both on-demand and live streaming using profiles that define certain operating parameters. In on-demand, the MPD remains fixed and these constraints are used: each Representation is a single Segment, SubSegments are aligned across Adaptation Sets, and SubSegments must begin with SAPs. With live, the MPD may be updated during a Presentation's lifetime (e.g., with new Periods and Segments). The MPD is a well-formatted XML document based on a DASH-specified XML schema. An example of an on-demand MPD supporting two Representations of protected English audio at 64kb/s and 32kb/s is shown in Figure 5.

```
<?xml version="1.0"?>
<MPD
  xmlns:xsi=
       "http://www.w3.org/2001/XMLSchema-instance"
  xmlns="urn:mpeg:DASH:schema:MPD:2011"
  xsi:schemaLocation=
      "urn:mpeg:DASH:schema:MPD:2011 DASH-MPD.xsd"
  type="static"
  mediaPresentationDuration="PT3256S"
  minBufferTime="PT1.2S"
  profiles=
     "urn:mpeg:dash:profile:isoff-on-demand:2011">
<BaseURL>http://cdn1.example.com/</BaseURL>
<BaseURL>http://cdn2.example.com/</BaseURL>
 <Period>
  <!-- English Audio -->
  <AdaptationSet mimeType="audio/mp4"
  codecs="mp4a.0x40" lang="en"
  subsegmentAlignment="true"
  subsegmentStartsWithSAP="1">
  <ContentProtection schemeIdUri=
   "urn:uuid:706D6953-656C-5244-4D48-656164657221"/>
   <Representation id="1" bandwidth="64000">
    <BaseURL>7657412348.mp4</BaseURL>
   </Representation>
   <Representation id="2" bandwidth="32000">
    <BaseURL>3463646346.mp4</BaseURL>
   </Representation>
  </AdaptationSet>
 </Period>
</MPD>
```

Figure 5 An On-Demand MPD (an XML document)

This MPD (derived from example G.1 of [7]) indicates the meta-data included for a Presentation, including the relevant XML scheme and location, duration, applicable profile, and BaseURL (used as a prefix for all URLs). One Adaptation Set contains two audio Representations for bit rates 32kbps and 64kbps using mp4 audio (mp4a.0x40). This MPD also indicates the name of the content protection method employed.

While there is no specification as to the playback duration of the media in Segments, smaller duration Segments are usually preferable for live scenarios to minimize the interval between when a live event occurs and is displayed by the client. The time to transfer a Segment (or SubSegment, when Segment indexing is used) generally represents a lower bound on startup latency and generally a client tries to pick a Representation that closely matches the current download rate. The contents of segments (e.g., types of codecs used) are not restricted provided the encodings used are compatible with ISO BMFF.

4.4 Common Encryption

DASH is designed with the support of premium content in mind. At present, digital rights management (DRM) systems are required by content owners for display on devices such as smartphones and personal computers. While the various DRM systems have different key management procedures, they all eventually utilize some encryption algorithm.

Among the large collection of options for encryption, the Advanced Encryption Standard (AES) has become most popular. DASH specifies works with a variant of AES as the "common" method of encryption for supporting DRM with DASH. According to [10], AES in counter mode (AES-CTR) is the recommended method.

5. DASH and Other Streaming Technologies

From the servers' points of view, DASH provides flexibility and options that may be optimized to support different delivery and/or user experience goals. For example, DASH has enough flexibility to:

- minimize service access time to ensure that clients need not wait long after initial tune-in or any seeking operations.
- minimize the end-to-end delay in live services, e.g. by the adaptation of segment durations.

- maximize delivery efficiency, e.g. by providing appropriate Adaptation Sets and Representations.
- adjust to CDN properties, e.g., the desired number and sizes of files, the handling of HTTP requests, etc.
- re-use legacy content, e.g., but augmenting content in basic MP4 files with the meta-data and structures needed to take advantage of DASH

DASH's functionality is a superset of proprietary Adaptive HTTP Streaming solutions, in particular Apple HTTP Live Streaming [11] and Microsoft Smooth Streaming [12]. For example, DASH can mimic Apple HTTP Live Streaming using these parameters:

- Segments have a duration of a roughly constant 10 seconds within each Representation
- Each Segment is provided in one fragment so only the first loop in the 'sidx' box of the Segment is provided.
- Each Representation is complete, i.e. audio and video components are multiplexed together in each Segments.
- Playlist-based URIs for lists of Segments with regularly updated MPDs (live case).

DASH can also mimic typical SmoothStreaming scenarios, e.g., by utilizing the following parameters:

- Segments have a duration of a roughly constant 1 or 2 seconds within each Representation
- Each Segment is represented by one fragment, and only the first loop of the 'sidx' is used to provide the Media Presentation time of each Segment.
- Media components are provided in separate Representations with alternatives in same Adaptation Set and complementary components in separate Adaptation Sets.
- A compact template-based method is used for generating URIs used to identify Segments within the MPD/manifest.

Other configurations are based on experience with PD services, especially re-using existing media content, where Segments may be accessed with HTTP partial GET requests, improving bandwidth efficiency and CDN adaptation.

A side-by-side comparison provided by Knowlton in 2011 highlights the characteristics of several adaptive streaming solutions [13], including Microsoft's, Adobe's and Apple's. DASH compares favorably to these, by offering a superset of the functionality of each system, delivered in the form of an international standard. In particular, DASH offers DVR-type recording, compatibility with web and caching infrastructure, multiple options for DRM, advertisement insertion, and variable segment durations.

6. DASH Summary

DASH provides a universal and flexible solution for Adaptive HTTP Streaming. It is the only international standard on this topic, and is based on existing technologies: codecs, encapsulation formats, DRM and delivery protocols. It focuses on the formats required to provide interoperability between various clients and servers using conventional HTTP protocols and infrastructure. Specifically, it includes the syntax and semantics of the MPD, the formats for Segments and the use of URIs and lower layer protocols. However, it also leaves several important factors unconstrained, in order to foster the greatest level of innovation:

- Details of content encoding, e.g.,
  - Size and duration of Segments
  - Number of Representations and associated bit rates
  - Frequency and position of SAPs
  - Choice of multiplexed or individual components
  - Choice and operation of DRM, if used
- Detailed client behavior, e.g.,
  - How and when to download segments
  - How and when to switch among Representations in an Adaptation Set
  - How to use HTTP/1.1 capabilities (e.g., whether request pipelining is used)
- Method for transporting the MPD (HTTP is not required)

7. Qualcomm Stream Manager

As a special value-added capability, Qualcomm implements a DASH client designed for high performance and good quality of experience on SnapDragon-based systems. The Qualcomm Stream Manger (QSM) is responsible for dynamic adaptation by changing Representations based on network conditions and codec capabilities.

Once the MPD is acquired, a client can use DASH through any IP network that enables HTTP connections. The network may be managed or unmanaged, wired or wireless, public or private. The major challenge is to enable an efficient and desirable streaming experience for the user. To do so, the client will need to be adaptive, responding to dynamic conditions such as changes in the network latency or delivery capacity. By careful observations of the data contained in the MPD and Segments of a Presentation, along with observation of the operating environment, an intelligent client is able to issue requests for Segments at the appropriate times, and change Representations at optimal times. This ultimately provides the media player (a presumed separate application) with data used in providing the best possible user experience.

The QSM manages several key aspects of the DASH client access process, including the measurement of network performance, and the selection of Representations among options from an Adaptation Set expressed in the DASH MPD. It also handles the processing of meta-data in order to support random seek operations.

Internally, QSM stores the Representations present in various Adaptation sets, and selects among them using a set of plug-in filters that monitor network performance. QSM is capable of switching between different filter sets, depending on the operating environment and configuration parameters.

QSM is also responsible for deciding the ordering and request sizes for media data and meta-data (e.g., by using byte range requests). When supporting live streaming, for example, QSM is careful to periodically check for MPD updates. It also has to adapt to situations where Segments are not available in a timely fashion. For example, QSM may have to abandon the downloading of a live Presentation Segment whose desired playback time has lapsed.

7.1 A Demonstration of DASH with QSM

The Mobile World Congress (MWC) demonstration shows the Qualcomm DASH implementation contrasted with a stock Android 4.0 ("Ice Cream Sandwich") HLS implementation. Both the DASH & HLS clients receive the same available bandwidth, and access video from the same server. Their performance is compared as video is streamed to the clients, and shows how they react to a series of different bandwidth scenarios. One is a canned trace, designed to provide bandwidth to support each of the available video representations. The other is a realistic HSPA+ field trace, from data collected in Hong Kong.

7.2 DASH/QSM Performance versus HLS

To better understand the relative performance of the Qualcomm DASH client versus alternatives, we compare the selected media representation rate (utilized bandwidth) of the Qualcomm client (QC) versus an HLS client (HC) given the same network environment. The HLS client is the streaming client shipped by default on Android 4.0 (Ice Cream Sanwich). Figure 6 shows the results of a 10-minute run comparing the bandwidth levels of QC and HC as the available network capacity is systematically varied between 500Kbps and 2.8Mbps.

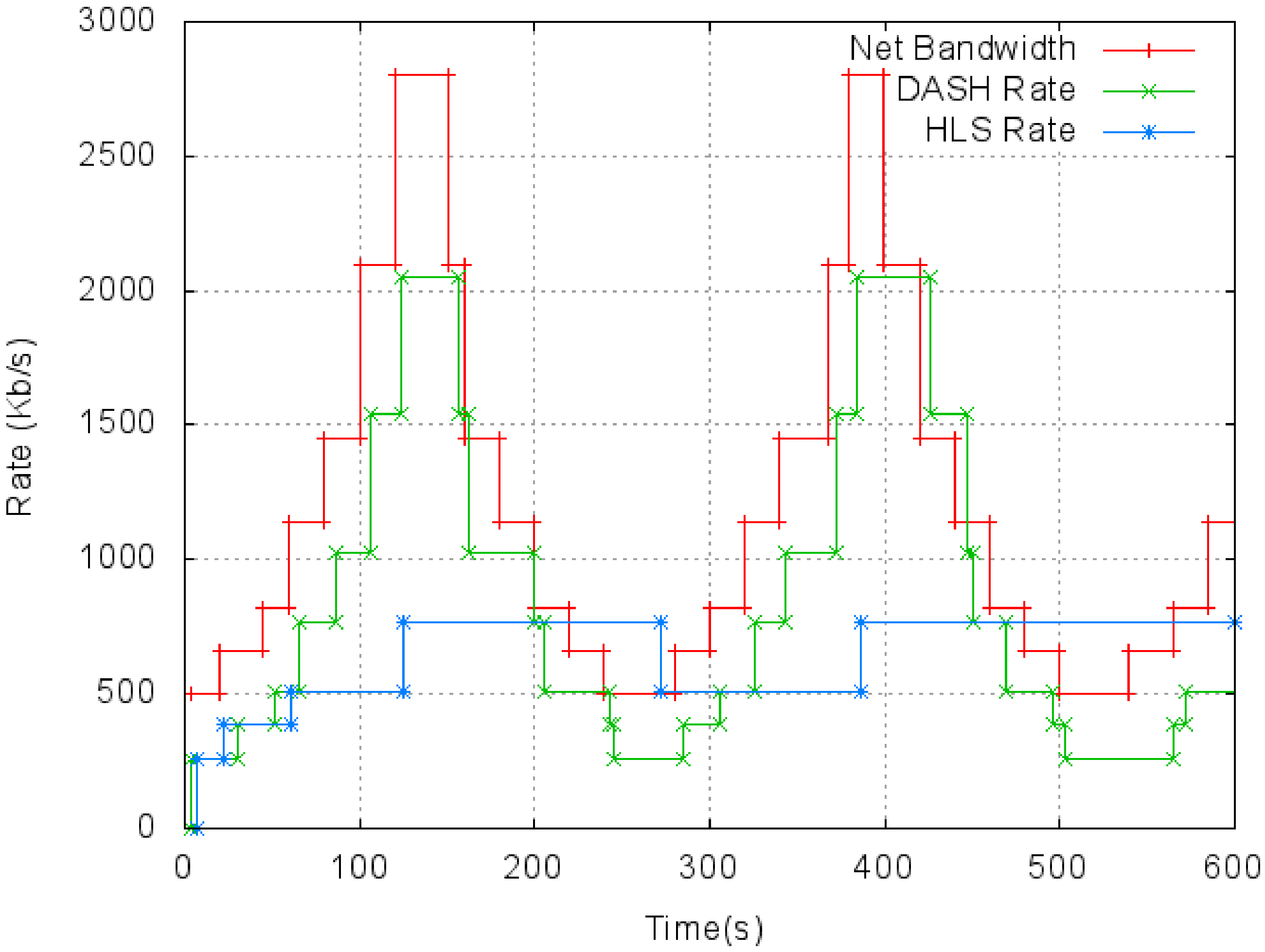


Figure 6 Qualcomm DASH client and HLS performance (controlled network case)

After an initial stabilization phase of about 20 seconds, both HC and QC utilize 250Kbps as the network operates at 500Kbps. As the network bandwidth is increased or decreased 27 times in a stair-step fashion about every 20 seconds, QC adjusts its bandwidth usage with a typical 5-10 second lag, with a total of 26 adjustments. HC makes far fewer adjustments (5), and has an apparent lag of as much as 100 seconds (at time interval [280,380]). In the most extreme case, HC utilizes only 27% of the available rate, whereas QC utilizes 74% of the available rate under the same circumstances.

8. CONCLUSIONS AND OUTLOOK

DASH defines the first standard on Adaptive Streaming over HTTP. Specific design principles have been taken into account that enables flexible deployments when using the formats defined in DASH. Major players in the market, including those that offer proprietary solutions today, have developed the specification.

Qualcomm has developed a DASH client incorporating the Qualcomm Stream Manager, which employs customized algorithms to optimize the user experience on SnapDragon platforms. DASH is flexible enough to replace existing proprietary adaptive streaming systems, and the Qualcomm client's performance meets or exceeds existing alternatives.

DASH also serves as a baseline for several other organizations, including the Open IPTV Forum. MPEG is looking at backward-compatible extensions to integrate additional media including 3D-TV or scalable video coding. Furthermore, initial interoperability testing has started. Currently there is great anticipation that DASH will become the common core package of functions supporting an industry-standard for Dynamic Adaptive Streaming over HTTP.

9. References

[1] Sandvine, "Global Internet Phenomena Spotlight : Netflix Rising," 2011.

[2] Cisco, "Cisco Visual Networking Index: Forecast and Methodology, 2010–2015," 2011.

[3] Cisco. (2012, January) VNI Forecast. [Online]. http://ciscovni.com/vni_forecast/index.htm

[4] A. Rao and R. Lanphier H. Schulzrinne, "Real Time Streaming Protocol (RTSP)," Internet RFC 2326, Apr 1998.

[5] S. Casner, R. Frederick and V. Jacobson H. Schulzrinne, "RTP: A Transport Protocol for Real-Time Applications," Internet RFC 3550, July 2003.

[6] J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach and T. Berners-Lee R. Fielding, "Hypertext Transfer Protocol -- HTTP/1.1," Internet RFC 2616, June 1999.

[7] ISO/IEC, "Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats," 23009-1, 2012.

[8] ISO/IEC, "Information technology — Coding of audio-visual objects — Part 12: ISO base media file format," 15444-12 (technically identical to ISO/IEC 14496-12), 2004.

[9] ISO/IEC, "Information technology — JPEG 2000 image coding system — Part 12: ISO base media file format, AMENDMENT 3: DASH support and RTP reception hint track processing," 15444-12 / AM3, 2011.

[10] ISO/IEC, "Information technolog — MPEG systems technologies — Part 7: Common encryption in ISO base media file format files," 23001-7, 2011.

[11] and W. May R. Pantos (Ed.), "HTTP Live Streaming," IETF draft-panthos-http-live-streaming, work in progress Sep. 2011.

[12] A. Zambelli, "IIS Smooth Streaming Technical Overview," Mar. 2009.

[13] C. Knowlton. (2011, June) Adaptive Streaming Comparison. [Online]. http://learn.iis.net/page.aspx/792/adaptive-streaming-comparison

[14] 3GPP, "Transparent end-to-end packet switched streaming service (PSS); 3GPP file format (3GP)," 26.244 (Release 6),.