Computing QoE-Relevant Adaptive Video Streaming Metrics Using Discrete-Time Analysis

Susanna Schwarzmann, Paula Breitbach, Thomas Zinner
FG INET, TU Berlin
Email: {susanna, paula, zinner}@inet.tu-berlin.de

***Abstract*—HTTP Adaptive Streaming (HAS) is the de-facto standard for video delivery over the Internet. Splitting the video clip into small segments and providing multiple quality levels per segment allows the client to dynamically adapt the quality to current network conditions. The performance of HAS, and as a consequence the user Quality of Experience (QoE), is influenced by a multitude of parameters. This includes adjustable settings like quality switching thresholds, the initial buffer level, or the maximum buffer, as well as video characteristics like segment duration or the variation of segment sizes along the video. Finding an appropriate tuning of those parameters still remains a challenge, which is mainly tackled by performing testbed measurements or simulative analysis. Due to the large problem space and the complex interactions of the involved influence factors, a holistic comparison of a multitude of parameter settings is extremely time intensive. To tackle this problem, we propose to enhance a GI/GI/1 system with pq-policy, which models video buffer behavior, with the capability to switch between different quality levels. This allows to investigate all relevant QoE influence factors for HAS-based video delivery. In a first evaluation, we illustrate the impact of different quality switching thresholds on the QoE influence factors for varying network conditions.**

***Index Terms*—Adaptive video streaming; QoE; DASH; Discrete-time analysis; Modeling;**

I. Introduction

Online video streaming has become the prevalent way of video consumption and a large fraction of the global Internet traffic can be attributed to on-demand video content [1]. MPEG dynamic adaptive streaming over HTTP (DASH) [2] is a widely adopted standard for Internet video delivery and allows the adaptation of the video quality to the available throughput and client capabilities. The content is split typically into segments of 2 to 10 seconds length and encoded into multiple quality levels [3]. The segments are provided via HTTP and the location and properties of the segments are summarized in an XML-based media presentation description (MPD) file. A DASH client, e.g. a set-top box or browser, first requests the MPD file and afterwards downloads and displays the segments in a quality dictated by the client's internal quality adaptation strategy.

The adaptation strategy considers a combination of parameters, like the client's device screen size, user preferences, measured throughput or current buffer level, to decide which quality level to choose for which segment. Its objective is to maximize the Quality of Experience (QoE) of the user. New strategies are coming up regularly and are being discussed in the research community [4]–[11]. They with regard to their throughput forecasting behavior, which allows them to improve the played back video quality, while reducing video stallings and quality switches. Besides the adaptation strategies, thresholds for the initial buffer time or the segment duration have a high impact on the perceived QoE [12].

So far, comparisons between quality adaptation strategies or player- and coding-relevant parameters have mainly been conducted using measurements in dedicated testbeds or by service providers within their infrastructure. Due to the large problem space, it is time consuming to do holistic comparisons between different mechanisms and parameter settings. Instead, such comparisons are done for specific use-cases which are considered to be relevant. Recently a couple of queueing-based models [13], [14] have been developed. These models are based on certain assumptions regarding the adaptation strategy and other relevant parameters, but allow to easily compute QoE metrics like the stalling probability for a large set of different network scenarios and parameter settings. However, these models do not take quality switching into account and thus do not allow to compute further QoE relevant metrics like the switching frequency, the switching amplitude, or the average video quality. Hence, there is currently no model which allows to take different video qualities, segment durations, and network conditions as input factors into account and to compute all QoE-relevant HAS metrics.

To close this gap, we enhance a discrete-time model from literature [13] by including explicit quality switching based on the state of the video buffer. Firstly, we present a formal discrete-time model for adaptive streaming systems. Secondly, we implement this model and perform an investigation of the impact of the switching thresholds on the outlined QoE metrics for different network throughput variations.

The rest of the work is structured as follows. Section II introduces DASH, presents the QoE-influencing DASH parameters, and summarizes related work on modeling DASH behavior. Section III describes the proposed model and its computation of QoE-relevant metrics. We perform an exemplary evaluation applying the proposed model in Section IV. Section V concludes the paper.

II. Background and Related Work

A. Dynamic Adaptive Streaming over HTTP

Dynamic Adaptive Streaming over HTTP (DASH) enables the adaptation of video quality to current network conditions throughout the video play back. The video is split in small segments of equal length, typically, the segment durations range between 2 and 10 seconds. Each of the video segments is available in several representations, which differ in terms of resolution and encoding bitrate. The Media Presentation Description (MPD) lists all available qualities, the segments' duration, and the URL to the specific video segments. The video client downloads the MPD and runs a DASH heuristic, which decides about the next segment's quality to request. This is either done based on the current video buffer level or based on throughput measurements. Accordingly, one distinguishes between buffer-based and throughput-based DASH heuristics. Some hybrid adaptation logics consider both, the client's buffer and the throughput. The heuristics decide so to maximize the video play back quality, whilst simultaneously avoiding rebuffering.

B. DASH Parameters and QoE Influencing Factors

Video re-buffers, i.e. video stallings, have a strong negative effect on a user's QoE. Hofeld et al. [15] showed that users rather accept a larger initial waiting time than video interruptions during playback. Besides the waiting times, video QoE is influenced by the play back quality throughout the video, as well as the frequency and amplitude of quality switches [3]. State-of-the-art HAS systems aim at tuning certain parameters, so to maximize a user's satisfaction with video services. One of these parameters to influence HAS performance is the initial buffer threshold. It determines the minimum video time that needs to be buffered in order to start the playback. This parameter constitutes a trade-off between initial waiting time and the probability of video stallings. A low initial buffer threshold allows to quickly start playing, but stallings are likely to occur due to the small buffer. Another factor is the placement of quality switching thresholds. These thresholds describe the value of buffer state or throughput that trigger to increase or decrease the quality level. Aggressive heuristics implement low thresholds in order to deliver high quality, hazarding the consequence of an increased stalling probability. The maximum buffer limits the video time stored at the client. The more a client is allowed to buffer, the lower is its risk to run into video interruptions. However, large buffers decrease the promptness of quality adaptations and increase the traffic that is wasted in case the user aborts the video stream. Besides these adjustable parameters, the logic and techniques applied in HAS adaptation heuristics influence the performance. For example, the heuristics differ in the way they monitor and smooth the current throughput, and implement different techniques to determine the next quality, i.e. applying complex machine-learning algorithms versus simple if-else-statements. Apart from the heuristic- and player-specific parameter settings, there are three more factors affecting the streaming quality. These are related to the preparation of the video content. The first one is the duration of video segments. Segments of short duration allow for a more fine granular quality adaptation, however, the shorter the segments, the lower is the encoding efficiency [16]. The second one is the number of available video representations. While a high number of video qualities on the one hand facilitates less noticeable quality switches and a very fine-granular adaptation, it imposes high storage costs for the content provider on the other hand. The third one is the bitrate used throughout the encoding of one quality layer. While a constant bitrate results in a similar size for all segments within one quality layer, the visual quality might vary among different scenes of the video, as complex scenes with high motion require a higher bitrate.

In order to be able to set these parameters so to optimize user QoE, their impact and interactions must be will understood. Measurements and simulation often do not scale, due to large problem space and hence are only able to cover specific scenarios and use-cases. Analytical models constitute a good method to do holistic evaluations to study the impact of certain parameters on QoE influencing factors.

C. Models for HAS Behavior

Efforts have been made towards modeling HAS behavior using Markov models. Hoßfeld et al. [14] presents an $M/M/1/\infty$ model with pq-policy to investigate the impact of user profiles on the QoE of adaptive streaming. They use mean-value analysis to appropriately dimension the video buffer so to meet the trade-off between initial delay and buffered time for different user characteristic, e.g. watching a complete video versus browsing videos.

De Cicco et al. [17] formalizes the behavior of an Akamai video streaming session. The system is modeled as a hybrid automaton, using upon others the video level, the current rate, and the playout buffer as state variables. Using their model, the authors show that stalling can be avoided by properly tuning switching thresholds and that a proper setting of the ratio between idle states and segment downloading can avoid large buffering, which results in network resource wasting in case the user aborts the video.

Burger et al. [13] models the video buffer as a $GI/GI/1$ queue with pq-policy using discrete time-analysis. Thereby, the video portion buffered at the client is considered as the amount of unfinished work in the system. The video playback conforms the service time, i.e. draining the buffer. It is assumed that the inter-arrival times, which correspond the segments' download durations, are independent. The model allows to evaluate the impact of video characteristics (e.g. segment duration, bitrate variation), network dynamics, and buffer policies on the streaming performance. However, this work does not model the quality switching behavior of DASH. Admittedly, the model allows for evaluating metrics like stalling probability and average buffer, but not for evaluating those QoE influencing factors that are related to video quality, i.e., the average quality or the number and amplitudes of quality switches. Correspondingly, the model in it its current state does not yet allow to examine the impact of the number of quality levels, or the setting of quality switching thresholds, on HAS performance.

Notion	Description
U_n	RV for the buffer level immediately upon arrival of segment
$A_n^{(i)}$	RV for the inter-arrival time of segment n of quality level i
B_n	RV for the play time of segment n
$C_n^{(i)}$	RV for the average bitrate of segment n of quality level i
D_n	RV for the average throughput received for downloading segment n
q	Maximum buffer, i.e. pause buffering threshold
p	Continue buffering threshold
qt_i	Buffer threshold for requesting quality layer i

TABLE I

NOTIONS USED IN THE MODEL DESCRIPTION AND THEIR MEANING

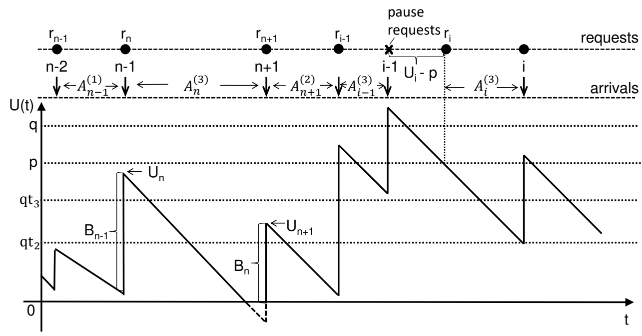

Fig. 1. Sample state process of GI/GI/1 buffer with pq-policy and switching thresholds.

In this paper, we build upon the work of Burger et al. and extend the existing model by including the DASH switching behavior.

III. PROPOSED MODEL

The model proposed in this paper is based on the work conducted in [13]. As every model, it abstracts reality and is based on a couple of assumptions. Besides assumptions like negligible round trip times and protocol overhead, or the playback availability of segments solely when completely downloaded, this also includes segment arrivals and service times to be independent. We extend this model by mimicking switching between different quality layers based on the buffer thresholds. Hence, the selected quality for the next video segment, and therewith the bitrate of this segment and the downloading time, depend on the current buffer state. Thus, the proposed extension constitutes a relaxation of the independence assumption for segment inter-arrivals.

For the following subsections, we use the notions illustrated in Table I. The respective probability mass functions are denoted by $u_n(k)$, $a_n^{(i)}(k)$, $b_n(k)$, $c_n^{(i)}(k)$, and $d_n(k)$.

A. Model Description

We describe the proposed model exemplary considering three quality layers ($i = 1, 2, 3$) using Figure 1. The time t is depicted along the x-axis, the y-axis represents the buffered time. Upon arrival of segment $n-2$, the subsequent segment $n-1$ is requested, denoted as r_{n-1}. As the current buffer $U(t)$ is below the threshold qt_2, quality 1 is chosen for segment $n-1$, resulting in a download time of $A_{n-1}^{(1)}$. Upon arrival of segment $n-1$, the buffer exceeds threshold qt_3, triggering segment n to be requested in quality 3. After the segment's download duration, i.e. $A_n^{(3)}$, segment $n+1$ is requested in quality 2, as the buffer exceeds qt_2, but is below qt_3. Once the maximum buffer q is reached, the client enters idle state, i.e. the request is paused until the buffer falls below threshold p for continuing buffering.

The current video playback buffer level, which corresponds the unfinished work in case of a GI/GI/1 queue immediately upon the $(n-1)$-th segment arrival, is denoted by U_n. Assuming that the video player requests the next segment n with quality level i immediately upon arrival of the $(n-1)$-th segment, the inter-arrival time of the segment $A_n^{(i)}$ equals the time to download the segment.

$$A_n^{(i)} = \frac{C_n^{(i)} \cdot B_n}{D_n} . \tag{1}$$

Upon arrival of segment n, its playtime B_n is added to the buffer level.

Note that the distribution of A_n^i has to be calculated by a ratio distribution in order to consider the bitrate of the segment and the download bandwidth. Since we are utilizing discrete-time analysis, we can compute the ratio distribution by iterating over all possible combinations.

The service rate of the video player is determined by the video bitrate which corresponds to the amount of video data played out per time unit, e.g., bits per second. Hence, the average bitrate corresponds to the service rate of video playtime by the video player, thus we denote $E[C_n^i] = \mu^i$. The average throughput corresponds to the arrival rate of video playtime in the queue, thus we denote $E[D_n] = \lambda$. In the parameter study, we consider the bandwidth provisioning factor a, i.e., the offered load on the video player, which corresponds the ratio of available bandwidth to the lowest quality level's bitrate.

$$a = \frac{E[D_n]}{E[C_n^1]} = \frac{\lambda}{\mu^1} \tag{2}$$

A fluent video playback is generally only possible if $a > 1$. If more video data is delivered than is played out, the video buffer will increase, and better qualities will be downloaded, based on the bitrates of the quality levels and the available throughput.

To derive the discrete time model of the buffer level in the GI/GI/1 queue with pq-policy and quality switching thresholds, we introduce the following notations.

Assuming we have N qualities, we choose thresholds $qt_1, \ldots, qt_N$ with

$$qt_1 = 0 < qt_2 < qt_3 < \ldots < qt_{N-1} < qt_N \leq p.$$

According to the buffer level U_n immediately after the arrival of segment $n-1$, the quality for the next segment request is determined. If $qt_i \leq U_n < qt_{i+1}$, the player requests the next segment in quality i. For the first quality the condition reduces to $U_n < qt_2$ and for quality N we need to have $U_n \geq qt_N$.

The last inequality $qt_N \leq p$ ensures that the quality does not decrease when pausing the requests during the idle state.

We can define the disjoint sets $Q_i,\ i = 1, \ldots, N$ representing all buffer levels that result in the download of the next segment in quality i.

$$\begin{aligned}
&Q_1 = \{0, 1, \ldots, qt_2 - 1\} \\
&Q_i = \{qt_i, qt_i + 1, \ldots, qt_{i+1} - 1\}, \quad i = 2, \ldots, N-1 \\
&Q_N = \{qt_N, \ldots\}
\end{aligned}$$

The set Q_i contains the buffer levels upon which quality i is requested.

Following the notation of [13], we introduce the conditional random variables

$$\begin{aligned}
\tilde{U}_{n,i} &= U_n | U_n \in Q_i, U_n < q \qquad i = 1, \ldots, N \\
\tilde{U}_{n,N,1} &= U_n | U_n \in Q_N, U_n \geq q
\end{aligned}$$

and

$$\begin{aligned}
U_{n+1,i} &= U_{n+1} | U_n \in Q_i, U_n < q \qquad i = 1, \ldots, N \\
U_{n+1,N,1} &= U_{n+1} | U_n \in Q_N, U_n \geq q
\end{aligned}$$

with corresponding distributions $\tilde{u}_{n,i}, \tilde{u}_{n,N,1}, u_{n+1,i}$, and $u_{n+1,N,1}$. The first two probability mass functions are the normalized restriction of $u_n(k)$ to a certain range of buffer levels, each of which corresponds to one quality.

$$\begin{aligned}
\tilde{u}_{n,i}(k) &= P(U_n = k | U_n \in Q_i, U_n < q), \quad i = 1, \ldots, N \\
\tilde{u}_{n,N,1}(k) &= P(U_n = k | U_n \in Q_N, U_n \geq q)
\end{aligned}$$

These distributions can be easily calculated. With $qt_1 := 0$, we get for $i = 1, \ldots, N-1$:

$$\tilde{u}_{n,i}(k) = \frac{\sigma_{qt_i}(\sigma^{qt_{i+1}}(u_n(k)))}{P(qt_i \leq U_n < qt_{i+1})}.$$

For $i = N$, we have:

$$\tilde{u}_{n,N}(k) = \frac{\sigma_{qt_N}(\sigma^{q}(u_n(k)))}{P(qt_N \leq U_n < q)}$$

$$\tilde{u}_{n,N,1}(k) = \frac{\sigma_{q}(u_n(k))}{P(U_n \geq q)},$$

where we use the $\sigma-$operator that truncates the distribution to a certain range.

$$\sigma_m(u(k)) = \begin{cases} u(k), & k \geq m \\ 0, & k < m \end{cases}$$

$$\sigma^m(u(k)) = \begin{cases} u(k), & k < m \\ 0, & k \geq m \end{cases}$$

With the sweep operator π we define another operator that takes the probability mass below 0 or above p and adds it to 0 or p respectively.

$$\pi_0(u(k)) = \begin{cases} u(k), & k > 0 \\ u(0) + \sum\limits_{i<0} u(i), & k = 0 \\ 0, & k < 0 \end{cases}$$

$$\pi^p(u(k)) = \begin{cases} u(k), & k < p \\ u(p) + \sum\limits_{i \geq p} u(i), & k = p \\ 0, & k > p \end{cases}$$

We can define a similar operator for a random variable X:

$$\Pi_0(X) = \begin{cases} X, & X \geq 0 \\ 0, & X < 0 \end{cases}$$

We use π_0 or Π_0, since negative buffer levels are not possible.

Next, we derive $U_{n+1,i}$ and $U_{n+1,N,1}$ and their respective probability mass functions.

$$U_{n+1,i} = \Pi_0(\tilde{U}_{n,i} - A_n^{(i)}) + B_n, \quad \text{for } i = 1, \ldots, N.$$

The buffer level $U_{n,i}$, given that U_n is in quality range i, is reduced by the corresponding inter-arrival time $A_n^{(i)}$. Since the buffer level can't drop below zero, we apply the Π_0 operator to the result. Then the $n-th$ segment arrives and its playtime B_n is added to the buffer level.

The probability mass function of $U_{n+1,i}$ is given by:

$$u_{n+1,i}(k) = (\pi_0[\tilde{u}_{n,i}(k) * a_n^{(i)}(-k)] * b_n(k)).$$

If we have $qt_N \leq U_n$ (i.e. the buffer level corresponds to the highest quality), U_n is either below the buffering-pause threshold q or it exceeds the threshold q. The first case is included in the previous formula with $i = N$. In the second case, the segment request is paused until the buffer level falls below threshold p. $U_{n+1,N,1}$ is then given by:

$$U_{n+1,N,1} = \Pi_0(p - A_n^{(N)}) + B_n$$

with distribution:

$$u_{n+1,N,1}(k) = (\pi_0[\pi^p(\tilde{u}_{n,N,1}(k)) * a_n^{(N)}(-k)] * b_n(k)).$$

From the above equations, we derive with $qt_1 := 0$:

$$\begin{aligned}
u_{n+1}(k) = \pi_0\Big[\sum_{i=1}^{N-1} \big(\sigma_{qt_i}(\sigma^{qt_{i+1}}(u_n(k))) * a_n^{(i)}(-k)\big) + \\
\sigma_{qt_N}(\sigma^{q}(u_n(k))) * a_n^{(N)}(-k) + \\
\pi^p(\sigma_q(u_n(k))) * a_n^{(N)}(-k)\Big] * b_n(k).
\end{aligned}$$

Additionally, we are interested in the buffer level not directly after a segment arrives but before. Furthermore, we allow the buffer level to be negative, which means we omit the operator Π_0. This corresponding random variable is denoted by $\hat{U}_n$. The distribution is given by:

$$\begin{aligned}
\hat{u}_n(k) = \sum_{i=1}^{N-1} \big(\sigma_{qt_i}(\sigma^{qt_{i+1}}(u_n(k))) * a_n^{(i)}(-k)\big) + \\
\sigma_{qt_N}(\sigma^{q}(u_n(k))) * a_n^{(N)}(-k) + (\pi^p(\sigma_q(u_n(k))) * a_n^{(N)}(-k)).
\end{aligned}$$

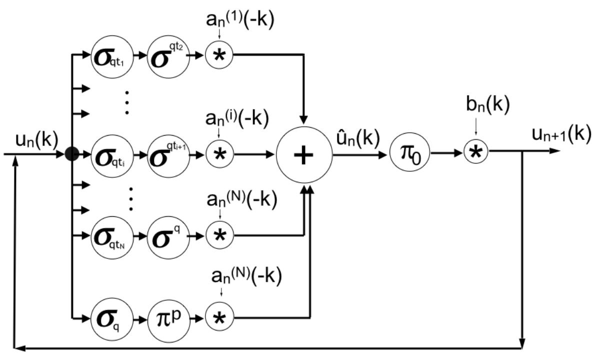

Fig. 2. Computational diagram of the buffer model

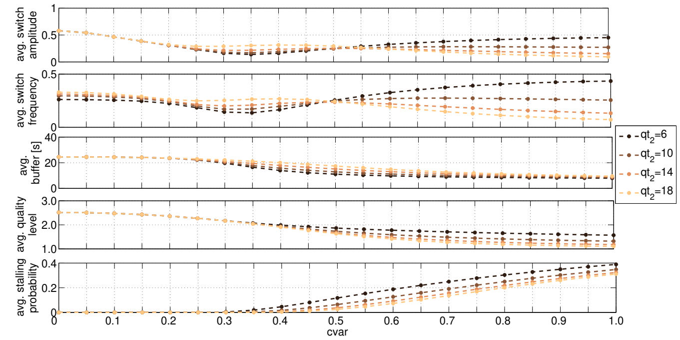

Fig. 3. Impact of the quality switching threshold qt_2 on QoE-IFs for different $cvar$ values of the available bandwidth

B. Metrics

In the following, we present the computation of the QoE-relevant metrics. For the sake of brevity, we limit the description on the steady state probabilities. Thereby, $u(k), \hat{u}(k), a(k), b(k)$ denote the steady state distribution of the corresponding random variables. We define $qt_1 := 0$ and $qt_{N+1} := q$ to shorten the notation.

Stalling probability A stalling event occurs, when the buffer is empty and the next segment hasn't arrived yet. We can calculate the stalling probability by summing up the probability mass of all negative buffer levels of $\hat{u}(k)$.

$$p_{st} = \sum_{i<0} \hat{u}(i) \tag{3}$$

Stalling duration The stalling duration corresponds to the time that passes between the buffer runs empty and the arrival of the next segment.

$$L = -\sum_{i<0} i \cdot \hat{u}(i) \tag{4}$$

Average buffer level To calculate the average buffer level, $u(k)$ and $\pi_0(\hat{u}(k))$ has to be taken into account, since $u(k)$ is the buffer level after segment arrival and $\hat{u}(k)$ is the buffer level immediately before a segment arrives. Instead, we calculate the average buffer level upon segment arrival by the following formula, where we sum over all possible buffer levels i.

$$\bar{u} = \sum_{i} i \cdot u(i) \tag{5}$$

Switching amplitude A switch of amplitude j means that the quality of segment n is j steps lower or higher than the quality of segment $n-1$. The probability for a switch of amplitude j for $j = 0, \ldots, N-1$ is given by the following formula, where we define $Q_i = \varnothing$ for $i < 1$ or $i > N$.

$$p_{amp}(j) = \sum_{i=1}^{N} \sum_{k \in Q_{i+j} \cup Q_{i-j}} (\pi_0[\sigma_{qt_i}(\sigma^{qt_{i+1}}(u(k))) * a^{(i)}(-k)] * b(k)) + \sum_{k \in Q_{N+j} \cup Q_{N-j}} (\pi_0[\pi^p(\sigma_q(u(k))) * a^{(N)}(-k)] * b(k)) \tag{6}$$

Switching probability The probability to observe a switch from one quality to another when the next segment arrives can be calculated by:

$$p_{switch} = \sum_{i=1}^{N} \sum_{k \notin Q_i} (\pi_0[\sigma_{qt_i}(\sigma^{qt_{i+1}}(u(k))) * a^{(i)}(-k)] * b(k)) + \sum_{k \notin Q_N} (\pi_0[\pi^p(\sigma_q(u(k))) * a^{(N)}(-k)] * b(k)). \tag{7}$$

Average quality The average quality, where the quality is between 1 and N, is given by:

$$\bar{Q} = \sum_{i=1}^{N} i \sum_{k \in Q_i} u(k). \tag{8}$$

IV. Model Applicability to Study the Influence of DASH Parameters

In the following, we exemplary illustrate the model's applicability by studying the impact of the quality switching threshold on relevant QoE influence factors. We consider three quality levels, whereby the threshold to switch to quality layer 2, i.e. qt_2, is set to 6, 10, 14, and 18 seconds. The threshold for requesting the third quality level, i.e. qt_3, is set to 25 seconds and does not change throughout the study. The video segment duration is set to 5 seconds. For the bitrates of the three quality layers, it holds $0.7 * q_1 = q_2 = 1.3 * q_2$, whereby the average bitrate of quality level 2 is 5000 $kbps$ with a standard deviation of 500 $kbps$. We set a bandwidth provisioning factor to $a = 1.5$, i.e. the available bandwidth is the 1.5-fold of the lowest quality's bitrate. The coefficient of variation of the available bandwidth, i.e. $cvar$, ranges from 0 to 1 in steps of 0.05.

The plots in Figure 3 illustrate, from top to bottom, the average amplitude of quality switches, the frequency of quality changes, the average video buffer, the average quality level, and the stalling probability.

For $cvar = \{0, 0.05, 0.1, 0.15, 0.2\}$, the behavior is similar for all threshold configurations of qt_2. For $cvar$ ranging between 0.25 and 0.5, the average buffer values start to drift apart, whereby a higher threshold qt_2 indicates a larger buffer.

Within this region ($cvar$ between 0.25 and 0.5), it is also observable that $qt_2 = 6$ shows the lowest switching frequency and amplitude, whereby $qt_2 = 18$ shows the highest values for these metrics. This is due to the fact that the average buffer, when setting qt_2 to 18 seconds, lies between 22.5 seconds and 17.39 seconds. Hence, the average buffer is close to the the switching threshold of $qt_2 = 18$, and as a consequence, quality switches are triggered with higher probability. As the buffer constantly decreases with increasing values of $cvar$, the buffer approaches the values of the lower switching thresholds. As a result, beginning with $cvar = 0.55$, the switching frequency increases with decreasing qt_2.

In general, the buffer shrinks with increasing bandwidth variations. Accordingly, the probability of stallings increases, especially in cases where quality is rather adapted in an aggressive ($qt_2 = 6$), than in a conservative manner ($qt_2 = 18$). Although small quality thresholds bring a high quality on average, they should be avoided if the network is likely to show high variability. The results point out that higher values for threshold qt_2 can cushion network dynamics and lead to less stalling, while they provide a similar quality as the lower thresholds for qt_2 in constant scenarios, i.e. $cvar = 0$.

The threshold qt_2 determines the number of video interruptions in networks with high variability, but at the same time has hardly impact on the average quality in scenarios with static network conditions. Accordingly it is generally better to set the first threshold to a larger value and thus increase the quality in a quite conservative manner.

Together with the standardized ITU-T model P.1203 [18], which retrieves the QoE value on MOS scale from the relevant HAS QoE-IFs, the proposed discrete-time analysis model can be used to optimize specific parameter setting with respect to the user-perceived quality.

V. Conclusion

Due to the high complexity of adaptive video streaming systems, it is cumbersome and time-consuming to perform a holistic analysis of all involved parameters. Hence, theoretical models providing an appropriate abstraction, while still replicating the adaptive video streaming behavior, are required. In this work, we extend a GI/GI/1 buffer model with pq-policy by including the switching behavior of adpative video streaming systems. This allows to capture all relevant QoE metrics for adaptive video streaming systems. As a first step, we studied the influence of the switching threshold on QoE-IFs under varying network conditions. The evaluations reveal the dependence of this thresholds with respect to the network characteristics and clearly show how the thresholds can be used to tune the trade-off between a high video quality and stalling.

For future work, we firstly plan to validate the derived model with testbed-based measurements to outline the significance of the model. Secondly, we will extensively study the interaction and influence of different network and video characteristics, as well as player configurations on the HAS performance. This will allow us to identify key influence parameters as well as parameter ranges which have a significant impact on the QoE relevant influence factors. Thereby, these studies allow to find an optimal adjustment for parameters like buffer thresholds, number of video qualities or segment durations for different network scenarios and video types.

References

[1] Cisco, "Cisco visual networking index: Forecast and methodology, 2016–2021," 2017.

[2] T. Stockhammer, "Dynamic adaptive streaming over http–: standards and design principles," in *Proceedings of the second annual ACM conference on Multimedia systems*. ACM, 2011, pp. 133–144.

[3] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hossfeld, and P. Tran-Gia, "A survey on quality of experience of http adaptive streaming," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 469–492, 2015.

[4] Y. Hou, L. Xue, S. Li, and J. Xing, "User-experience-oriented fuzzy logic controller for adaptive streaming," *The Computer Journal*, 2018.

[5] A. Bentaleb, A. C. Begen, S. Harous, and R. Zimmermann, "Want to play dash?: a game theoretic approach for adaptive streaming over http," in *Proceedings of the 9th ACM Multimedia Systems Conference*. ACM, 2018, pp. 13–26.

[6] W. ur Rahman and K. Chung, "Saba: segment and buffer aware rate adaptation algorithm for streaming over http," *Multimedia Systems*, pp. 1–21, 2018.

[7] ——, "A novel adaptive logic for dynamic adaptive streaming over http," *Journal of Visual Communication and Image Representation*, vol. 49, pp. 433–446, 2017.

[8] P. K. Yadav, A. Shafiei, and W. T. Ooi, "Quetra: A queuing theory approach to dash rate adaptation," in *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 2017, pp. 1130–1138.

[9] P. Juluri, V. Tamarapalli, and D. Medhi, "Sara: Segment aware rate adaptation algorithm for dynamic adaptive streaming over http," in *Communication Workshop (ICCW), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1765–1770.

[10] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "Bola: Near-optimal bitrate adaptation for online videos," in *INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE*. IEEE, 2016, pp. 1–9.

[11] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive," *IEEE/ACM Transactions on Networking (TON)*, vol. 22, no. 1, pp. 326–340, 2014.

[12] D. Stohr, A. Frömmgen, A. Rizk, M. Zink, R. Steinmetz, and W. Effelsberg, "Where are the sweet spots?: A systematic approach to reproducible dash player comparisons," in *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, 2017, pp. 1113–1121.

[13] V. Burger, T. Zinner, L. Dinh-Xuan, F. Wamser, and P. Tran-Gia, "A generic approach to video buffer modeling using discrete-time analysis," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 2s, p. 33, 2018.

[14] T. Hoßfeld, C. Moldovan, and C. Schwartz, "To each according to his needs: Dimensioning video buffer for specific user profiles and behavior," in *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*. IEEE, 2015, pp. 1249–1254.

[15] T. Hoßfeld, S. Egger, R. Schatz, M. Fiedler, K. Masuch, and C. Lorentzen, "Initial delay vs. interruptions: Between the devil and the deep blue sea," in *Quality of Multimedia Experience (QoMEX), 2012 Fourth International Workshop on*. IEEE, 2012, pp. 1–6.

[16] S. Lederer, C. Müller, and C. Timmerer, "Dynamic adaptive streaming over http dataset," in *Proceedings of the 3rd Multimedia Systems Conference*. ACM, 2012, pp. 89–94.

[17] L. De Cicco, G. Cofano, and S. Mascolo, "A hybrid model of the akamai adaptive streaming control system," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 3321–3326, 2014.

[18] ITU-T, "Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport," International Telecommunication Union, Recommendation P.1203, 2017.