DEMO PAPER: LIBDASH - AN OPEN SOURCE SOFTWARE LIBRARY FOR THE MPEG-DASH STANDARD

Christopher Mueller[*†], Stefan Lederer*[*†], Joerg Poecher[†], and Christian Timmerer*[*†]

*Alpen-Adria-Universität Klagenfurt
Universitätsstraße 65-67
9020 Klagenfurt am Wörthersee, Austria
{firstname.lastname}@itec.aau.at

†bitmovin Softwareentwicklung OG
Lakeside Science and Technology Park, B01
9020 Klagenfurt am Wörthersee, Austria
{firstname.lastname}@bitmovin.net

ABSTRACT

Dynamic Adaptive Streaming over HTTP (DASH) is an ISO/IEC MPEG standard which enables the convenient and smooth transportation of multimedia data to heterogeneous end devices over networks with variable bandwidth conditions. This kind of streaming technology is mainly used with HTTP 1.0 and 1.1 respectively, which both have some drawbacks. Therefore, the IETF has started the development of HTTP 2.0, which is based on Google's SPDY proposal and already supported by several major companies, e.g., Facebook, Twitter, Akamai, Mozilla and obviously Google. Furthermore, Content Centric Networking (CCN) is another novel approach for future networks that is considered as an revolutionary approach compared to HTTP 2.0. The CCN communication paradigm is completely different and does not rely on direct connections between hosts, it rather focuses on the content. This paper demonstrates DASH with HTTP 2.0/SPDY and CCN using our universal libdash library. Moreover, different mechanisms of DASH will be shown that can be used to provide on-demand and live content in an efficient and comfortable way.

Index Terms— DASH, HTTP 2.0, SPDY, CCN

1. INTRODUCTION

Multimedia is nowadays omnipresent in the Internet and major content providers such as Netflix, Hulu, Vudu as well as Amazon are utilizing the current HTTP-based Internet infrastructure for their services. Interestingly, this streaming approach performs pretty well without any support from the underlying network. Nevertheless, in networks based on the Internet Protocol (IP) such streaming systems may face certain disadvantages. The varying bandwidth conditions must be handled on the application layer as the network does not guarantee any Quality of Service (QoS). That is, bandwidth-adaptive streaming adopting MPEG-DASH [1] but the network and its elements were initially designed for best effort file delivery and not for real-time multimedia streaming. However, due to the flexible design of the network it does not prohibit the use cases of modern network applications [2] such as real-time multimedia streaming, rich Internet services, interactive communication, etc.

One of the major protocols of modern networks is nowadays HTTP that has been used to deploy a variety of services, which are performing surprisingly well (including, real-time multimedia streaming). Nevertheless, there is still room for improvements and the Internet Engineering Taskforce (IETF) Hypertext Transfer Protocol Bis (httpbis) working group [3] has recently started the development process of HTTP/2.0. Three proposals have been submitted to the IETF [4], i.e., Google SPDY [5], Microsoft HTTP Speed+Mobility [6] and Network-Friendly HTTP Upgrade [7], where Google's SPDY proposal has been chosen as working draft for HTTP/2.0 and which now should further enhance the performance of traditional use cases as well as modern use cases. This network update is rather evolutionary and does not need changes on the network architecture side compared to the approach from Jacobsen et al. [8] named Content Centric Networking (CCN) which is a revolutionary approach that would need drastic changes on the network side.

CCN moves the focus of traditional end-to-end connections to the content. This concept could eventually replace IP but it is also possible to deploy it on top of IP. In comparison to IP-based networks, where clients setup connections between each other to exchange content, CCN directly requests the content via a naming scheme and without any connection setup. This means, that a client which wants to consume some content simply sends an interest for this content into the network and the network responds with the corresponding content. This approach is revolutionary and requires many changes within the network architecture.

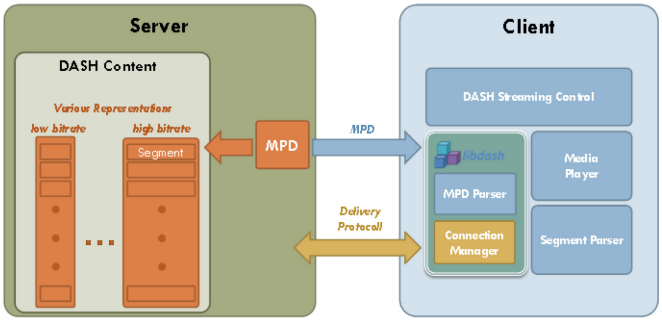

Figure 1. High-Level MPEG-DASH Architecture.

Acknowledgments: This work was supported by the Internet Foundation Austria (Netidee), and in part by the EC in the context of the ALICANTE (FP7-ICT-248652), SocialSensor (FP7-ICT-287975), and QUALINET (COST IC 1003) projects and partly performed in the Lakeside Labs research cluster at AAU.

2. LIBDASH

The general architecture of MPEG-DASH is depicted in Figure 1 where the orange parts are standardized, i.e., the MPD and segment formats. The delivery of the MPD, DASH streaming control, media player, and segment parser, are depicted in blue in Figure 1. These parts are not standardized and allow the differentiation of industry solutions, due to the performance or different features that can be integrated at that level. Our open source DASH library *libdash* is depicted at the client and encapsulates the MPD parsing and HTTP module which is responsible to handle the HTTP download per request through the streaming logic. Therefore, the library provides interfaces for these modules to access the MPD and the downloadable media segments. The download order of these media segments is not handled by libdash and left open to the DASH streaming control, which is an independent component in this architecture, but it can also be included in the media player. In a typical deployment, a DASH server provides segments in several bitrates and resolutions. The client initially receives the MPD through libdash, which provides then a convenient object-oriented interface to that MPD. The MPD contains the temporal relationships for the various qualities and segments. Based on that information, the client can download individual media segments through libdash at any point in time. Therefore, varying bandwidth conditions can be handled by switching to the corresponding quality level at segment boundaries in order to provide a smooth streaming experience. This adaptation is not part of libdash and the MPEG-DASH standard and will be left to the application which is using libdash.

3. DEMONSTRATION SETUP

The demonstration setup depicted in Figure 2 contains a fully featured DASH player that is based on libdash and a QT-based frontend. This player is open source available and part of the libdash suite, which are both platform independent and running on Windows, Linux, Mac as well as embedded platforms. With this player different mechanism of DASH are demonstrated, e.g., on-demand and live content, segment lists, templates, timelines, and separated video and audio tracks as well as multiplexed content. This also includes the playback of all test vectors of the DASH264 interoperability guidelines of the DASH Industry Forum [9].

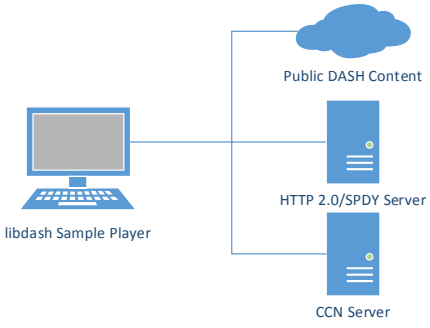

Figure 2. Demonstration Setup

Furthermore, the flexible architecture of libdash and its well-defined interfaces enable the usage of different network protocols such as HTTP 2.0/SPDY or CCN, which we have integrated into our testbed. The performance of these protocols will be shown in the demonstration setup using the DASH content of [10]. The DASH-based streaming via SPDY is enabled using the Spdylay library [11] on the client side and an Apache Web server using the mod_spdy module [12] on the server. The streaming via CCN uses the CCNx implementation [13] on the server as well as the client side. The needed extensions of libdash to enable DASH streaming via those protocols will be made publicly available on our Web site.

12. REFERENCES

[1] ISO/IEC DIS 23009-1.2, "Information Technology — Dynamic Adaptive Streaming over HTTP (DASH) — Part 1: Media Presentation Description and Segment Formats".

[2] L. Popa, A. Ghodsi, and I. Stoica. 2010. "HTTP as the narrow waist of the Future Internet". In Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks (Hotnets-IX). ACM, New York, NY, USA

[3] IETF Hypertext Transfer Protocol Bis Working Group, http://datatracker.ietf.org/wg/httpbis/, (last access: Apr 2013).

[4] HTTP/2.0 Call for Expression of Interest, http://trac.tools.ietf.org/wg/httpbis/trac/wiki/Http2CfI, (last access: Apr 2013).

[5] M. Belshe, R. Peon, "SPDY Protocol", Google, http://tools.ietf.org/html/draft-mbelshe-httpbis-spdy-00 (last access: Apr 2013).

[6] R. Trace, A. Foresti, S. Singhal, O. Mazahir, H. Nielson, B. Raymor, R. Rao, G. Motenegro, "HTTP Speed+Mobility", Microsoft, http://tools.ietf.org/html/draft-montenegro-httpbis-speed-mobility-02, (last access: Apr 2013).

[7] W. Tarreau, A. Jeffries, A. de Croy, P-H. Kamp, "Proposal for a Network-Friendly HTTP Upgrade", http://tools.ietf.org/html/draft-tarreau-httpbis-network-friendly-00, (last access: Apr 2013).

[8] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass N. H. Brigges, R. L. Braynard, "Networking Named Content", *In Proceedings of the ACM International Conference on emerging Networking Experiments and Technologies*, Rome, Italy, Dec. 2009.

[9] DASH Industry Forum, URL: www.dashif.org, (last access: Apr. 2013).

[10] S. Lederer, C. Mueller, C. Timmerer, "Dynamic Adaptive Streaming over HTTP Dataset", *ACM Multimedia Systems*, Chapel Hill, North Carolina, USA, Feb. 2012.

[11] Spdylay, http://spdylay.sourceforge.net/, (last access: Apr. 2013).

[12] Apache mod_spdy module, http://code.google.com/p/mod-spdy/, (last access: Apr. 2013).

[13] CCNx Project, URL: http://www.ccnx.org, (last access: Apr. 2013).