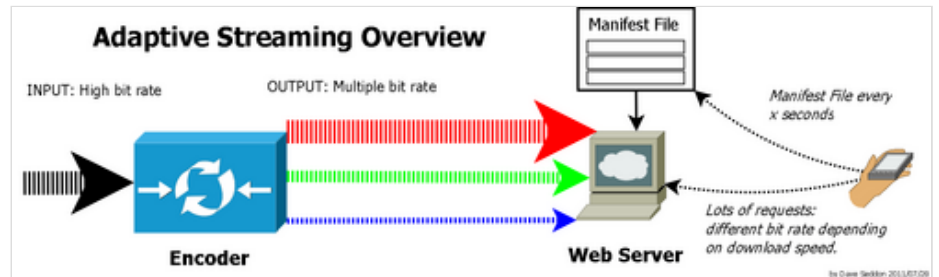




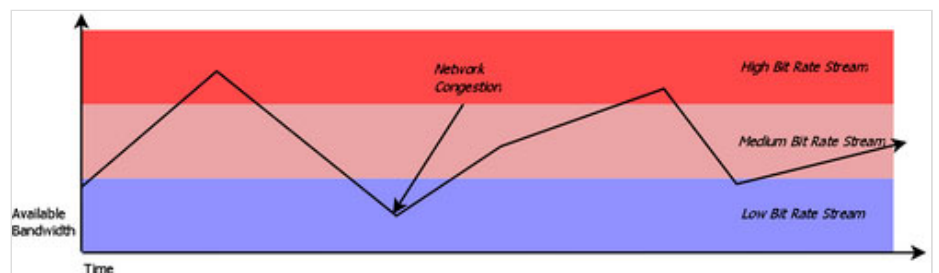
Adaptive bitrate streaming

Adaptive bitrate streaming is a technique used in streaming multimedia over computer networks.

While in the past most video or audio streaming technologies utilized streaming protocols such as RTP with RTSP, today's adaptive streaming technologies are based almost exclusively on HTTP,^[1] and are designed to work efficiently over large distributed HTTP networks.



Adaptive streaming overview



Adaptive streaming in action

Adaptive bitrate streaming

works by detecting a user's bandwidth and CPU capacity in real time, adjusting the quality of the media stream accordingly.^[2] It requires the use of an encoder which encodes a single source media (video or audio) at multiple bit rates. The player client^[3] switches between streaming the different encodings depending on available resources.^[4] This results in providing very little buffering, faster start times and a good experience for both high-end and low-end connections.^[5]

More specifically, adaptive bitrate streaming is a method of video streaming over HTTP where the source content is encoded at multiple bit rates. Each of the different bit rate streams are segmented into small multi-second parts.^[6] The segment size can vary depending on the particular implementation, but they are typically between two and ten seconds.^{[4][6]} First, the client downloads a manifest file that describes the available stream segments and their respective bit rates. During stream start-up, the client usually requests the segments from the lowest bit rate stream. If the client finds that the network throughput is greater than the bit rate of the downloaded segment, then it will request a higher bit rate segment. Later, if the client finds that the network throughput has deteriorated, it will request a lower bit rate segment. An adaptive bitrate (ABR) algorithm in the client performs the key function of deciding which bit rate segments to download, based on the current state of the network. Several types of ABR algorithms are in commercial use: throughput-based algorithms use the throughput achieved in recent prior downloads for decision-making (e.g., throughput rule in dash.js (<https://reference.dashif.org/dash.js>)), buffer-based algorithms use only the client's current

buffer level (e.g., BOLA^[7] in dash.js (<https://reference.dashif.org/dash.js>)), and hybrid algorithms combine both types of information (e.g., DYNAMIC^[8] in dash.js (<https://reference.dashif.org/dash.js>)).

Current uses

Post-production houses, content delivery networks and studios use adaptive bit rate technology in order to provide consumers with higher quality video using less manpower and fewer resources. The creation of multiple video outputs, particularly for adaptive bit rate streaming, adds great value to consumers.^[9] If the technology is working properly, the end user or consumer's content should play back without interruption and potentially go unnoticed. Media companies have been actively using adaptive bit rate technology for many years now and it has essentially become standard practice for high-end streaming providers; permitting little buffering when streaming high-resolution feeds (begins with low-resolution and climbs).

Benefits of adaptive bitrate streaming

Traditional server-driven adaptive bitrate streaming provides consumers of streaming media with the best-possible experience, since the media server automatically adapts to any changes in each user's network and playback conditions.^[10] The media and entertainment industry also benefit from adaptive bitrate streaming. As the video space grows, content delivery networks and video providers can provide customers with a superior viewing experience. Adaptive bitrate technology requires additional encoding, but simplifies the overall workflow and creates better results.

HTTP-based adaptive bitrate streaming technologies yield additional benefits over traditional server-driven adaptive bitrate streaming. First, since the streaming technology is built on top of HTTP, contrary to RTP-based adaptive streaming, the packets have no difficulties traversing firewall and NAT devices. Second, since HTTP streaming is purely client-driven, all adaptation logic resides at the client. This reduces the requirement of persistent connections between server and client application. Furthermore, the server is not required to maintain session state information on each client, increasing scalability. Finally, existing HTTP delivery infrastructure, such as HTTP caches and servers can be seamlessly adopted.^{[11][12][13][14]}

A scalable CDN is used to deliver media streaming to an Internet audience. The CDN receives the stream from the source at its Origin server, then replicates it to many or all of its Edge cache servers. The end-user requests the stream and is redirected to the "closest" Edge server. This can be tested using libdash^[15] and the Distributed DASH (D-DASH) dataset,^[16] which has several mirrors across Europe, Asia and the US. The use of HTTP-based adaptive streaming allows the Edge server to run a simple HTTP server software, whose licence cost is cheap or free, reducing software licensing cost, compared to costly media server licences (e.g. Adobe Flash Media Streaming Server). The CDN cost for HTTP streaming media is then similar to HTTP web caching CDN cost.

History

Adaptive bit rate over HTTP was created by the DVD Forum at the WG1 Special Streaming group in October 2002. The group was co-chaired by [Toshiba](#) and [Phoenix Technologies](#), The expert group count with the collaboration of [Microsoft](#), [Apple Computer](#), [DTS Inc.](#), [Warner Brothers](#), [20th Century Fox](#), [Digital Deluxe](#), [Disney](#), [Macromedia](#) and [Akamai](#). The technology was originally called [DVDoverIP](#) and was an integral effort of the DVD ENAV book.^[17] The concept came from storing MPEG-1 and MPEG-2 DVD TS Sectors into small 2KB files, which will be served using an HTTP server to the player. The MPEG-1 segments provided the lower bandwidth stream, while the MPEG-2 provided a higher bit rate stream. The original XML schema provided a simple playlist of bit rates, languages and url servers. The first working prototype was presented to the DVD Forum by Phoenix Technologies at the [Harman Kardon Lab](#) in Villingen Germany.

Implementations

Adaptive bit rate streaming was introduced by Move Networks in 2006 ^[18] and is now being developed and utilized by [Adobe Systems](#), [Apple](#), [Microsoft](#) and [Octoshape](#).^[19] In October 2010, Move Networks was awarded a patent for their adaptive bit rate streaming (US patent number 7818444).^[20]

Dynamic Adaptive Streaming over HTTP (DASH)

Dynamic Adaptive Streaming over HTTP (DASH), also known as MPEG-DASH, is the only adaptive bit-rate HTTP-based streaming solution that is an international standard^[21] MPEG-DASH technology was developed under [MPEG](#). Work on DASH started in 2010 and became a Draft International Standard in January 2011 and an International Standard in November 2011.^{[21][22][23]} The MPEG-DASH standard was published as [ISO/IEC 23009-1:2012](#) (http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=57623) in April, 2012.

MPEG-DASH is a technology related to [Adobe Systems HTTP Dynamic Streaming](#), [Apple Inc. HTTP Live Streaming \(HLS\)](#) and [Microsoft Smooth Streaming](#).^[24] DASH is based on Adaptive HTTP streaming (AHS) in [3GPP Release 9](#) and on HTTP Adaptive Streaming (HAS) in [Open IPTV Forum Release 2](#).^[25] As part of their collaboration with MPEG, 3GPP Release 10 has adopted DASH (with specific codecs and operating modes) for use over wireless networks.^[25]

The goal of standardizing an adaptive streaming solution is to assure the market that the solution can work universally, unlike other solutions that are more specific to certain vendors, such as Apple's HLS, Microsoft's Smooth Streaming, or Adobe's HDS.

Available implementations are the HTML5-based *bitdash* MPEG-DASH player^[26] as well as the open source C++-based DASH client access library *libdash* of [bitmovin GmbH](#),^[15] the DASH tools of the [Institute of Information Technology \(ITEC\)](#) at [Alpen-Adria University Klagenfurt](#),^{[3][27]} the multimedia framework of the [GPAC group](#) at [Telecom ParisTech](#),^[28] and the *dash.js*^[29] player of the [DASH-IF](#).

Apple HTTP Live Streaming (HLS)

HTTP Live Streaming (HLS) is an HTTP-based media streaming communications protocol implemented by Apple Inc. as part of QuickTime X and iOS. HLS supports both live and Video on demand content. It works by breaking down media streams or files into short pieces (media segments) which are stored as MPEG-TS or fragmented MP4 files. This is typically done at multiple bitrates using a stream or file segmenter application, also known as a packager. One such segmenter implementation is provided by Apple.^[30] Additional packagers are available, including free / open source offerings like Google's Shaka Packager ^[31] and various commercial tools as well - such as Unified Streaming.^[32] The segmenter is also responsible for producing a set of playlist files in the M3U8 format which describe the media chunks. Each playlist is specific to a given bitrate, and contains the relative or absolute URLs to the chunks for that bitrate. The client is then responsible for requesting the appropriate playlist depending on available bandwidth.

HTTP Live Streaming is a standard feature in the iPhone 3.0 and newer versions.^[33]

Apple has submitted its solution to the IETF for consideration as an Informational Request for Comments.^[34] This was officially accepted as RFC 8216 (<https://www.rfc-editor.org/rfc/rfc8216>) A number of proprietary and open source solutions exist for both the server implementation (segmenter) and the client player.

HLS streams can be identified by the playlist URL format extension of m3u8 or MIME type of application/vnd.apple.mpegurl.^[35] These adaptive streams can be made available in many different bitrates and the client device interacts with the server to obtain the best available bitrate which can reliably be delivered.

Playback of HLS is supported on many platforms including Safari and native apps on macOS / iOS, Microsoft Edge on Windows 10, ExoPlayer on Android, and the Roku platform. Many Smart TVs also have native support for HLS. Playing HLS on other platforms like Chrome / Firefox is typically achieved via a browser / JavaScript player implementation. Many open source and commercial players are available including hls.js, video.js http-streaming, BitMovin, JWPlayer, THEOplayer, etc.

Adobe HTTP Dynamic Streaming (HDS)

"HTTP Dynamic streaming is the process of efficiently delivering streaming video to users by dynamically switching among different streams of varying quality and size during playback. This provides users with the best possible viewing experience their bandwidth and local computer hardware (CPU) can support. Another major goal of dynamic streaming is to make this process smooth and seamless to users, so that if up-scaling or down-scaling the quality of the stream is necessary, it is a smooth and nearly unnoticeable switch without disrupting the continuous playback."^[36]

The latest versions of Flash Player and Flash Media Server support adaptive bit-rate streaming over the traditional RTMP protocol, as well as HTTP, similar to the HTTP-based solutions from Apple and Microsoft,^[37] HTTP dynamic streaming being supported in Flash Player 10.1 and later.^[38] HTTP-

based streaming has the advantage of not requiring any firewall ports being opened outside of the normal ports used by web browsers. HTTP-based streaming also allows video fragments to be cached by browsers, proxies, and CDNs, drastically reducing the load on the source server.

Microsoft Smooth Streaming (MSS)

Smooth Streaming is an IIS Media Services extension that enables adaptive streaming of media to clients over HTTP.^[39] The format specification is based on the ISO base media file format and standardized by Microsoft as the Protected Interoperable File Format.^[40] Microsoft is actively involved with 3GPP, MPEG and DECE organizations' efforts to standardize adaptive bit-rate HTTP streaming. Microsoft provides Smooth Streaming Client software development kits for Silverlight and Windows Phone 7, as well as a Smooth Streaming Porting Kit that can be used for other client operating systems, such as Apple iOS, Android, and Linux.^[41] IIS Media Services 4.0, released in November 2010, introduced a feature which enables Live Smooth Streaming H.264/AAC videos to be dynamically repackaged into the Apple HTTP Adaptive Streaming format and delivered to iOS devices without the need for re-encoding. Microsoft has successfully demonstrated delivery of both live and on-demand 1080p HD video with Smooth Streaming to Silverlight clients. In 2010, Microsoft also partnered with NVIDIA to demonstrate live streaming of 1080p stereoscopic 3D video to PCs equipped with NVIDIA 3D Vision technology.^[42]

Common Media Application Format (CMAF)

CMAF is a presentation container format used for the delivery of both HLS and MPEG-DASH. Hence it is intended to simplify delivery of HTTP-based streaming media. It was proposed in 2016 by Apple and Microsoft and officially published in 2018.^[43]

QuavStreams Adaptive Streaming over HTTP

QuavStreams Adaptive Streaming is a multimedia streaming technology developed by Quavlive. The streaming server is an HTTP server that has multiple versions of each video, encoded at different bitrates and resolutions. The server delivers the encoded video/audio frames switching from one level to another, according to the current available bandwidth. The control is entirely server-based, so the client does not need special additional features. The streaming control employs feedback control theory.^[44] Currently, QuavStreams supports H.264/MP3 codecs muxed into the FLV container and VP8/Vorbis codecs muxed into the WEBM container.

Uplynk

Uplynk delivers HD adaptive bitrate streaming to multiple platforms, including iOS, Android, Windows Mac, Linux, and Roku, across various browser combinations, by encoding video in the cloud using a single non-proprietary adaptive streaming format. Rather than streaming and storing multiple formats for different platforms and devices, Uplynk stores and streams only one. The first studio to use this technology for delivery was Disney–ABC Television Group, using it for video encoding for web, mobile and tablet streaming apps on the ABC Player, ABC Family and Watch Disney apps, as well as the live Watch Disney Channel, Watch Disney Junior, and Watch Disney XD.^{[45][46]}

Self-learning clients

In recent years, the benefits of self-learning algorithms in adaptive bitrate streaming have been investigated in academia. While most of the initial self-learning approaches are implemented at the server-side^{[47][48][49]} (e.g. performing admission control using reinforcement learning or artificial neural networks), more recent research is focusing on the development of self-learning HTTP Adaptive Streaming clients. Multiple approaches have been presented in literature using the SARSA^[50] or Q-learning^[51] algorithm. In all of these approaches, the client state is modeled using, among others, information about the current perceived network throughput and buffer filling level. Based on this information, the self-learning client autonomously decides which quality level to select for the next video segment. The learning process is steered using feedback information, representing the Quality of Experience (QoE) (e.g. based on the quality level, the number of switches and the number of video freezes). Furthermore, it was shown that multi-agent Q-learning can be applied to improve QoE fairness among multiple adaptive streaming clients.^[52]

Criticisms

HTTP-based adaptive bit rate technologies are significantly more operationally complex than traditional streaming technologies. Some of the documented considerations are things such as additional storage and encoding costs, and challenges with maintaining quality globally. There have also been some interesting dynamics found around the interactions between complex adaptive bit rate logic competing with complex TCP flow control logic.^{[11][53] [54] [55][56]}

However, these criticisms have been outweighed in practice by the economics and scalability of HTTP delivery: whereas non-HTTP streaming solutions require massive deployment of specialized streaming server infrastructure, HTTP-based adaptive bit-rate streaming can leverage the same HTTP web servers used to deliver all other content over the Internet.

With no single clearly defined or open standard for the digital rights management used in the above methods, there is no 100% compatible way of delivering restricted or time-sensitive content to any device or player. This also proves to be a problem with digital rights management being employed by any streaming protocol.

The method of segmenting files into smaller files used by some implementations (as used by HTTP Live Streaming) could be deemed unnecessary due to the ability of HTTP clients to request byte ranges from a single video asset file that could have multiple video tracks at differing bit rates with the manifest file only indicating track number and bit rate. However, this approach allows for serving of chunks by any simple HTTP server and so therefore guarantees CDN compatibility. Implementations using byte ranges such as Microsoft Smooth Streaming require a dedicated HTTP server such as IIS to respond to the requests for video asset chunks.

See also

- Multiple description coding
- Hierarchical modulation – alternative with reduced storage and authoring demands

References

1. Saamer Akhshabi; Ali C. Begen; Constantine Dovrolis (2011). *An Experimental Evaluation of Rate-Adaptation Algorithms in Adaptive Streaming over HTTP*. In Proceedings of the second annual ACM conference on Multimedia systems (MMSys '11). New York, NY, USA: ACM.
2. A. Bentaleb, B. Taani, A. Begen, C. Timmermer, and R. Zimmermann, "A Survey on Bitrate Adaptation Schemes for Streaming Media over HTTP", In *IEEE Communications Surveys & Tutorials* (IEEE COMST), Volume 1 Issue 1, pp. 1-1, 2018. (<https://ieeexplore.ieee.org/document/8424813/>)
3. DASH at ITEC, VLC Plugin, DASHEncoder and Dataset (<http://www-itec.uni-klu.ac.at/dash/>) by C. Mueller, S. Lederer, C. Timmerer
4. "Proceedings Template – WORD" (<http://www-itec.uni-klu.ac.at/bib/files/p37-mueller.pdf>) (PDF). Retrieved 16 December 2017.
5. Gannes, Liz (10 June 2009). "The Next Big Thing in Video: Adaptive Bitrate Streaming" (<https://web.archive.org/web/20100619225207/http://pro.gigaom.com/2009/06/how-to-deliver-as-much-video-as-users-can-take/>). Archived from the original (<http://pro.gigaom.com/2009/06/how-to-deliver-as-much-video-as-users-can-take/>) on 19 June 2010. Retrieved 1 June 2010.
6. "mmsys2012-final36.pdf" (<http://www-itec.uni-klu.ac.at/bib/files/p89-lederer.pdf>) (PDF). Retrieved 16 December 2017.
7. Spiteri, Kevin; Urgaonkar, Rahul; Sitaraman, Ramesh K. (2016). "BOLA: Near-optimal bitrate adaptation for online videos. IEEE INFOCOM, 2016, by Spiteri, Urgaonkar, and Sitaraman, IEEE INFOCOM, April 2016". arXiv:1601.06748 (<https://arxiv.org/abs/1601.06748>). doi:10.1109/TNET.2020.2996964 (<https://doi.org/10.1109%2FTNET.2020.2996964>). S2CID 219792107 (<https://api.semanticscholar.org/CorpusID:219792107>). {{cite journal}}: Cite journal requires |journal= (help)
8. "From Theory to Practice: Improving Bitrate Adaptation in the DASH Reference Player, by Spiteri, Sitaraman and Sparacio, ACM Multimedia Systems Conference, June 2018" (https://people.cs.umass.edu/~ramesh/Site/PUBLICATIONS_files/abr-dashjs.pdf) (PDF).
9. Marshall, Daniel (18 February 2010). "Show Report: Video Processing Critical to Digital Asset Management" (<https://web.archive.org/web/20111004213015/http://www.elementaltechnologies.com/blog/show-report-video-processing-critical-digital-asset-management>). Elemental Technologies. Archived from the original (<http://www.elementaltechnologies.com/blog/show-report-video-processing-critical-digital-asset-management>) on 4 October 2011. Retrieved 15 October 2011.
10. Seufert, Michael; Egger, Sebastian; Slanina, Martin; Zinner, Thomas; Hoßfeld, Tobias; Tran-Gia, Phuoc (2015). "A Survey on Quality of Experience of HTTP Adaptive Streaming" (<https://opus.bibliothek.uni-augsburg.de/opus4/frontdoor/index/index/docId/107128>). *IEEE Communications Surveys & Tutorials*. **17** (1): 469–492. doi:10.1109/COMST.2014.2360940 (<https://doi.org/10.1109%2FCOMST.2014.2360940>). S2CID 18220375 (<https://api.semanticscholar.org/CorpusID:18220375>).
11. Saamer Akhshabi; Ali C. Begen; Constantine Dovrolis. "An Experimental Evaluation of Rate-Adaptation Algorithms in Adaptive Streaming over HTTP" (https://web.archive.org/web/20111017140109/http://www.cc.gatech.edu/~sakhshab/Saamer_MMSys11.pdf) (PDF). Archived from the original (http://www.cc.gatech.edu/~sakhshab/Saamer_MMSys11.pdf) (PDF) on 17 October 2011. Retrieved 15 October 2011. {{cite journal}}: Cite journal requires |journal= (help)
12. Anthony Vetro. "The MPEG-DASH Standard for Multimedia Streaming Over the Internet" (<http://ns1.cs.sfu.ca/teaching/13/880/DASH-mm-2011.pdf>) (PDF). Retrieved 10 July 2015. {{cite journal}}: Cite journal requires |journal= (help)
13. Jan Ozer (28 April 2011). "What Is Adaptive Streaming?" (<https://www.streamingmedia.com/Articles/Editorial/What-Is-.../What-Is-Adaptive-Streaming-75195.aspx>). Retrieved 10 July 2015. {{cite journal}}: Cite journal requires |journal= (help)

14. Jeroen Famaey; Steven Latré; Niels Bouten; Wim Van de Meerssche; Bart de Vleeschauer; Werner Van Leekwijck; Filip De Turck (May 2013). "On the merits of SVC-based HTTP Adaptive Streaming" (<https://ieeexplore.ieee.org/document/6573013>): 419–426. Retrieved 10 July 2015. {{cite journal}}: Cite journal requires |journal= (help)
15. libdash: Open-source DASH client library (<http://www.bitmovin.net/libdash>) by bitmovin
16. "Distributed DASH Dataset | ITEC – Dynamic Adaptive Streaming over HTTP" (http://www-itec.uni-klu.ac.at/dash/?page_id=958). Itec.uni-klu.ac.at. Retrieved 16 December 2017.
17. *DVD Book Construction* (<http://www.dvdforum.org/tech-dvdbook.htm>), DVD Forum, May 2005
18. Yang, Hongyun (2014). "Opportunities and Challenges of HTTP Adaptive Streaming" (https://article.nadiapub.com/IJFGCN/vol7_no6/16.pdf) (PDF). *International Journal of Future Generation Communication and Networking*. **7** (6): 165–180.
19. Gannes, Liz (10 June 2009). "The Lowdown on Apple's HTTP Adaptive Bitrate Streaming" (<http://newteevee.com/2009/06/10/the-lowdown-on-apples-http-adaptive-bitrate-streaming/>). Archived (<https://web.archive.org/web/20100619175521/http://newteevee.com/2009/06/10/the-lowdown-on-apples-http-adaptive-bitrate-streaming/>) from the original on 19 June 2010. Retrieved 24 June 2010.
20. "Move Gets Streaming Patent; Are Adobe & Apple Hosed? – Online Video News" (<https://web.archive.org/web/20111022155719/http://gigaom.com/video/move-gets-streaming-patent-are-adobe-apple-hosed-2/>). Gigaom.com. 15 September 2010. Archived from the original (<http://gigaom.com/video/move-gets-streaming-patent-are-adobe-apple-hosed-2/>) on 22 October 2011. Retrieved 15 October 2011.
21. "MPEG ratifies its draft standard for DASH" (https://web.archive.org/web/20120820233136/http://mpeg.chiariglione.org/meetings/geneva11-1/geneva_press.htm). MPEG. 2 December 2011. Archived from the original (http://mpeg.chiariglione.org/meetings/geneva11-1/geneva_press.htm) on 20 August 2012. Retrieved 26 August 2012.
22. Timmerer, Christian (26 April 2012). "HTTP streaming of MPEG media – blog entry" (<http://multimediacommunication.blogspot.com/2010/05/http-streaming-of-mpeg-media.html>). Multimediacommunication.blogspot.com. Retrieved 16 December 2017.
23. "ISO/IEC DIS 23009-1.2 Dynamic adaptive streaming over HTTP (DASH)" (http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=57623). Iso.org. Retrieved 16 December 2017.
24. Updates on DASH (<http://multimediacommunication.blogspot.com/2011/02/dynamic-adaptive-streaming-over-http.html>) – blog entry
25. ETSI 3GPP 3GPP TS 26.247; Transparent end-to-end packet-switched streaming service (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH) (<http://www.3gpp.org/ftp/Specs/html-info/26247.htm>)
26. "bitdash HTML5 MPEG-DASH player" (<https://web.archive.org/web/20160710145025/http://www.dash-player.com/>). Dash-player.com. 22 January 2016. Archived from the original (<http://www.dash-player.com>) on 10 July 2016. Retrieved 16 December 2017.
27. "A VLC media player plugin enabling dynamic adaptive streaming over HTTP" (<http://www-itec.uni-klu.ac.at/bib/files/p723-muller.pdf>) (PDF). Retrieved 16 December 2017.
28. "GPAC Telecom ParisTech" (<https://web.archive.org/web/20120224084539/http://gpac.wp.institut-telecom.fr/2011/02/02/mp4box-fragmentation-segmentation-splitting-and-interleaving/>). Archived from the original (<http://gpac.wp.institut-telecom.fr/2011/02/02/mp4box-fragmentation-segmentation-splitting-and-interleaving/>) on 24 February 2012. Retrieved 28 March 2013.
29. "dash.js" (<https://github.com/Dash-Industry-Forum/dash.js>). Github.com. Retrieved 16 December 2017.
30. *Mac Developer Library* (<https://developer.apple.com/library/mac/documentation/networkinginternet/conceptual/streamingmediaguide/UsingHTTPLiveStreaming/UsingHTTPLiveStreaming.html>), Apple, retrieved 2 June 2014

31. *Shaka Packager Github Repository* (<https://github.com/shaka-project/shaka-packager>), Google, retrieved 3 January 2023
32. *Unified Streaming* (<https://www.unified-streaming.com/>), Unified Streaming, retrieved 3 January 2023
33. Prince McLean (9 July 2009). "Apple launches HTTP Live Streaming standard in iPhone 3.0" (http://www.appleinsider.com/articles/09/07/08/apple_launches_http_live_streaming_standard_in_iphone_3_0.html). AppleInsider. Retrieved 15 October 2011.
34. R. Pantos, *HTTP Live Streaming* (<http://tools.ietf.org/html/draft-pantos-http-live-streaming>), IETF, retrieved 11 October 2011
35. *RFC 8216* (<https://www.rfc-editor.org/rfc/rfc8216.html#section-4#section-4>). sec. 4. doi:10.17487/RFC8216 (<https://doi.org/10.17487%2FRFC8216>).
36. Hassoun, David. "Dynamic streaming in Flash Media Server 3.5 – Part 1: Overview of the new capabilities" (https://web.archive.org/web/20140330211949/http://www.adobe.com/devnet/adobe-media-server/articles/dynstream_advanced_pt1.html). *Adobe Developer Connection*. Adobe Systems. Archived from the original (http://www.adobe.com/devnet/flashmediaserver/articles/dynstream_advanced_pt1.html) on 30 March 2014.
37. "HTTP Dynamic Streaming" (<http://www.adobe.com/products/httpdynamicstreaming/>). Adobe Systems. Retrieved 13 October 2010.
38. "FAQ HTTP Dynamic Streaming" (<http://www.adobe.com/products/hds-dynamic-streaming/faq.html>). Adobe Systems. Retrieved 12 January 2015.
39. "Smooth Streaming" (<https://www.iis.net/download/smoothstreaming>). IIS.net. Archived (<https://web.archive.org/web/20100615150921/http://www.iis.net/download/SmoothStreaming>) from the original on 15 June 2010. Retrieved 24 June 2010.
40. Chris Knowlton (8 September 2009), *Protected Interoperable File Format* (<https://learn.iis.net/page.aspx/685/protected-interoperable-file-format>), Microsoft, retrieved 15 October 2011
41. "Microsoft End-to-End Platform Powers Next-Generation Silverlight and IIS Media Experiences Across Multiple Screens" (<http://www.microsoft.com/presspass/press/2010/apr10/04-08nab2010pr.msp>). Microsoft. 8 April 2010. Retrieved 30 July 2011.
42. "First Day of IBC" (<https://web.archive.org/web/20110202053648/http://team.silverlight.net/announcement/first-day-of-ibc/>). Microsoft. Archived from the original (<http://team.silverlight.net/announcement/first-day-of-ibc/>) on 2 February 2011. Retrieved 22 January 2011.
43. Traci Ruether (23 January 2019). "What Is CMAF?" (<https://www.wowza.com/blog/what-is-cmaf>). Retrieved 13 January 2022.
44. Luca De Cicco; Saverio Mascolo; Vittorio Palmisano. "Feedback Control for Adaptive Live Video Streaming" (<http://c3lab.poliba.it/images/6/6d/MMSYS2011.pdf>) (PDF). MMSYS2011. Retrieved 9 September 2012.
45. Dean Takahashi (16 January 2013). "Uplynk creates a cheap and efficient way for Disney to stream videos" (<https://venturebeat.com/2013/01/16/uplynk-creates-a-cheap-and-efficient-way-for-disney-to-stream-videos/>). *VentureBeat*. Retrieved 16 December 2017.
46. Dreier, Troy (16 January 2013). "UpLynk Emerges from Stealth Mode; DisneyABC Is First Customer – Streaming Media Magazine" (<http://www.streamingmedia.com/Articles/News/Online-Video-News/UpLynk-Emerges-from-Stealth-Mode%3B-Disney-ABC-Is-First-Customer-87154.aspx>). Streamingmedia.com. Retrieved 16 December 2017.
47. Y. Fei; V. W. S. Wong; V. C. M. Leung (2006). "Efficient QoS provisioning for adaptive multimedia in mobile communication networks by reinforcement learning" (<https://ieeexplore.ieee.org/document/1363846>). *Mobile Networks and Applications*. **11** (1): 101–110. CiteSeerX 10.1.1.70.1430 (<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.70.1430>). doi:10.1007/s11036-005-4464-2 (<https://doi.org/10.1007%2Fs11036-005-4464-2>). S2CID 13022779 (<https://api.semanticscholar.org/CorpusID:13022779>).

48. V. Charvillat; R. Grigoras (2007). "Reinforcement learning for dynamic multimedia adaptation". *Journal of Network and Computer Applications*. **30** (3): 1034–1058. doi:10.1016/j.jnca.2005.12.010 (<https://doi.org/10.1016%2Fj.jnca.2005.12.010>).
49. D. W. McClary; V. R. Syrotiuk; V. Lecuire (2008). "Adaptive audio streaming in mobile ad hoc networks using neural networks". *Ad Hoc Networks*. **6** (4): 524–538. doi:10.1016/j.adhoc.2007.04.005 (<https://doi.org/10.1016%2Fj.adhoc.2007.04.005>).
50. V. Menkovski; A. Liotta (2013). "Intelligent control for adaptive video streaming". *IEEE International Conference on Consumer Electronics (ICCE)*. Washington, DC. pp. 127–128. doi:10.1109/ICCE.2013.6486825 (<https://doi.org/10.1109%2FICCE.2013.6486825>).
51. M. Claeys; S. Latré; J. Famaey; F. De Turck (2014). "Design and evaluation of a self-learning HTTP adaptive video streaming client" (<https://biblio.ugent.be/publication/5733061>). *IEEE Communications Letters*. **18** (4): 716–719. doi:10.1109/lcomm.2014.020414.132649 (<https://doi.org/10.1109%2Flcomm.2014.020414.132649>). hdl:1854/LU-5733061 (<https://hdl.handle.net/1854%2FLU-5733061>). S2CID 26955239 (<https://api.semanticscholar.org/CorpusID:26955239>).
52. S. Petrangeli; M. Claeys; S. Latré; J. Famaey; F. De Turck (2014). "A multi-agent Q-Learning-based framework for achieving fairness in HTTP Adaptive Streaming". *IEEE Network Operations and Management Symposium (NOMS)*. Krakow. pp. 1–9. doi:10.1109/NOMS.2014.6838245 (<https://doi.org/10.1109%2FNOMS.2014.6838245>).
53. Pete Mastin (28 January 2011). "Is adaptive bit rate the yellow brick road, or fool's gold for HD streaming?" (<https://web.archive.org/web/20110907004353/http://www.fierceonlinevideo.com/story/adaptive-bit-rate-yellow-brick-road-or-fools-gold-hd-streaming/2011-01-28>). Archived from the original (<http://www.fierceonlinevideo.com/story/adaptive-bit-rate-yellow-brick-road-or-fools-gold-hd-streaming/2011-01-28>) on 7 September 2011. Retrieved 15 October 2011.
54. Luca De Cicco; Saverio Mascolo. "An Experimental Investigation of the Akamai Adaptive Video Streaming" (http://c3lab.poliba.it/images/d/d8/Akamai_wima2010.pdf) (PDF). Retrieved 29 November 2011. {{cite journal}}: Cite journal requires |journal= (help)
55. "Adaptive streaming: a comparison" (<https://web.archive.org/web/20140419012434/http://www.quavlive.com/en/adaptive-streaming/>). Archived from the original (<http://www.quavlive.com/en/adaptive-streaming/>) on 19 April 2014. Retrieved 17 April 2014. {{cite journal}}: Cite journal requires |journal= (help)
56. Chris Knowlton (28 January 2010). "Adaptive Streaming Comparison" (<https://www.iis.net/learn/media/smooth-streaming/adaptive-streaming-comparison>). {{cite journal}}: Cite journal requires |journal= (help)

Further reading

- The Next Big Thing in Video: Adaptive Bitrate Streaming (<http://pro.gigaom.com/2009/06/how-to-deliver-as-much-video-as-users-can-take/>) Archived (<https://web.archive.org/web/20100619225207/http://pro.gigaom.com/2009/06/how-to-deliver-as-much-video-as-users-can-take/>) 19 June 2010 at the [Wayback Machine](#)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Adaptive_bitrate_streaming&oldid=1284230850"