Quality of Experience of Adaptive HTTP Streaming in Real-World Environments

Christian Timmerer[†,‡], Matteo Maiero[†], Benjamin Rainer[†], Stefan Petscharnig[†]
Daniel Weinberger[‡], Christopher Mueller[‡], Stefan Lederer[‡]
[†]Institute of Information Technology (ITEC), Alpen-Adria-Universität Klagenfurt, Austria, www.aau.at
[‡]bitmovin GmbH, Austria, www.bitmovin.net
Email: {*firstname.lastname*}@itec.aau.at, {*firstname.lastname*}@bitmovin.net

1. Introduction

Real-time entertainment services such as streaming video and audio are currently accounting for more than 60% of the Internet traffic, e.g., in North America's fixed access networks during peak periods [1]. Interestingly, these services are all delivery over-the-top (OTT) of the existing networking infrastructure using the Hypertext Transfer Protocol (HTTP) which resulted in the standardization of MPEG Dynamic Adaptive Streaming over HTTP (DASH) [2]. The MPEG-DASH standard enables smooth multimedia streaming towards heterogeneous devices and commonly assumes the usage of HTTP-URLs to identify the segments available for the clients [3].

In this paper we focus on the Quality of Experience (QoE) of DASH-based services. We provide a general definition of QoE and which parameters are important for media services based on MPEG-DASH. The core of the paper comprises results of a QoE evaluation of different adaptation logics proposed in the research literature and also one commercially available implementation from bitmovin within real-world environments.

2. Quality of Experience for Dynamic Adaptive Streaming over HTTP

Quality of Experience

The term Quality of Experience (QoE) can be seen as an evolution from the term Quality of Service (QoS), both defined by the ITU-T in P.10/G.100. QoS is defined as the "totality of characteristics of a telecommunications service that bear on its ability to satisfy stated and implied needs of the user of the service" whereas QoE is defined as "the overall acceptability of an application or service, as perceived subjectively by the end-user". Although this definition was largely used (but not necessarily agreed), one could easily understand that acceptability is only one aspect of quality, as one may accept a service – depending on the context – but not necessarily be happy or satisfied. Therefore, the COST Action IC1003 – QUALINET goes a step beyond and defines QoE as "*the degree of delight or annoyance of the user of an application or service. It results from the fulfillment of his or her expectations with respect to the utility and/or enjoyment of the application or service in the light of the user's personality and current state*" [4].

The QUALINET white paper even goes further and defines influence factors as "any characteristic of a user, system, service, application, or context whose actual state or setting may have influence on the Quality of Experience for the user" which are grouped into human, system, and context influence factors. Additionally, features of QoE are provided depending on the level of direct perception, interaction, the usage situation, and service. A QoE feature is thus defined as "a perceivable, recognized and namable characteristic of the individual's experience of a service which contributes to its quality".

As the definitions above are very generic we will next describe what it means for DASH-based services.

QoE factors impacting DASH

Different application domains may have different requirements in terms of QoE. Thus, there is a need to provide specializations of a generally agreed definition of QoE (see above) pertaining to the respective application domain taking into account its requirements formulated by means of influence factors and features of QoE. Consequently, an application-specific QoE definition can be provided by selecting the influence factors and features of QoE reflecting the requirements of the application domain and incorporating them into the generally agreed definition of QoE.

For DASH-based services the main QoE influence factors can be described as *initial/start-up delay*, *buffer underruns* also known as *stalls*, *quality switches*, and *media throughput*.

The **initial** or **start-up delay** comprises the time between service/content request and start of the actual playout which typically involves processing time both at the server and client, network time for sending the MPD request and receiving first segments, and initial buffer time before the playout starts. In general, the start-up delay shall be low but it also depends on the use case. For example, the QoE of live streams or short movie clips is more sensitive to start-up delay than full-length video on demand content.

A **stall** occurs when the video/picture freezes that is typically caused due to **buffer underuns** and playback

is resumed if enough segments have been re-buffered. In practice, users experiencing stalls usually report a very low QoE and, thus, stalls shall be avoided at all, even if it means increasing the start-up delay.

In changing network conditions **quality switches** occur to avoid buffer underruns (and stalls) in order to guarantee a smooth video playback. However, if it happens too often (e.g., every second) or with a high amplitude (e.g., switching from a very high quality to a very low quality representation) it may negatively impact the QoE.

Finally, the overall **media throughput** at the client measured in media bits per second and a higher media throughput usually means higher QoE but it should be never used alone and always in conjunction with the above metrics as we will see in the experimental results.

The above-mentioned parameters focus on the context, specifically on delivery and device characteristics but QoE is about the users consuming content and services. Thus, it is important to understand the way how the content is provided for DASH-based services as is directly influences the QoE. In particular, this means how many different **representations** are available and in which qualities (incl. bitrate, resolution, etc.) and the actual **segment length** (e.g., 2s vs. 10s). Additional parameters are the available languages, existence of subtitles, closed caption, or any other means that help impaired users to consume the content more conveniently. In this paper we focus on the context parameters, different segment lengths, and assume a broad range of different representations available from which the client can select.

3. QoE Evaluation of DASH-based Services

Methodology

The ***test sequence*** is based on the DASH dataset [5] where we adopt the Big Buck Bunny sequence that we encoded with bitcodin – a cloud-based transcoding as-a-service, http://www.bitcodin.com/ – in order to get the representations with a bitrate of 100, 150, 200, 350, 500, 700, 900, 1100, 1300, 1600, 1900, 2300, 2800, 3400, 4500 kbps and resolutions ranging from 192×108 to 1920×1080. The configuration provides a good mix of resolutions and bitrates for both fixed and mobile network environments. In fact, we provide two versions, one with a segment length of 2s and the other with 10s that are the most common segment sizes currently adopted by actual deployments (i.e., Apple HLS uses 10s whereas others like Microsoft and Adobe use 2s).

For the actual ***MPEG-DASH client*** we adopt our bitdash streaming framework – http://www.dash-player.com/ – and compare it with ten different adaptation algorithms reported in the research literature.

For the ***objective evaluation*** we adopt the setup according to [6] where the bandwidth and delay between a server and client are shaped using a shell script, that invokes the Unix program TC with netEM and a token bucket filter. In particular, the delay was set to 80ms and the bandwidth follows a predefined trajectory (alternatively, real-world bandwidth traces could be used [6]). The delay corresponds to what can be observed within long-distance fixed line connections or reasonable mobile networks and, thus, is representative for a broad range of application scenarios. The bandwidth trajectory contains both abrupt and step-wise changes in the available bandwidth to properly test all the adaptation logics under different conditions.

For the ***subjective evaluation*** we adopt a crowdsourcing approach that uses the Microworker platform – https://microworkers.com/ – to run such campaigns and to recruit participants, which are actually referred to as microworkers. The content server is located in Europe and, thus, we limit participants to Europe in order to reduce network effects due to proxies, caches, or content distribution networks (CDNs) that we cannot control.

At the end of the subjective evaluation, each microworker needs to hand in a proof that she/he has successfully participated which is implemented using a unique identification number. We set the compensation to US$ 0.4, which is the minimum compensation for this type of campaign at the time of writing this paper.

The stimulus is the same as for the objective evaluation but we added another sequence – an excerpt from Tears of Steel, also available at [5] in order to mitigate any bias that may be introduced when using only one type of content. The content configuration is the same as for Big Buck Bunny but we used only one segment size of 2s.

The goal of this evaluation setup is to provide objective metrics which are collected at the client to be analyzed during the evaluation. These metrics include the observed bitrate, selected quality representation, buffer level, start-up delay, and stalls (re-buffering due to underruns).

The subjective evaluation methodology comprises an introduction, a pre-questionnaire, the main evaluation, and a post-questionnaire. The introduction explains the structure of the task and how to assess the actual QoE asking the microworker to provide an honest response. The pre-questionnaire collects demographic data like country of residence that we use later to filter participants. The main evaluation comprises a Web site presenting the stimulus (both sequences) with a gray background as recommended in Rec. ITU-R BT.500-11. The content is actually streamed over the open

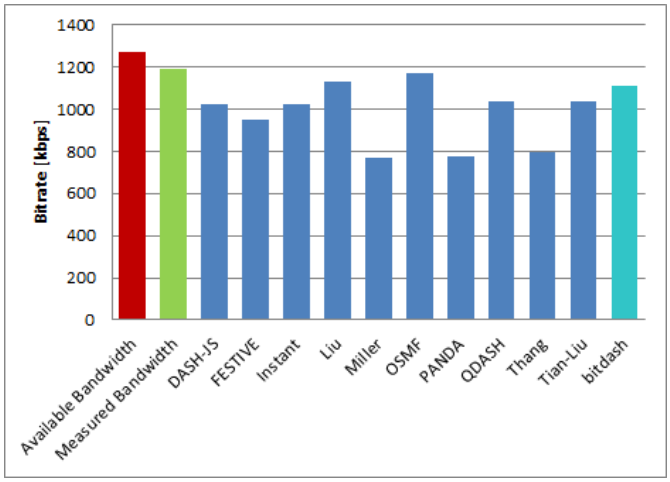

Figure 1: Average Media Throughput/Bitrate of all Adaptation Logics (higher is better).

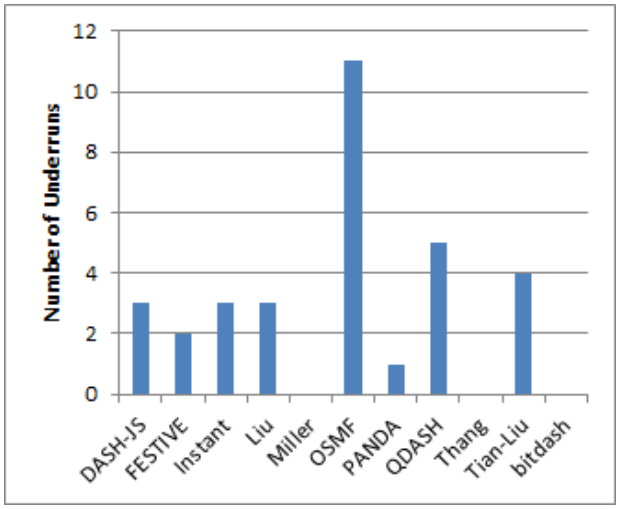

Figure 2: Number of Buffer Underruns/Stalls (lower is better).

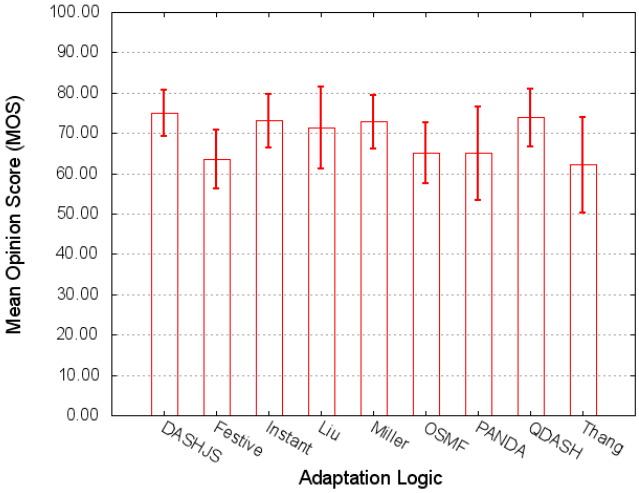

Figure 3: Mean Opinion Score (MOS) per Adaptation Logic with a 95% Confidence Interval.

Internet to which the microworker is connected using a JavaScript-based DASH client with one of the available adaptation logics. The selection of the adaptation logic is uniformly distributed (p=1/10) among the participants and the size of DASH client is fixed to a resolution of 1280×720 pixels. After the stimulus presentation, participants rate the QoE using a slider with a continuous scale from 0 to 100. The slider is initially set to 50 (middle position) and the time for rating the QoE is limited to eight seconds. The stimulus – both sequences – is presented in random order to the participants. Finally, the post-questionnaire gathers any feedback from the participants using a free text field.

In addition to the QoE rating we gather various objective metrics such as number of stalls (i.e., buffer underruns), and the average media throughput of the client.

This methodology enables a subjective evaluation of different DASH adaptation logics within real-world environments as opposed to controlled environments and, thus, provides a more realistic evaluation of adaptive HTTP streaming systems. However, using crowdsourcing requires a more careful evaluation of the participant's feedback. Therefore, we filtered participants using browser fingerprinting, stimulus presentation time, actual QoE rating, and feedback from the pre-questionnaire.

In the following sections we provide the results of the objective and subjective evaluations.

Results

The average media throughput in terms of bitrate [kbps] is shown in Figure 1. The "Available Bandwidth" on the left side of the figure shows the average bandwidth according to the predefined bandwidth trajectory used in the evaluation. The "Measured Bandwidth" by the clients is shown next to it, which is typically a bit lower than the available bandwidth due to the network overhead. The results of the different adaptation logics is shown subsequently and bitdash – on the very right side of the figure – is among the top three implementations, namely 1. OSMF (1170.65 kbps), 2. Liu (1129.69 kbps), and 3. bitdash (1109.43 kbps). However, taking into account the average media throughput only is a fallacy when investigating the number of stalls as depicted in Figure 2. Interestingly, among the top three, only bitdash does not produce any stall whereas the client with the best average media throughput produces the highest number of stalls, obviously not good for high QoE.

For the subjective evaluation, in total 220 microworkers participated in the subjective quality assessment from which 19 participants were excluded from the evaluation (due to issues during the crowdsourcing test as outlined in Section 3.3). From

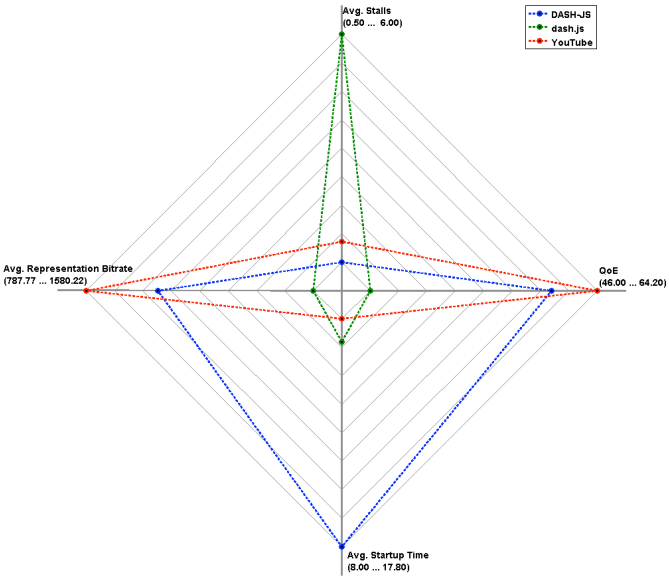

Figure 4: Overview of the Results for the Evaluated DASH-enabled Web Clients [7].

the remaining 201 participants were 143 male and 58 female with an average age of 28. The results presented in this section reflect the behavior of the adaptation logics in a real-world environment with subjects spread across Europe accessing the test sequences over the open Internet.

Figure 3 depicts the QoE in terms Mean Opinion Score (MOS) per adaptation logic (95% confidence interval). Interestingly, DASH-JS (and also Instant) provides the highest MOS value but due to overlapping confidence intervals relatively little can be stated whether it performs significantly better than the other algorithms. However, it provides a good indication about its effectiveness in a real-world environment. OSMF does not have the lowest MOS value despite its worse performance during the objective evaluation. In particular, Thang has the lowest MOS value – during the objective evaluation – although it does not cause any stalls but comes with a relatively low media throughput for both segment sizes.

Finally, we would like to share insights from a different study comparing DASH-JS, dash.js (DASH-IF reference player available at http://dashif.org/), and YouTube [7].

Figure 4 shows an overview of the results along four dimensions: average representation bitrate (i.e., media throughput at the client), average startup time (or startup delay), average number of stalls, and the QoE in terms of Mean Opinion Score (MOS). DASH-JS maintains the lowest number of stalls (0.5 stalls on average) and the average representation bitrate is about 1,330 kbit/s. However, DASH-JS has the highest average startup time. The reason for this high startup time is that DASH-JS estimates the initial bandwidth when downloading the MPD and, thus, may select a higher bitrate in the beginning than the other clients. dash.js is outperformed by the other two DASH-enabled Web clients in three of the four dimensions. In particular, dash.js provides the lowest average representation bitrate, the highest number of stalls, and the lowest QoE. YouTube outperforms all other clients in three cases, specifically in the representation bitrate, startup time, and QoE. Furthermore, Figure 4 indicates a correlation between the number of stalls and the QoE and that the representation bitrate impacts the also QoE but is not solely responsible for the QoE.

4. Conclusions

In this paper we have presented means for the Quality of Experience (QoE) evaluation of MPEG-DASH clients using objective/subjective measures and in controlled/real-world environments. An important finding is that the average media throughput/bitrate at the client cannot be used alone to describe the performance of MPEG-DASH clients and needs to be combined with other metrics such as the number of stalls. Interestingly, the start-up delay does not necessarily influence the QoE but buffer underruns or stalls will definitely and also significantly impact the media experience and, thus, shall be avoided at all.

The findings presented in this paper provide useful insights for current and future deployments of adaptive media streaming services based on the MPEG-DASH.

Acknowledgment: FFG project AdvUHD-DASH.

References

[1] Sandvine, "Global Internet Phenomena Report 2H 2014", *Sandvine Intelligent Broadband Networks*, 2014.

[2] Sodogar, I., "The MPEG-DASH Standard for Multimedia Streaming over the Internet", *IEEE Multimedia*, vol. 18, no. 4, Oct.-Dec. 2011, pp. 62-67.

[3] Timmerer, C., Mueller, C., Lederer, S., "Adaptive Media Streaming over Emerging Protocols", *Broadcast Engineering Conference (BEC), NAB2014*, 2014.

[4] Le Callet, P., Möller, S., Perkis, A., (eds), "Qualinet White Paper on Definitions of Quality of Experience", *European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003)*, Lausanne, Switzerland, Version 1.2, March 2013.

[5] Lederer, S., Mueller, C., Timmerer, C., "Dynamic Adaptive Streaming over HTTP Dataset", *In Proc. of ACM MMSys'12*, 2012.

[6] Mueller, C., Lederer, S., Timmerer, C., "An Evaluation of Dynamic Adaptive Streaming over HTTP in Vehicular Environments", *In Proc. of ACM MoVid'12*, 2012.

[7] Rainer, B., Timmerer, T., "Quality of Experience of Web-based Adaptive HTTP Streaming Clients in Real-World Environments using Crowdsourcing", *In Proc. of VideoNext'14*, 2014.