MPEG DASH: A Technical Deep Dive and Look at What’s Next

Andy Salo
RGB Networks

Abstract

The MPEG DASH standard was ratified in December 2011 and published by the International Organization for Standards (ISO) in April 2012. This paper will review the technical aspects of the new MPEG DASH standard in detail, including: how DASH supports live, on-demand and time-shifted (NDVR) services; how the two primary video formats – ISO-base media file format (IBMFF) and MPEG-2 TS – compare and contrast; how the new standard supports digital rights management (DRM) methods; and how Media Presentation Description (MPD) XML files differ from current adaptive streaming manifests. In addition, the paper will discuss how MPEG DASH is likely to be adopted by the industry, what challenges must still be overcome, and what the implications could be for cable operators and other video service providers (VSPs).

INTRODUCTION

For much of the past decade, it was quite difficult to stream live video to a mobile device. Wide bandwidth variability, unfavorable firewall configurations and lack of network infrastructure support all created major roadblocks to live streaming. Early, more traditional streaming protocols, designed for small packet formats and managed delivery networks, were anything but firewall-friendly. Although HTTP progressive download was developed partially to get audio and video streams past firewalls, it still didn’t offer true streaming capabilities.

Now, the advent of adaptive streaming over HTTP technology has changed everything, reshaping video delivery to PCs, laptops, game consoles, tablets, smartphones and other mobile devices, as well as such key home devices as Web-connected TVs and pure and hybrid IP set-top boxes (STBs). As a result, watching video online or on the go is no longer a great novelty, nor is streaming Internet-delivered content to TV screens in the home. Driven by the explosion in video-enabled devices, consumers have swiftly moved through the early-adopter phase of TV Everywhere service, reaching the point where a growing number expect any media to be available on any device over any network connection at any time. Increasingly, consumers also expect the content delivery to meet the same high quality levels they have come to know and love from traditional TV services.

Even though the emergence of the three main adaptive streaming protocols from Adobe, Apple and Microsoft over the past three and a half years has made multiscreen video a reality, significant problems still remain. Each of the three proprietary platforms is a closed system, with its own manifest format, content formats and streaming protocols. So, content creators and equipment vendors must craft several different versions of their products to serve the entire streaming video market, greatly driving up costs and restricting the market’s overall development.

In an ambitious bid to solve these nagging problems, MPEG has recently adopted a new standard for multimedia streaming over the Internet. Known as MPEG Dynamic Adaptive Streaming over HTTP, or MPEG DASH, the new industry standard attempts to create a universal delivery format for streaming media by incorporating the best elements of the three main proprietary streaming solutions. In doing so, MPEG DASH aims to provide the long-sought interoperability between different network servers and different consumer electronics devices, thereby fostering a common ecosystem of content and services.

This paper will review the technical aspects of the new MPEG DASH standard in detail, including: how DASH supports live, on-demand and time-shifted (NDVR) services; how the two primary video formats (ISO-base media file format (IBMFF) and MPEG-2 TS) compare and contrast; how the standard supports DRM methods; and how Media Presentation Description (MPD) XML files differ from current adaptive streaming manifests. In addition, the paper will discuss how MPEG DASH is likely to be adopted by the industry, what challenges must still be overcome, and what the implications could be for cable operators and other video service providers (VSPs).

AN ADAPTIVE STREAMING PRIMER

As indicated previously, the delivery of streaming video and audio content to consumer electronics devices has come a long way over the past few years. Thanks to the introduction of adaptive streaming over HTTP, multimedia content can now be delivered more easily than ever before. In particular, adaptive streaming offers two critical features for video content that have made the technology the preferred choice for mobile delivery.

First, adaptive streaming over HTTP breaks down, or segments, video programs into small, easy-to-download chunks. For example, Apple's HTTP Live Streaming (HLS) protocol typically segments video content into 10-second chunks, while Microsoft's Smooth Streaming (MSS) protocol and Adobe's HTTP Dynamic Streaming (HDS) usually break video content into even smaller chunks of five seconds or less.

Second, adaptive streaming encodes the video content at multiple bitrates and resolutions, creating different chunks of different sizes. This is the truly 'adaptive' part of adaptive streaming, as the encoding enables the mobile client to choose between various bitrates and resolutions and then adapt to larger or smaller chunks automatically as network conditions keep changing.

In turn, these two key features of adaptive streaming lead to a number of benefits:

1. Video chunks can be cached by proxies and easily distributed to content delivery networks (CDNs) or HTTP servers, which are simpler and cheaper to operate than the special streaming servers required for 'older' video streaming technologies.

2. Bitrate switching allows clients to adapt dynamically to network conditions.

3. Content providers no longer have to guess which bitrates to encode for end devices.

4. The technology works well with firewalls because the streams are sent over HTTP.

5. Live and video-on-demand (VoD) workflows are almost identical. When a service provider creates a live stream, the chunks can easily be stored for later VoD delivery.

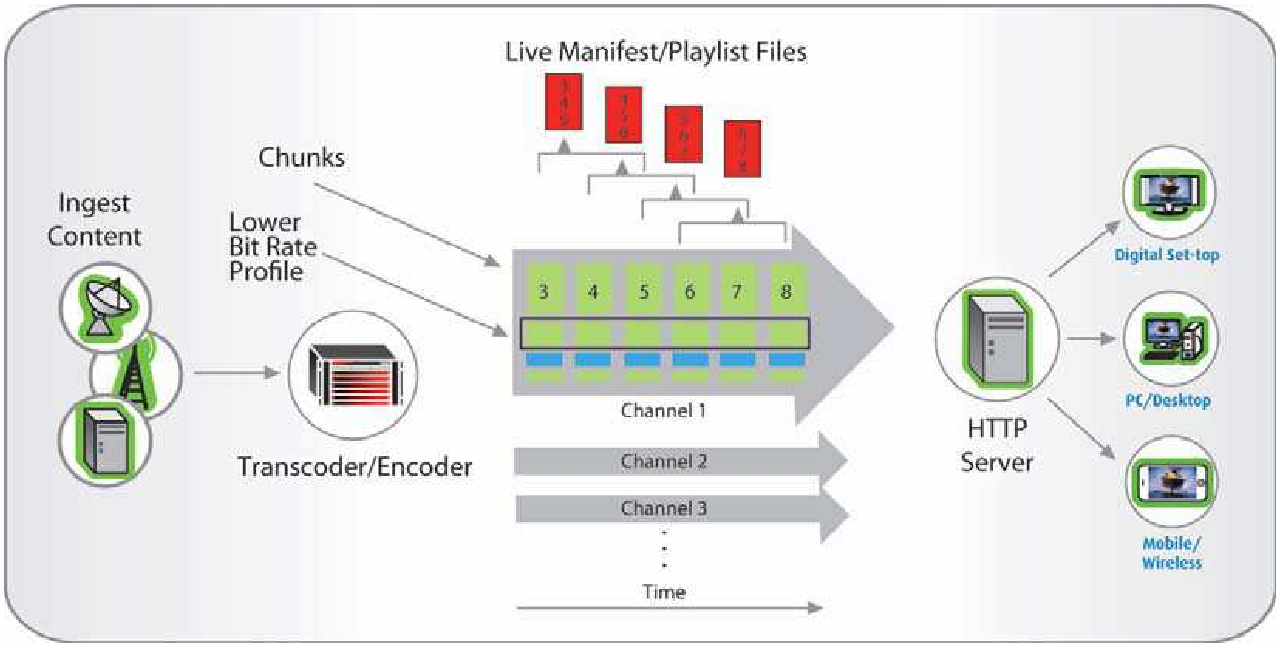


Figure 1: Content Delivery Chain for Live Adaptive Streaming

Sensing the promise of adaptive streaming technology, several major technology players have sought to carve out large shares of the rapidly growing market. Most notably, the list now includes such prominent tech companies as Adobe, Apple and Microsoft.

While the streaming of video using HTTP-delivered fragments goes back many years (and seems lost in the mists of time), Move Networks caught the attention of several media companies with its adaptive HTTP streaming technology in 2007. Move was quickly followed by Microsoft, which entered the market by releasing Smooth Streaming in October 2008 as part of its Silverlight architecture. Earlier that year, Microsoft demonstrated a prototype version of Smooth Streaming by delivering live and on-demand streaming content from such events as the Summer Olympic Games in Beijing and the Democratic National Convention in Denver.

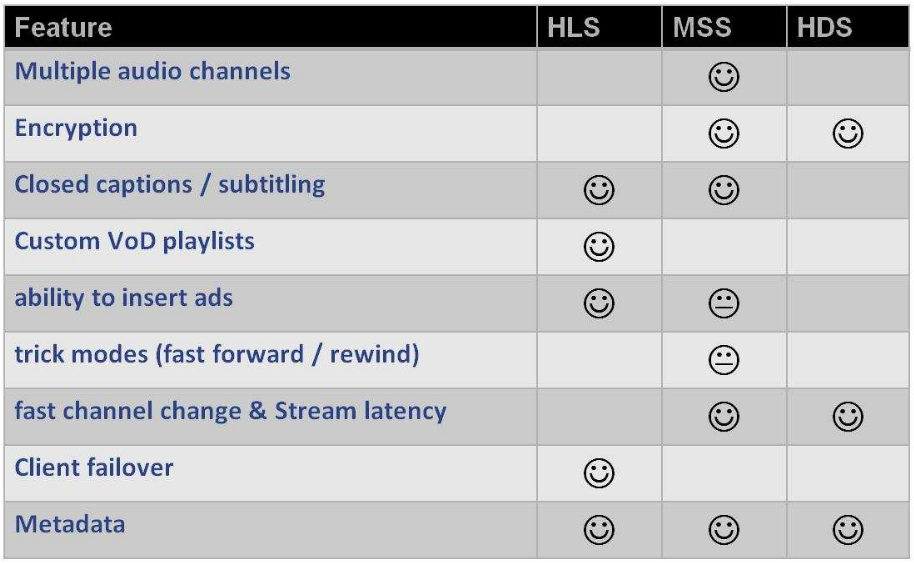
Smooth Streaming has all of the typical characteristics of adaptive streaming. The video content is segmented into small chunks and then delivered over HTTP. Usually, multiple bitrates are encoded so that the client can choose the best video bitrate to deliver an optimal viewing experience based on network conditions.

Apple came next with HLS, originally unveiling it with the introduction of the iPhone 3.0 in mid-2009. Prior to the iPhone 3, no streaming protocols were supported natively on the iPhone, leaving developers to wonder what Apple had in mind for native streaming support. In May 2009, Apple proposed HLS as a standard to the Internet Engineering Task Force (IETF), and the draft is now in its eighth iteration.

HLS works by segmenting video streams into 10-second chunks; the chunks are stored using a standard MPEG-2 transport stream file format. The chunks may be created using several bitrates and resolutions – so-called profiles – allowing a client to switch dynamically between different profiles, depending on network conditions.

Adobe, the last of the Big Three, entered the adaptive streaming market in late 2009 with the announcement of HTTP Dynamic Streaming (HDS). Originally known as “Project Zeri,” HDS was introduced in June 2010. Like MSS and HLS, HDS breaks up video content into small chunks and delivers them over HTTP. Multiple bitrates are encoded so that the client can choose the best video bitrate to deliver an optimal viewing experience based on network conditions.

HDS is closer to Microsoft Smooth Streaming than it is to Apple’s HLS protocol. Primarily, this is because HDS, like MSS, uses a single aggregate file from which MPEG-4 container fragments are extracted and delivered. In contrast, HLS uses individual media chunks rather than one large aggregate file.



Feature	HLS	MSS	HDS
Multiple audio channels		☺	
Encryption		☺	☺
Closed captions / subtitling	☺	☺	
Custom VoD playlists	☺		
ability to insert ads	☺	😐	
trick modes (fast forward / rewind)		😐	
fast channel change & Stream latency		☺	☺
Client failover	☺		
Metadata	☺	☺	☺

Figure 2: Feature Comparison of Three Major Adaptive Streaming Platforms

THE DUELING STREAMING PLATFORM PROBLEM

The three major adaptive streaming protocols – MSS, HLS and HDS – have much in common. Most importantly, all three streaming platforms use HTTP streaming for their underlying delivery method, relying on standard HTTP Web servers instead of special streaming servers. They all use a combination of encoded media files and manifest files that identify the main and alternative streams and their respective URLs for the player. And their respective players all monitor either buffer status or CPU utilization and switch streams as necessary, locating the alternative streams from the URLs specified in the manifest.

The overriding problem with MSS, HLS and HDS is that these three different streaming protocols, while quite similar to each other in many ways, are different enough that they are not technically compatible. Indeed, each of the three proprietary commercial platforms is a closed system with its own type of manifest format, content formats, encryption methods and streaming protocols, making it impossible for them to work together.

Take Microsoft Smooth Streaming and Apple's HLS. Here are three key differences between the two competing platforms:

1. HLS makes use of a regularly updated "moving window" metadata index file that tells the client which chunks are available for download. Smooth Streaming uses time codes in the chunk requests so that the client doesn't have to keep downloading an index file. This leads to a second difference:

2. HLS requires a download of an index file every time a new chunk is available. That makes it desirable to run HLS with longer duration chunks, thereby minimizing the number of index file downloads. So, the recommended chunk duration with HLS is 10 seconds, while it is just two seconds with Smooth Streaming.

3. The "wire format" of the chunks is different. Although both formats use H.264 video encoding and AAC audio encoding, HLS makes use of MPEG-2 Transport Stream files, while Smooth Streaming makes use of "fragmented" ISO MPEG-4 files. The "fragmented" MP4 file is a variant in which not all the data in a regular MP4 file is included in the file. Each of these formats has some advantages and disadvantages. MPEG-2 TS files have a large installed analysis toolset and have pre-defined signaling mechanisms for things like data signals (e.g. specification of ad insertion points). But fragmented MP4 files are very flexible and can easily accommodate all kinds of data, such as decryption information, that MPEG-2 TS files don't have defined slots to carry.

Or take Adobe HDS and Apple's HLS. These two platforms have a number of key differences as well:

1. HLS makes use of a regularly updated "moving window" metadata index (manifest) file that tells the client which chunks are available for download. Adobe HDS uses sequence numbers in the chunk requests so the client doesn't have to keep downloading a manifest file.

2. In addition to the manifest, there is a bootstrap file, which in the live case gives the updated sequence numbers and is equivalent to the repeatedly downloaded HLS playlist.

3. Because HLS requires a download of a manifest file as often as every time a new chunk is available, it is desirable to run HLS with longer duration chunks, thus minimizing the number of manifest file downloads. More recent Apple client versions appear to check how many segments are in the playlist and only re-fetch the manifest when the client runs out of segments. Nevertheless, the recommended chunk duration with HLS is still 10 seconds, while it is usually just two to five seconds with Adobe HDS.

4. The "wire format" of the chunks is different. Both formats use H.264 video encoding and AAC audio encoding. But HLS makes use of MPEG-2 TS files, while Adobe HDS (and Microsoft SS) make use of "fragmented" ISO MPEG-4 files.

Due to such differences, there is no such thing as a universal delivery standard for streaming media today. Likewise, there is no universal encryption standard or player standard. Nor is there any interoperability between the devices and servers of the various vendors. So, content cannot be re-used and creators and equipment makers must develop several different versions of their products to serve the entire streaming video market, greatly driving up costs and restricting the market's overall development.

INTRODUCING MPEG DASH: A STANDARDS-BASED APPROACH

Seeing the need for a universal standard for the delivery of adaptive streaming media, MPEG decided to step into the void three years ago. In April 2009, the organization issued a Request for Proposals for an HTTP streaming standard. By that July, MPEG had received 15 full proposals. In the following two years, MPEG developed the specification with the help of many experts and in collaboration with other standards groups, such as the Third Generation Partnership Project (3GPP) and the Open IPTV Forum (OIPF).

The resulting MPEG standardization of Dynamic Adaptive Streaming over HTTP is now simply known as MPEG DASH.

MPEG DASH is not a system, protocol, presentation, codec, middleware, or client specification. Rather, the new standard is more like a neutral enabler, aimed at providing several formats that foster the efficient and high-quality delivery of streaming media services over the Internet.

As described by document ISO/IEC 23009-1, MPEG DASH can be viewed as an amalgamation of the industry's three prominent adaptive streaming protocols – Adobe HDS, Apple HLS and Microsoft Smooth Streaming. Like those three proprietary platforms, DASH is a video streaming solution where small chunks of video streams/files are requested using HTTP and then spliced together by the client. The client entirely controls the delivery of services.

In other words, MPEG DASH offers a standards-based approach for enabling a host of media services that cable operators and telcos have traditionally offered in broadcast and IPTV environments and extending those capabilities to adaptive bitrate delivery, including live and on-demand content delivery, time-shifted services (NDVR, catch-up TV), and targeted ad insertion. DASH enables these features through a number of inherent capabilities, and perhaps most importantly, through a flexibility of design and implementation. Its capabilities and features include:

- Multiple segment formats (ISO BMFF and MPEG-2 TS)
- Codec independence
- Trick mode functionality
- Profiles: restriction of DASH and system features (claim & permission)
- Content descriptors for protection, accessibility, content rating, and more
- Common encryption (defined by ISO/IEC 23001-7)
- Clock drift control for live content
- Metrics for reporting the client session experience

A Tale of Two Containers – MPEG-2 TS and ISO BMFF

Under the MPEG DASH standard, the media segments can contain any type of media data. However, the standard provides specific guidance and formats for use with two types of segment container formats – MPEG-2 Transport Stream (MPEG-2 TS) and ISO base media file format (ISO BMFF).

MPEG-2 TS is the segment format that HLS currently uses, while ISO BMFF (which is basically the MPEG-4 format) is what Smooth Streaming and HDS currently use.

This mix of the two container formats employed by the three commercial platforms allows for a relatively easy migration of existing adaptive streaming content from the proprietary platforms to MPEG DASH. That's because the media segments can often stay the same; only the index files must be migrated to a different format, which is known as Media Presentation Description.

Media Presentation Description (MPD) – Definition and Overview

At a high level, MPEG DASH works nearly the same way as the three other major adaptive streaming protocols. DASH presents available stream content to the media player in a manifest (or index) file – called the Media Presentation Description (MPD) – and then supports HTTP download of media segments. The MPD is analogous to an HLS m3u8 file, a Smooth Streaming Manifest file or an HDS f4m file. After the MPD is delivered to the client, the content – whether it's video, audio, subtitles or other data – is downloaded to clients over HTTP as a sequence of files that is played back contiguously.

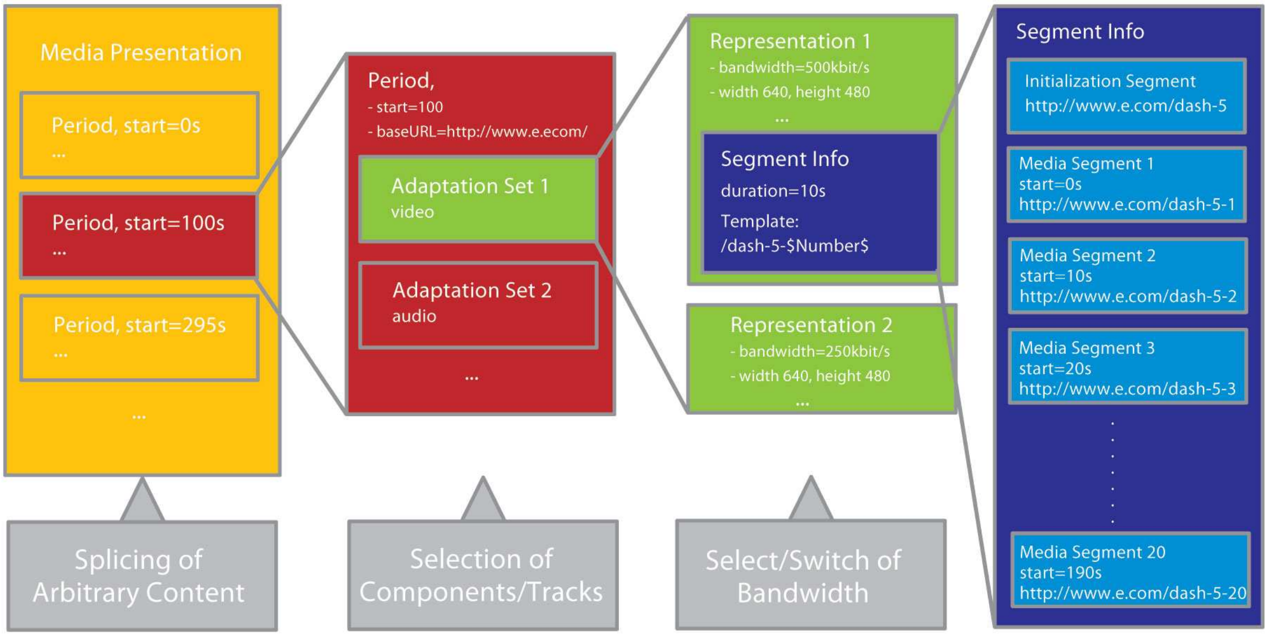


Figure 3: Media Presentation Data Model
(Diagram originally developed by Thomas Stockhammer, Qualcomm)

Like a manifest file in the three commercial platforms, the MPD in MPEG DASH describes the content that is available, including the URL addresses of stream chunks, byte-ranges, different bitrates, resolutions, and content encryption mechanisms. The tasks of choosing which adaptive stream bitrate and resolution to play and switching to different bitrate streams according to network conditions are performed by the client (again, similar to the other adaptive streaming protocols). In fact, DASH does not prescribe any client-specific playback functionality; rather, it just addresses the formatting of the content and associated MPDs.

To see what an MPEG DASH MPD file looks like compared to an HLS m3u8 file, consider the following example. The files contain much of the same information, but they are formatted and presented differently.

Figure 4: Comparison of MPEG DASH MPD and HLS m3u8 Files

Index.m3u8 (top level m3u8)

```
#EXTM3U
#EXT-X-STREAM-INF:PROGRAM- ID=1,BANDWIDTH=291500,RESOLUTION=320x180
stream1.m3u8
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=610560,RESOLUTION=512x288
stream2.m3u8
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=2061700,RESOLUTION=1024x576
stream3.m3u8
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=4659760,RESOLUTION=1280x720
stream4.m3u8
```

Index.mpd

```
<?xml version="1.0" encoding="utf-8"?>
<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"
  type="static"
  mediaPresentationDuration="PT12M34.041388S"
  minBufferTime="PT10S"
  profiles="urn:mpeg:dash:profile:isoff-live:2011">

  <Period>
    <AdaptationSet
      mimeType="audio/mp4"
      segmentAlignment="0"
      lang="eng">
      <SegmentTemplate
        timescale="10000000"
        media="audio_eng=$Bandwidth$-$Time$.dash"
        initialisation=" audio_eng=$Bandwidth$.dash">
        <SegmentTimeline>
          <S t="667333" d="39473889" />
          <S t="40141222" d="40170555" />

       ...

          <S t="7527647777" d="12766111" />
        </SegmentTimeline>
      </SegmentTemplate>
      <Representation id="audio_eng=96000" bandwidth="96000" codecs="mp4a.40.2"
audioSamplingRate="44100" />
    </AdaptationSet>
    <AdaptationSet
      mimeType="video/mp4"
      segmentAlignment="true"
      startWithSAP="1"
      lang="eng">
      <SegmentTemplate
        timescale="10000000"
        media="video=$Bandwidth$-$Time$.dash"
        initialisation="video=$Bandwidth$.dash">
        <SegmentTimeline>
          <S t="0" d="40040000" r="187" />
          <S t="7527520000" d="11678333" />
        </SegmentTimeline>
      </SegmentTemplate>

      <Representation id="video=299000" bandwidth="299000" codecs="avc1.42C00D"
width="320" height="180" />
      <Representation id="video=480000" bandwidth="480000" codecs="avc1.4D401F"
width="512" height="288" />
 codecs="avc1.4D401F" width="1024" height="576" />
      <Representation id="video=4300000" bandwidth="4300000"
codecs="avc1.640028" width="1280" height="720" />
    </AdaptationSet>
  </Period>
</MPD>
```

MPEG DASH’S PRIME CAPABILITIES – OVERVIEW

As mentioned earlier, MPEG DASH offers a great number of capabilities for adaptive streaming. This section goes into greater detail about many of the prime capabilities.

Codec Independence: Simply put, MPEG DASH is audio/video agnostic. As a result, the standard can work with media files of MPEG-2, MPEG-4, H.264, WebM and various other codecs and does not favor one codec over another. It also supports both multiplexed and unmultiplexed encoded content. More importantly, DASH will support emerging standards, such as HEVC (H.265).

Trick Mode Functionality: MPEG DASH supports VoD trick modes for pausing, seeking, fast forwarding and rewinding content. For instance, the client may pause or stop a Media Presentation.

In this case, the client simply stops requesting Media Segments or parts thereof. To resume, the client sends requests to Media Segments, starting with the next sub-segment after the last requested sub-segment.

DASH’s treatment of trick modes could prove to be a major improvement over the way that the three existing streaming protocols handle these on-demand functions now.

Profiles: Restriction of DASH and System Features (Claim & Permission): MPEG DASH defines and allows for the creation of various profiles. A profile is a set of restrictions of media formats, codecs, protection formats, bitrates, resolutions, and other aspects of the content. For example, the DASH spec defines a profile for ISO BMFF basic on-demand.

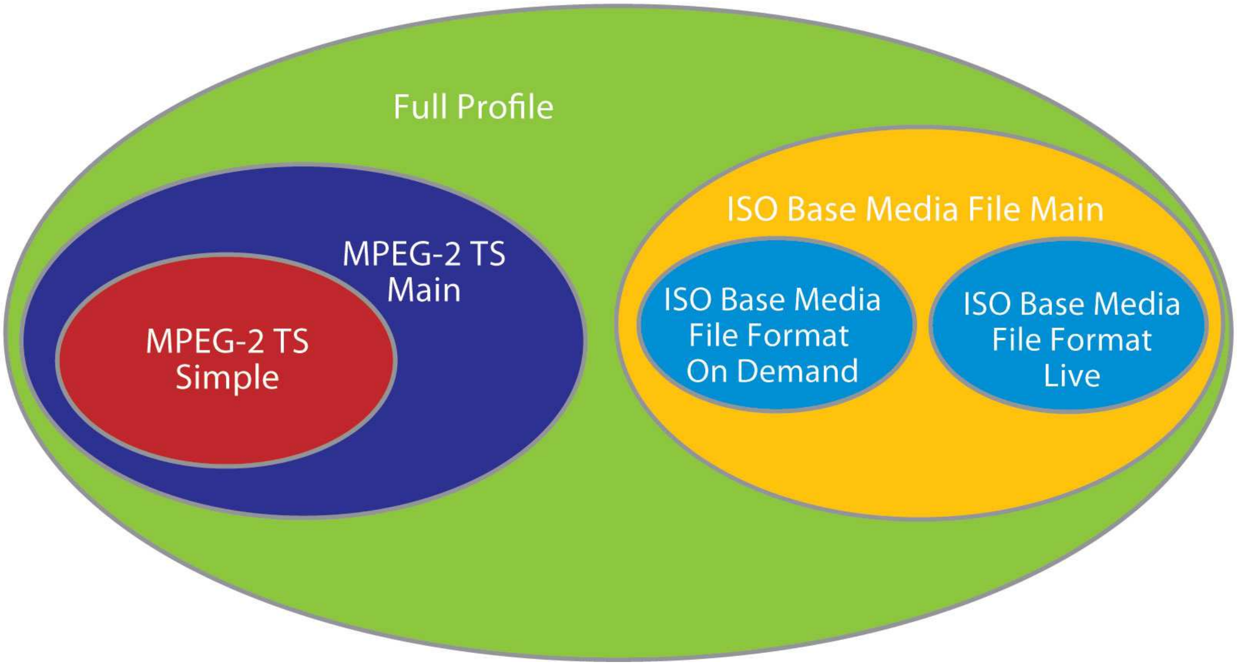

Full Profile
MPEG-2 TS Main
MPEG-2 TS Simple
ISO Base Media File Main
ISO Base Media File Format On Demand
ISO Base Media File Format Live

Figure 5: Describing MPEG DASH Profiles
(Diagram originally developed by Thomas Stockhammer, Qualcomm)

Content Descriptors for Protection, Accessibility, Content Rating: MPEG DASH offers a flexible set of descriptors for the media content that is being streamed. These descriptors spell out such elements as the rating of the content, the role of various components, accessibility features, DRM methods, camera views, frame packing, and the configuration of audio channels, among other things.

Common Encryption (defined by ISO/IEC 23001-7): One of the most important features of MPEG DASH is its use of Common Encryption, which standardizes signaling for what would otherwise be a number of non-interoperable, albeit widely used, encryption methods. Leveraging this standard, content owners or distributors can encrypt their content just once and then stream it to different clients with different DRM license systems. As a result, content owners can distribute their content freely and widely, while service providers can enjoy access to an open, interoperable ecosystem of vendors. In fact, Common Encryption is also used as the underlying standard for Ultraviolet, the Digital Entertainment Content Ecosystem's (DECE's) content authentication system. Common Encryption will be discussed in a bit more detail later in this paper.

Clock Drift Control for Live Content: In MPEG DASH, each media segment can include an associated Coordinated Universal Time (UTC) time, so that a client can control its clock drift and ensure that the encoder and decoder remain closely synchronized. Without this, a time difference between the encoder and decoder could cause the client play-back buffer to starve or overflow, due to different rates of video delivery and playback.

Metrics for Reporting the Client Session Experience: MPEG DASH has a set of well-defined quality metrics for tracking the user's session experience and sending the information back to the server.

MULTIPLE DRM METHODS & COMMON ENCRYPTION

As mentioned earlier, one of MPEG DASH's most important features is its use of Common Encryption, which standardizes signaling for a number of different, widely used encryption methods. Common Encryption (or "CENC") describes methods of standards-based encryption, along with key mapping of content to keys. CENC can be used by different DRM systems or Key Management Servers (KMS) to enable decryption of the same content, even with different vendors' equipment. It works by defining a common format for the encryption-related metadata required to decrypt the protected content. The details of key acquisition and storage, rights mapping, and compliance rules are not specified in the standard and are controlled by the DRM server. For example, DRM servers supporting Common Encryption will identify the decryption key with a key identifier (KID), but will not specify how the DRM server should locate or access the decryption key.

Using this standard, content owners or distributors can encrypt their content just once and then stream it to the various clients with their different DRM license systems. Each client receives the content decryption keys and other required data using its particular DRM system. This information is then transmitted in the MPD, enabling the client to stream the commonly encrypted content from the same server.

As a result, content owners can distribute their content freely and widely without the need for multiple encryptions. At the same time, cable operators and other video service providers can enjoy access to an open, interoperable ecosystem of content producers and equipment vendors.

USE CASES

The MPEG DASH spec supports both simple and advanced use cases of dynamic adaptive streaming. Moreover, the simple use cases can be gradually extended to more complex and advanced cases. In this section, we'll detail three such common use cases:

Live and On-Demand Content Delivery: MPEG DASH supports the delivery of both live and on-demand media content to subscribers through dynamic adaptive HTTP streaming. Like Adobe's HDS, Apple's HLS and Microsoft's Smooth Streaming platforms, DASH encodes the source video or audio content into file segments using a desired format. The segments are subsequently hosted on a regular HTTP server. Clients then play the stream by requesting the segments in a profile from a Web server, downloading them via HTTP.

MPEG DASH's great versatility in supporting both live and on-demand content has other benefits as well. For instance, these same capabilities also enable video service providers to deliver additional time-shifted services, such as network-based DVR (NDVR) and catch-up TV services, as explained below.

Time-Shifted Services (NDVR, catch-up TV, etc.): MPEG DASH supports the flexible delivery of time-shifted services, such as NDVRs and catch-up TV. For the enabling of time-shifted services, VoD assets, rather than live streams, are required. VoD assets formatted for MPEG DASH can be created using a transcoder. Additionally, a device commonly referred to as a Catcher can "catch" a live TV program and create a VoD asset, suitable for streaming after the live event. Because the VoD asset can be streamed in MPEG DASH in the same manner as the live content, the asset can be re-used and monetized by the operator.

Targeted Ad Insertion: Wherever there is video service, there is usually some kind of advertising content to monetize the service. ‘Traditional” ad insertion methods rely on a set of technologies based on the widely used protocols for distributing UDP/IP video: ad servers, ad splicers, and an ecosystem based on zoned ad delivery. But as video delivery transport has evolved via the new set of adaptive HTTP-based delivery protocols from Apple, Microsoft and Adobe, the ad insertion ecosystem has had to evolve to employ new, targeted technologies for insertion and delivery of revenue-generating commercials. The difficulty of inserting ads with the three existing delivery methods is that the protocols don’t support the same ad insertion methods, due to the inherent nature of how the protocols work.

MPEG DASH offers the dramatic potential to help enable adaptive bitrate advertising on many different types of client devices. DASH supports the dynamic insertion of advertising content into multimedia streams. In both live and on-demand use cases, commercials can be inserted either as a period between different multimedia periods or as a segment between different multimedia segments. As in the case with VoD trick modes, this would represent a significant improvement over the way that the three leading streaming protocols currently handle targeted ad insertion.

It is worth emphasizing that DASH supports a network-centric approach to ad insertion, as opposed to a client-centric approach in which the client pre-fetches ads and splices them locally based on interactions with external ad management systems. In DASH, the information about when ads play, which ads play, and how ads are delivered is transmitted through the MPD, which is created and distributed from the network.

PROSPECTS FOR INDUSTRY ADOPTION – CATALYSTS & CHALLENGES

With the development, ratification and introduction of the MPEG DASH platform, MPEG is attempting to rally the technology community behind a universal delivery standard for adaptive streaming media. Many tech companies have already enlisted in the effort, joining the new MPEG DASH Promoters Group to drive the broad adoption of the standard.

Not surprisingly, equipment vendors and content publishers are especially enthusiastic about the new standard. For instance, content publishers savor the opportunity to produce just a single set of media files that could run on all DASH-compatible electronics devices.

The key to MPEG DASH’s success, though, will be the participation of the three major proprietary players – Adobe, Apple, and Microsoft – that now divvy up the adaptive streaming market. While all three companies have contributed to the standard, their levels of support for DASH vary greatly. In particular, Apple’s backing is still in question because of the competitive advantages that its HLS platform stands to lose if DASH becomes the universal standard.

Besides such competitive issues, MPEG DASH faces potential intellectual property rights challenges as well. For example, it is still not clear if DASH will be saddled with royalty payments and, if so, where those royalties might be applied. This section will look at the intellectual property rights and other issues that may yet bedevil the new standard.

Unresolved Intellectual Property Rights Issues: In addition to the competitive issues, there are some unresolved intellectual property rights issues with MPEG DASH. For instance, when companies seek to contribute intellectual property to the MPEG standards effort, the contribution is usually accepted only if the property owner agrees to Reasonable and Non-Discriminatory (RAND) terms. In the case of DASH, though, it is not clear that all of the intellectual property rights (IPR) in the standard are covered by RAND terms.

Non-Interoperable DASH Profiles: Although MPEG DASH may have a single, unified name, it actually consists of a collection of different, non-interoperable profiles. So DASH doesn’t solve the problem of different, non-interoperable implementations unless DASH clients support all profiles. This would basically be equivalent to having a client that supports HLS, HDS and Smooth Streaming (which, incidentally, would also address the interoperability problem). Thus, the adoption of DASH doesn’t immediately imply a unified, interoperable ecosystem – a DASH world may suffer from the same interoperability issues that HLS, Smooth Streaming and HDS create today.

CONCLUSION

Now that MPEG DASH has been published by the ISO, it seems well on its way to becoming a solid, broadly accepted standard for the streaming media market. Three years in the making, DASH is poised to provide a universal platform for delivering streaming media content to multiple screens. Designed to be very flexible in nature, it promises to enable the re-use of existing technologies (containers, codecs, DRM, etc.), seamless switching between protocols, and perhaps most importantly, a high-quality experience for end users.

Furthermore, most of the tech industry’s major players have already lined up firmly behind DASH. The list of prominent supporters includes Akamai, Dolby, Samsung, Thomson, Netflix and, most notably, such leading streaming media providers as Microsoft and Adobe. Apple stands out as one of the few major tech players that haven’t fully enlisted in the effort yet. So there’s a great deal of hope in the industry that MPEG DASH could actually bring in all of the major players and realize its full market potential.

Yet several critical hurdles remain in the way of DASH’s dash to destiny. For one thing, Apple, Adobe and Microsoft must throw their full weight behind the standard and agree to make the switch from their proprietary HLS protocols in the future despite some clear competitive disadvantages of doing so. For another, all industry stakeholders must agree to make their intellectual property contributions to the standard royalty-free.

Neither of these developments will likely happen overnight. So it’s not clear yet if MPEG DASH will end up superseding the existing adaptive streaming formats as a true universal industry standard or merely co-existing with one or more of them in a still-fragmented market. As usual, the outcome will depend on what the major vendors decide to do. It will also depend on whether cable operators and other video service providers shift their multiscreen deployments and content offerings to DASH or continue on their current streaming paths. Only time will tell.