Close to optimum? User-centric evaluation of adaptation logics for HTTP adaptive streaming

Tobias Hoßfeld, Michael Seufert, Christian Sieber, Thomas Zinner, Phuoc Tran-Gia

Angaben zur Veröffentlichung / Publication details:

Hoßfeld, Tobias, Michael Seufert, Christian Sieber, Thomas Zinner, and Phuoc Tran-Gia. 2014. "Close to optimum? User-centric evaluation of adaptation logics for HTTP adaptive streaming." *PIK - Praxis der Informationsverarbeitung und Kommunikation* 37 (4): 275–85. https://doi.org/10.1515/pik-2014-0029.

Tobias Hoßfeld*, Michael Seufert, Christian Sieber, Thomas Zinner and Phuoc Tran-Gia

Close to Optimum?

User-centric Evaluation of Adaptation Logics for HTTP Adaptive Streaming

Abstract: HTTP Adaptive Streaming (HAS) is the de-facto standard for over-the-top (OTT) video streaming services. It allows to react to fluctuating network conditions on short time scales by adapting the video bit rate in order to avoid stalling of the video playback. With HAS the video content is split into small segments of a few seconds playtime each, which are available in different bit rates, i.e., quality level representations. Depending on the current conditions, the adaptation algorithm on the client side chooses the appropriate quality level and downloads the respective segment. This allows to avoid stalling, which is seen as the worst possible disturbance of HTTP video streaming, to the most possible extend. Nevertheless, the user perceived Quality of Experience (QoE) may be affected, namely by playing back lower qualities and by switching between different qualities. Therefore, adaptation algorithms are desired which maximize the user's QoE for the currently available network resources. Many downloading strategies have been proposed in literature, but a solid user-centric comparison of these mechanisms among each other and with the global optimum is missing. The major contributions of this work are as follows. A proper analysis of the influence of quality switches and played out representations on QoE is conducted by means of subjective user studies. The results suggest that, in order to optimize QoE, first, the quality level of the video stream has to be maximized and second, the number of quality switches should be minimized. Based on our findings, a QoE optimization problem is formulated and the performance of our proposed algorithm is compared to other algorithms and to the QoE-optimal adaptation.

Keywords: HTTP Adaptive Streaming (HAS), Quality of Experience (QoE), User Studies, HAS QoE Model, Adaptation Mechanism, Test-bed Experiments, Optimization Problem

DOI 10.1515/pik-2014-0029

*** Corresponding author: Tobias Hoßfeld:**
E-Mail: tobias.hossfeld@uni-due.de
Michael Seufert: E-Mail: seufert@informatik.uni-wuerzburg.de
Christian Sieber: E-Mail: c.sieber@tum.de
Thomas Zinner: E-Mail: zinner@informatik.uni-wuerzburg.de
Phuoc Tran-Gia: E-Mail: trangia@informatik.uni-wuerzburg.de

1 Introduction

Video streaming has evolved to the dominating application in the current Internet and its share is expected to grow even further within the near future [1]. Over-the-top (OTT) video distribution networks like YouTube, Hulu, or Netflix typically use a HTTP/TCP progressive streaming approach. This allows for the use of the advantages of HTTP, i.e., the HTTP delivery structure, an easy Network Address Translation (NAT) and firewall traversal, as well as the advantages of TCP, i.e., congestion control and guaranteed packet delivery. The buffering of content at the client's end further allows to overcome limitations of network resources on short time scales and to assure a continuous playout of the video content. If this is not possible, e.g., in case of live video streaming, limited network resources may lead to buffer underuns and the interruption of the playback, i.e., stalling.

To overcome this problem and to allow for a flexible adaptation of the video quality to the available network resources and device capabilities, HTTP Adaptive Streaming (HAS) has been designed. The video content is available in multiple bit rates, i.e., quality levels, and split into small segments each containing a few seconds of playtime. The client measures the current bandwidth and/or buffer status and requests the next part of the video in an appropriate bit rate, such that stalling is avoided and the available bandwidth is best possibly utilized. Hence, the control intelligence which segment to stream has moved from the servers to the clients. The HAS streaming technology is adopted by a wide range of applications and video content providers [2] and is standardized in ISO/IEC 23009-1 [3].

From the network perspective HAS enables the efficient and easy use of existing content distribution and network infrastructure components such as Content Distribution Networks (CDNs), HTTP caches, NAT devices, and firewalls. The protocol is typically used with single layer codecs like H.264/AVC, however, recent studies [4] showed that with scalable video codecs (SVC) like H.264/SVC a more efficient usage of the infrastructure and a better played out video quality can be achieved. Further, a scalable video codec allows more download flexibility since already downloaded parts of the video clip can be enhanced at a later time.

Much research in the HAS area tries to find the best downloading strategy in order to maximize the user perceived quality. Recently, several download algorithms for HAS have been proposed, both for single layer and multi layer codecs. These algorithms try to improve on typically investigated performance parameters, such as initial delay, stalling delays and frequencies, played out video quality and switching frequency. However, a solid user-centric comparison of these mechanisms among each other and with the theoretical optimum is missing.

The contribution of this paper is fourfold. First, we discuss existing adaptation logics including our proposed BIEB algorithm for H.264/SVC HAS streaming [5] which implements 'Bandwidth Independent Efficient Buffering'. Second, we conduct a subjective user study to identify the impact of video qualities and quality switching frequencies on the user perceived quality. Based on these results we model a Quality of Experience (QoE)-optimal adaptation strategy as a mixed integer program, which is the third contribution of this paper. Finally, we perform a user-centric comparison of the discussed adaptation logics among each other and with the global optimum.

The remainder of this paper is structured as follows. Section II gives an overview on the principle of HTTP adaptive video streaming and introduces existing adaptation logics from literature. For the sake of completeness, the Bandwidth Independent Efficient Buffering (BIEB) algorithm as proposed in [5] is described in detail and pseudocode for its implementation is given in Section III. The QoE influence factors on HAS are then summarized in Section IV, before the subjective user tests and the obtained user ratings are discussed. As a result of this QoE study, an optimal QoE adaptation is found. Section V models the optimal QoE adaptation strategy with mixed integer programming and formulates the optimization functions based on the QoE findings from the subjective tests. Section VI compares the different adaptation algorithms from literature with the optimal solution with respect to the user perceived quality. Finally, Section VII concludes this work and highlights future steps.

II HTTP Adaptive Streaming

The principle of HTTP Adaptive Streaming is first introduced in Section II-A, before existing adaptation logics are briefly summarized in Section II-B. In particular, the proposed algorithms *KLU* [2], *TUB* [6], and *Tribler* [7] are revisited.

A Principle of HTTP Adaptive Streaming

HTTP video streaming is a popular Internet service but suffers from several drawbacks when network conditions and video requirements are badly aligned. If the available bandwidth is smaller than the video bit rate, playout buffer depletes which will eventually cause stalling, i.e., interruption of playback due to insufficient data, which deteriorates the QoE severely (e.g., [8], [9]). However, if the video bit rate is smaller than the available bandwidth the video can be played out smoothly, but resources are spared which could be utilized for a better video quality. HAS tackles this misalignment by flexibly selecting the video quality which is delivered to the end users.

To allow for adaptation, a video clip has to be available in different video bit rates, i.e., in different quality level representations. The media is split into small segments of a few seconds duration, such that switching the quality is possible at fixed, short time intervals. On the client side, the current network conditions and/or buffer status are monitored and the adaptation logic decides which video part to download next. It requests the next segment in an appropriate bit rate, such that stalling is avoided and the available bandwidth is best possibly utilized. [10] evaluated HAS under vehicular mobility and found that quality adaptation could effectively reduce stalling by 80% when throughput decreased, and that it achieved a higher utilization of the available bandwidth when throughput increased. Also in non-mobile environments, HAS mitigates the drawbacks of classical HTTP video streaming. Thus, it is very popular nowadays which manifests in an increasing number of video applications employing different HAS solutions as default video streaming technology, and standardization efforts, such as MPEG-DASH [3].

B Existing Adaptation Logics

Many adaptation logics were already proposed in related literature. Three algorithms have been selected for comparison to our proposed algorithm. These are the prototype implementation of Müller et al. from Klagenfurt University (*KLU* [2]) and the algorithm proposed by Miller et al. from TU Berlin (*TUB* [6]) which were both designed for single-layer content like AVC. Furthermore, a chunk selection strategy based on Tribler by Oechsner et al. (*Tribler* [7]) is considered which was developed for multi-layered content like SVC. The selected three adaptation logics are described briefly in the following.

KLU uses the current bandwidth, the current buffer level, and the average bit rate of each representation for its decision. The current bandwidth is compared to the average bit rates of each representation, and the representation with the highest bit rate less or equal the estimation is selected. The bandwidth estimation is a function of the current buffer level and the throughput measured for the last segment. The estimation is decreased if the buffer level is less than 35% and increased if the buffer level is equal or higher than 50%.

TUB decides based on the average bit rate of each representation, the current bandwidth, and the current buffer level. For a buffer level less or equal to a configured minimum threshold the algorithm switches to the lowest representation. For a level less than a configured low threshold, the algorithm switches to the next lower representation, if the bandwidth is less than the average bit rate of the current representation. For a high buffer level, the algorithm increases the quality level by one, if the bandwidth is higher than the average bit rate of the next segment. Thus, in contrast to KLU, TUB tries to adapt stepwise only by one quality level at a time. The current bandwidth is estimated by the throughput of recent segments, but the estimation is modified depending on the current buffer level. To minimize the initial delay, a fast-start phase was introduced in addition to the normal mode of operation.

Tribler relies on two configuration parameters, that are the threshold of base layer segments and the maximum number of segments. Starting from the current segment, it downloads only the lowest quality (i.e., base layer) of the following segments. If the threshold of base layer segments is reached, the algorithm tries to download all quality levels of all succeeding segments up to a maximum number of segments.

It is important to mention that existing algorithms select among the available chunks just based on technical parameters like bandwidth or bit rate, but do not take the expected video quality perceived by the end user into account. SVC encoding introduces some overhead but SVC-based algorithms allow that different representations of the same time slot can be requested independently and incrementally. Single-layer strategies, on the other hand, can also request different representation of the same time slot, but only completely downloaded segments can be used for decoding. Thus, SVC-based algorithms may cope better with highly variable bandwidth, e.g., in mobile scenarios, than single-layer codecs [11]. In the following, our proposed SVC-based adaptation algorithm (*BIEB* [5]) is presented, and the algorithms are compared among each other and with respect to the QoE-optimal adaptation.

III BIEB Algorithm for Video Adaptation

The BIEB ('Bandwidth Independent Efficient Buffering') algorithm[1] is an adaptation logic for SVC videos. In contrast to other algorithms, it does not rely on estimations of the available bandwidth and does not postulate a constant bit-rate of the content, however, it assumes a constant size ratio between the segments of each representation. A comprehensive presentation is given in [5] but the most important aspects will be described in the following.

BIEB uses a number of parameters during playback. The size ratio between the segments of each representation is given by $br(r) = \frac{B_{avg,r}}{B_{avg,1}}$ with $B_{avg,r}$ being the average bit-rate of representation r without the dependency layers needed for decoding. r_{curr} is the currently selected representation, 1 is the lowest (i.e., the base layer) and r_{max} the highest representation, respectively.

We define γ, a base number of segments which statically increases the desired buffer level for all selected representations. According to the size ratio $br(r)$, BIEB computes the number of segments which should be additionally buffered for each quality level r. For example, if the enhancement layer is three times the size of the base layer, three times more segments of the base layer should be buffered than segments of the enhancement layer, in addition to γ segments.

Thus, the desired buffer level $\beta(r, r_{curr})$ for each selected representation r is given by

$$\beta(r, r_{curr}) = \begin{cases} \gamma + br(r_{curr} - r + 1) & \text{if} \quad r_{curr} - r + 1 \leq r_{max}, \\ \gamma + (r_{curr} - r - r_{max} + 2) \cdot br(r_{max}) & \end{cases}$$

else.

The algorithm loops from the minimum quality level to the current quality until it encounters a level where the number of currently buffered segments $\delta(r)$ is less than the desired number for this level $\beta(r, r_{curr})$. If all quality levels, including the current, have already reached their desired number of segments, the algorithm increases the base numbers for each quality and switches to the next higher level. The pseudo code for the algorithm is given in Algorithm 1. In Figure 1, the steady phase is depicted relative to the current playback position p_{curr} with three segments missing (displayed faded out) to the desired buffer levels $\beta(r, r_{curr})$ for all selected representations. Note that in the growing phase the buffer level requirements are increased

1 An implementation of BIEB algorithm is available at http://git.io/GTAuGg.

past the amount required for the next representation (cf. r_{curr} +2 in $\beta(r, r_{curr} + 2)$). This is done to inhibit any short-term throughput fluctuations from causing unwanted quality switches. In the case of $r_{curr} = \{r_{max} - 1, r_{max}\}$, virtual layers for $r = \{1, 2\}$ might be needed whose size is estimated as a multiple of $br(r_{max})$.

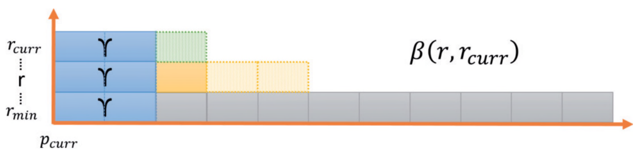

Fig. 1: BIEB algorithm: Desired buffer levels with some segments already in the buffer (displayed filled) and three missing segments (displayed faded out) before entering the growing phase.

For the BIEB algorithm, [5] presents objective results and compares its performance with the other adaptation logic presented in Section II-B. However, it is unclear how far away the algorithms are from the QoE-optimal adaptation strategy, i.e., the adaptation strategy which maximizes the subjectively perceived quality under given network conditions. Therefore, QoE influence factors are discussed in the following and the optimal adaptation is modeled.

$r = 1$;
/* Get r_{curr} */
$r_{curr} = r_{max}$;
while $\delta(r_{curr}) == 0$ **and** $r_{curr} > 1$ **do**
  $r_{curr} - -$;
end
/* Steady Phase */
while $r \leq r_{curr}$ **do**
  if $\delta(r) < \beta(r, r_{curr})$ **then**
    request next segment of representation r;
    exit;
  end
  $r + +$;
end
/* Growing Phase */
$r = 1$;
while $r \leq r_{curr}$ **do**
  if $\delta(r) < \beta(r, r_{curr} + 2)$ **then**
    request next segment of representation r;
    exit;
  end
  $r + +$;
end
/* Quality Increase */
if $r \neq r_{max}$ **then**
  $r_{curr} = r_{curr} + 1$;
  request segment $p_{curr} + \gamma$ of representation r
else
  /* idle until p_{curr} increases */
end

Algorithm 1: BIEB Adaptation Algorithm. An implementation is available at http://git.io/GTAuGg.

IV On Qoe of HTTP Adaptive Streaming

The different influence factors on HTTP adaptive video streaming are considered in Section IV-A. An understanding of those factors is required in order to design a subjective user study for quantifying the impact of those factors on QoE. Section IV-B introduces the setup of the user tests, which were conducted in a crowdsourcing environment [12], summarizes the demographics of the test participants, and quantifies QoE in terms of mean opinion scores of the individual user ratings. The derived QoE model formulates the rationale of the optimization problem for the QoE optimal adaptation of HAS.

A QoE Influence Factors

Quality of Service (QoS) in telecommunication networks is usually described objectively by network parameters like packet loss, delay, or jitter. However, a good QoS does not necessarily mean that all customers perceive a good service quality. Hence, QoE was introduced [13] which explicitly relies on subjective criteria. For classical HTTP video streaming, [8], [14] showed that the subjectively perceived quality is most influenced by initial delay and stalling. HAS, in contrast, trades off stalling or delay for adaptation (e.g., a small video chunk size leads to less stalling but more quality switches [10], [15]). However, the changing of the delivered video quality during playback introduces an additional impact on QoE [9], [16].

Several works investigated this impact of adaptation on QoE. In the following, some general findings of image quality adaptation are presented. However, it must be noted that also adaptations in other dimensions are possible, e.g., switching of resolution or frame rate. [17] found that the frequency of quality switches should be kept as small as possible. If a switch cannot be avoided, its amplitude should be kept as small as possible. Thus, a stepwise reduction of image quality is preferred to one single decrease. [18] investigated rapid alternation of base layer and enhancement layer in adaptive video streaming to mobile devices. They also confirmed the frequency effect and the amplitude effect, and additionally found that the content played an important role how adaptation impacts QoE. [19] investigated the impact of changing the quantization parameter of H.264 video streams. They found that QoE falls slowly when the quantization parameter starts to increase, i.e., the video bit rate decreases and the image quality gets worse. Only after reaching a high quantization parameter the perceived quality drops faster.

B Subjective User Studies and QoE Results

As related work suggested that the played out quality level and the number of switches have a significant impact on QoE, a subjective study was designed to confirm these findings. To have a diverse and large user base, a crowdsourcing experiment was conducted in cooperation with microworkers.com, a large international platform for distributing tasks over the Internet to anonymous workers on the basis of monetary compensation. The platform allows researchers to create a task, define a compensation, and make it available to the crowd. The experiments were setup utilizing the QualityCrowd2 framework [20] and designed according to the best practices for QoE tests in crowdsourcing as suggested in [12] in order to cope with remoteness and reliability of participants. The framework allows web-based quality assessment of video content through common web servers and common web browsers on the client side, respectively. To obtain the QoE model for adaptive video streaming, a user study with approximately 100 test subjects was conducted.

Before being able to start the experiment, every participant was asked to complete a short demographic survey. The majority of the users accessed the campaign's web-site from Asia (70%) and from Europe (26%). 42% of the participants were between the age of 22 and 25. The age-groups 18 to 21 and 26 to 30 were represented with 18% each. As occupation, 47% of the test subjects specified to be a student, followed by 32% who stated to be in employment. 40% of the participants completed a 4-year college and 17% a 2-year college. 17% indicated that high school was their highest education. Almost all test persons use the Internet daily (97%) mainly utilizing a fixed line (85% fixed line, 15% mobile access) access technology. A majority of participants (61%) visits video websites several times a day and primarily access the Internet from work (64% at work, 36% at home). 31% specified to wear prescription glasses.

After the demographic survey, a short introduction was presented to the user explaining with pictures how to watch and rate the test sequences. After the users acknowledged the introduction, the test sequences were presented to the participants sequentially. Each test sequence was completely downloaded to the browser cache to prevent any stalling. On completion of the download, a play button was activated for the user to start the playback. After the playback of the video sequence, the user was asked *Did you notice any change in quality during playback? If yes, did you feel annoyed by them?* and was presented a 5-point ACR slider with the options *Imperceptible (did not notice any)*, *Perceptible but not annoying (did notice, but did not care)*, *Slightly annoying*, *Annoying* and *Very annoying*.

As test sequence a 15 second (360 frames) video from the movie „Tears of Steel“, an open-source short movie produced and published by the Blender Foundation, was used. The scene depicts two persons standing on a small bridge and contains a low level of detail and motion (SI: 8.5, TI: 5.37). We encoded the test sequence into two quality levels by downscaling the source material to 640x360 and 160x90. Note that in the browser of the user, the two quality levels were both scaled to a window size of 320x180. Six different representation switching patterns were presented to the user in random order. Two patterns with zero switches were presented, one which only showed the higher quality to the user and one only showing the lower quality level. The other four patterns started and ended on the highest layer, but included 2, 4, 8, and 14 quality switches, which were uniformly distributed over the 15 seconds of the sequence.

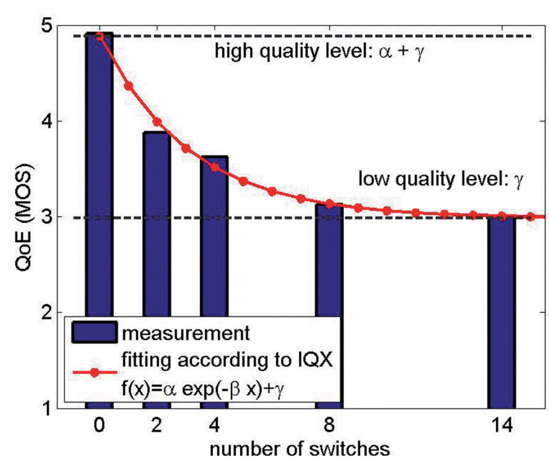

Fig. 2: Subjective user ratings as MOS for a video with two representations.

Figure 2 shows the QoE results of the conducted experiments. It presents MOS for the test sequences depending on the number of quality switches x and the corresponding fitted function $f(x)$. Switching between the high and low quality level representation has a negative influence on QoE. However, the user perceived quality is bounded by the MOS of the highest layer and the lowest layer. Thus, if there is no switch, QoE follows as $f(0) = \alpha + \gamma$, which represents the MOS of the high layer. When the number of switches increases and, thus, the low quality representation is delivered more frequently, $\lim_{x\to\infty} f(x) = \gamma$. The IQX hypothesis [21] suggests an exponential relation between QoE (in terms of MOS) and the number x of quality switches

$$f(x) = \alpha e^{-\beta x} + \gamma. \tag{1}$$

From the subjective user studies we obtain the following fitting $f(x)=1.90e^{-0.32x} + 2.98$. However, it has to be noted that more subjective tests have to be conducted to consider more than two layers, to examine additional influence factors, and to provide a generic QoE model for HTTP adaptive streaming. Nevertheless, from the results of the QoE study the following rule of thumb can be concluded. To maximize QoE for a single user, the video time played out in its highest quality level should be maximized, while the number of switches should be minimized. This is the basic rationale of the optimization problems formulated in this paper.

V Modeling Optimal Qoe Adaptation

Modeling of the optimal QoE adaptation follows a two-step approach in order to take into account the QoE findings. The optimal adaptation strategy is formulated as a mixed integer program which is based on [22]. In the first step the video time on highest quality level is maximized and in a second step the number of switches is minimized while stalling is avoided at any time. It has to be noted that [22] optimizes the downloaded data volume rather than the video time on highest quality level which we suggest for QoE optimal adaptation. Section V-A introduces the variables and parameters as used in the two-step mixed integer program in Section V-B.

A Definition of Variables and Parameters

First of all, the used notation is presented and is also summarized in Table I. It is assumed that a video is available in $R = \{1,\ldots,r_{max}\}$ representations and split into n segments. Each segment S_{ij} contains data for τ seconds of the video representation $j \in R$, and has to be played out at time D_i, $i = 1,\ldots,n$. During the time $[0,t]$ the user receives an amount of data $V(t) = v$. In compliance with the available download volume, the client downloads segments and plays them out before their respective deadline, such that no stalling occurs. In this work, no start-up/initial delay is considered. This means, after the first segment has been downloaded, the video playout begins.

These variables are sufficient to formulate the optimization problems. The boolean target variable x_{ij} indicates if the client downloads segment S_{ij} or not, and serves as input to the optimization function. Thus, the optimal assignment x_{ij} describes the outcome of an optimal adaptation strategy. This assignment is realizable under the given conditions without stalling, however, no indications of the optimal decisions are contained, i.e., the optimal assignment does not indicate when to download which segment.

In order to remove dependencies on the actual bandwidth conditions and video characteristics, the results presented in this work are normalized. The available bandwidth was adjusted, such that a video of duration $n\tau$ with total size $S^* = \sum_{i=1}^{n} S_{ir_{max}}$ of the highest quality representation r_{max} can be downloaded completely without stalling and initial delay. In other words, the received download volume at $n\tau$ equals the total size of the highest quality representation, i.e., $V(n\tau) = S^*$.

Table I: Notations and variables used for the optimization problem.

variable	explanation
R	available representations, i.e. $R = \{1,\ldots,r_{max}\}$
n	number of segments
τ	duration of a segment
S_{ij}	size of segment i from representation j
w_{ij}	weighting factor indicating the QoE value of segment i for representation j
D_i	playback deadline for segment i
$V(t)$	total amount of data received by a client during the time $[0,t]$
$x_{ij} \in \{0, 1\}$	target variable indicating if client downloads segment i from representation j ($x_{ij} = 1$) or not ($x_{ij} = 0$)
W_{opt}	optimal quality value for single user without stalling

B Optimal Adaptation Strategy as Mixed Integer Program

Based on [22], the optimal adaptation strategy can be formulated as mixed integer program. For the definition of the optimization problem, the target variable $x_{ij} \in \{0, 1\}$ is introduced indicating if the client downloads segment i from representation j ($x_{ij} = 1$) or not ($x_{ij} = 0$). The playout of a segment has different impact on QoE depending on the selected representation. Therefore, in order to optimize for QoE, a value function w_{ij} is introduced which indicates the quality value of a segment i in representation j, i.e., the contribution of a segment to the overall perceived quality. In this work, the used value function weights each segment of representation $j \in \{1,\ldots,r_{max}\}$ by j, which results in an optimization of the mean representation number.

It has to be noted that [22] suggests to maximize the downloaded volume and to minimize the quality switches which leads to a different quality value function. The rationale behind this assumption is the fact that a representation in a higher quality requires a larger volume than a lower quality representation. However, a low quality representation of segment k may be larger in practice than the high quality representation of another segment i, i.e., $S_{i1} > S_{kr_{max}}$, $r_{max} > 1$. In that case, which can occur due to different motion patterns and scenes in the video, the optimization would not select the highest possible quality layer.

Two optimization problems 1 and 2 can be formulated which take into account the QoE results which have shown that the quality layer has to be maximized first, and the number of switches have to be minimized in a second step. This two-step approach will lead to an optimal QoE without requiring a dedicated QoE model that maps parameters to QoE.

Optimization Problem 1: Maximize quality value for single user without stalling.

$$\text{maximize } W = \sum_{i=1}^{n}\sum_{j=1}^{r_{max}} w_{ij}x_{ij} \quad \textit{with} \;\; x_ij \in \{0,1\} \tag{2}$$

$$\text{subject to} \sum_{j=1}^{r_{max}} x_{ij} = 1 \qquad \forall i = 1,\ldots,n \tag{3}$$

$$\sum_{i=1}^{k}\sum_{j=1}^{r_{max}} S_{ij}x_{ij} \le V(D_k) \;\; \forall k = 1,\ldots,n \tag{4}$$

Optimization problem 1 will maximize the downloaded quality value depending on the value function w_{ij}. Constraint (3) ensures that for each segment one representation is downloaded and constraint (4) ensures that all segments i are downloaded before their deadline D_i. In this respect, $V(D_i)$ represents the maximum amount of data the client can download until the playback deadline of segment i. In the following, the optimal quality value W of problem 1 will be denoted by W_{opt}.

Optimization Problem 2: Minimize switches for single user without stalling at given target quality W_{opt}.

$$\text{minimize } \frac{1}{2}\sum_{i=1}^{n-1}\sum_{j=1}^{r_{max}} (x_{ij} - x_{i+1,j})^2 \;\; \textit{with} \;\; x_{ij} \in \{0,1\} \tag{5}$$

$$\text{subject to} \sum_{j=1}^{r_{max}} x_{ij} = 1 \qquad \forall i = 1,\ldots,n \tag{6}$$

$$\sum_{i=1}^{k}\sum_{j=1}^{r_{max}} S_{ij}x_{ij} \le V(D_k) \;\; \forall k = 1,\ldots,n \tag{7}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{r_{max}} w_{ij}x_{ij} \ge W_{opt} \tag{8}$$

Similarly, constraints (6) and (7) in optimization problem 2 are the same as constraints (3) and (4) in optimization problem 1. Additionally, constraint (8) ensures that minimizing the number of quality switches does not decrease the overall quality value below the optimum W_{opt}.

Problem 1 is known as Multiple-Choice Nested Knapsack Problem (MCNKP, [23]), while problem 2 is a Quadratic MCNKP. It is known that MCNKP is NP-hard, but pseudo-polynomial time algorithms exist which were employed by using the software gurobi[2].

VI User-centric Evaluation of Adaptation Logics

The goal of this work is the user-centric evaluation of the four HAS adaptation algorithms in a demanding realistic scenario to stress their quality adaptation and compare their performance to the QoE-optimal adaptation. The complete „Tears of Steel“ movie was chosen as example video content, which has a playback length of about 12 minutes and features high image quality with fast-paced actions scenes and slow-paced character close-ups in a science fiction scenario. The movie was transcoded into H.264/SVC with spatial scalability using the JSVM reference software (version 9.19.15). The GoP size was set to 8 frames, the IDR and intra period to 24 frames and the QP factor was set to 24. Three spatial resolutions were configured, 1280x720, 640x360, and 320x180. The encoded movie shows averages bitrates of 0.26 Mbps, 0.95 Mbps, and 2.67 Mbps, and a maximum bitrate of 1.28 Mbps, 3.37 Mbps, and 10.46 Mbps for the three spatial layers.

For usage with MPEG-DASH, a segment duration τ of 2 seconds (48 frames) was chosen which leads to n = 367 segments in total. Three inter-dependent representations

2 http://www.gurobi.com/

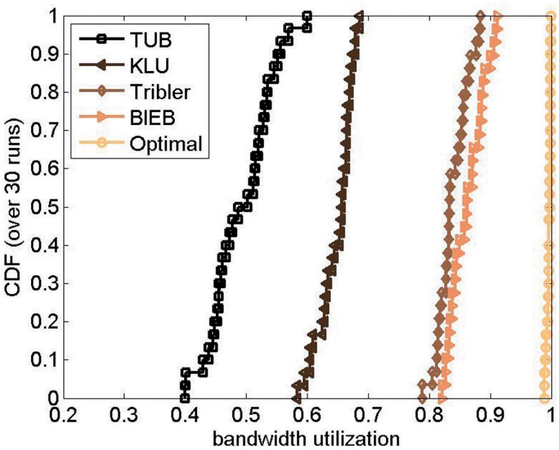

Fig. 3: Bandwidth utilization.

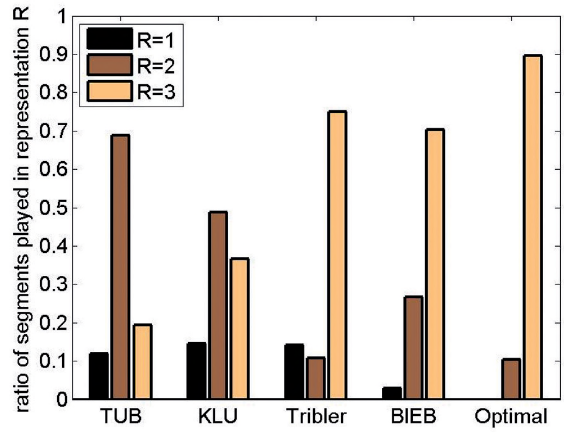

Fig. 5: Played out segments.

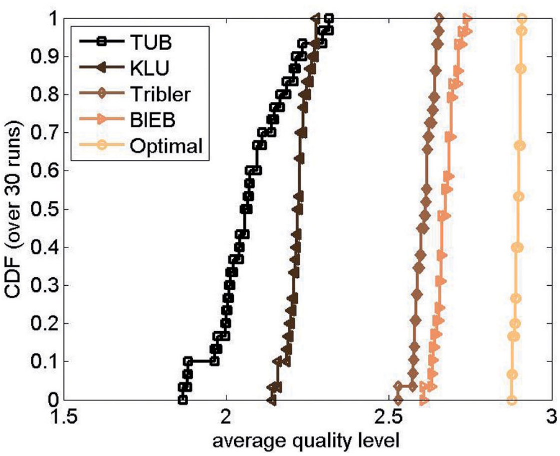

Fig. 4: Average quality level of different adaptation logics.

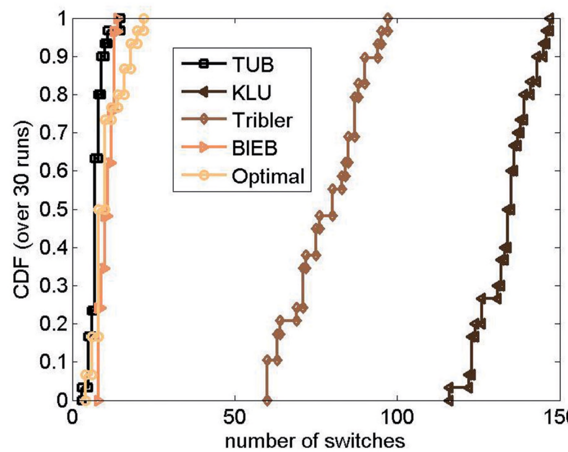

Fig. 6: Number of quality switches.

$R = \{1, 2, 3\}$ were created from the SVC segments by dissecting the SVC bitstream along the spatial scalability. Note that as scalable video coding is used here, for decoding the segment S_{ij} the segments $S_{i,1}, \ldots, S_{i,(j\text{-}1)}$ are also required. In the following, segment size is defined as the sum of the segment plus all required lower layer segments ($S_{iz} = \sum_{j=1}^{z} S_{ij}$). A total volume of 238.57 MB is required to download the video content in the highest quality, 84.86 MB and 26.52 MB for the medium and lowest quality level, respectively. The DASH segments have an average size from the lowest to the highest layer of 75.77 KB, 242.47 KB, and 681.64 KB with a standard deviation of 37.15 KB, 127.09 KB, and 419.74 KB.

For evaluation under realistic network conditions, a bandwidth pattern is used, which was recorded in a vehicular mobility scenario by Müller et al. [2]. The bandwidth pattern was recorded in and around Klagenfurt, Austria driving on a highway while connected to the Internet with a mobile UTMS stick and measuring the throughput of a large HTTP download. The mean measured bandwidth was 2.81 Mbps but it was adjusted over time in such a way that after the video duration $n\tau$, the video is completely downloaded in its highest representation, i.e., $V(n\tau) = \sum_{i=1}^{n} S_{ir_{max}}$. This resulted in a mean adjusted bandwidth of 2.67 kBps.

In order to emulate different network conditions for the different simulation runs while keeping the total download volume constant, the bandwidth trace from the vehicular mobility scenario was permuted to obtain 30 different bandwidth patterns. For each bandwidth pattern, measurements of the four adaptation algorithms were performed in the test-bed at the University of Würzburg in December 2012. Moreover, the optimization problem was solved for each pattern, which provides the QoE-optimal adaptation result for each bandwidth pattern.

Figure 3 depicts the CDF of the bandwidth utilization by the adaptation algorithms. BIEB and Tribler were always able to use at least 82 % and 78 % of the available bandwidth, whereas KLU and TUB were using only down to 58 % and 40 %, respectively. Also the latter two algorithms' maximum utilization of 56 % and 62 % is low compared to the other adaptation logics. The optimal solution would almost always utilize the whole available bandwidth, but Tribler and BIEB can come close up to around 90 % utilization.

However, the bandwidth utilization alone does not reflect the perceived quality of the algorithm. Therefore, according to the QoE results above, the average quality level and the number of switches have to be considered. Figure 4 shows the CDF of the average quality level achieved by each algorithm. BIEB and Tribler again perform best which is a rather intuitive result that could be expected from the algorithms' bandwidth utilization. In the 30 runs they achieve a mean average quality level of 2.67 and 2.61, which is close to the mean average optimum of 2.90. TUB and KLU do not reach such high average quality levels which results in mean values of 2.07 and 2.22, respectively.

Figure 5 shows the playback quality in terms of ratio of played segments of each representation. The optimal adaptation would playout the highest representation for 90 % the time and would never play out the lowest quality representation. In terms of highest representation, Tribler comes closest with 73% of the time playing out the best image quality, followed by BIEB with 68%. KLU and TUB select the highest representation only 37% and 19% of the time but use the medium representation 49% and 70% of the time. This again shows that BIEB and Tribler clearly outperform KLU and TUB regarding the playback quality and come close to the QoE-optimal adaptation.

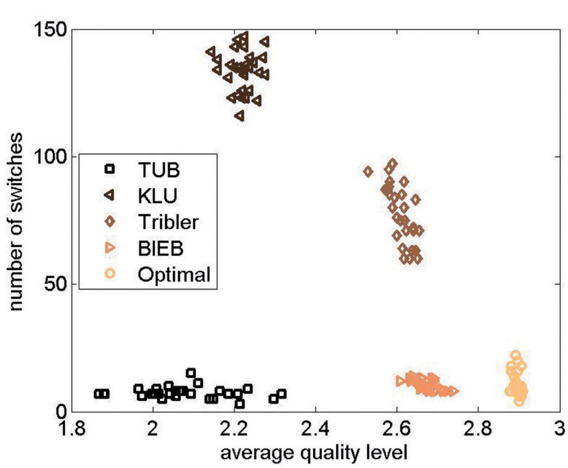

Fig. 7: Scatter plot for the 30 trial runs: average quality level vs. number of switches for the different adaptation algorithms and the optimal solution.

Considering the number of switches which are introduced by each adaptation logic a different result can be seen in Figure 6. TUB and BIEB need only few switches and are very close to the optimum. Note that the optimum can require more switches because the average quality level is optimized in first place before the number of switches is minimized. KLU and Tribler need significantly more switches and on average change the quality of the playback every 5.15 s and 6.81 s, respectively. In contrast, BIEB and TUB adapt every 61.33 s and 92 s, respectively, which are reasonable values to have only a low impact on the QoE. To put it in a nutshell, KLU and Tribler are very aggressive algorithms and try to immediately adapt to the current network condition. This, however, leads to a high quality switching frequency. BIEB and TUB are more conservative resulting in a low frequency and low impact on the QoE.

Figure 7 combines the previous findings in a single plot and shows the number of switches over the achieved average quality level. Thus, the optimal values are located in the lower right corner of the plot. It can be seen that each adaptation logic forms an own cluster which suggests that the performance results presented in this analysis are consistent over all runs. Here again it is obvious that BIEB shows the overall best performance from an user-centric point of view and is close to the QoE-optimal adaptation. Although Tribler also leads to a high playback quality, the QoE suffers from the high number of quality switches. KLU performs worse considering playback quality and frequency. Although TUB shows only medium playback quality, it has the advantage of a low number of quality switches.

VII Conclusions and Outlook

In this paper, a user-centric SVC-based algorithm for HTTP adaptive streaming was presented and compared to existing algorithms regarding achieved Quality of Experience. It was confirmed with a subjective study that the quality level and the number of switches in played out video stream are the key influence parameters of perceived quality. These results could be used to formulate an optimization problem whose solution are QoE-optimal adaptation for given bandwidth conditions and video content. The comparison of the algorithms was conducted by means of measurements in a test-bed for a mobile scenario. The results show that the BIEB mechanism outperforms the other algorithms in terms of video quality, switching frequency, and utilization of the available resources for the investigated network scenario. Moreover, a comparison with the optimal adaptation showed that the BIEB mechanism comes close to what is possible in the investigated mobile scenario.

In future work, the BIEB algorithm will be objectively and subjectively evaluated for different scenarios. Besides more network scenarios, the impact of other parameters like the segment length or the specific number and quality of the SVC layers will also be part of our investigations. Moreover, the impact of cross-traffic or background traffic as well as multi-user scenarios are also of interest. To conclude, we aim at refining the BIEB algorithm to provide a QoE as close as possible to the theoretical optimum for a wide range of possible parameters and network scenarios.

Funding: This work was partly funded by Deutsche Forschungsgemeinschaft (DFG) under grants HO 4770/1-1 and TR257/31-1 and in the framework of the EU ICT Project SmartenIT (FP7-2012-ICT-317846). The authors alone are responsible for the content.

References

1 Cisco, „Cisco Visual Networking Index: Forecast and Methodology, 2012–2017," Cisco, Tech. Rep., 2013.

2 C. Müller, S. Lederer, and C. Timmerer, „An Evaluation of Dynamic Adaptive Streaming over HTTP in Vehicular Environments," in *Proceedings of the 4th Workshop on Mobile Video (MoVID 2012)*, Chapel Hill, NC, USA, Feb. 2012.

3 International Standards Organization/International Electrotechnical Commission (ISO/IEC), „23009-1:2012 Information Technology – Dynamic Adaptive Streaming over HTTP (DASH) – Part 1: Media Presentation Description and Segment Formats," 2012.

4 Y. Sanchez, T. Schierl, C. Hellge, D. Hong, D. D. Vleeschauwer, W. V. Leekwijck, Y. Lelouedec, and T. Wiegand, „Improved caching for httpbased video on demand using scalable video coding," in *Proceedings of IEEE Consumer Communications and Networking Conference (CCNC 2011)*, Las Vegas, NV, USA, Jan. 2011.

5 C. Sieber, T. Hoßfeld, T. Zinner, P. Tran-Gia, and C. Timmerer, „Implementation and User-centric Comparison of a Novel Adaptation Logic for DASH with SVC," in *IFIP/IEEE International Workshop on Quality of Experience Centric Management (QCMan), Best Paper Award*, Ghent, Belgium, May 2013.

6 K. Miller, E. Quacchio, G. Gennari, and A. Wolisz, „Adaptation algorithm for adaptive streaming over http," in *Proceedings of the 19th International Packet Video Workshop (PV 2012)*, Munich, Germany, May 2012.

7 S. Oechsner, T. Zinner, J. Prokopetz, and T. Hoßfeld, „Supporting scalable video codecs in a p2p video-on-demand streaming system," in *21th ITC Specialist Seminar on Multimedia Applications-Traffic, Performance and QoE*, 2010.

8 T. Hoßfeld, R. Schatz, M. Seufert, M. Hirth, T. Zinner, and P. Tran-Gia, „Quantification of YouTube QoE via Crowdsourcing," in *Proceedings of the IEEE International Workshop on Multimedia Quality of Experience-Modeling, Evaluation, and Directions (MQoE 2011)*, Dana Point, CA, USA, 2011.

9 O. Oyman and S. Singh, „Quality of Experience for HTTP Adaptive Streaming Services," *IEEE Communications Magazine*, vol. 50, no. 4, pp. 20–27, 2012.

10 J. Yao, S. S. Kanhere, I. Hossain, and M. Hassan, „Empirical Evaluation of HTTP Adaptive Streaming Under Vehicular Mobility," in *Proceedings of the 10th International IFIP TC 6 Networking Conference: Networking 2011*, Valencia, Spain, 2011.

11 J. Famaey, S. Latre, N. Bouten, W. Van de Meerssche, B. De Vleeschauwer, W. Van Leekwijck, and F. De Turck, „On the merits of SVC-based HTTP Adaptive Streaming," in *IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, Ghent, Belgium, 2013.

12 T. Hossfeld, C. Keimel, M. Hirth, B. Gardlo, J. Habigt, K. Diepold, and P. Tran-Gia, „Best Practices for QoE Crowdtesting: QoE Assessment with Crowdsourcing," *Transactions on Multimedia*, vol. 16, Feb. 2014.

13 P. Le Callet, S. Möller, and A. Perkis, Eds., *Qualinet White Paper on Definitions of Quality of Experience (2012)*. Lausanne, Switzerland: European Network on Quality of Experience in Multimedia Systems and Services (COST Action IC 1003), 2012.

14 R. K. P. Mok, E. W. W. Chan, X. Luo, and R. K. C. Chan, „Inferring the QoE of HTTP Video Streaming from User-Viewing Activities," in *Proceedings of the ACM SIGCOMM Workshop on Measurements Up the STack (W-MUST)*, Toronto, ON, Canada, 2011.

15 O. Abboud, T. Zinner, K. Pussep, S. Al-Sabea, and R. Steinmetz, „On the Impact of Quality Adaptation in SVC-based P2P Video-on-Demand Systems," in *Proceedings of the 2nd Annual ACM Conference on Multimedia Systems (MMSys 2011)*, Santa Clara, CA, USA, 2011.

16 C. Alberti, D. Renzi, C. Timmerer, C. Mueller, S. Lederer, S. Battista, and M. Mattavelli, „Automated QoE Evaluation of Dynamic Adaptive Streaming over HTTP," in *Proceedings of the 5th International Workshop on Quality of Multimedia Experience (QoMEX 2013)*, Klagenfurt, Austria, 2013.

17 M. Zink, J. Schmitt, and R. Steinmetz, „Layer-encoded Video in Scalable Adaptive Streaming," *IEEE Transactions on Multimedia*, vol. 7, no. 1, pp. 75–84, 2005.

18 P. Ni, R. Eg, A. Eichhorn, C. Griwodz, and P. Halvorsen, „Flicker Effects in Adaptive Video Streaming to Handheld Devices," in *Proceedings of the 19th ACM International Conference on Multimedia (MM 2011)*, Scottsdale, AZ, USA, 2011.

19 K. D. Singh, Y. Hadjadj-Aoul, and G. Rubino, „Quality of Experience Estimation for Adaptive HTTP/TCP Video Streaming Using H. 264/AVC," in *Proceedings of IEEE Consumer Communications and Networking Conference (CCNC 2012)*, Las Vegas, NV, USA, 2012.

20 C. Keimel, J. Habigt, C. Horch, and K. Diepold, „QualityCrowd: A Framework for Crowd-based Quality Evaluation," in *Proceedings of the 29th IEEE Picture Coding Symposium (PCS 2012)*, Krakow, Poland, May 2012.

21 M. Fiedler, T. Hossfeld, and P. Tran-Gia, „A Generic Quantitative Relationship Between Quality of Experience and Quality of Service," *IEEE Network*, vol. 24, no. 2, pp. 36–41, 2010.

22 K. Miller, N. Corda, S. Argyropoulos, A. Raake, and A. Wolisz, „Optimal Adaptation Trajectories for Block-Request Adaptive Video Streaming," in *Proceedings of the 20th International Packet Video Workshop (PV 2013)*, San Jose, CA, USA, Dec. 2013.

23 E. Y.-H. Lin, „A Bibliographical Survey on Some Well-known Non-Standard Knapsack Problems," *INFOR -Information Systems and Operational Research*, vol. 36, no. 4, pp. 274–317, 1998.

Tobias Hoßfeld: University of Duisburg-Essen, Chair of Modeling of Adaptive Systems, Schützenbahn 70, D-45127 Essen

Thomas Zinner: University of Würzburg, Institute of Computer Science, Am Hubland, D-97074 Würzburg

Michael Seufert: University of Würzburg, Institute of Computer Science, Am Hubland, D-97074 Würzburg

Phuoc Tran-Gia: University of Würzburg, Institute of Computer Science, Am Hubland, D-97074 Würzburg

Christian Sieber: Technische Universitat München, Institute for Communication Networks, D-80290 München