

Dynamic Adaptive Streaming over HTTP

M. G. Michalos¹, S. P. Kessanidis² and S. L. Nalmpantis^{3*}

¹School of Computing, Information Systems & Mathematics, Kingston University, London, United Kingdom.

²University of Piraeus, Dpt of Digital Systems, Greece.

³Dpt of Electrical Engineering, Kavala Institute of Technology, Kavala, Greece.

Received 21 August 2012; Accepted 29 September 2012

Abstract

In this paper, an in depth elaboration and analysis of adaptive streaming is taking place. Current implementation techniques of Dynamic Adaptive Streaming over HTTP from Adobe, Apple and Microsoft are being presented along with all detailed transmission procedures including server, distribution and client components. Specifically, all exploited protocols along with their attributes are being described and then the streaming media preparation and the delivery process are being represented. Following, a simulation of a DASH employment is taking place using JW Player 5 for Flash and HTML5 installed on a server with specific streaming video of 4 quality levels. Finally, measurements are taken through Quality Monitor plugin and reasoned out for a better comprehension of DASH experience.

Keywords: HTTP, DASH, dynamic adaptive streaming.

1. Introduction

Moving backwards in the early days of the internet where the first websites landed, the next most amazing element to embed in web pages apart from plain text, were pictures or even, later on, animation GIFs. Now days, with internet's massive evolution, web elements seem to meet no frontier. Music, Videos, High Definition Videos or even live streaming are some of the most popular web objects to be implemented on websites. In addition to this rapid web evolution comes the enormous outspread of handheld devices such as smartphones, tablets, netbooks, notebooks and laptops where all web appliances must be in compliance with.

Video embed for web today can be considered more as a necessity than a feature. Although it has been embraced by all biggest websites all over the world, it isn't flawless, like every other multimedia streaming service available on the web. Taking all the above under consideration, a new delivery service has been developed in order to render the best possible video streaming and it was named Dynamic Adaptive Streaming over HTTP (DASH). DASH is a streaming technology which set off in 2010 developing in parallel with MPEG [2] and managed to standardize only until recently, late 2011. Despite its infant development, the DASH experience isn't very far from most of the Internet users as the famous YouTube is based on the DASH initiative.

2. Protocol approach

Since Hyper Text Transfer Protocol (HTTP) has been chosen to deliver the video streaming, a protocol approach

should be materialized in order to better understand how DASH takes advantage of the existing network infrastructure. Internet Protocol (IP) is the keystone to the Internet structure because it is responsible for transferring datagrams (packets) across the Internet network. TCP on the other hand is a protocol that is responsible for carrying out reliable interconnectivity and with no loss of packets network connections.

TCP may sound like a guaranteed mean of streaming but through a closer look it isn't. Its error correction techniques may result on displaying a visual or audio impairment while the eventual and undesired method of playback would be to pause the streaming media and press play when packet transmission recurs [11]. Also, through TCP, video should initially be downloaded completely and then viewed while through DASH user has the ability to choose the part of the video that desires to view. This part of DASH is being fulfilled by the Real Time Messaging Protocol (RTMP) [2]. RTMP is based on TCP but dodges its lack of pre-downloading the video providing the ability of seeking through the video timeline and watching any part of it at the same time. Last, but not least HTTP is the protocol that bridges the website files through the World Wide Web (WWW) with the end user.

It is important to mention that HTTP serves almost all websites available on the Internet while it is responsible for delivering any kinds of files throughout the web such as images and html, php, xhtml and images. HTTP consists of two messages, Response and Request as shown in Fig. 1. HTTP Get is the most considerable Request HTTP message which is responsible for retrieving the address (URL) of a file from the server. But, apart from the messages, there are some concrete processes that reinforce DASH and emanate from HTTP. More specifically, HTTP downloading, progressive downloading and HTTP pseudo-streaming are

* E-mail address: stefnal@sch.gr

integral procedures that without their presence, DASH wouldn't be available [1].

HTTP downloading represents the simplest video carrying procedure as the user selects the video, downloads it and then views it. Although it sounds as a reliable procedure the disadvantage is that the user must download the whole video first in order to view it. Moving forward to HTTP progressive downloading [9], it is a procedure that many Internet users are familiar with since it includes multimedia downloading while the specific file is being viewed or generally played. It is in fact a not so trustworthy service as it requires a very fast Internet connection in order for the downloading to stay ahead of the playback. In order to overcome this clog, HTTP pseudo-streaming was invented. This feature of streaming can recognize the client's internet connection as well as other hardware features (CPU, RAM memory, HD compatibility etc.) and delivers the best available video quality by taking under consideration all the above factors [9]. The final outcome would be the best video quality streaming in compliance with the desired, most smooth viewing experience.

HTTP transcends in some other features also. Firstly, as standardized protocol it can overcome any firewall protections thus, making it even more easy to be approached by clients that are not so comfortable with the Internet use [12]. Secondly and most important, HTTP is a stateless protocol which means that a server assigns a unique connection to every client in order to view a video and when the client is done, server closes the connection without saving any recent information. To sum up, HTTP is a widely used protocol since it delivers the Internet to almost every user worldwide and it gives full control of the streaming procedure.

3. What's DASH all about?

DASH is all about delivering video to the internet user in an adaptive mode. This means that the stream is being delivered to the client by recognizing and adapting to network's capacities every time a new request takes place. Most internet users don't possess a fixed line, thus not having a stable bandwidth for downloading media. This is where DASH takes over by chopping the file into smaller pieces, the segments and downloading them in a dynamic way thus the streaming is in a continuous, without interruption playback mode no matter which part of the stream is being watched while the rest is being downloaded [8]. 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 [10]. 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 [3]. MDP includes a wide variety of operations which mostly concentrate in dividing the file to be delivered in segments. Then, MPD is responsible for allocating a

particular server address which then will be called by the client using this specific URL address [14].

3rd Generation Partnership Project (3GPP) took over the standardization of the DASH process. MPD process requires segments definition and this is where 3GPP participates by undertaking the naming. Thus, all segments defined in the preparation process receive 3GP file format (.3gp) and a specific URL is being allocated in order for them to be smoothly retrieved [14]. 3GP file format was the initial labeling for the segment files but more implementations came along afterwards and more file formats had to be invented according to every implementation sort for segments.

After cropping the file into segments and defining them through 3GPP standards, all MPD information is contained into a manifest file where the location of the media is being declared [14]. 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 XML manifest files created by the segmenter to the client.

Last, but not least, 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 byte ranges and other adaptive data without having to download the whole segments [10].

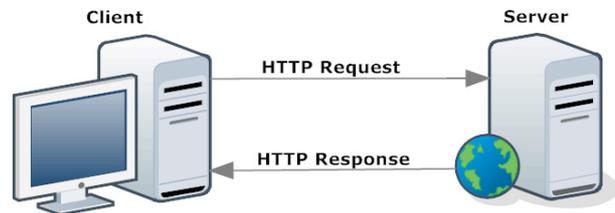


Fig. 1. HTTP Request/Response scheme.

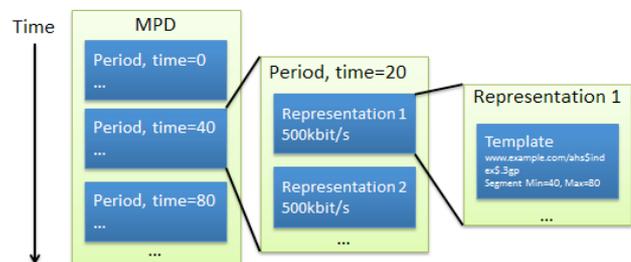


Fig. 2. MPD format [13]

4. Current implementations

Although Move Networks was the company that recognized and found the adaptive streaming technology, it was only until 2010 that it was awarded the Fundamental Patent for inventing adaptive streaming. The only difference between the process described above and Move Networks' patent is that the streaming file is being chopped in streamlets and not

segments. Nevertheless, giants of computer society such as Adobe, Apple and Microsoft had made a significant growth on adaptive streaming over HTTP by offering developers with tools for developing media delivery websites. Their appliances are already being enjoyed by millions of users in personal computers, tablets, even mobile phones taking into consideration the fact that YouTube and Vimeo are based on the aforementioned company's implementation foundations.

4.1. Adobe Adaptive Streaming

Through Flash, Adobe has developed streaming technologies for several years now but she soon acknowledged the need for adaptive playback and pioneered in adaptive web media delivery [4]. Adobe upholds adaptive streaming by presuming upon both RTMP and HTTP protocols but by implementing different server applications. RTMP requires Flash Media Server instatement as server component and Flash Media Player as client component. On this settlement, RTMP supports multiple files which are used to generate a multi-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 server component and Flash Player on the client component side [4]. Adobe HTTP supports standardized formats of VP6 and H264 file types which are chopped into fragments and used from a specific manifest, the FMF which is just a similar format to the one of XML.

4.2. Apple HTTP Live streaming

Apple developed HTTP Live Streaming (HLS), a protocol referring to media streaming communications that was primarily established in QuickTime player and the iPhone [5]. HLS applies DASH process in its simplest form by clipping the stream into smaller HTTP-based file downloads. Client may select any stream available which includes media encoded in a variety of data rates, allowing this way the most 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 [5].

Apple has documented an Internet Draft which was submitted to the IETF as a proposed standard. The specific RFC [16] is still in draft mode and as this paper is written it has reached Draft Version 08. It is important to mention that HLS has found profitable ground through very noted company's applications such as Adobe's Flash Media Server, 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 are iPhone, iPad and iPod Touch.

4.3. Microsoft Live Smooth Streaming

Live Smooth Streaming is an adaptive media streaming over HTTP which mainly is an outspread of Internet Information Services (IIS) Media Services web server application. As of 2009, Microsoft's DASH approach on Live Smooth Streaming specification was based on the ISO Base Media File Format. It was also standardized as the Protected Interoperable File Format (PIFF) and the manifest file is based on XML file types [6]. The XML manifest file is used to convey the table of segments URLs to the client which contain audio and video material of fragmented MP4. The only difference with the rest of the DASH implementations

is that the specific segments may contain irrespectively audio and video material [6]. This way, both media files that refer to a specific segment may be downloaded distinctively depending on the network's available quality.

Though Server Component is Microsoft's IIS with the extension of Smooth Streaming, a Client Component is also indispensable. Thus, the client must install to his browser Microsoft's Silverlight Player which supports H.264 and VC-1 video as well as AAC and WMA audio of codec-agnostic material. Microsoft has also developed Smooth Streaming Porting Kit which is a used for other operating systems than Microsoft such as Apple iOS, Google Android and Linus. Last but not least, Microsoft has managed to evolve collaboration with NVIDIA graphics which resulted to demonstrate both live and on demand 1080p 3D HD video with Smooth Streaming to clients outfitted with NVIDIA 3D vision equipment.

5. Simulation of Adaptive Streaming over HTTP using JW Player

Since DASH has been established as video technique not only for home entertainment delivery but mobile as well, many open source embeddable media players have been evolved. JW Player is an embeddable media player that supports most ordinary video and audio format and comes with a lot of plugins and modification options.

JW Player supports both Flash and HTML5 making streaming media accessible across multiple devices and internet browsers. It's important to mention player's accessibility as it is supported using Flash Player or HTML5 by all major browsers such as Google Chrome, Mozilla Firefox, Microsoft Internet Explorer, Opera, Safari and most widespread mobile operating systems such as Apple iOS (only HTML5) and Google Android. It also supports an easy to use Application Programming Interface (API) based on JavaScript for flexible manipulation by developers. Furthermore, it can be enhanced through Flash and JavaScript plugins allowing customized video delivery with amplified streaming experience. In addition, it supports video delivery and bitrate switching using HTTP Pseudo streaming and RTMP streaming protocols but unfortunately, for the time being, only through Flash exploitation. Last but not least, JW Player supports FLV/F4V, H.264/MP4, MP3 and AAC video and audio formats through Flash while HTML5 upholds formats that are being supported by the browsers.

5.1. Simulation appliance

In order to simulate Adaptive Streaming over HTTP, a new server account has been allocated with a domain at a datacenter in Texas USA. JW Player has been published in a single webpage named index.html to the specific domain through a common File Transfer Protocol (FTP) program. Then, a video has been recorded with a Canon 550D digital camera of 51" in which was subjoined an audio sequence of 44.1KHz and 128kbps. The specific video has been converted to .flv file in order to comply with the simplest mean of video delivery and to reciprocate with all browsers.

Moreover, the initial file has been compressed to multiple versions of size and bitrate as shown on Table 1.

Table 1. Multiple video characteristics used for adaptive streaming simulation.

Filename	Resolution/Bitrate	Resolution size	Ratio	Level
240p.flv	240	320x240	4:3	1
360p.flv	360	640x360	4:3	2
480p.flv	480	640x480	4:3	3
720p.flv	720	1280x720	16:9	4

After uploading the videos through FTP, index.html has been granted with the following HTML code.

```

<script type="text/javascript"
src="jwplayer.js"></script>
<div id="container">Loading the player...</div>
<script type="text/javascript">
jwplayer("container").setup({
flashplayer: "player.swf",
plugins: {
"qualitymonitor": {}
},
levels: [
{ bitrate: 240, file:
"http://www.website.com/240p.flv", width: 320 },
{ bitrate: 360, file:
"http://www.website.com/360p.flv", width: 640 },
{ bitrate: 480, file:
"http://www.website.com/480p.flv", width: 640 },
{ bitrate: 720, file:
"http://www.website.com/720p.flv", width: 1280 } ],
provider: "http",
"http.startparam"."starttime" });
</script>
    
```

As described above, it is evident from the HTML source code that both JavaScript and Flash are being supported through jwplayer.js and player.swf respectively [7]. Furthermore, JW Player supports HTTP streaming through the “provider” command where HTTP is being declared. Also, all bitrates along with the video source and width are being mentioned. Another command that is being referred in the source code is a plugin called quality monitor. Among many plugins that JW Player supports, Quality Monitor is a real time depicter of video streaming features such as bandwidth used by video delivery (Green line), dropped frames in case of frames rate alteration (Red line), width (which in this case is still Blue line) and level (White line) [7]. Level represents the multiple versions of video quality as shown in Table 1.

5.2. Simulation results

Simulation has been conducted through personal computers, laptops and netbooks that use the internet for regular use and of course, to view streams of common videos. The video, as described above has been published and measurements have been taken through a variety of browsers, internet connections and areas of Peloponnese contingent, Attica and Thessaloniki.

In Fig. 3 and 4 one can see a smooth playback which is being carried out in a very close range of internet line bandwidth. Specifically Quality monitor consumes bandwidth line of 1024 kbps and 1124 kbps for Fig. 3 and Fig. 4 respectively. Although it’s small difference of only 100 kbps, stream in Fig. 4 displays better and smoother

playback, thus consolidating on level 3 video of 360p. Stream in Figure stabilizes in level 4 on 240p and despite the small difference of bandwidth, one can see that bandwidth starts at a low range and also steadies quite low in contrast with Fig. 4.

In Fig. 5 one can see that playback from Level 1 drops to Level 4 and then swarms to Level 3. It is clear by the green line that bandwidth of user’s line is quite low but, surprisingly blue line that represents width rises up to 1680 pixels. This happens because user’s hardware is quite advanced and since the bandwidth line can’t keep up in order to serve maximum streaming experience JW Player maintains bitrate but proliferates window width in order to render better viewing quality.

On the other hand, Fig. 6 represents a smooth playback of the stream as it consolidates at Level 2 of 480p where all lines seem to maintain a stabilized flow.

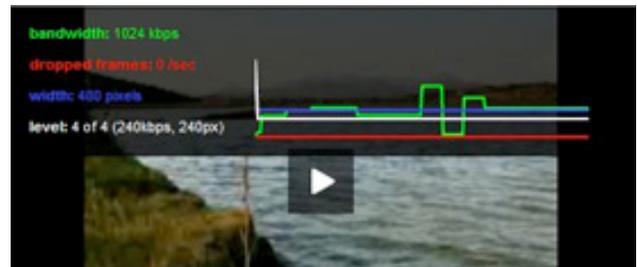


Fig. 3. Level 4 playback in Attica with 24Mbits aDSL line.



Fig. 4. Level 3 Playback in Tripolis with 24Mbits aDSL line.

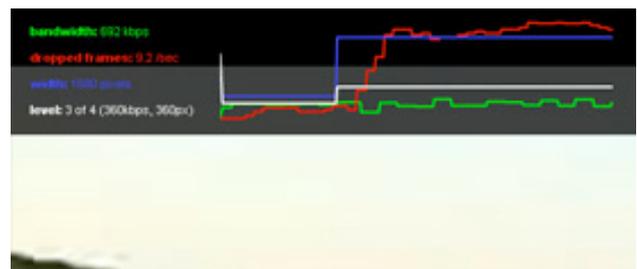


Fig. 5. Level 3 playback in Tripolis with 24Mbits aDSL line.

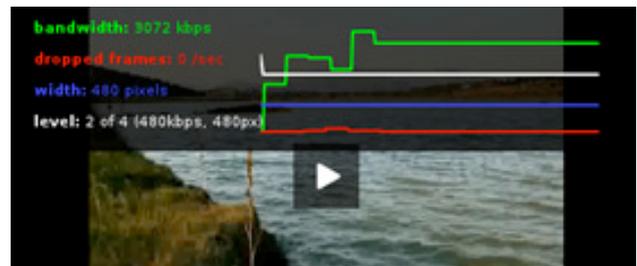


Fig. 6. Level 2 Playback in Thessaloniki with 24Mbits aDSL line.

In Fig. 7 and 8 one can notice a very interesting playback occasion. Level 1 video is 720p of data rates which is considered an HD stream. Although both streams seem to behave very spacious with bandwidth consumption, none

swarms to Level 1 playback. This is happening because both users' hardware doesn't cope with HD playback thus, it intrudes 480p playback.



Fig. 7. Level 2 playback in Attica with 24Mbps aDSL line.

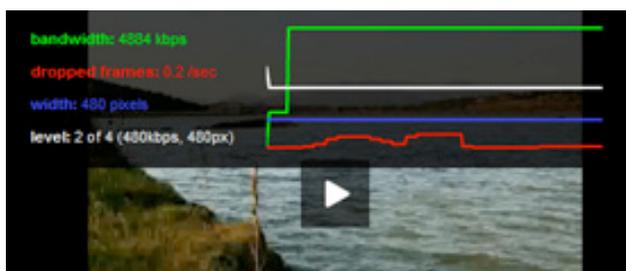


Fig. 8. Level 2 Playback in Attica with 24Mbps aDSL line.

6. Conclusion and Simulation Evaluation

From the simulation performed, one can realize that adaptive streaming over HTTP is an accessible multimedia networking technology from a developer's but also a user's perspective. Developers can elaborate small streaming distribution websites with open source programming toolkits while users can already have access to DASH through Content Distribution Networks (CDNs) where movies or live streaming media are being stored for broadcast. Larger delivery networks are being elaborated with more complex and expensive toolkits in order to serve greater amount of users and data.

Adaptive streaming technology proved to veil the chasm that full media downloading has been performing until recently. In addition to that, the already established and firewall-compatible HTTP has been chosen to deliver streaming media. It is a fact though that most of the adaptive process is taking place on the user's side as implementations take advantage of the internet bandwidth and the available computer hardware.

Over the recent years, Video on Demand (VoD) services has raised gaining more enthusiasts. A future evolution of DASH could include Scalable Video Coding (SVC) [15] as well as Variable Bitrate (VBR) streams. Specifically for the second instance, it is very important to implement such a process since adaptivity may stick in a non-representative bitrate of the media [1].

References

1. A brief tutorial on adaptive streaming. (2011). EBU Technical. [Internet]. <<http://tech.ebu.ch/news/>>. [Accessed 22nd December 2011].
2. Christopher Müller and Christian Timmerer. (2011). A Test-Bed for the Dynamic Adaptive Streaming over HTTP featuring Session Mobility. ACM Multimedia Systems 2011. February 23 – February 25, San Jose, California, USA. ACM, 2011.
3. Frank Hartung, Sinan Kesici and Daniel Catrein. (2011). DRM Protected Dynamic Adaptive HTTP Streaming. ACM Multimedia Systems 2011. February 23 – February 25, San Jose, California, USA. ACM, 2011.
4. HTTP Dynamic Streaming. (2011). Adobe. <<http://www.adobe.com/>>. [Accessed 23rd of December 2011].
5. HTTP Live Streaming Overview. (2011). Apple. <<http://www.apple.com/>>. [Accessed 27th December 2011].
6. IIS Smooth Streaming Technical Overview. (2011). Microsoft. <<http://www.microsoft.com/>>. [Accessed 25th December 2011].
7. JW Player 5 for Flash and HTML5 (2011). Longtail. [Internet]. <<http://www.longtailvideo.com/players/>>. [Accessed 20th December 2011].
8. Kashaf Mazhar, Compliance Procedures for Dynamic Adaptive Streaming over HTTP (DASH). Master's Degree Project, The Royal Institute of Technology, 2011.
9. Luciano Romero. A Dynamic Adaptive HTTP Streaming Video Service for Google Android. Master's Degree Project, The Royal Institute of Technology, 2011.
10. Muhammad Siraj. HTTP Based Adaptive Streaming over HSPA. Master's Degree Project, The Royal Institute of Technology, 2011.
11. Saamer Akhshabi, Ali C. Begen and Constantine Dovrolis. (2011). An Experimental Evaluation of Rate-Adaptation Algorithms in Adaptive Streaming over HTTP. ACM Multimedia Systems 2011. February 23 – February 25, San Jose, California, USA. ACM, 2011.
12. Sodagar Iraj. (2011). The MPEG-DASH Standard for Multimedia Streaming Over the Internet, Multimedia, IEEE, Vol. 18 (4), pp. 62 – 67.
13. Thomas Stockhammer. (2011). Dynamic Adaptive Streaming over HTTP – Standards and Design Principles. ACM Multimedia Systems 2011. February 23 – February 25, San Jose, California, USA. ACM, 2011.
14. Thorsten Lohmar, Torbjörn Einarsson, Per Fröjdh, Frédéric Gabin and Markus Kampmann. (2011). Dynamic Adaptive HTTP Streaming of Live Content. International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM 2011), June 20 – June 23, Lucca, Italy. IEEE, 2011.
15. Yago Sánchez, Thomas Schierl, Cornelius Hellge, Thomas Wiegand and Danny De Vleeschauwer. (2011). iDASH: Improved Dynamic Adaptive Streaming over HTTP using Scalable Video Coding. ACM Multimedia Systems 2011. February 23 – February 25, San Jose, California, USA. ACM, 2011.
16. R. Pantos, and W. May. (2012). HTTP Live Streaming. [Online]. Available: <http://tools.ietf.org/html/draft-pantos-http-live-streaming-06>.