Dynamic Adaptive Streaming over HTTP – Standards and Design Principles

Thomas Stockhammer
Qualcomm Incorporated
c/o Nomor Research
Brecherspitzstraße 8
81541 Munich, Germany
+49 89 978980 02
stockhammer@nomor.de

ABSTRACT
In this paper, we provide some insight and background into the Dynamic Adaptive Streaming over HTTP (DASH) specifications as available from 3GPP and in draft version also from MPEG. Specifically, the 3GPP version provides a normative description of a Media Presentation, the formats of a Segment, and the delivery protocol. In addition, it adds an informative description on how a DASH Client may use the provided information to establish a streaming service for the user. The solution supports different service types (e.g., On-Demand, Live, Time-Shift Viewing), different features (e.g., adaptive bitrate switching, multiple language support, ad insertion, trick modes, DRM) and different deployment options. Design principles and examples are provided.

Categories and Subject Descriptors
H.4.m [**Information Systems Applications**]: Miscellaneous.

General Terms
Standardization.

Keywords
3GPP, video, mobile video, standards, streaming.

1. INTRODUCTION
Internet access is becoming a commodity on mobile devices. With the recent popularity of smart phones, smartbooks, connected netbooks and laptops the Mobile Internet use is dramatically expanding. According to recent studies [7], expectations are that between 2009 and 2014 the mobile data traffic will grow by a factor of 40, i.e., it will more than double every year. Figure 1 shows that the video traffic will by then account for 66% of the total amount of the mobile data. At the same time mobile users expect high-quality video experience in terms of video quality, start-up time, reactivity to user interaction, trick mode support, etc., and the whole ecosystem including content providers, network operators, service providers, device manufacturers and technology providers need to ensure that these demands can be met. Affordable and mature technologies are required to fulfil the users' quality expectations. One step into this direction is a common, efficient and flexible distribution platform that scales to the rising demands. Standardized components are expected to support the creation of such common distribution platforms.

MMSys'11, February 23–25, 2011, San Jose, California, USA.

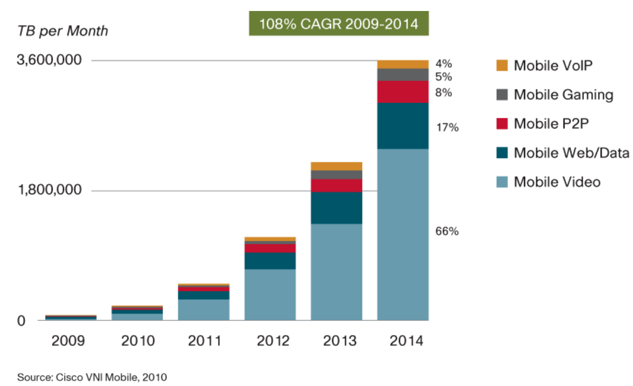

Figure 1 Video Will Account for 66 Percent of Global Mobile Data Traffic by 2014 (Source [7], Figure 2)

Traditional streaming generally uses a stateful protocol, e.g., the Real-Time Streaming Protocol (RTSP): Once a client connects to the streaming server the server keeps track of the client's state until the client disconnects again. Typically, frequent communication between the client and the server happens. Once a session between the client and the server has been established, the server sends the media as a continuous stream of packets over either UDP or TCP transport. In contrast, HTTP is stateless. If an HTTP client requests some data, the server responds by sending the data and the transaction is terminated. Each HTTP request is handled as a completely standalone one-time transaction.

Alternatively to streaming, progressive download may be used for media delivery from standard HTTP Web servers. Clients that support HTTP can seek to positions in the media file by performing byte range requests to the Web server (assuming that it also supports HTTP/1.1 [4]). Disadvantages of progressive download are mostly that (i) bandwidth may be wasted if the user decides to stop watching the content after progressive download has started (e.g., switching to another content), (ii) it is not really bitrate adaptive and (iii) it does not support live media services. Dynamic Adaptive Streaming over HTTP (DASH) addresses the weaknesses of RTP/RTSP-based streaming and progressive download.

2. DESIGN PRINCIPLES

HTTP-based progressive download does have significant market adoption. Therefore, HTTP-based streaming should be as closely aligned to HTTP-based progressive download as possible, but take into account the above-mentioned deficiencies.

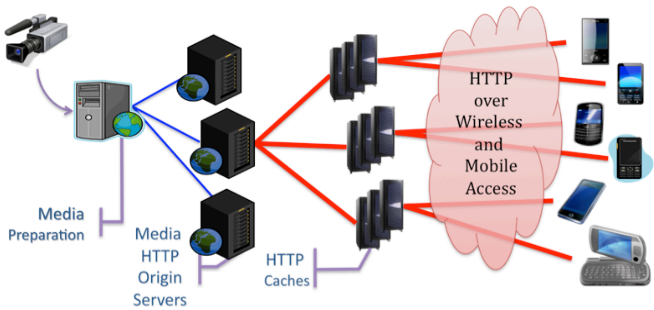

Figure 2 Example Media Distribution Architecture

Figure 2 shows a possible media distribution architecture for HTTP-based streaming. The media preparation process typically generates segments that contain different encoded versions of one or several of the media components of the media content. The segments are then hosted on one or several media origin servers typically, along with the media presentation description (MPD). The media origin server is preferably an HTTP server such that any communication with the server is HTTP-based (indicated by a bold line in the picture). Based on this MPD metadata information that describes the relation of the segments and how they form a media presentation, clients request the segments using HTTP GET or partial GET methods. The client fully controls the streaming session, i.e., it manages the on-time request and smooth playout of the sequence of segments, potentially adjusting bitrates or other attributes, for example to react to changes of the device state or the user preferences.

Massively scalable media distribution requires the availability of server farms to handle the connections to all individual clients. HTTP-based Content Distribution Networks (CDNs) have successfully been used to serve Web pages, offloading origin servers and reducing download latency. 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 base to launch streaming services that build on this existing infrastructure. This can reduce capital and operational expenses, and reduces or eliminates decisions about resource provisioning on the nodes. This principle is indicated in Figure 2 by the intermediate HTTP servers/caches/proxies. Scalability, reliability, and proximity to the user's location and high-availability are provided by general-purpose servers. The reasons that lead to the choice of HTTP as the delivery protocol for streaming services are summarized below:

1. HTTP streaming is spreading widely as a form of delivery of Internet video.
2. There is a clear trend towards using HTTP as the main protocol for multimedia delivery over the Open Internet.
3. HTTP-based delivery enables easy and effortless streaming services by avoiding NAT and firewall traversal issues.
4. HTTP-based delivery provides reliability and deployment simplicity due as HTTP and the underlying TCP/IP protocol are widely implemented and deployed.
5. HTTP-based delivery provides the ability to use standard HTTP servers and standard HTTP caches (or cheap servers in general) to deliver the content, so that it can be delivered from a CDN or any other standard server farm.
6. HTTP-based delivery provides the ability to move control of "streaming session" entirely to the client. The client basically only opens one or several or many TCP connections to one or several standard HTTP servers or caches.
7. HTTP-based delivery provides the ability to the client to automatically choose initial content rate to match initial available bandwidth without requiring the negotiation with the streaming server.
8. HTTP-based delivery provides a simple means to seamlessly change content rate on-the-fly in reaction to changes in available bandwidth, within a given content or service, without requiring negotiation with the streaming server.
9. HTTP-based streaming has the potential to accelerate fixed-mobile convergence of video streaming services as HTTP-based CDN can be used as a common delivery platform.

Based on these considerations, 3GPP had identified the needs to provide a specification for a scalable and flexible video distribution solution that addresses mobile networks, but is not restricted to 3GPP radio access networks (RANs). 3GPP has taken the initiative to specify an Adaptive HTTP Streaming solution in addition to the already existing RTP/RTSP-based streaming solutions and the HTTP-based progressive download solution.

Specifically the solution is designed

- to support delivery of media components encapsulated in ISO base media file format box structure,
- to address delivery whereas presentation, annotation and user interaction is largely out-of-scope,
- to permit integration in different presentation frameworks.

The 3GPP sub-group SA4 working on codecs and protocols for media delivery started the HTTP streaming activity in April 2009 and completed the Release-9 specification work early March 2010. The 3GPP Adaptive HTTP Streaming (AHS) has been integrated into 3GPP Transparent end-to-end Packet-switched Streaming Service (PSS). Specifically, 3GPP TS 26.234 [1] (PSS Codecs and Protocols) clause 12 specifies the 3GPP Adaptive HTTP Streaming solution, and 3GPP TS 26.244 [2] (3GP File Format) clauses 5.4.9, 5.4.10, and 13 specify the encapsulation formats for segments. The Release-9 work is now under maintenance mode and some minor bug fixes and clarifications were agreed during the year 2010 and have been integrated into the latest versions of 3GPP TS 26.234 and 3GPP TS 26.244.

The solution supports features such as

- fast initial startup and seeking,
- bandwidth-efficiency,
- adaptive bitrate switching,
- adaptation to CDN properties,
- re-use of HTTP-server and caches,
- re-use of existing media playout engines,
- support for on-demand, live and time-shift delivery services,
- simplicity for broad adoption.

3GPP has also sought alignment with other organizations and industry fora that work in the area of video distribution. For example, as the Open IPTV Forum (OIPF) based their HTTP Adaptive Streaming (HAS) solution [13] on 3GPP. 3GPP recently also addressed certain OIPF requirements and integrated appropriate features in the Release-9 3GPP Adaptive HTTP Streaming specification. Also MPEG's draft DASH solution is heavily based on 3GPP's AHS. Finally, 3GPP has ongoing work in Release-10, now also referred to as DASH. This work will extend the Release-9 3GPP AHS specification in a backward-compatible way. Close coordination with the ongoing MPEG DASH activities is organized.

3. 3GPP Adaptive HTTP Streaming

3.1 Overview

3GPP Adaptive HTTP Streaming, since Release-10 referred to as as 3GP-DASH, is the result of a standardization activity in 3GPP SA4 Figure 3 shows the principle of the 3GP-DASH specification. The specification provides

- a normative definition of a Media Presentation, with Media Presentation defined as a structured collection of data that is accessible to the DASH Client through Media Presentation Description,
- a normative definition of the formats of a Segment, with a Segment defined as an integral data unit of a media presentation that can be uniquely referenced by a HTTP-URL (possibly restricted by a byte range),
- a normative definition of the delivery protocol used for the delivery of Segments, namely HTTP/1.1,
- an informative description on how a DASH client may use the provided information to establish a streaming service for the user.

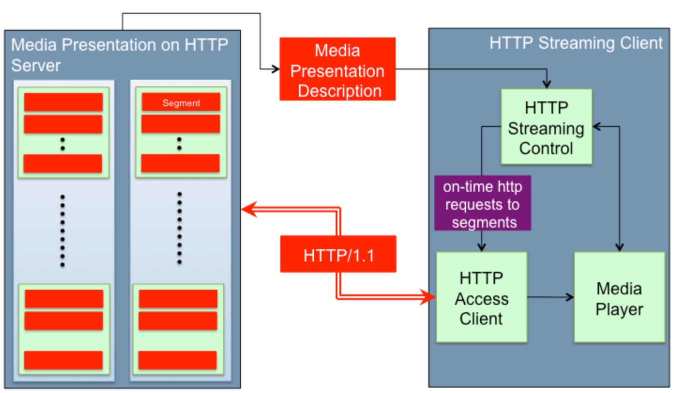

Figure 3 Solution overview – 3GP-DASH

DASH in 3GPP is defined in two levels:

1. Clause 12.2 in TS 26.234 [1] provides a generic framework for Dynamic Adaptive Streaming independent of the data encapsulation format for media segments.
2. Clause 12.4 in TS 26.234 [1] provides a specific instantiation of this framework with the 3GP/ISO base media file format by specifying the segment formats, partly referring to the formats in TS 26.244 [2].

This approach makes the framework defined in 3GPP extensible, for example to any other segment formats, codecs and DRM solutions.

3G-DASH supports multiple services, among others:

- On-demand streaming,
- Linear TV including live media broadcast,
- Time-shift viewing with network Personal Video Recording (PVR) functionalities.

Specific care was taken in the design that the network side can be deployed on standard HTTP servers and distribution can be provided through regular Web infrastructures such as HTTP-based CDNs. The specification also leaves room for different server/network-side deployment options as well as for optimized client implementations.

The specification also defines provisions to support features such as

- Initial selection of client- and/or user-specific representations of the content,
- Dynamic adaptation of the played content to react to environmental changes such as access bandwidth or processing power,
- Trick modes such as seeking, fast forward or rewind,
- Simple insertion of pre-encoded advertisement or other content in on-demand and live streaming services,
- Efficient delivery of multiple languages and audio tracks,
- Content protection and content security, etc.

The remainder of this section provides further background information on the concept of a Media Presentation, the usage of HTTP, as well as segment types and formats in the 3GPP instantiation. A summary of the normative specification is also provided.

3.2 Media Presentation

The concept of a Media Presentation is introduced in TS 26.234 [1], clause 12.2. A Media Presentation is a structured collection of encoded data of some media content, e.g., a movie or a program. The data is accessible to the DASH Client to provide a streaming service to the user. As shown in Figure 4:

- A Media Presentation consists of a sequence of one or more consecutive non-overlapping Periods.
- Each Period contains one or more Representations from the same media content.
- Each Representation consists of one or more Segments.
- Segments contain media data and/or metadata to decode and present the included media content.

Period boundaries permit to change a significant amount of information within a Media Presentation such as server location, encoding parameters, or the available variants of the content. The Period concept has been introduced among others for splicing of new content, such as ads, and for logical content segmentation. Each Period is assigned a start time, relative to start of the Media Presentation.

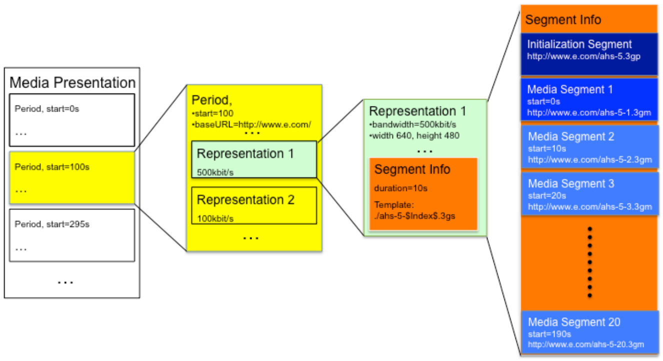

Figure 4 Media Presentation Data Model

Each Period itself consists of one or more Representations. A Representation is one of the alternative choices of the media content or a subset thereof typically differing by the encoding choice, e.g., by bitrate, resolution, language, or codecs.

Each Representation includes one or more media components, where each media component is an encoded version of one individual media type such as audio, video or timed text. Each Representation is assigned to a group. Representations in the same group are alternatives to each other. The media content within one Period is represented by either one Representation from group zero, or the combination of at most one Representation from each non-zero group.

A Representation consists of at most one Initialisation Segment and one or more Media Segments. Media components are time-continuous across boundaries of consecutive Media Segments within one Representation. Segments represent a unit that can be uniquely referenced by an HTTP-URL (possibly restricted by a byte range). Thereby, the Initialisation Segment contains information for accessing the Representation, but no media data. Media Segments contain media data and must fulfil some further requirements, namely:

- Each Media Segment is assigned a start time in the media presentation to enable downloading the appropriate Segments in regular play-out mode or after seeking. This time is generally not accurate media playback time, but only approximate such that the client can make appropriate decisions on when to download the Segment such that it is available in time for play-out.
- Media Segments may provide random access information, i.e., presence, location and timing of Random Access Points (RAPs).
- A Media Segment, when considered in conjunction with the information and structure of the MPD, contains sufficient information to time-accurately present each contained media component in the Representation without accessing any previous Media Segment in this Representation provided that the Media Segment contains a RAP. The time-accuracy enables seamlessly switching Representations and jointly presenting multiple Representations.
- Media segments may also contain information for randomly accessing subsets of the Segment by using partial HTTP GET requests.

A Media Presentation is described in a Media Presentation Description (MPD), and MPDs may be updated during the lifetime of a Media Presentation. In particular, the MPD describes accessible Segments and their timing. The MPD is a well-formatted XML document and the 3GPP Adaptive HTTP Streaming specification defines an XML schema to define MPDs. An MPD may be updated in specific ways such that an update is consistent with the previous instance of the MPD for any past media. A graphical presentation of the XML schema is provided in Figure 5. The mapping of the data model to the XML schema is highlighted. For the details of the individual attributes and elements please refer to TS 26.234 [1], clause 12.2.5.

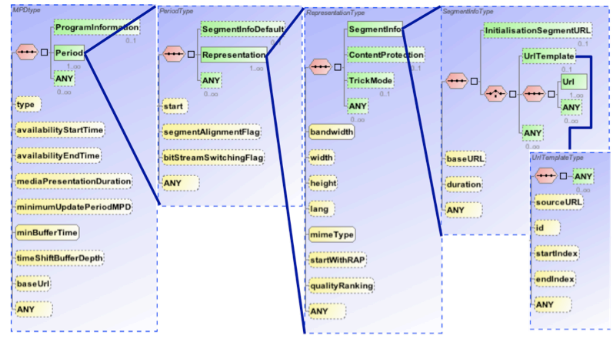

Figure 5 MPD XML-Schema

DASH also supports live streaming services. In this case, the generation of segments typically happens on-the-fly. Due to this clients typically have access to only a subset of the Segments, i.e., the most recent MPD describes a time window of accessible Segments for this instant in time. By providing updates of the MPD, the server may describe new Segments and/or new Periods such that the updated MPD is compatible with the previous MPD.

Therefore, for live streaming services a Media Presentation is typically described by the initial MPD and all MPD updates. To ensure synchronization between client and server, the MPD provides access information in Universal Time Clock (UTC) time. As long as server and client are synchronized to UTC time, the synchronization between server and client can be ensured by the use of the UTC times in the MPD.

Time-shift viewing and network PVR functionality are supported in a straightforward manner, as segments may be accessible on the network over a long period of time.

3.3 Usage of HTTP

The 3GPP DASH specification is written such that it enables delivering content from standard HTTP servers to an HTTP-Streaming client and enables caching content by standard HTTP caches. Therefore, the streaming server and streaming client comply to HTTP/1.1 as specified in RFC2616 [7] and HTTP-Streaming Clients are expected to use the HTTP GET method or the partial GET method for downloading media segments. No further details on caches and proxies are specified, as they are transparent to protocol.

3.4 Segments based on 3GP File Format

Beyond the general adaptive streaming framework, 3GPP DASH specifies an instantiation that uses segment formats based on the 3GP file format as specified in TS 26.244 [2]. Each Representation may either consist of

- one Initialisation Segment and at least one Media Segment, but typically a sequence of Media Segments, or
- one self-Initialising Media Segment.

An *Initialisation Segment* provides the client with the metadata that describes the media content and is basically a file conformant with the 3GPP file format without any media data. An Initialisation Segment consists of the "ftyp" box, the "moov" box, and optionally the "pdin" box. The "moov" box contains no samples. This reduces the start-up time significantly as the Initialisation Segment needs to be downloaded before any Media Segment can be processed, but may be downloaded asynchronously before the Media Segments.

Self-Initialising Media Segments comply with the 3GP Adaptive-Streaming Profile as specified in TS 26.244, clause 5.4.9. For 3GP files conforming to this profile the 'moov' box is in the beginning of the file after the 'ftyp' and a possibly present 'pdin' box, all movie data is contained in Movie Fragments, and the 'moov' box is followed by one or more 'moof' and 'mdat' box pairs. In addition 3GP files conforming to this profile may contain any of the new boxes specified in TS 26.244 [2], clause 13, namely the segment type ('styp') box, the track fragment adjustment ('tfad') box and the segment index ('sidx') box. Self-Initialising media segments are assigned start time 0 relative to the Period start time, so no additional information is necessary for each segment. However, the additional boxes 'tfad' and 'sidx' may be used for accurate timing of each component within the 3GP file after random access and seeking within the 3GP file. More details on the boxes for DASH are provided further below.

A *Media Segment* may start with a 'styp' box and a sequence of one or more whole self-contained movie fragments. The 'styp' box is used for file branding of segments. In addition, each 'traf' box may contain a 'tfad' box for track alignment to permit random access to the start of the segment or any fragment within the segment. Furthermore, each Media Segment may contain one or more 'sidx' boxes. The 'sidx' provides global timing for each contained track, time/byte range offsets of the contained movie fragments, as well as time offsets of random access points, if any.

Note that the codecs in 3GPP AHS are identical to 3GPP PSS codecs as specified in TS 26.234, clause 7.2 for speech, 7.3 for audio, 7.4 for video and 7.9 for timed text. However, there is no restriction for use of any other codecs as long as the codecs can be encapsulated in ISO base media file format.

3.5 Segment Indexing

Segment Indexing is an important concept to permit byte range access to subsets of segments. This permits fast access to substructures in the segment for fast switching, simple random access, etc. Each segment index 'sidx' box documents a subsegment defined as one or more consecutive movie fragments, ending either at the end of the containing segment, or at the beginning of a subsegment documented by the next 'sidx'.

The 'sidx' contains timing information to place the segment into the global time line of the media presentation. Beyond, it contains a loop that provides an index of the subsegment, i.e., the duration of each sub-segment and the offset from the first byte following the enclosing 'sidx', to the first byte of the referenced box.

By downloading only a small portion in the beginning of the media segment, e.g., by using a byte range request, the segment index boxes may be fetched. Segment index boxes may be used for several purposes:

1. To provide a mapping of each track contained in the media segment to the media presentation timeline, such that synchronous playout of media components within and across Representations is enabled as well as to permit switching across Representations.
2. To enable fast navigation through segments, possibly using byte range requests and to minimize the download of media data during a seeking process.
3. To locate the position of random access points within segments without downloading unnecessary media data before the random access point.

Figure 6 shows a simple example of a single segment index. In this case, the first loop of the segment index box provides the exact timing for each media component starting in fragment F1. Furthermore, the second loop provides time-byte offset information as well as random access information for selected fragments, namely fragment F1, F3, and F5.

Figure 6 A Simple Example of Segment Index (Legend: S, in yellow is a segment index box; F, in blue, is a movie fragment with its data; The arrows, in red above, shows the fragment documented by the first loop; Blue arrows below show the second-loop time-byte index pointers)

As Segments may be of very different size, the first 'sidx' box may or may not describe all details of the following movie fragments in this segment. To avoid large 'sidx' boxes in case of large segments, the information may be provided in a nested manner, in such a way that 'sidx' boxes may reference not only the start of a movie fragment, but also other 'sidx' boxes.

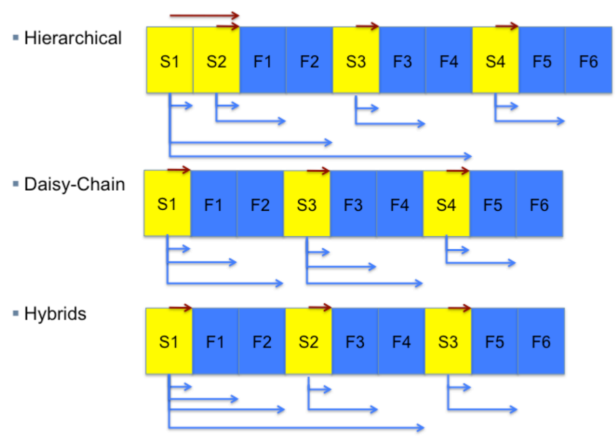

Figure 7 Nested Segment Indices (same legend as in Figure 6)

Several examples for nested segment indices are provided in Figure 7. In the 'hierarchical' case, the first loop for S1 points to the first movie fragment, but the first pointer in the second loop for S1 references a 'sidx' box. Also, any other pointer in the second loop for S1 references a 'sidx' box. Only the second level (S2/3/4) points directly to movie fragments. Therefore, with downloading the 'sidx' box S1, fast coarse navigation through the segment is enabled. Further refinements are done in the lower level. In the daisy chain case, S1 and S3 reference both movie fragments and other 'sidx' boxes: Such a construction enables fast navigation to the initial fragments of the segments, whereas later fragments may require some sequential resolution. The syntax is flexible enough to also support any hybrids of strictly hierarchical and daisy-chain constructions.

3.6 Summary

The 3GP-DASH specification provides a universal and flexible solution for Adaptive HTTP Streaming. The solution is based on existing technologies, including codecs, encapsulation formats, content protection and delivery protocols. 3GP-DASH focuses on the specification of the interface between standard HTTP servers that host media presentations and the HTTP Streaming client. Specifically, 3GPP AHS specifies

- syntax and semantics of Media Presentation Description,
- format of Segments, and
- the delivery protocol for segments.

3GP-DASH also provides an informative overview on how a DASH Client may use the provided information to establish a streaming service to the user.

However, 3GP DASH permits flexible configurations to address different use cases and delivery scenarios. Among others, 3GP-DASH does not specify

- Details on content provisioning, for example
    - Size and duration of segments can be selected flexibly and individually for each Representation,
    - Number of Representations and the associated bitrates can be selected based on the content requirements,
    - Frequency and position of random access points are not restricted by the streaming solution,
    - Other attributes and encoding parameters of each Representation, etc, are not restricted.
    - Multiplexed components and individual component may be used.
- Normative client behaviour to provide streaming service, e.g.
    - Prescriptions of how and when to download segments
    - Representation selection and switching procedures among different Representations
    - Usage of HTTP/1.1 on how to download segments, etc
- The transport of MPD, while being possible through HTTP, may also be delivered by other means.

To emphasize the flexibility in terms of use cases and deployment options, some example deployments are provided in section 4.

4. DEPLOYMENT OPTIONS

3GP-DASH provides a significant amount of options and flexibility for deploying Adaptive HTTP Streaming services on both, the service provisioning end as well as the client side. Figure 8 shows a possible deployment architecture. Content preparation is done offline and ingested into an HTTP-Web serving cloud with origin and cache servers. The ingestion tool may adapt to specifics of CDNs, for example adapt the segment duration/size to CDN properties, use load-balancing and geo-location information, etc. By providing access to the MPD, the access client can access the streaming service through any IP network that enables HTTP connections. The network may be managed or unmanaged, wired or wireless, and multiple access networks may even be used in parallel. The major intelligence to enable an efficient and high-quality streaming service is in the HTTP Streaming client. With the access to the MPD, the client is able to issue requests to segments at the appropriate times and provide the media player with data for best user experience.

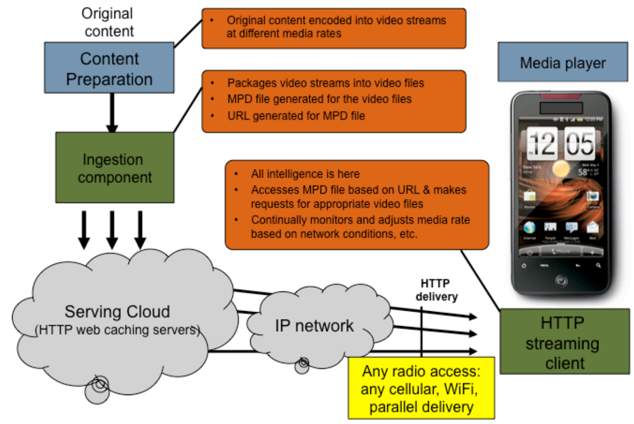

Figure 8 Example Deployment Architecture

3GP-DASH provides flexibility and options such that on the content preparation and ingestion side, the service may be optimized to support different delivery and/or user experience aspects, such as

- minimization of service access time, i.e., to ensure that the client have fast access after tune-in and after any seeking operations.
- minimization of the end-to-end delay in live services, e.g., by the adaptation of segment durations.
- maximization of delivery efficiency, e.g., by ensuring that the client has options for close-to-playout time downloading, or by providing appropriate bandwidth Representations, etc.
- adjustment to CDN properties, for example the desired file sizes, the amount of files, the handling of HTTP requests, etc.
- the reuse of encoded legacy content, for example media stored in MP4 files, media encoded with/without coordination between encoders of different bitrates, coordinated random access point placement, etc.

The 3GP-DASH solution includes some of the design options from proprietary Adaptive HTTP Streaming solutions, in particular Apple HTTP Live Streaming [8] and Microsoft Smooth Streaming [10]. For example, Apple HTTP Live Streaming configuration may typically be mapped to

- Constant segment duration of roughly 10 seconds for each Representation,
- Each segment is provided in one fragment so only the first loop in the 'sidx' box of the segment may be provided,
- Each Representation is complete and assigned to group 0, i.e., typically audio and video are multiplexed within a single segments,
- Playlist-based segment lists with regularly updated MPDs to address the live generation and publishing of segments,

Typical MS SmoothStreaming deployments may be mapped to the 3GPP AHS specification by providing

- constant segment size of roughly 1 or 2 seconds for each Representation,

- each Segment is represented by one movie 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 group and complementary components in separate groups,
- a template-based segment list generation is applied to support compact MPD/manifest representation.

Additional design principles results from from implementation experience of progressive download services, especially the reuse of existing media content, or DFSplash [9] for which segments may be easily accessed with HTTP partial GET requests for optimized user experience, bandwidth efficiency and CDN adaptation.

Table 1 provides a comparison of adaptive streaming solutions based on a collection in [11]. 3GP-DASH is added to the row. It is obvious that the flexibility of 3GP-DASH can address the features of proprietary adaptive streaming solutions.

Table 1 Adaptive Streaming Comparison, based on collection in [11]

Feature	MS IIS [10]	Apple [8]	3GP- DASH
On-Demand & Live	Yes	yes	Yes
Live DVR	Yes	no	Yes
Delivery Protocol	HTTP	HTTP	HTTP
Scalability via HTTP Edge Caches	Yes	yes	Yes
Origin Server	MS IIS	HTTP	HTTP
Stateless Server Connection	Yes	yes	yes
Media Container	MP4	MP2 TS	3GP/MP4
DRM Support for Live and VOD	PlayReady	no	OMA DRM
Add Insertion Support	Yes	no	yes
Supported Video Codecs	Agnostic	H.264 BL	Agnostic
Default Segment Duration	2 sec	10 sec	flexible
End-to-End Latency	>1.5sec	30sec	flexible
File Type on Server	contiguous	fragmented	both
3GPP Adaptation	No	No	in work
Specification	proprietary	proprietary	standard

A few service examples are provided in section 5.

5. SERVICE EXAMPLES

5.1 On-Demand Adaptive Streaming Service

Assume that a streaming service provider offers a popular piece of content as on-demand streaming. The content has duration of around 1 hour and 30 minutes and one video and one audio component. The on-demand content is accessible for a long time. The service is deployed on a CDN that is optimized to file sizes of around 2MByte.

The media content is offered for different access bitrates with video resolution 320x240 with H.264/MPEG AVC baseline profile with level 1.3 and low-complexity AAC audio in a 3GP container. The service offering should permit seeking and fast forward. It is especially important to support fast start-up at the beginning of the service or after seeking.

An example for a valid MPD for this service offering is provided below:


```
<?xml version="1.0" encoding="UTF-8"?>
<MPD
 type="OnDemand"
 mediaPresentationDuration="PT1H27M48.2S"
 minBufferTime="PT3S"         availabilityStartTime="1971-03-
06T23:36:47+01:00"
 availabilityEndTime="2051-03-06T09:30:47Z"

xsi:schemaLocation="urn:3GPP:ns:PSS:AdaptiveHTTPStreamingMPD:
2009 3GPP-MPD.xsd"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns="urn:3GPP:ns:PSS:AdaptiveHTTPStreamingMPD:2009">
 <ProgramInformation                                moreInfor-
mationURL="http://www.example.com">
  <Title>Example 2: Adaptive On-Demand Streaming</Title>
 </ProgramInformation>
 <Period start="PT0S">
  <SegmentInfoDefault                  sourceUrlTemplatePeri-
od="http://www.example.com/rep-$RepresentationID$/seg-
$Index$.3gs"/>
  <Representation  mimeType='video/3gpp;  codecs="avc1.42E00C,
mp4a.40.2"' bandwidth="96000" width="320" height="240">
   <SegmentInfo duration="PT160S">
    <InitialisationSegmentURL
sourceURL="http://www.example.com/rep-96/seg-init.3gs"/>
    <UrlTemplate id="96"/>
   </SegmentInfo>
   <TrickMode alternatePlayoutRate="8.0"/>
  </Representation>
  <Representation  mimeType='video/3gpp;  codecs="avc1.42E00C,
mp4a.40.2"' bandwidth="192000" width="320" height="240">
   <SegmentInfo duration="PT80S">
    <InitialisationSegmentURL
sourceURL="http://www.example.com/rep-192/seg-init.3gs"/>
    <UrlTemplate id="192"/>
   </SegmentInfo>
  </Representation>
  <Representation  mimeType='video/3gpp;  codecs="avc1.42E00C,
mp4a.40.2"' bandwidth="256000" width="320" height="240">
   <SegmentInfo duration="PT62.5S">
    <InitialisationSegmentURL
sourceURL="http://www.example.com/rep-256/seg-init.3gs"/>
    <UrlTemplate id="256"/>
   </SegmentInfo>
  </Representation>
  <Representation  mimeType='video/3gpp;  codecs="avc1.42E00C,
mp4a.40.2"' bandwidth="320000" width="320" height="240">
   <SegmentInfo duration="PT50S">
    <InitialisationSegmentURL
sourceURL="http://www.example.com/rep-320/seg-init.3gs"/>
     <UrlTemplate id="320"/>
   </SegmentInfo>
  </Representation>
  <Representation  mimeType='video/3gpp;  codecs="avc1.42E00C,
mp4a.40.2"' bandwidth="384000" width="320" height="240">
   <SegmentInfo duration="PT40S">
    <InitialisationSegmentURL
sourceURL="http://www.example.com/rep-384/seg-init.3gs"/>
    <UrlTemplate id="384"/>
   </SegmentInfo>
  </Representation>
 </Period>
</MPD>
```


The Segments for each representation are encoded according to the Initialisation Segment and Media Segment Format of 3GP DASH. The ‘sidx’ is provided to permit accurate seeking by the use of byte range requests. Fragments within the Media Segments are typically of 1 second duration, but not each fragment starts at a RAP.

A typical message flow is shown in Figure 9. It is assumed that the client has access to the MPD. Then according to the client behavior specified in [1] the client may parse the MPD and may create a list of accessible Segments for each Representation as defined in Table 2 - **Table 6**.

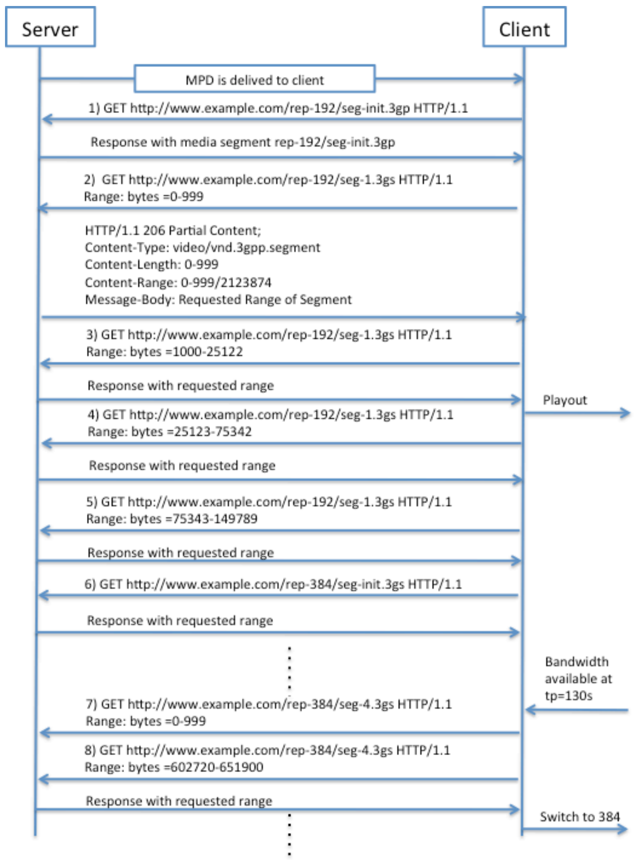

Figure 9 Message flow for Adaptive On-Demand Service

Table 2 Segment list for Representation with Id=96

Index	Start Time	URL
1	0s	http://www.example.com/rep-96/seg-1.3gs
2	160s	http://www.example.com/rep-96/seg-2.3gs
3	320s	http://www.example.com/rep-96/seg-3.3gs
…	…	….
33	5120s	http://www.example.com/rep-96/seg-33.3gs

Table 3 Segment list for Representation with Id =192

Index	Start Time	URL
1	0s	http://www.example.com/rep-192/seg-1.3gs
2	80s	http://www.example.com/rep-192/seg-2.3gs
3	160s	http://www.example.com/rep-192/seg-3.3gs
…	…	….
66	5200s	http://www.example.com/rep-192/seg-66.3gs

Table 4 Segment list for Representation with Id =256

Index	Start Time	URL
1	0s	http://www.example.com/rep-256/seg-1.3gs
2	62.5s	http://www.example.com/rep-256/seg-2.3gs
3	125s	http://www.example.com/rep-256/seg-3.3gs
…	…	….
85	5250s	http://www.example.com/rep-256/seg-85.3gs

Table 5 Segment list for Representation with Id=320

Index	Start Time	URL
1	0s	http://www.example.com/rep-320/seg-1.3gs
2	160s	http://www.example.com/rep-320/seg-2.3gs
3	320s	http://www.example.com/rep-320/seg-3.3gs
…	…	….
106	5250s	http://www.example.com/rep-320/seg-106.3gs

Table 6 Segment list for Representation with Id=384

Index	Start Time	URL
1	0s	http://www.example.com/rep-384/seg-1.3gs
2	40s	http://www.example.com/rep-384/seg-2.3gs
3	80s	http://www.example.com/rep-384/seg-3.3gs
…	…	….
132	5240s	http://www.example.com/rep-384/seg-132.3gs

An example message flow is provided in Figure 9.

1) The client initially requests the Initialisation Segment for the 192 kbit/s Representation.
2) The client also requests the first 1000 bytes of the first Media Segment of this Representation and after receipt it analyses the 'sidx' box and finds out the size and duration of the first fragment.
3) The client requests the first fragment by using byte ranges and starts play-out of the sequence.
4) The client requests a larger chunks of the segments with possible multiple fragments to gradually fill the buffer, but ensuring the continuous playout. The same may be achieved by parallel requests to multiple byte ranges.
5) The client continues to fill the data until the buffer is sufficiently filled by requesting possibly larger pieces of segment.
6) Once in stable buffer mode, the client prepares in parallel for switching representations, for example by downloading Initialisation Segment for Representation with id=384.
7) Once the client discovers that sufficient bandwidth is available, e.g., at tp=130s, it prepares to switch to a new representation by first identifying the segment that contains a possible switch time, in this case http://www.example.com/rep-384/seg-4.3gs, and downloads the 'sidx' box for scheduling further downloads.
8) A suitable byte range containing a RAP is chosen to start downloading the new Representation and eventually switch to the new Representation.

The message flow does not address fast seeking, but seeking would be performed in a similar manner as at start-up with the difference that the accessible bandwidth is known and search for a RAP using the 'sidx' box is necessary.

5.2 Adaptive Live Streaming Service

Assume that a service provider offers a popular live program as linear TV. The service is deployed on a CDN with an end-to-end delay being in a range of around 30 seconds. The media content is typically offered for different access bitrates and different device capabilities. The live service is typically split into Periods where live content is distributed and Periods during which pre-canned program is inserted, for example for ad insertion.

The program is available with a maximum time shift buffer of 90 minutes. More concrete the meeting starts at 9am at the West Coast, but the service is up from 8:45 showing a black screen only. At 9am, the live session is started which is interrupted by a canned ad starting at 11:01:22.12 seconds. After 15 minutes, the live session will restart. Further details are not known and the MPD is continuously updated. At some time before lunch break, the exact timing for the next ad break is scheduled. The duration of this break is unknown at this time. In the evening the session will terminate.

An example for a valid MPD for this service offering at time NOW of 2010-04-26T08:53:00-08:00 is provided below:

```
<?xml version="1.0" encoding="UTF-8"?>
<MPD
 type="Live"
 minBufferTime="PT3S"
 availabilityStartTime="2010-04-26T08:45:00-08:00"
 minimumUpdatePeriodMPD="PT5M0S"
 timeShiftBufferDepth="PT1H30M0S"

xsi:schemaLocation="urn:3GPP:ns:PSS:AdaptiveHTTPStreamingMPD:
2009 3GPP-MPD.xsd"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns="urn:3GPP:ns:PSS:AdaptiveHTTPStreamingMPD:2009">
 <ProgramInformation                                moreInfor-
mationURL="http://www.example.com">
  <Title>Example 3: 3GPP SA4 Meeting in Vancouver as Live
Broadcast</Title>
  <Source>3GPP</Source>
 </ProgramInformation>
 <Period start="PT0S">
  <Representation mimeType='video/3gpp; codecs="avc1.42E00B"'
bandwidth="10000" width="320" height="240">
   <SegmentInfo duration="PT60S">
    <InitialisationSegmentURL        sourceURL="http://www.ad-
server.com/1-day-black/QVGA/0.3gp"/>
    <UrlTemplate    sourceURL="http://www.ad-server.com/1-day-
black/QVGA/$Index$.3gs"/>
   </SegmentInfo>
  </Representation>
 </Period>
 <Period start="PT15M0S">
  <SegmentInfoDefault duration="PT10S" sourceUrlTemplatePeri-
od="http://www.example.com/Period-2010-04-26T08-45-00/rep-
$RepresentationID$/seg-$Index$.3gs"/>
  <Representation  mimeType='video/3gpp;  codecs="avc1.42E00C,
mp4a.40.2"' bandwidth="192000" width="320" height="240">
   <SegmentInfo>
    <InitialisationSegmentURL
sourceURL="http://www.example.com/rep-QVGA-LQ/seg-init.3gp"/>
    <UrlTemplate id="QVGA-LQ"/>
   </SegmentInfo>
  </Representation>
  <Representation  mimeType='video/3gpp;  codecs="avc1.42E00C,
mp4a.40.2"' bandwidth="384000" width="320" height="240">
   <SegmentInfo>
    <InitialisationSegmentURL
sourceURL="http://www.example.com/rep-QVGA-HQ/seg-init.3gp"/>
    <UrlTemplate id="QVGA-HQ"/>
   </SegmentInfo>
  </Representation>
  <Representation  mimeType='video/3gpp;  codecs="avc1.64001E,
mp4a.40.2"' bandwidth="512000" width="640" height="480">
   <SegmentInfo>
    <InitialisationSegmentURL
sourceURL="http://www.example.com/rep-VGA-LQ/seg-init.3gp"/>
    <UrlTemplate id="VGA-LQ"/>
   </SegmentInfo>
  </Representation>
  <Representation  mimeType='video/3gpp;  codecs="avc1.64001E,
mp4a.40.2"' bandwidth="1024000" width="640" height="480">
   <SegmentInfo>
    <InitialisationSegmentURL
sourceURL="http://www.example.com/rep-VGA-HQ/seg-init.3gp"/>
    <UrlTemplate id="VGA-HQ"/>
   </SegmentInfo>
  </Representation>
 </Period>
 <Period start="PT2H01M22.12S">
  <SegmentInfoDefault duration="PT10S" sourceUrlTemplatePeri-
od="http://www.ad-server.com/15min-
Ads/$RepresentationID$/$Index$.3gs"/>
  <Representation  mimeType='video/3gpp;  codecs="avc1.42E00C,
mp4a.40.2"' bandwidth="256000" width="320" height="240">
   <SegmentInfo>
    <InitialisationSegmentURL        sourceURL="http://www.ad-
server.com/15min-Ads/QVGA/0.3gp"/>
    <UrlTemplate id="QVGA"/>
   </SegmentInfo>
  </Representation>
  <Representation  mimeType='video/3gpp;  codecs="avc1.64001E,
mp4a.40.2"' bandwidth="512000" width="640" height="480">
   <SegmentInfo>
    <InitialisationSegmentURL        sourceURL="http://www.ad-
server.com/15min-Ads/VGA/0.3gp"/>
    <UrlTemplate id="VGA"/>
   </SegmentInfo>
  </Representation>
 </Period>
 <Period start="PT2H16M22.12S">
  <SegmentInfoDefault duration="PT10S" sourceUrlTemplatePeri-
od="http://www.example.com/Period-2010-04-26T11-01-22/rep-
$RepresentationID$/seg-$Index$.3gs"/>
  <Representation  mimeType='video/3gpp;  codecs="avc1.42E00C,
mp4a.40.2"' bandwidth="192000" width="320" height="240">
   <SegmentInfo>
    <InitialisationSegmentURL
sourceURL="http://www.example.com/rep-QVGA-LQ/seg-0.3gp"/>
    <UrlTemplate id="QVGA-LQ"/>
   </SegmentInfo>
  </Representation>
  <Representation  mimeType='video/3gpp;  codecs="avc1.42E00C,
mp4a.40.2"' bandwidth="384000" width="320" height="240">
   <SegmentInfo>
    <InitialisationSegmentURL
sourceURL="http://www.example.com/rep-QVGA-HQ/seg-0.3gp"/>
    <UrlTemplate id="QVGA-HQ"/>
   </SegmentInfo>
  </Representation>
  <Representation  mimeType='video/3gpp;  codecs="avc1.64001E,
mp4a.40.2"' bandwidth="512000" width="640" height="480">
   <SegmentInfo>
    <InitialisationSegmentURL
sourceURL="http://www.example.com/rep-QVGA-LQ/seg-0.3gp"/>
    <UrlTemplate id="VGA-LQ"/>
   </SegmentInfo>
  </Representation>
  <Representation  mimeType='video/3gpp;  codecs="avc1.64001E,
mp4a.40.2"' bandwidth="1024000" width="640" height="480">
   <SegmentInfo>
    <InitialisationSegmentURL
sourceURL="http://www.example.com/rep-QVGA-LQ/seg-0.3gp"/>
    <UrlTemplate id="VGA-HQ"/>
   </SegmentInfo>
  </Representation>
 </Period>
</MPD>
```

At time 2010-04-26T12:26:00-08:00 the client has accesses to an updated MPD as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<MPD
 type="Live"
 minBufferTime="PT3S"
 availabilityStartTime="2010-04-26T08:45:00-08:00"
 minimumUpdatePeriodMPD="PT5M0S"             timeShiftBuffer-
Depth="PT1H30M0S"

xsi:schemaLocation="urn:3GPP:ns:PSS:AdaptiveHTTPStreamingMPD:
2009 3GPP-MPD.xsd"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns="urn:3GPP:ns:PSS:AdaptiveHTTPStreamingMPD:2009">
 <ProgramInformation                                moreInfor-
mationURL="http://www.example.com">
  <Title>Example 3: 3GPP SA4 Meeting in Vancouver as Live
Broadcast</Title>
  <Source>3GPP</Source>
 </ProgramInformation>
 <Period start="PT2H01M22.12S">
  <SegmentInfoDefault duration="PT10S" sourceUrlTemplatePeri-
od="http://www.ad-server.com/15min-
Ads/$RepresentationID$/$Index$.3gs"/>
  <Representation  mimeType='video/3gpp;  codecs="avc1.42E00C,
mp4a.40.2"' bandwidth="256000" width="320" height="240">
   <SegmentInfo>
    <InitialisationSegmentURL        sourceURL="http://www.ad-
server.com/15min-Ads/QVGA/0.3gs"/>
    <UrlTemplate id="QVGA"/>
   </SegmentInfo>
  </Representation>
  <Representation  mimeType='video/3gpp;  codecs="avc1.64001E,
mp4a.40.2"' bandwidth="512000" width="640" height="480">
   <SegmentInfo>
    <InitialisationSegmentURL        sourceURL="http://www.ad-
server.com/15min-Ads/VGA/0.3gs"/>
    <UrlTemplate id="VGA"/>
```

```
    </SegmentInfo>
   </Representation>
  </Period>
  <Period start="PT2H16M22.12S">
   <SegmentInfoDefault duration="PT10S" sourceUrlTemplatePeri-
od="http://www.example.com/Period-2010-04-26T11-01-22/rep-
$RepresentationID$/seg-$Index$.3gs"/>
   <Representation  mimeType='video/3gpp;  codecs="avc1.42E00C,
mp4a.40.2"' bandwidth="192000" width="320" height="240">
    <SegmentInfo>
     <InitialisationSegmentURL
sourceURL="http://www.example.com/rep-QVGA-LQ/seg-0.3gs"/>
     <UrlTemplate id="QVGA-LQ"/>
    </SegmentInfo>
   </Representation>
   <Representation  mimeType='video/3gpp;  codecs="avc1.42E00C,
mp4a.40.2"' bandwidth="384000" width="320" height="240">
    <SegmentInfo>
     <InitialisationSegmentURL
sourceURL="http://www.example.com/rep-QVGA-HQ/seg-0.3gs"/>
     <UrlTemplate id="QVGA-HQ"/>
    </SegmentInfo>
   </Representation>
   <Representation  mimeType='video/3gpp;  codecs="avc1.64001E,
mp4a.40.2"' bandwidth="512000" width="640" height="480">
    <SegmentInfo>
     <InitialisationSegmentURL
sourceURL="http://www.example.com/rep-VGA-LQ/seg-0.3gs"/>
     <UrlTemplate id="VGA-LQ"/>
    </SegmentInfo>
   </Representation>
   <Representation  mimeType='video/3gpp;  codecs="avc1.64001E,
mp4a.40.2"' bandwidth="1024000" width="640" height="480">
    <SegmentInfo>
     <InitialisationSegmentURL
sourceURL="http://www.example.com/rep-VGA-HQ/seg-0.3gs"/>
     <UrlTemplate id="VGA-HQ"/>
    </SegmentInfo>
   </Representation>
  </Period>
  <Period start="PT4H15M18.3S">
   <SegmentInfoDefault duration="PT10S" sourceUrlTemplatePeri-
od="http://www.ad-server.com/120min-
Ads/$RepresentationID$/$Index$.3gs"/>
   <Representation  mimeType='video/3gpp;  codecs="avc1.42E00C,
mp4a.40.2"' bandwidth="256000" width="320" height="240">
    <SegmentInfo>
     <InitialisationSegmentURL       sourceURL="http://www.ad-
server.com/15min-Ads/QVGA/0.3gs"/>
     <UrlTemplate id="QVGA"/>
    </SegmentInfo>
   </Representation>
   <Representation  mimeType='video/3gpp;  codecs="avc1.64001E,
mp4a.40.2"' bandwidth="512000" width="640" height="480">
    <SegmentInfo>
     <InitialisationSegmentURL       sourceURL="http://www.ad-
server.com/15min-Ads/VGA/0.3gs"/>
     <UrlTemplate id="VGA"/>
    </SegmentInfo>
   </Representation>
  </Period>
</MPD>
```

At time 2010-04-26T18:45:00-08:00 the client has access to another update of the MPD as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<MPD
 type="Live"
 minBufferTime="PT3S"
 availabilityStartTime="2010-04-26T08:45:00-08:00"
 mediaPresentationDuration="PT12H0M0S"
 timeShiftBufferDepth="PT1H30M0S"

xsi:schemaLocation="urn:3GPP:ns:PSS:AdaptiveHTTPStreamingMPD:
2009 3GPP-MPD.xsd"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns="urn:3GPP:ns:PSS:AdaptiveHTTPStreamingMPD:2009">
 <ProgramInformation                              moreInfor-
mationURL="http://www.example.com">
  <Title>Example  3:  3GPP  SA4  Meeting  in  Vancouver  as  Live
Broadcast</Title>
  <Source>3GPP</Source>
 </ProgramInformation>
 <Period start="PT8H22M45S">
  <SegmentInfoDefault duration="PT10S" sourceUrlTemplatePeri-
od="http://www.example.com/Period-2010-04-26T17-07-45/rep-
$RepresentationID$/seg-$Index$.3gs"/>
  <Representation  mimeType='video/3gpp;  codecs="avc1.42E00C,
mp4a.40.2"' bandwidth="192000" width="320" height="240">
   <SegmentInfo>
    <InitialisationSegmentURL
sourceURL="http://www.example.com/rep-QVGA-LQ/seg-0.3gs"/>
    <UrlTemplate id="QVGA-LQ"/>
   </SegmentInfo>
  </Representation>
  <Representation  mimeType='video/3gpp;  codecs="avc1.42E00C,
mp4a.40.2"' bandwidth="384000" width="320" height="240">
   <SegmentInfo>
    <InitialisationSegmentURL
sourceURL="http://www.example.com/rep-QVGA-HQ/seg-0.3gs"/>
    <UrlTemplate id="QVGA-HQ"/>
   </SegmentInfo>
  </Representation>
  <Representation  mimeType='video/3gpp;  codecs="avc1.64001E,
mp4a.40.2"' bandwidth="512000" width="640" height="480">
   <SegmentInfo>
    <InitialisationSegmentURL
sourceURL="http://www.example.com/rep-VGA-LQ/seg-0.3gs"/>
    <UrlTemplate id="VGA-LQ"/>
   </SegmentInfo>
  </Representation>
  <Representation  mimeType='video/3gpp;  codecs="avc1.64001E,
mp4a.40.2"' bandwidth="1024000" width="640" height="480">
   <SegmentInfo>
    <InitialisationSegmentURL
sourceURL="http://www.example.com/rep-VGA-HQ/seg-0.3gs"/>
    <UrlTemplate id="VGA-HQ"/>
   </SegmentInfo>
  </Representation>
 </Period>
 <Period start="PT10H45M00S">
  <Representation mimeType='video/3gpp; codecs="avc1.42E00B"'
bandwidth="10000" width="320" height="240">
   <SegmentInfo duration="PT60S">
    <InitialisationSegmentURL       sourceURL="http://www.ad-
server.com/1-day-black/QVGA/0.3gs"/>
    <UrlTemplate    sourceURL="http://www.ad-server.com/1-day-
black/QVGA/$Index$.3gs"/>
   </SegmentInfo>
  </Representation>
 </Period>
</MPD>
```

The Segments for each representation are encoded according to the Initialisation Segment and Media Segment format as specified in 3GP-DASH. The ‘sidx’ is provided to permit accurate seeking and time-alignment of media segments.

A typical message flow is aligned with the ones for On-Demand streaming. Updates of the MPD need to be considered. The accessible segment lists for the different times are provided.

At time 2010-04-26T08:53:00-08:00, only one Period and one Representation is available. This is the accessible segment list:

Index	Start Time	URL
1	0s	http://www.ad-server.com/1-day-black/QVGA/1.3gs
2	60s	http://www.ad-server.com/1-day-black/QVGA/2.3gs
3	120s	http://www.ad-server.com/1-day-black/QVGA/3.3gs
…	…	….
7	480s	http://www.ad-server.com/1-day-black/QVGA/7.3gs

At time 2010-04-26T12:26:00-08:00 as the time shift buffer is 90 minutes, media segments back to 2010-04-26T10:56:00-08:00 are available, i.e., media segments from two periods. We present the segment list of the generic Representation in each Period.

For Period starting at <Period start="PT2H01M22.12S">, i.e., at 2010-04-26T10:46:22.12-08:00, the following media segments are available:

Index	Start Time	URL
58	570s	http://www.ad-server.com/15min-Ads/$RepresentationID$/58.3gs
59	580s	http://www.ad-server.com/1-day-black/$RepresentationID$/59.3gs

60	590s	http://www.ad-server.com/1-day-black/$RepresentationID$/60.3gs
…	…	….
90	890s	http://www.ad-server.com/1-day-black/$RepresentationID$/90.3gs

For Period starting at <Period start="PT2H16M22.12S"> i.e., at 2010-04-26T11:01:22.12-08:00, the following media segments are available

Index	Start Time	**URL with Base Url** http://www.example.com
1	0s	./Period-2010-04-26T11-01-22/rep-$RepresentationID$/seg-1.3gs
2	10s	./Period-2010-04-26T11-01-22/rep-$RepresentationID$/seg-2.3gs
3	590s	./Period-2010-04-26T11-01-22/rep-$RepresentationID$/seg-3.3gs
…	…	….
508	5070s	./Period-2010-04-26T11-01-22/rep-$RepresentationID$/seg-508.3gs

At time 2010-04-26T18:45:00-08:00 the client can access media segments back to 2010-04-26T17:15:00-08:00, i.e., media segments from only one periods. We present the segment list of the generic Representation in this Period.

For Period starting at <Period start="PT8H22M45S">i.e., at 2010-04-26T10:46:22.12-08:00, the following media segments are available due to the NOW constraints

Index	Start Time	**URL with Base Url** http://www.example.com
44	430s	./Period-2010-04-26T17-07-45/rep-$RepresentationID$/seg-44.3gs
45	440s	./Period-2010-04-26T11-01-22/rep-$RepresentationID$/seg-45.3gs
46	450s	./Period-2010-04-26T11-01-22/rep-$RepresentationID$/seg-46.3gs
…	…	….
583	5820s	./Period-2010-04-26T11-01-22/rep-$RepresentationID$/seg-583.3gs

6. CONCLUSIONS AND OUTLOOK

3GP-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 3GP-DASH. Major players in the market, including those that offer proprietary solutions today, participated in the development of the specification. 3GP-DASH also serves as baseline for several other organizations, in particular the Open IPTV Forum and MPEG. Especially MPEG [12] is considering backward-compatible extensions to the 3GP-DASH specification to integrate additional media such as multiview or scalable video coding. Furthermore, initial efforts in interoperability testing have started. Currently there is great hope that the foundations laid in 3GP-DASH build the core package of an industry-standard for Dynamic Adaptive Streaming over HTTP (DASH).

7. ACKNOWLEDGMENTS

Many thanks to all the colleagues in Qualcomm Incorporated and especially in 3GPP SA4 and MPEG DASH for the collaboration on the matter and their contributions to a hopefully successful and widely deployed standard.

8. REFERENCES

[1] 3GPP TS 26.234: "Transparent end-to-end packet switched streaming service (PSS); Protocols and codecs".

[2] 3GPP TS 26.244: "Transparent end-to-end packet switched streaming service (PSS); 3GPP file format (3GP)".

[3] 3GPP TS 26.245: "Transparent end-to-end packet switched streaming service (PSS); Timed text format".

[4] IETF RFC 2616: "Hypertext Transfer Protocol – HTTP/1.1", Fielding R. et al., June 1999.

[5] ISO/IEC 14496-12:2005 | 15444-12:2005: "Information technology – Coding of audio-visual objects – Part 12: ISO base media file format" | "Information tech-nology – JPEG 2000 image coding system – Part 12: ISO base media file format".

[6] IETF RFC 3986: "Uniform Resource Identifiers (URI): Generic Syntax", Berners-Lee T., Fielding R. and Masinter L., January 2005

[7] Cisco White Paper: Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2009-2014, http://bit.ly/bwGY7L

[8] Apple HTTP live Streaming: http://tools.ietf.org/id/draft-pantos-http-live-streaming-04.txt

[9] DF-Splash Overview: http://www.digitalfountain.com/ufiles/library/DF-Splash-Service-Overview.pdf

[10] Microsoft Smooth Streaming: http://go.microsoft.com/?linkid=9682896

[11] Adaptive Streaming Comparison, http://learn.iis.net/page.aspx/792/adaptive-streaming-comparison

[12] ISO/IEC JTC1/SC29/WG11 N11338, Call for Proposals on HTTP Streaming of MPEG Media, April 2010, Dresden, Germany.

[13] OIPF Specification Volume 2a - HTTP Adaptive Streaming V2.0, 2010/09/07.