World Applied Sciences Journal 35 (2): 262-269, 2017
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2017.262.269

An Effective Analysis of MPEG DASH-Dynamic Adaptive Streaming over HTTP, a Streaming Standard for Multimedia

[1]B. Kalpana amd [2]P. Rangarajan

[1]Department of Information Technology, R.M.D Engineering College, Kavaraipettai, Tamil Nadu - 601206
[2]Department of Computer Science, R.M.D Engineering College Kavaraipettai, Tamil Nadu - 601206

Abstract: In this paper, we provide some insight and background into MPEG DASH-Dynamic Adaptive Streaming over HTTP, a streaming standard for multimedia over internet. It provides interoperability between various servers and devices and is instrumental for market growth. It supports a broad range of devices, such as PCs, TVs, laptops, set-top boxes, game consoles, tablets and mobiles phones. Previous adaptive streaming technologies - such as Apple HLS, Microsoft Smooth Streaming, Adobe HDS, etc. - have been released by vendors with limited support of company-independent streaming servers as well as playback clients. In this paper effective analysis of Dynamic Adaptive Streaming over HTTP with other streaming techniques such as Adobe, Apple and Microsoft is presented along with all their detailed transmission procedures including server, distribution and client components. Specifically, all exploited protocols along with their attributes are being described with streaming media preparation and the delivery process.

Key words: HTTP · DASH · Dynamic adaptive streaming

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. Early deployments of HTTP streaming used progressive download, where the client opens a TCP connection to a server and progressively downloads the multimedia content. As soon as enough data is available on the client, the client could eventually start with the decoding and rendering respectively. However, in case of bandwidth fluctuations, clients cannot react, resulting in service interruptions also referred to as stalls. Probably one of the first adaptive HTTP streaming solutions employed an explicit adaptation loop inspired by RTP-based streaming where clients perform bandwidth measurements and push the information towards the server. The server analyzes these reports and modifies the progressive download session on-demand. All existing adaptive HTTP streaming technologies, such as the proprietary Adobe HTTP Dynamic Streaming (HDS), Apple HTTP Live Streaming (HLS), Microsoft Smooth Streaming (MSS) and the only international standardized solution MPEG Dynamic Adaptive Streaming over HTTP (MPEG-DASH) follow nearly the same principle [1].

The basic idea is to generate multiple versions of the same content (e.g., different bitrates or spatial resolutions) and chop these versions into segments (e.g., two seconds).The segments are provided on a web server and can be downloaded through HTTP standard compliant GET requests. Typically, the relationship between the different versions is described by a manifest, which is provided to the client prior to the streaming session. The manifest represents the different qualities of the media content and the individual segments of each quality with HTTP Uniform Resource Locators (URLs). This structure provides the binding of the segments to the bitrate (resolution, etc.) among others (e.g., start time, duration of segments). As a consequence, each client will first request the manifest that contains the temporal and structural information for the media content and based on that information it will request the individual segments that fit best for its requirements. The adaptation to the

Corresponding Author: B. Kalpana, Department of Information Technology, R.M.D Engineering College, Kavaraipettai, Tamil Nadu - 601206, India.

bitrate or spatial resolution is done on the client side for each segment, e.g., the client can switch to a higher bitrate - if bandwidth permits - on a per segment basis, or to a lower bitrate - if bandwidth decreases. This has several advantages because the client knows its capabilities such as the received throughput, delay, device capabilities (e.g., screen resolution), etc. best.

MPEG DASH-Dynamic Adaptive Streaming over HTTP: MPEG-DASH (Dynamic Adaptive Streaming over HTTP, ISO/IEC 23009-1) is a vendor independent, international standard ratified by MPEG and ISO. Previous adaptive streaming technologies - such as Apple HLS, Microsoft Smooth Streaming, Adobe HDS, etc. - have been released by vendors with limited support of company-independent streaming servers as well as playback clients. As such a vendor-dependent situation is not desired, standardization bodies started a harmonization process, resulting in the ratification of MPEG-DASH in 2012.DASH is a streaming technique for delivering video to the internet user in anadaptive mode. This means that the stream is being deliveredto the client by recognizing and adapting to network'scapacities every time a new request takes place. Most internet users don't possess a fixed line, thus not having astable bandwidth for downloading media. This is whereDASH takes over by chopping the file into smaller pieces, the segments and downloading them in a dynamic way thusthe streaming is in a continuous, without interruptionplayback mode no matter which part of the stream is beingwatched while the rest is being downloaded [2]. Still, if the network used is proven to be inadequate and undesirable breaks interrupt, DASH seamlessly changes stream to a lower quality video, which is also stored in the server. An HTTP streaming system consists of Server Components, Distribution Components and Client Components. Concerning the Server Components, if a live streaming is taking place, an encoder is necessary that would encode media to be afterwards encapsulated for transference [3]. This mostly takes place if a live streaming is being delivered. Then the file is being elaborated in the segmenter where according to the file's duration, a group of multiple files is being generated. Segmenter is part of HTTP pseudo streaming in which video delivery includes a preparation stage. More particularly, DASH includes the Media Preparation Description (MPD) and file format definition [4]. MDP includes a wide variety of operations whichmostly concentrate in dividing the file to be delivered insegments. Then, MPD is responsible for allocating apartícular server address which then will be called by theclient using this specific URL address [5]. 3rd Generation Partnership Project (3GPP) took over the standardization of the DASH process. MPD process requiressegments definition and this is where 3GPP participates byundertaking the naming. Thus, all segments defined in thepreparation process receive 3GP file format (.3gp) and aspecific URL is being allocated in order for them to besmoothly retrieved [5]. 3GP file format was the initiallabeling for the segment files but more implementationscame along afterwards and more file formats had to beinvented according to every implementation sort for segments. After cropping the file into segments and defining themthrough 3GPP standards, all MPD information is containedinto a manifest file where the location of the media is being declared [5]. This file can basically be considered as an encoder of multiple files and XML was the best file type chosen to represent the manifest. Then, HTTP engages to retrieve all files required to substantiate the stream.Distribution components for the case of DASH refer to a simple HTTP web based server. Server is responsible for storing the stream and transferring the appropriate XMLmanifest files created by the segmenter to the client. The client component refers to the XML manifest file which identifies the URL of the stream.XML file, as mentioned above, also refer to all information associated with segments as well as their bit rate and other playback intelligence. Taking under consideration the client potentials, the client can call for specific segment types regarding bte ranges and other adaptive data without having to download the whole segments [3].

In recent years, MPEG-DASH has been integrated into new standardization efforts, e.g., the HTML5 Media Source Extensions (MSE) enabling the DASH playback via the HTML5 video and audio tag, as well as the HTML5 Encrypted Media Extensions (EME) enabling DRM-protected playback in web browsers. Furthermore, DRM-protection with MPEG-DASH is harmonized across different systems with the MPEG-CENC (Common Encryption) and MPEG-DASH playback on different SmartTV platforms is enabled via the integration in Hybrid broadcast broadband TV (HbbTV 1.5 and HbbTV 2.0). The usage of the MPEG-DASH standard has also been simplified by industry efforts around the DASH Industry Forum and their DASH-AVC/264 recommendations, as well as forward looking approaches such as the DASH-HEVC/265 recommendation on the usage of H.265/HEVC withinMPEG-DASH.

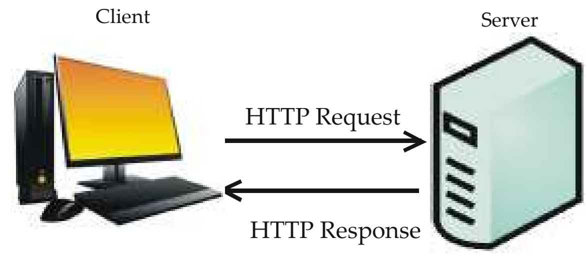

Fig. 1: HTTP Request/Response scheme

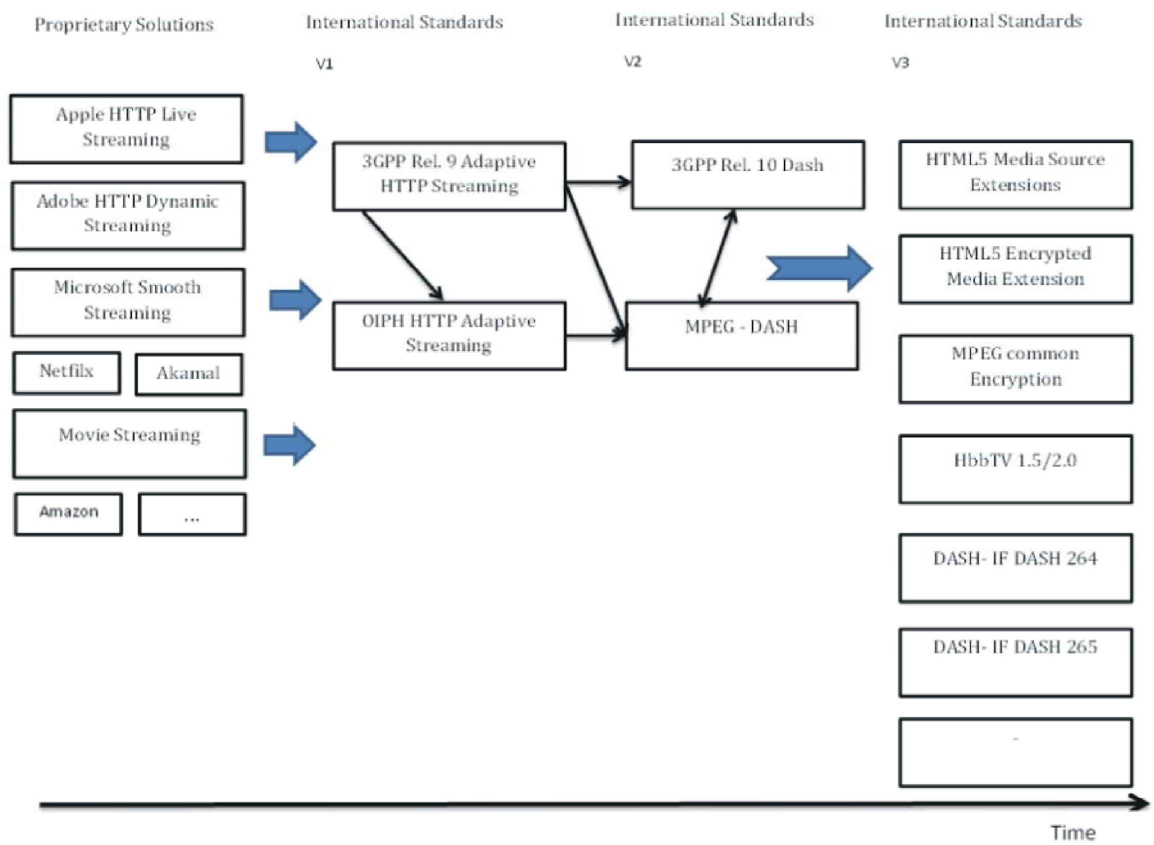

Fig. 2: Comparison of streaming techniques

Today, MPEG-DASH is gaining more and more deployments, accelerated by services such as Netflix or Google which recently switched to this new standard shown in Figure 2. With these two major sources of internet traffic, 50 % of total internet traffic is already in MPEG-DASH [6].

Current Implementation of MPEG-DASH: All HTTP-based adaptive streaming technologies have two components, the encoded A/V streams themselves and manifest files that identify the streams for the player and contain their URL addresses. For DASH, the actual A/V streams are called the Media Presentation, while the manifest file is called the Media Presentation Description. The media presentation defines the video sequence with one or more consecutive periods that break up the video from start to finish. Each period contains multiple adaptation sets that contain the content that comprises the audio/video experience. This content can be mixed, in which case there might be one adaptation set, or represented in elementary streams, as shown in Figure 1, enabling features like multiple language support for audio.

Each adaptation set contains multiple representations, each a single stream in the adaptive streaming experience. In the figure, Representation 1 is 640x480@500Kbps, while Representation 2 is 640x480@250Kbps.

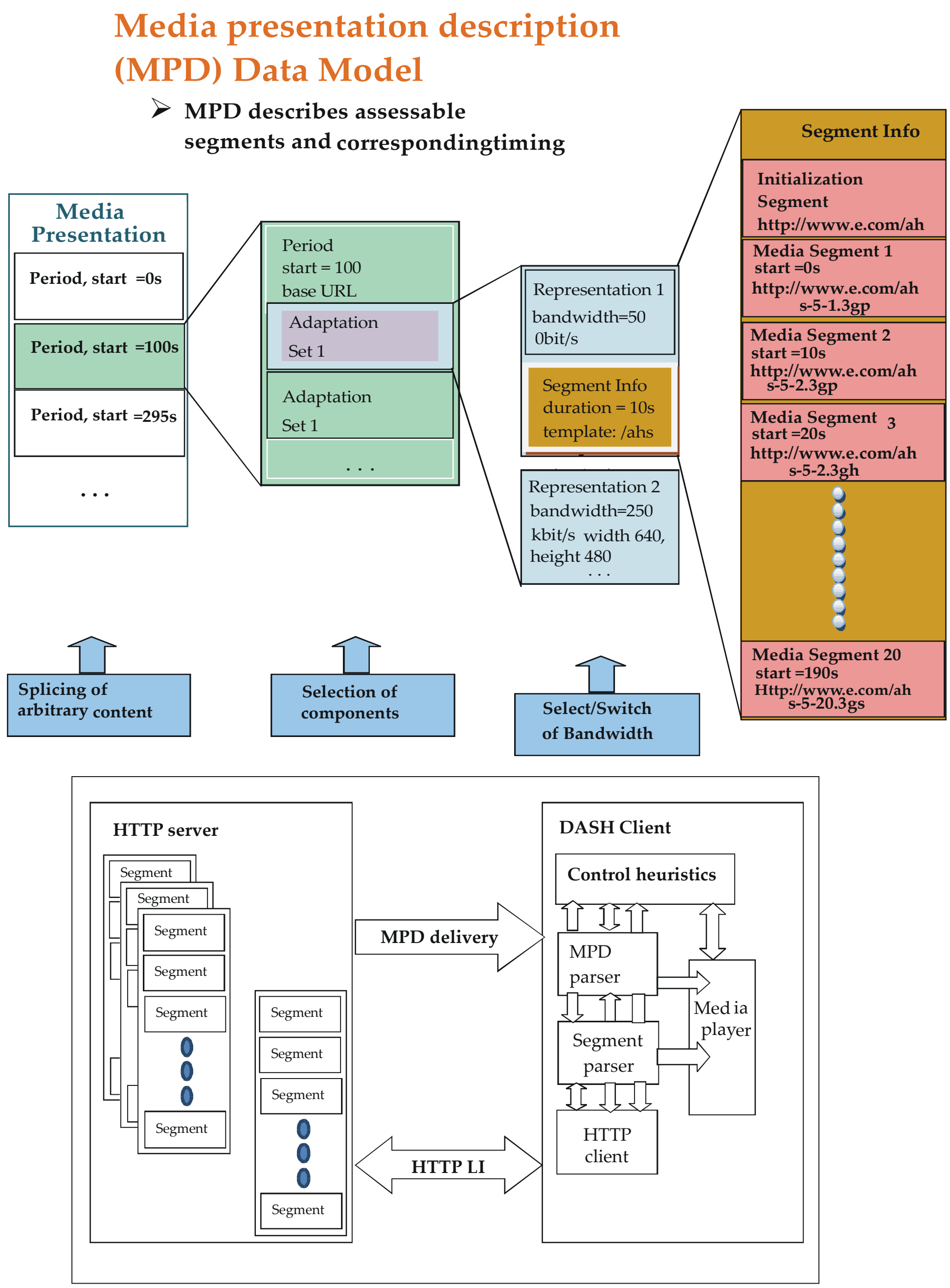

Fig. 3: Media Presentation Description

Each representation is divided into media segments, essentially the chunks of data that all HTTP-based adaptive streaming technologies use. Data chunks can be presented in discrete files, as in HLS, or as byte ranges in a single media file. Presentation in a single file helps improve file administration and caching efficiency as compared to chunked technologies that can create hundreds of thousands of files for a single audio/video event [7].

The DASH manifest file, called the Media Presentation Description as shown in Fig. 3, is an XML file that identifies the various content components and the location of all alternative streams. This enables the DASH player to identify and start playback of the initial segments, switch between representations as necessary to adapt to changing CPU and buffer status and change adaptation sets to respond to user input, like enabling/disabling subtitles or changing languages.

Adobe Adaptive Streaming: Adobe HTTP Dynamic Streaming (HDS) is a proprietary solution for on-demand and live delivery of high-quality content. Unlike UDP-based protocols such as RTP, Adobe HDS uses an adaptive bit-rate technique to deliver standard MP4 media over regular HTTP connections. Because it is adaptive, the streaming settings can be changed based on the current network conditions and the client hardware. HDS has support for file encryption and can be used to deliver HD quality video up to 1080p with bit-rates up to 6Mbps. HDS can use either H.264 or VP6 video and AAC or MP3 audio [8].

Adobe Media Server provides a high-performance, turnkey delivery server that can protect and deliver any content with a very simple publishing workflow, with support for multi bitrate delivery [9].

HTTP Dynamic Streaming was developed by Adobe as an alternative to their RTMP protocol. HDS allows for adaptive streaming over HTTP to any device that's compatible with Adobe Flash or Air. A big benefit to streaming with HDS instead of RTMP does not have to rely on a FMS, which significantly decreases the cost of operating the stream. Adobe has released a module for Apache, the most popular Open Source HTTP server that allows it to act as a streaming origin server. Because of the 97% market penetration that Flash Player has, HDS is a great choice for streaming to desktop computers. However, the Flash Player is not supported by Android and iOS, which limits practical use for broadcasting to mobile devices. RTMP requires Flash Media Server instatement as server component andFlash Media Player as client component. On this settlement, RTMP supports multiple files which are used to generate amulti-bitrate playback. It also supports bit rate alteration with file reuse of existing multiple bitrate encoded media.On the other hand HTTP requires Adobe Air as servercomponent and Flash Player on the client component side. Adobe HTTP supports standardized formats of VP6 andH246 file types which are chopped into fragments and usedfrom a specific manifest, the FMF which is just a similarformat to the one of XML [9].

Apple HTTP Live Streaming: HLS stands for HTTP Live Streaming and is a protocol developedby Apple for their iOS devices and QuickTime player. Support was added to Android 3.0 (Honeycomb) which has made HLS the ideal candidate for streaming to the widest range of mobile devices. A large number of client video players are available including the default HTML5 player and most mobile browsers.

HLS can be deployed using most HTTP servers (including Apache) or a number of commercial streaming servers such as Adobe FMS and Wowza. Also, many streaming services and CDNs can stream to HLS-compatible players via transmuxing, or dynamically repackaging existing video streams into their HLS compatible packages.

Apple developed HTTP Live Streaming (HLS), a protocolreferring to media streaming communications that wasprimarily established in QuickTime player and the iPhone. HLS applies DASH process in its simplest form byclipping the stream into smaller HTTP-based file downloads.Client may select any stream available which includes mediaencoded in a variety of data rates, allowing this way themost appropriate data rate adaptation. Apple's manifest is reflected on a playlist which contains the list of available qualities. This playlist is divided into smaller sub-playlists which include URLs for each M3U segment. Apple has documented an Internet Draft which wassubmitted to the IETF as a proposed standard. The specificRFC [10] is still in draft mode It is important to mention that HLS has found profitable ground through very noted company's applications such as Adobe's Flash MediaServer, Microsoft's Internet Information Services and Google's Android. Apple has also implemented the HLS protocol to all of its current iOS handheld devices which areiPhone, iPad and iPod Touch [11].

Microsoft Live Smooth Streaming: HTTP Smooth Streaming is Microsoft's foray into adaptive HTTP streaming that runs on their IIS web server and Silverlight player. The Silverlight player detects local bandwidth and CPU conditions and dynamically switches bitrates to offer uninterrupted streaming. HSS supports multiple audio and video codecs and is highly customizable. It is often used for very large-scale streams, such as NBC's online stream of the Olympic Games and is part of Netflix's streaming stack [12].

Streaming to Apple devices is possible when using the H.264 video codec. IIS is able to transmux HSS fragments into ones compatible with iOS 3.0 and later devices.

TV undeniably is moving to the mobile web. That is why adaptive HTTP streaming will be the backbone of any streaming strategy that aims at enabling superior viewing experiences on any device

Live Smooth Streaming is an adaptive media streaming overHTTP which mainly is an outspread of Internet InformationServices (IIS) Media Services web server application. As of 2009, Microsoft's DASH approach on Live SmoothStreaming specification was based on the ISO Base MediaFile Format. It was also standardized as the Protected Interoperable File Format (PIFF) and the manifest file isbased on XML file types [13]. The XML manifest file is used to convey the table of segments URLs to the client whichcontain audio and video material of fragmented MP4. Theonly difference with the rest of the DASH implementationsis that the specific segments may contain irrespectivelyaudio and video material [12]. This way, both media files that refer to a specific segment may be downloaded distinctivelydepending on the network's available quality. Though Server Component is Microsoft's IIS with theextension of Smooth Streaming, a Client Component is alsoindispensable. Thus, the client must install to his browser Microsoft's Silver light Player which supports H.264 andVC-1 video as well as AAC and WMA audio of codec agnostic material. Microsoft has also developed Smooth Streaming Porting Kid which is a used for other operating systems than Microsoft such as Apple iOS, Google Android and Linus. Microsoft has managed to evolve collaboration with NVIDIA graphics which resulted to demonstrate both live and on demand 1080p 3D HD videowith Smooth Streaming to clients outfitted with NVIDIA 3Dvision equipment.

Adaptive Streaming Techniques Implementation: MPEG-DASH and also Apple HLS can be used with ordinary HTTP-Servers such as Apache, Nginx, IIS, etc. Adobe, as well as Microsoft, are using server side mechanisms that need additional logic on the server.

MPEG-DASH is an international standard, ratified in 2012 and currently adopted by YouTube, Netflix, etc. Several members of different companies such as Microsoft, Adobe, Apple, Samsung, Akamai, Cisco, Dolby, Ericsson, Harmonic, Qualcomm, Netflix, Intel, Bitmovin, InterDigital, etc. have contributed to the standardization. The Apple HLS IETF Internet-Draft is driven by one company that is able to change its direction from one day to another. However, since May 2009, Apple has made no effort to move the HLS IETF Internet Draft to an international IETF standard.

Switching between multiple audio channels is especially important for multi-language content. MPEG-DASH supports this feature as well as Apple HLS and Microsoft Smooth Streaming. MPEG Common Encyrption (CENC) allows content encrypted once, to be compatible with multiple DRM systems. This is possible, as nearly every DRM system supports AES as content encryption method and only the license key exchange between the client and the server is different.

Ad insertion is possible in all formats through chunk substitution. This means that individual chunks of the original video will simply be replaced by chunks that contain advertisements. MPEG-DASH enables a standardized interface through periods that enables insertion in an efficient way which means that ordinary HTTP servers can still be used and no additional, proprietary logic is required to redirect requests for specific chunks to chunks that contain the advertisement [14].

Fast channel switching is a feature that is directly related to the chunk size, as smaller chunks allow faster channel change times than bigger chunks. Apple HLS typically uses 10 second chunks and is optimized for that chunk size. Adobe, Microsoft as well as MPEG-DASH are designed to work with 2 and 4 seconds chunks that allow faster channel change. Additionally, the overhead of the MP4 format used in MPEG-DASH and Microsoft has significantly lower overhead than the MPEG-2 Transport Stream (MP2TS) format used in Apple HLS [15]. The tradeoff between large and small chunk sizes is that small chunks sizes enable fast channel switching, reducing startup latency for some systems (some players start the playback only if they have received the first chunk fully) and allow a more flexible adaptation behavior. On the other hand, small chunk sizes reduce coding efficiency as Group of Pictures (GOP) need to be smaller and therefore the codec can probably not fully benefit from more temporal redundancies when larger GOPs are also allowed. Additionally, decreasing the chunk size introduces overhead on the network layer as every chunk needs to be requested through HTTP and therefore the protocol overhead gets introduced for each chunk [16]. Apple HLS v8, MSS and MPEG-DASH can be used with separate audio video content.

Adaptive Streaming Feature Comparison

FEATURE	Adobe HDS	Apple HLS	Microsoft Smooth	MPEG DASH
Deployment of standard HTTP Servers	×	✓	×	✓
Official International Standard (e.g., ISO/IEC MPEG)	×	×	×	✓
Multiple Audio Channels (e.g., Language, comments etc)	×	✓	✓	✓
Flexible content protection with common Encryption (DRM)	✓	✓	✓	✓
Closed captions/Subtitles	✓	✓	✓	✓
Efficient Ad Insertion	×	×	×	✓
Fast channel Switching	✓	×	✓	✓
Support multiple CDNs in parallel	×	×	×	✓
HTMLS Support	×	×	×	✓
Support in Hbb TV (Version 1.5)	×	×	×	✓
HEVC Ready (UHD/4K)	×	×	×	✓
Agnostics to video Codecs	×	×	×	✓
Agnostics to audio Codecs	×	×	×	✓
ISO Base Media File Format Segments	✓	×	✓	✓
MPEG-2 TS Segments	×	✓	×	✓
Segment format Extension beyond MPEG	×	×	×	✓
Support for multiplexed (Audio +Video content)	✓	✓	✓	✓
Support for non - multiplexed(Separate Audio, Video) content	×	✓	×	✓
Definition for Quality Metrics	×	×	×	✓
Client Logging & Reporting	×	×	×	✓
Client Failover	×	✓	×	✓
Remove and add Quality levels during Streaming	×	×	×	✓
Multiple video views	×	×	×	✓
Efficient trick Modes	×	×	×	✓

Conclusion and Simulation Evaluatio: DASH is an extraordinarily attractive technology for web producers, a single standard that should allow them to encode once and then securely distribute to a universe of players, from mobile to OTT and to the desktop via plug-ins or HTML5. In addition to not resolving the HTM5 codec issue, it's also unclear whether publishers will be charged for the privilege of producing files using the DASH spec, which could be a significant negative.

Mozilla has already indicated that they probably won't support the specification as currently written and Apple and Adobe have not affirmed if or when they will support the technology. An optimist would assume that the value of DASH to the streaming media marketplace would compel all stakeholders to make their contributes royalty free and convince Apple, Adobe and Mozilla to support the specification soon after its release. Until all this plays out, though, DASH may play a significant role in some markets, but won't reach its full potential.

At this point, HDS and HLS are more popular delivery methods than MPEG-DASH; they're just isn't broad support for MPEG DASH at this time. HDS and HLS are also more full-featured than MPEG DASH — but only for now.

However, these problems will be solved in the future. The major pros of MPEG DASH are good support among encoders, growing standardization; lower overhead will continue to drive more and more adoption of the technology.

REFERENCES

1. Timmerer, C. and C. Griwodz, 2012. Dynamic adaptive streaming over HTTP: from content creation to consumption. In Proceedings of the 20th ACM international conference on Multimedia, pp: 1533-1534). ACM.
2. De Cicco, L., G. Cofano and S. Mascolo, 2016. A hybrid model of the Akamai adaptive streaming control system. Nonlinear Analysis: Hybrid Systems, 21: 139-154.
3. Hwang, J., J. Lee and C. Yoo, 2016. Eliminating bandwidth estimation from adaptive video streaming in wireless networks. Signal Processing: Image Communication, 47: 242-251.
4. Michalos, M.G., S.P. Kessanidis and S.L. Nalmpantis, 2012. Dynamic adaptive streaming over HTTP. Journal of Engineering Science and Technology Review, 5(2): 30-34.
5. Xiang, S., M. Xing, L. Cai and J. Pan, 2015. Dynamic rate adaptation for adaptive video streaming in wireless networks. Signal Processing: Image Communication, 39: 305-315.

6. Liu, C., I. Bouazizi, M.M. Hannuksela and M. Gabbouj, 2012. Rate adaptation for dynamic adaptive streaming over HTTP in content distribution network. Signal Processing: Image Communication, 27(4): 288-311.
7. Tanwir, S. and H. Perros, 2016. Modeling live adaptive streaming over HTTP.Computer Communications, 85: 74-88.
8. Rainer, B., D. Posch and H. Hellwagner, XXXX. Investigating the Performance of Pull-based Dynamic Adaptive Streaming in NDN.
9. González, S., W. Castellanos, P. Guzmán, P. Arce and J.C. Guerri, 2016. Simulation and experimental testbed for adaptive video streaming in ad hoc networks. Ad Hoc Networks.
10. Sullivan, G.J., J.R. Ohm, W.J. Han and T. Wiegand, 2012. Overview of the high efficiency video coding (HEVC) standard. IEEE Transactions on Circuits and Systems for Video Technology, 22(12): 1649-1668.
11. Stockhammer, T., 2011. Dynamic adaptive streaming over HTTP--: standards and design principles. In Proceedings of the second annual ACM conference on Multimedia Systems, pp: 133-144). ACM.
12. Meng, S., J. Sun, Y. Duan and Z. Guo, 2016. Adaptive Video Streaming With Optimized Bitstream Extraction and PID-Based Quality Control. IEEE Transactions on Multimedia, 18(6): 1124-1137.
13. Bouten, N., R.D.O. Schmidt, J. Famaey, S. Latré, A. Pras and F. De Turck, 2015. QoE-driven in-network optimization for Adaptive Video Streaming based on packet sampling measurements. Computer networks, 81: 96-115.
14. Wu, T., S. Petrangeli, R. Huysegems, T. Bostoen and F. De Turck, 2016. Network-based video freeze detection and prediction in HTTP adaptive streaming. Computer Communications.
15. Meng, S., J. Sun, Y. Wang and Z. Guo, 2014. A PID-based quality control algorithm for SVC video streaming. In 2014 IEEE International Conference on Communications (ICC) (pp: 1693-1698). IEEE.
16. Guo, L., Y. Shi and W. Duan, 2005. A bandwidth allocation algorithm based on historical QoS metric for adaptive video streaming. In International Conference on Networking (pp: 917-925). Springer Berlin Heidelberg.