Modeling Adaptive Video Streaming Using Discrete-Time Analysis

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

***Abstract*—HTTP Adaptive Streaming (HAS) is the de-facto standard for video delivery over the Internet. Video clips are split into segments and the quality can dynamically be adapted by choosing between multiple quality levels per segment. Based on client-centric parameters like the video buffer state or the throughput, an adaptive bitrate (ABR) algorithm decides the quality of the segments. Besides the applied ABR algorithm, a multitude of parameters influence the Quality-of-Experience (QoE) like network and video characteristics, buffer thresholds, number of provided quality levels, and segment durations. This results in a highly complex interaction between the different system parameters.**

However, the interdependence of these parameters has not been explored in a holistic manner, yet. In this paper, a generic performance model for throughput-based and buffer based ABR algorithms is proposed. Using discrete-time analysis, the model allows to compute relevant HAS metrics such as video interruptions and playback quality. To highlight the practical applicability of the model, an extensive evaluation is presented, in which the probabilistic model's results are compared with actual measurements of simplistic buffer-based and rate-based ABR algorithms. The results indicate that the model is sufficiently expressive to model the behavior in various settings, while still remaining computationally tractable.

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

I. Introduction

Online video streaming has become the prevalent way of video consumption. A large fraction of the global Internet traffic can be attributed to on-demand video content [1]. HTTP adaptive streaming (HAS) is widely adopted and allows a dynamic adaptation of the video quality to the available bandwidth. The content is split into segments of (typically) 2 to 10 seconds length and encoded under multiple quality levels [2]. The properties of the segments are summarized in an XML-based media presentation description (MPD) file. The HAS client requests the MPD file and afterwards downloads the segments in a quality dictated by the internal adaptive bitrate (ABR) strategy at the client-side.

The ABR mechanism considers client-related metrics to decide the quality of the next segment. The metrics most prominently used depend on throughput measurements or the current video buffer. Accordingly, both rate-based and buffer-based ABR strategies were studied [3], [4]. Besides the specific implementation of the ABR scheme, various player- and video-specific parameters influence the overall HAS performance, like video stallings, quality switches or the average playback video quality. These parameters include quality switching thresholds, upper and lower buffer bounds, segment durations, and the number of provided quality levels. To tune these parameters, both testbed measurements and simulations have been widely adopted [3], [4]. However, given the vast number of potential parameter combinations, these methods do not sufficiently scale to completely study the whole parameter space. Consequently, current evaluations are performed for specific use-case and only cover a subset of the large problem space.

As no *holistic evaluation methodologies* exist so far, we have only a limited understanding of the effects that various parameters have on the performance metrics of adaptive video streaming across the various operational settings. In this context, parameters include internal influence factors like number of quality levels, quality switching thresholds, or segment durations, while operational settings refer to external influence factors like network characteristics or video content. Therefore, it remains unclear, even for a single operational setting, which parameter configurations yield the best performance. Furthermore, it is practically infeasible to identify *relevant* factors and interdependencies between these factors significantly influencing the performance, which is required for a comprehensive understanding of adaptive video streaming.

This problem has recently gained attention by the research community, and so far two different queueing-based approaches have been developed [5], [6]. These models are based on certain assumptions and do not cover all of the previously outlined external and internal influence factors. Nevertheless, they allow a fast computation of a subset of the relevant HAS performance metrics like the stalling probability and duration. In fact, neither different video qualities and ABR strategies in an explicit manner are captured, nor metrics like average quality or switching frequency care reflected. To enable the incorporation of these HAS-inherent parameters, a preliminary discrete-time queueing model to support switching between different qualities was proposed in [7]. However, as the model relies on buffer properties for computing the quality level, it can only mimic a buffer-based ABR mechanism while not supporting rate-based ABR behaviors. Furthermore, a empirical comparison of the model with testbed measurements in order to show its applicability has not been conducted yet.

The main contribution of this paper is of a theoretic nature: we generalize previous models to obtain a discrete-time model suitable *both* for buffer-based and rate-based adaptive streaming systems, while still being computationally tractable.

The proposed model allows the study of all QoE-relevant performance metrics, like stalling probabilities and durations as well as quality switching frequencies and amplitudes. By considering a vast number of parameter combinations for different bandwidth configurations, the practical applicability of the presented model is shown by comparing the model output with testbed measurements.

The rest of the work is structured as follows. Section II discusses differnet HAS schemes together with their QoE-influencing parameters and summarizes related work on modeling HAS behavior. Section III describes the proposed model with both quality adaptation schemes and its computation of QoE-relevant metrics. We perform an exemplary evaluation and validate the results in Section IV. Section V concludes the paper.

II. Background and Related Work

In the following an overview of the most prominent HAS realization, namely Dynamic Adaptive Streaming over HTTP (DASH) is given. Then, existing formal models to capture HAS behavior and their limitations are discussed.

A. Dynamic Adaptive Streaming over HTTP

Dynamic Adaptive Streaming over HTTP (DASH) enables the adaptation of video quality to current network conditions throughout the playback. 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 URLs for the specific video segments. The video client downloads the MPD and runs an ABR algorithm, which decides about the next segment's quality to request. The ABR heuristics' goal is generally to maximize the playback quality whilst simultaneously avoiding rebuffering.

B. ABR Mechanisms and Parameters

Several ABR strategies have been proposed: most either use the current video buffer level or throughput measurements. Accordingly, one classically distinguishes between buffer-based and throughput-based DASH heuristics. However, some hybrid adaptation logics consider both, the client's buffer and the throughput. In the following, we shortly discuss each of these ABR strategies together with the involved parameters.

a) Buffer-based approaches: Buffer-based approaches rely on the video part which that has been downloaded and not yet been played back. Basically, a large video buffer allows to request segments of high quality, otherwise the quality is kept low to support a fast download. ABR mechanisms building on this concept are presented in [4], [8]–[10], with BOLA [4] being the most prominent one. The implementations typically differ in their definition of quality switching thresholds, which crucially determine the streaming performance. Basic mechanisms only consider the number of quality levels and set the thresholds in a static manner. BOLA, however, tunes these thresholds dynamically based on an optimization function that additionally considers the bitrate of the quality levels.

b) Rate-based approaches: Rate-based ABR mechanisms monitor the throughput during segment download. In order to compensate short-time fluctuations, measured values are smoothed, e.g., by an exponentially weighted moving average filter. The ABR mechanisms requests the next segment of the highest bitrate not exceeding the measured available network bandwidth. Many implementations decrease the measured bandwidth by a certain factor to obtain a conservative estimation and to account for overhead. Probe and adapt (PANDA) [3], is one of the best known rate-based ABR mechanisms. Further approaches can be found in [11]–[13].

c) Hybrid approaches: Both, buffer-based and rate-based ABR mechanisms, suffer from certain weaknesses. Firstly, bandwidth estimation techniques applied in rate-based ABR mechanisms may not be reliable due to varying bandwidth demands of the client or changing network conditions. Secondly, buffer-based adaptation schemes suffer from long-term bandwidth fluctuations [14] and from performance degradations at the beginning of the video playback. To overcome these issues, hybrid ABR mechanisms like [15]–[17] rely both on the available bandwidth and the current buffer state for adapting the quality.

C. Models for HAS Behavior

Recently, first efforts have been made towards modeling HAS behavior using Markov models. $\mathsf{M}/\mathsf{M}/1/\infty$ models, for example, allow to easily compute relevant metrics, while highly abstracting from the system. Hoßfeld et al. [6] presents such a model, which applies a pq-policy: the buffer values of p and q constitute lower and upper bounds for segment requests, resulting in the typical HAS on-off behavior. The pq-model has been used to investigate the impact of user profiles on the Quality-of-Experience (QoE) of adaptive streaming. The authors use a mean-value analysis to appropriately dimension the video buffer to trade-off the initial delay and the buffered time for different user characteristics, e.g., watching a complete video versus browsing videos.

De Cicco et al. [18] give a formal model for the behavior of an Akamai video streaming session. The system is modeled as a hybrid automaton, using upon others the video quality 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 having wasted the scarce network resources when the user aborts the video playback.

Burger et al. [5] model the video buffer as a $\mathsf{GI}/\mathsf{GI}/1$ queue, i.e., allowing for arbitrary independent probability distributions. Considering also a pq-policy and employing discrete time-analysis, the video portion buffered at the client is modeled as well as the amount of unfinished work in the system. The playback corresponds to the the service time.The model allows to evaluate the impact of video characteristics,

among others, segment duration and bitrate variation, network dynamics, and buffer policies on the streaming performance. However, this work does not model the quality switching behavior of HAS. While the model allows for evaluating metrics like stalling probability and average buffer, it does not support the evaluation of factors particularly important to quantify the QoE, specifically, the average quality and the number and amplitudes of quality switches. Accordingly, the model presented by Burger et al. does not allow to adequately examine the impact of the number of quality levels, or the setting of quality switching thresholds, on HAS performance.

A preliminary idea how to improve the model of Burger et al. to support buffer-based video client behavior has been proposed in [7]. This approach allows for computing the probabilities of choosing certain quality levels as well as the buffer levels. This calculation is performed based on a set of quality switching thresholds, i.e., one threshold per quality level, which are provided as input to the model. Hence, the model extension allows to study the impact of player-specific switching thresholds or video-specific parameters, e.g., number of provided levels or bitrate characteristics, on the HAS-relevant metrics. As a first proof-of-concept, the impact of the threshold determining an up-switch from the lowest quality setting was evaluated under different network settings in [7].

In contrast to previous works, our approach generalizes the available models by also allowing to model rate-based client behavior. Furthermore, this work provides the first large-scale testbed-based validation of *the presented queuing models* under various parameter combinations.

III. Discrete-Time Models for ABR Mechanisms

The model proposed in this paper is based on the works conducted in [5] and [7]. Specifically, in [5] a general $\mathsf{GI}/\mathsf{GI}/1$ model was proposed to model HAS behavior without different quality settings while in our preliminary work [7] the model was extended to support several quality levels. In the following, we first introduce the model and its assumptions. We then present the model for buffer-based ABR solutions. Given this background, we show how also *rate-based* ABR-mechanisms can be modeled. Lastly, we discuss the realization of several key metrics to capture HAS-performance in both the buffer-based and rate-based model.

A. Model Overview and Notation

In the following the (probabilistic) HAS model is presented. We allow for an arbitrary number of quality levels $N \in \mathbb{N}$. The decision of which quality level to choose is always decided immediately upon having received the last video segment: given the current buffer level or the latest download rate either of the quality levels is chosen. Specifically, given *quality thresholds* qt_i for $i \in \{2, \dots, N\}$, the highest quality i is chosen such that the buffer level or rate lies above qt_i ($qt_1 = 0$). Furthermore, the model allows for considering buffering thresholds p and q, $p < q$, where once the buffer level q is exceeded the next video segment is delayed until the buffer undercuts the threshold p again, thereby limiting the amount of buffered video at any time.

Accordingly, each quality level i corresponds to a (discrete) set $Q_i \subseteq \mathbb{N}$ of buffer levels or rates upon which quality i is requested. For the buffer states, we have

$$\begin{aligned} Q_1 &= \{0, 1, \dots, qt_2 - 1\} \\ Q_i &= \{qt_i, \dots, qt_{i+1} - 1\},\ i \in \{2, \dots, N-1\} \\ Q_N &= \{qt_N, \dots, \max_{buf}\} \end{aligned} \quad \text{[buffer]} \ (1)$$

where $\max_{buf}$ denotes the maximal possible buffer size, e.g., the length of the video. For the rate variant, the sets are analogously defined using $\max_{rate}$ to denote the maximal rate:

$$\begin{aligned} Q_1 &= \{0, \dots, qt_2 - 1\} \\ Q_i &= \{qt_i, \dots, qt_{i+1} - 1\},\ i \in \{2, \dots, N-1\} \\ Q_N &= \{qt_N, \dots, \max_{rate}\} \end{aligned} \quad \text{[rate]} \ (2)$$

While the above parameters are specific to the ABR implementation and assumed to be fixed, the external factors, as for example the available bandwidth, are modeled as *random variables* (RV) drawn from *arbitrary* probability distributions. Specifically, we denote by B_n and D_n the random variables determining the playback time of segment n and the average throughput while downloading it, respectively. The required bitrate for the segment n of quality $i \in \{1, \dots, N\}$ is modeled as the RV $C_n^{(i)}$. Thus, the inter-arrival time of segment n of quality i is given via:

$$A_n^{(i)} = \frac{C_n^{(i)} \cdot B_n}{D_n}\,. \quad (3)$$

Figure 1 gives an exemplary overview of the model using three quality levels ($N = 3$). The time t is depicted along the x-axis and the y-axis represents the buffered playback time.

In the example for the segment $n-1$ the quality level 1 is chosen based on the above introduced control sets $\{Q_i\}_{i \in \mathcal{Q}}$. After the full download time of $A_{n-1}^{(1)}$ (but before the reception) two buffer levels are considered $\tilde{U}_{n-1}$ and $\hat{U}_{n-1}$: $\hat{U}_{n-1}$ allows for negative buffer values, which will indicate stalling, while $\tilde{U}_{n-1}$ denotes the actual buffer, which can at most be empty and hence must always be greater than 0. Specifically,

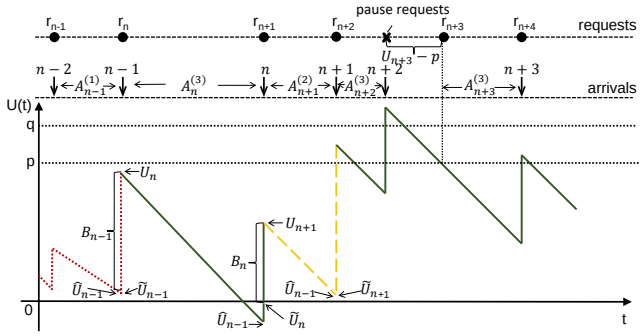


Fig. 1. Sample state process of GI/GI/1 buffer with pq-policy and switching behavior. The quality is determined by a control variable, e.g., the current buffer state or the estimated rate. Dotted lines denote lowest quality (1), dashed lines represent medium quality (2), and solid lines signify the highest quality (3).

$\hat{U}_{n-1}$ is obtained by subtracting the download time $A^{(1)}_{n-1}$ from U_{n-1} and $\tilde{U}_{n-1}$ is obtained by considering the maximum of $\hat{U}_{n-1}$ and 0. Hence, as $\hat{U}_{n-1} > 0$ holds in the example, $\hat{U}_{n-1} = \tilde{U}_{n-1}$ follows.

Accordingly, the segment's playtime B_{n-1} is added to the buffer level $\hat{U}_{n-1} = \tilde{U}_{n-1}$ to obtain the new buffer level U_n immediately after receiving the $n-1$-th request.

Now, the control variables grant downloading the next segment in the highest quality, resulting in a download duration of $A^{(3)}_n$. Now, the download duration $A^{(3)}_n$ is larger than the current buffer time U_n, and the virtual buffer $\hat{U}_n$ hence drops below zero while for the actual buffer $\tilde{U}_n = 0$ holds. Here, in this case the video playback is stalled for a duration of $\hat{U}_n$. After receiving the n-th request, the actual buffered playback time is set to B_n, i.e., $U_{n+1} = B_n$ holds, and the playback resumes as the buffered video value is now again positive. Then, after downloading segment $n+1$ in quality 2 with a time $A^{(2)}_{n+1}$, the threshold q is exceeded. Thus, segment request $n+2$ is delayed until the buffer reaches the threshold p.

Table I summarizes the above introduced notation and we note the following assumptions on which the model relies:

(1) Round trip times and protocol overhead are not modeled.
(2) Segments must be fully received before being played back.
(3) After stalling, the playback is immediately resumed once a segment arrives: there is no buffer limit for playback.
(4) Segment arrivals A_n and the service time B_n will be required to be independent probability distributions.

B. Stochastic Preliminaries

The above presented model essentially represents a complex $\mathsf{GI}/\mathsf{GI}/1$ queuing model. While in the example of Figure 1 the *random* variables, e.g., $A^{(i)}_n$ and U_n (cf. Table I) were depicted using specific values. In contrast, the model will work on the distributions of these random variables. We denote the underlying probability density functions by employing lower case letters: $a^{(i)}_n$, b_n, $c^{(i)}_n$, and d_n denote the density functions of the input random variables $A^{(i)}_n$, B_n, $C^{(i)}_n$, and D_n. While these density functions are part of the input of the model, the main task when analyzing the model will be to concisely compute the density functions pertaining to the buffer level random variables U_n, $\tilde{U}_n$, etc., as these will allow to compute the stalling probabilities (among others).

To facilitate the analysis, the following notation for arbitrary discrete probability density functions $f, g : \mathbb{Z} \to \mathbb{R}_{\geq 0}$ is used.

As we will need to take the latest buffer state or download rate into account, the following σ-operators are used to truncate the respective probability distributions to a certain range:

$$\big[\sigma_m(f)\big](k) = \begin{cases} f(k), & k \geq m \\ 0, & k < m \end{cases} \tag{4}$$

$$\big[\sigma^m(f)\big](k) = \begin{cases} f(k), & k < m \\ 0, & k \geq m \end{cases} \tag{5}$$

As the buffer computation might return negative buffer values or values exceeding threshold p, we additionally introduce the following sweep operators π_0 and π^p to concentrate the probability masses below 0 or above p to 0 or p, respectively.

$$\big[\pi_0(f)\big](k) = \begin{cases} f(k), & k > 0 \\ f(0) + \sum_{j<0} f(j), & k = 0 \\ 0, & k < 0 \end{cases} \tag{6}$$

$$\big[\pi^p(f)\big](k) = \begin{cases} f(k), & k < p \\ f(p) + \sum_{j\geq p} f(j), & k = p \\ 0, & k > p \end{cases} \tag{7}$$

Lastly, considering two random variables F and G with density functions f, g we note that the sum $H = F + G$ is distributed according to the density function $h = f * g$, where $*$ denotes the convolution of density functions, *when* F and G are independent. While the convolution inherently is an operation on functions and again yields a function, the notation $h(k) = f(k) * g(k)$ is often used in the literature on queuing theory to denote $h = f * g$ (cf. [5], [7], [19]), we adapt our notation accordingly. Indeed, one advantage of this notation is the simpler expression of subtractions of random variables: $H = F - G$ is then denoted by $h(k) = f(k) * h(-k)$, where $h(-k)$ denotes the probability density function $-k \mapsto h(k)$.

TABLE I
NOTATION

INPUT	FIXED	N	number of provided quality levels
		$\mathcal{Q}$	quality levels $\mathcal{Q} = \{1, \ldots, N\}$
		q	threshold after which buffering is paused
		p	threshold for resuming buffering after having stopped before
		qt_i	threshold (buffer or rate) for requesting quality $i \in \mathcal{Q}$, $qt_1 = 0$
		Q_i	set of states (buffer or rate) requesting quality $i \in \mathcal{Q}$
	RANDOM VAR. (RV)	$A^{(i)}_n$	inter-arrival time RV of segment n of quality $i \in \mathcal{Q}$
		B_n	playback time RV of segment n
		$C^{(i)}_n$	average bitrate RV of segment n of quality $i \in \mathcal{Q}$
		D_n	average throughput RV for downloading segment n
ANALYSIS		U_n	buffer RV *immediately after* arrival of segment $n-1$
		$\hat{U}_n$	virtual buffer RV *immediately before* arrival of segment n, which might attain negative values to denote stalling
		$\tilde{U}_n$	actual buffer RV ≥ 0 *immediately before* arrival of segment n

C. Buffer-based Model

In the following the buffer-based model is presented (cf. our preliminary work [7]). The core challenge in modeling the buffer-based behavior lies in computing the current buffer levels U_n, $\hat{U}_n$, and $\tilde{U}_n$. The computation of these is necessitated as the quality is chosen based on the buffer levels U_n and for example $\hat{U}$ is used to compute the stalling probability. For our analysis we assume $qt_N \leq p < q$ to hold and note that if $qt_N > p$ were to hold, then the highest quality could never be requested after pausing buffering.

Henceforth, our main goal is to derive concise (recursive) formulas for the probability density functions u_n, $\hat{u}_n$, $\tilde{u}_n$ of the respective buffer level random variables.

We start by considering the density function of $\hat{u}_n$ pertaining to the random variable $\hat{U}_n$. The virtual buffer value $\hat{U}_n$ is generally computed as a function of the previous buffer level U_n just after receiving segment $n-1$ and the arrival time $A^{(i)}_n$

of the segment n of quality i. Hence, both the chosen quality i, dictating the download time $A_n^{(i)}$, and the potential pausing of buffering depend on the buffer level U_n and the following cases have to be considered:

$$\hat{U}_n = \begin{cases} U_n - A_n^{(i)} & \text{if } \; qt_i \leq U_n < qt_{i+1}, i \in \mathcal{Q} \setminus \{N\} \quad (8) \\ U_n - A_n^{(N)} & \text{if } \; qt_N \leq U_n < q \quad (9) \\ p - A_n^{(N)} & \text{if } \; q \leq U_n \quad (10) \end{cases}$$

Above, the first $N-1$ cases (cf. Equation 8) capture the systems behavior when the buffer level U_n indicates to request a quality laying in $\{1, \ldots, N-1\}$. When the highest quality is chosen, the two cases of Equations 9 and 10 need to be considered: if the buffer level lies below q, then simply the next segment is downloaded according to the highest quality, while if q is exceeded, requesting the next segment is delayed until the buffer has reached the threshold p again (cf. Figure 1).

Considering the first $N-1$ cases first, the density of $U_n - A_n^{(i)}$ computes to $\hat{u}_n(k) = u_n(k) * a_n^{(i)}(-k)$. However, as these densities only hold under the restrictions of $qt_{i-1} \leq U_n < qt_i$, we obtain the following *partial* density functions:

$$\hat{u}_{n,i}(k) = \sigma_{qt_i}(\sigma^{qt_{i+1}}(u_n(k))) * a_n^{(i)}(-k)), \; i \in \mathcal{Q} \setminus \{N\}. \quad (11)$$

Above, the restriction on the *range* of U_n is realized by truncating the distribution $u_n(k)$ to the respective range between qt_i and qt_{i+1} using the respective σ-operators.

In a similar fashion, the density of the case of Equation 9 can be captured by the following partial density function:

$$\hat{u}_{n,N,<q}(k) = \sigma_{qt_N}(\sigma^{q}(u_n(k))) * a_n^{(N)}(-k). \quad (12)$$

Lastly, for Equation 10 the following density is obtained.

$$\hat{u}_{n,N,\geq q}(k) = \pi^{p}(\sigma_{q}(u_n(k))) * a_n^{(N)}(-k)\,. \quad (13)$$

Intuitively, by first applying the σ_q-operator only buffer levels exceeding q are considered, while the π^p operator then realigns the full probability mass to the value p.

Overall, the density function $\hat{u}_n$ can then be computed as the *summation* over the densities of Equations 11 to 13 (cf. [5], [7]). Hence, the following is obtained:

$$\hat{u}_n(k) = \begin{pmatrix} \sum_{i=1}^{N-1} \big(\sigma_{qt_i}(\sigma^{qt_{i+1}}(u_n(k))) * a_n^{(i)}(-k)\big) \\ + \quad \big(\sigma_{qt_N}(\sigma^{q}(u_n(k))) * a_n^{(N)}(-k)\big) \\ + \quad \big(\pi^{p}(\sigma_{q}(u_n(k)) * a_n^{(N)}(-k))\big) \end{pmatrix} \quad (14)$$

Given the ability to compute the density of $\hat{u}_n$ we can easily derive the densities of $\tilde{U}_n$ and U_n. Considering $\tilde{U}_n$, we note that $\tilde{U}_n = \max\{0, \hat{U}_n\}$ holds, as the actual buffer can never attain a negative value. Hence, the following density function is obtained using the π_0 sweep operator:

$$\tilde{u}_n(k) = \pi_0(\hat{u}_n(k))\,. \quad (15)$$

Lastly, as the buffer level U_{n+1} *immediately after* the arrival of segment n computes to $U_{n+1} = \tilde{U}_n + B_n$, the convolution of the respective densities is considered to obtain:

$$u_{n+1}(k) = \tilde{u}_n(k) * b_n(k)\,. \quad (16)$$

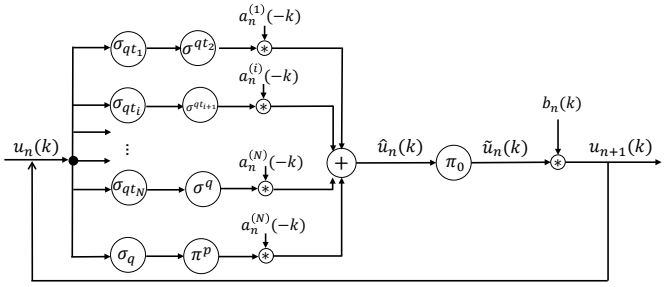


Fig. 2. Computational diagram for the buffer-based model.

Given Equation 16 and a density function u_n, the density u_{n+1} can now be computed as the employed Equations 14 and 15 only depend on the known distribution of u_n. Hence, given an initial distribution for u_1 (modeling the empty buffer), the exact density functions for *any* segment can be readily computed and Figure 2 summarizes the computation of the buffer distribution.

D. Rate-based Model

We now turn to the rate-based model. Again, the goal again is to derive the probability density functions u_n, $\hat{u}_n$, $\tilde{u}_n$. The main difference with respect to the buffer-based model is that the quality selection now only depends on the last previously observed data rate D_{n-1}: if $D_{n-1} \in Q_i$ holds, then the segment n is requested with quality i. Additionally, again the buffer thresholds p and q to control the size of the buffer are to be enforced: once U_n exceeds q, sending requests is delayed until the buffer reaches the lower threshold p again.

Given the fixed probability distributions $\{d_n\}_n$ for the download rate of the n-th segment, the probability that a segment will be requested using a specific quality computes to $P(D_{n-1} \in Q_i) = \sum_{k \in Q_i} d_{n-1}(k)$ for $i \in \mathcal{Q}$.

Analogously to the analysis of the buffer-based model, we start by deriving the density of $\hat{u}_n$ and $\tilde{u}_n$, which will then again be used to compute u_{n+1}. Considering $\hat{u}_n$, fewer cases need to be considered (cf. Equations 8 to 10), as the requested quality only depends on the previous data rate. Nevertheless, a distinction has to be made depending on whether the buffer exceeds the threshold q. For $U_n < q$, the density computes to:

$$\hat{u}_n^{<q}(k) = \sum_{i \in \mathcal{Q}} P(D_{n-1} \in Q_i) \cdot (u_n(k) * a_n^{(i)}(-k)) \quad (17)$$

In the other case, i.e., for $U_n \geq q$, the request is delayed until buffer level p is reached, and the density computes to:

$$\hat{u}_n^{\geq q}(k) = \sum_{i \in \mathcal{Q}} P(D_{n-1} \in Q_i) \cdot (\pi^p(\sigma_q(u_n(k))) * a_n^{(i)}(-k)) \quad (18)$$

To merge the above Equations 17 and 18, note that both only differ in whether the π^p operator is applied or not and their conditioning. Accordingly, to unify both, we introduce the following buffer distribution u'_n, where the probability mass above q is realigned to p:

$$u'_n(k) = \sigma^q(u_n(k)) + \pi^p(\sigma_q(u_n(k))) \quad (19)$$

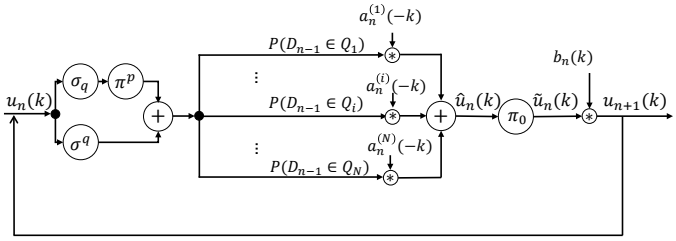


Fig. 3. Computational diagram for the rate-based model.

Using this adapted density function u'_n, the density $\hat{u}_n$ of the virtual buffer $\hat{U}_n$ computes to:

$$\hat{u}_n(k) = \sum_{i \in \mathcal{Q}} P(D_{n-1} \in Q_i) \cdot (u'_n(k) * a_n^{(i)}(-k)) \,. \tag{20}$$

Lastly, we note that the computation of the densities $\tilde{u}_n$ and u_{n+1} only rely on the computation of $\hat{u}_n$ (cf. Equations 15 and 16). Accordingly, the respective formulas still holds for the rate-based model, albeit now using $\hat{u}_n$ defined in Equation 20:

$$\tilde{u}_n(k) = \pi_0(\hat{u}_n(k)) \tag{21}$$

$$u_{n+1}(k) = \tilde{u}_n(k) * b_n(k) \,. \tag{22}$$

Figure 3 summarizes the above observations and represents the iterative computation of buffer densities.

E. Metric Computation

In the following, performance metrics are formulated based on the above derived density distributions pertaining to the random variables U_n, $\hat{U}_n$, and $\tilde{U}_n$. While the model generally allows to study the performance before and after the n-th segment arrival (by iteratively computing the respective densities), in the following only steady state performance metrics will be considered. To this end, we have to assume that the input probability distributions are either fixed, i.e., $b_n = b$, $c_n^{(i)} = c^{(i)}$, and $d_n = d$ hold for $n \geq 0$ or that the density series $\{b_n\}_n$, $\{c_n^{(i)}\}_n$, and $\{d_n\}_n$ converge, such that the steady state density functions $b = \lim_{n\to\infty} u_n$, etc., are well-defined. In either of the above cases, the buffer related density functions will eventually converge as well.

We first consider performance metrics based purely on the buffer levels and then discuss metrics pertaining to the quality.

1) Buffer-related Metrics: The first metric considered is the average buffer level. Since a precise computation of the buffer level at any point in time is cumbersome and involves the computation of recurrence times [**?**], we rely on the buffer level u at the segment arrival times. This metric can easily be computed, both for model and measurements. Accordingly, we estimate the average buffer level (at the segment arrivals) as:

$$\text{AVERAGE BUFFER LEVEL:} \quad \bar{u} = E(u) = \sum\nolimits_k k \cdot u(k) \tag{23}$$

We now consider the stalling probabilities and duration, both of which are indicated by negative $\hat{U}_n$ values. Specifically, whenever $\hat{U}_n$ lies below 0, stalling of exactly $-\hat{U}_n$ time units has occurred. Considering the steady state probabilities, the stalling probability computes to:

$$\text{STALLING PROBABILITY:} \quad p_{st} = \sum\nolimits_{k<0} \hat{u}(k) \tag{24}$$

As the stalling probability does not account for the actual stalling duration, we compute the duration by considering the negated expected value of $\hat{u}$ conditioned on the negative buffer levels and obtain:

$$\text{STALLING DURATION:} \quad L = -\sum\nolimits_{i<0} i \cdot \hat{u}(i) \tag{25}$$

2) Quality-related metrics: We now discuss quality-related metrics. In contrast to the above presented metrics, the quality-related metrics either depend on the buffer levels or the download durations, for the buffer-based and the rate-based models, respectively.

We first consider the average quality, which we define to lie in the range from 1 to N according to the definition of quality levels $\mathcal{Q}$. In the buffer-based model, the quality is dictated by the buffer level after having received a segment, i.e., u. For the rate-based model, the average quality is only determined by the steady state download rate d. Accordingly, we have:

$$\text{AVERAGE QUALITY:} \quad \bar{Q}_{buffer} = \sum_{i \in \mathcal{Q}} i \cdot \sum_{k \in Q_i} u(k) \tag{26}$$

$$\bar{Q}_{rate} = \sum_{i \in \mathcal{Q}} i \cdot \sum_{k \in Q_i} d(k) \tag{27}$$

Another performance metric of interest is the stability of quality levels in terms of the relative number of segments witnessing a quality level change. The following holds for the (steady state) probabilities of switching quality levels:

$$\text{SWITCHING PROBABILITY:} \quad P_{buffer}^{switch} = \lim_{n\to\infty} \sum_{i \in \mathcal{Q}} P(U_n \in Q_i, U_{n+1} \notin Q_i) \tag{28}$$

$$P_{rate}^{switch} = \lim_{n\to\infty} \sum_{i \in \mathcal{Q}} P(D_n \in Q_i, D_{n+1} \notin Q_i) \tag{29}$$

We first derive a concise formula for the buffer-based model, i.e., P_{buffer}^{switch}. As analyzed in Section III-C, the random variable U_{n+1}, respectively its density, depends on U_n (cf. Equation 16). As the different 'computational branches' of the computation of u_{n+1} (cf. Figure 2) need to be taken into account we first decompose P_{buffer}^{switch} into the sum:

$$\lim_{n\to\infty} \begin{pmatrix} \sum_{i=1}^{N} P(U_n \in Q_i,\ U_{n+1} \notin Q_i,\ U_n < q) \\ +\ \ P(U_n \in Q_N, U_{n+1} \notin Q_N, U_n \geq q) \end{pmatrix} \tag{30}$$

Now, conditioning on the quality level of U_n and considering the steady state distribution u, the switching probability P_{buffer}^{switch} can be computed as follows:

$$P_{buffer}^{switch} = \begin{pmatrix} \sum_{i=1}^{N} \sum_{k \notin Q_i} \left(\pi_0[\sigma_{qt_i}(\sigma^{qt_{i+1}}(u(k))) * a^{(i)}(\text{-}k)] * b(k)\right) \\ +\ \sum_{k \notin Q_N} \left(\pi_0[\pi^p(\sigma_q(u(k))) * a^{(N)}(\text{-}k)] * b(k)\right) \end{pmatrix} \tag{31}$$

Intuitively, above the σ-operators represent the conditioning of U_n having a specific quality (of steady state density u), while the inner formulas are obtained by following the computational specification to obtain the buffer level probabilities for

TABLE II
MODEL INPUT AND MEASUREMENT PARAMETERS

PARAMETER	VALUE
maximum buffer	40 seconds
buffer-based quality thresholds	$qt_2 = 10s$, $qt_3 = 20s$ $qt_4 = 30s$
rate-based quality thresholds	$qt_i = 1.15$-fold of avg. rate of level i
bandwidth provisioning factor a	$0.8, 1.0, 1.2, 1.4, 1.6, 2.0$
bandwidth coefficient of variation c_v	$0, 0.2, 0.4, 0.6$

TABLE III
CHARACTERISTICS OF THE TEST VIDEO SEQUENCE

LEVEL	BITRATE	FILESIZE AVG.	FILESIZE STD.	FILESIZE c_v
1	563	2837	1167	0.4115
2	1098	5510	2356	0.4276
3	1634	8192	3689	0.4503
4	2170	10868	5135	0.4726

U_{n+1} and then considering all buffer levels k where another quality level was chosen ($k \notin Q_i$).

For the rate-based approach the switching probability only depends on the (steady state) download rate and can be readily computed using basic probability theory:

$$P_{rate}^{switch} = \lim_{n\to\infty} \Big(\sum_{i=1}^{N} P(D_n \in Q_i, D_{n+1} \notin Q_i) \Big) \quad (32)$$

$$= \lim_{n\to\infty} \Big(1 - \sum_{i=1}^{N} P(D_n \in Q_i) \cdot P(D_{n+1} \in Q_i) \Big) \quad (33)$$

$$= 1 - \sum_{i\in\mathcal{Q}} \Big(\sum_{k\in Q_i} d(k) \Big)^2 . \quad (34)$$

IV. VALIDATION

In the following, we validate the results obtained by the model with testbed measurements. This is done for both cases, the buffer-based as well as the rate-based approach. We first describe the methodology, i.e., the measurement setup and parameter configurations. Afterwards, we describe the results.

A. Evaluation Methodology

This subsection describes the measurement setup and the configurational parameters used for the comparison between the results gained by measurements and analysis.

1) Measurement Setup: We validate the model with measurements from a virtual testbed setup. It consists of a video server and an HAS client, whereby each tenant operates within a dedicated Linux namespace[1]. Both namespaces are connected via a virtual interface, which is throttled using the linux traffic control[2]. We use the open source player Tapas [20]. For validating the rate-based model approach, we use the implementation of PANDA [3]. To validate the buffer-based approach, we implemented a simple heuristic, which selects the quality based on the current buffer state and predefined thresholds. For the sake of scalability, the videos are not rendered at the client, but the built-in fake decoder is used.

2) Video Characteristics: We use the first three minutes from the Big Buck Bunny clip provided by Blender foundation[3]. The test sequence consists of 48 segments with a duration of 5 seconds each. We encode the clip so to obtain four quality levels with the characteristics shown in Table III. The average filesize and standard deviation per quality level are provided as input to the model, which again uses these values for generating a negative binomial segment size distribution.

[1]http://man7.org/linux/man-pages/man7/namespaces.7.html
[2]https://linux.die.net/man/8/tc
[3]https://www.blender.org/foundation/

3) ABR *Configurations and Network Settings:* For the buffer-based approach, we set the quality switching thresholds to $qt_1 = 0$, $qt_2 = 10$, $qt_3 = 20$, and $qt_4 = 30$. In case of the rate-based approach, we set the safety margin, i.e., the percentage to which the throughput rate has to exceed the bitrate of a certain quality level, to 15%. Accordingly, the obtained thresholds are computed using the average bitrate per quality level as $qt_i = 1.15 \cdot C^{(i)}$, for $i \in \{2, 3, 4\}$, with $C^{(i)}$ denoting the average bitrate of quality level i. For the first threshold $qt_1 = 0$ holds. The maximum buffer threshold is set to $q = 40$. As we do not consider a segment request resume threshold, we set $p = q = 40$.

For the available average bandwidth, we define the bandwidth provisioning factor a. It corresponds to the ratio of the average bandwidth to the average bitrate of the lowest quality level. A fluent video playback is generally only possible if $a > 1$. We also evaluate different bandwidth variations as indicated by different coefficients of variation, c_v. For measurements with specific parameter settings for a and c_v, we generate custom network trace files whose bandwidth values are randomly generated on a per second basis using a negative-binomial distribution.

The parameter settings for different bandwidth provisioning factors and coefficients of variation, as well as the configurations of for the ABR are summarized in Table II.

4) Other Properties: In order to enhance the comparability of model and measurements, we deactivated the throughput estimation smoothing which is applied in the PANDA adaptation scheme. It is used to compensate short-time bandwidth fluctuations by taking past measurements into account. We furthermore consider the buffer only immediately after segment arrival, for both, measurements and model. The actual average buffers are expected to be lower. In order to account for overhead such as MPD file download or TCP/IP packet headers, we reduce the available bandwidth in the model by a factor of 0.07. This value is chosen as we obtain an average of 0.93 when dividing the received video bytes by the available bandwidth during measurements. Finally, we note that for each parameter combination, 10 testbed measurement runs have been performed.

B. Results

In the following we compare the results obtained by measurements with the model. Thereby, we distinguish between the two applied ABR schemes and give a short summary afterwards. For the measurements we utilize the encoded video clip mentioned above, while for the analytical computations we rely on the bitrate distribution of the segments. Hence,

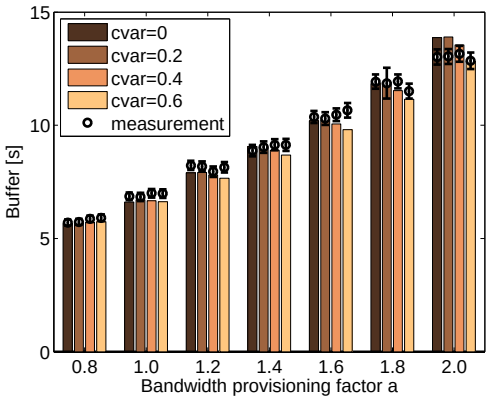


Fig. 4. Comparison of avg. buffer level as computed by the analytical model (bars) with buffer-based quality selection and the average buffer obtained from testbed measurements (dots) along with the 95% confidence intervals.

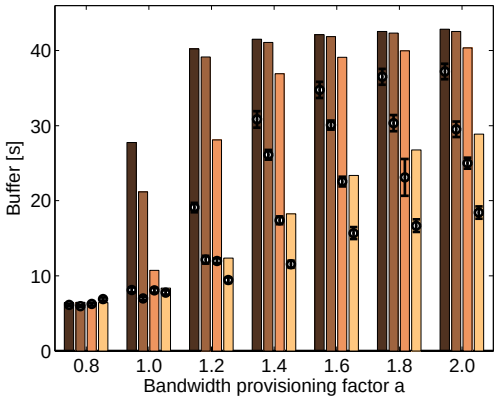
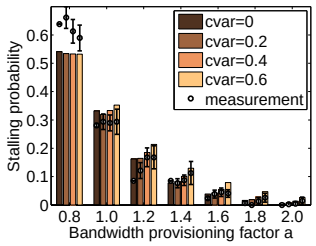
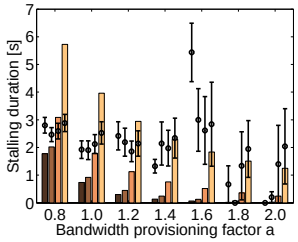


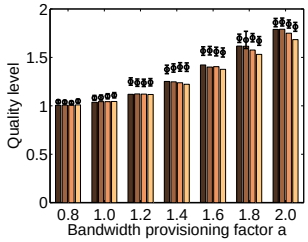
Fig. 6. Comparison of average buffer level as computed by the analytical model (bars) with rate-based quality selection and the average buffer obtained from testbed measurements (dots) along with the 95% confidence intervals.



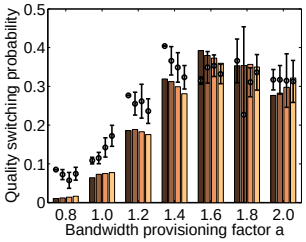
(a) Stalling probability



(b) Average stalling duration

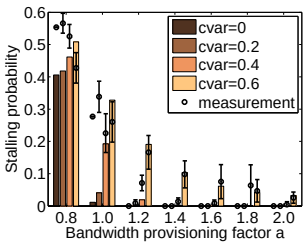


(c) Average video quality level

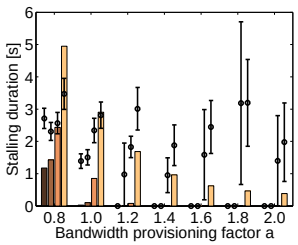


(d) Quality switching probability

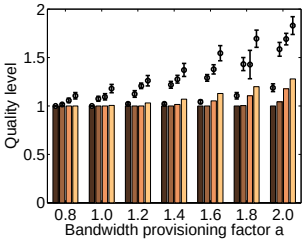
Fig. 5. Results of the buffer-based model validation. Bars denote the analytical model's output, dots denote the measurement results along with the 95% confidence intervals.



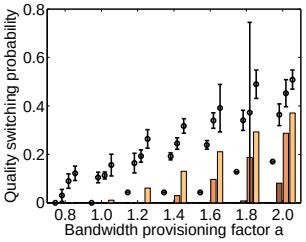
(a) Stalling probability



(b) Average stalling duration



(c) Average video quality level



(d) Quality switching probability

Fig. 7. Results of the rate-based model validation. Bars denote the analytical model's output, dots denote the measurement results along with the 95% confidence intervals.

the analytically computed results do not take the temporal correlation of the video clip into account.

1) Buffer-based Approach: Figure 4 shows the results pertaining to the video buffer using measurements and the analytical approach. The x-axis depicts the bandwidth provisioning factors, the y-axis denotes the buffer level in seconds. Values obtained from the analytical model are depicted as bars, whereas the colours express different c_v values. The dots represent the buffer levels obtained from the testbed measurements.

It can be seen that the model is quite accurate for all evaluated c_v parameters and provisioning factors a.

The results for stalling probability, average stalling duration, average video quality and quality switching probability are depicted in Figure 5. For stalling probability, average video quality and quality switching probability the model allows a quite accurate computation of the specific values, as depicted in the Figure 5a, Figure 5c and Figure 5d. For the stalling durations, depicted in Figure 5b, the measurement results are much higher compared to the analytical results, but also show larger confidence intervals. This can be attributed to the fact that the low number of stallings, particularly for bandwidth provisioning factors $a \geq 1.4$ leads to a small stalling probability and thereby a small sample size for stalling durations, since only 48 segments are downloaded within a single measurement run.

2) Rate-based Approach: Figure 6 depicts the buffer levels obtained from the rate-based model and compared with the buffer values obtained from testbed measurements. For a bandwidth provisioning factor of $a = 0.8$, the model's output is accurate for all coefficients of variation. For $a \geq 1.0$, the model tends to overestimate the buffer, however, the trend of decreasing buffer levels with increasing c_v is reflected.

Figure 7 depicts the testbed measurement results and model

outputs for HAS performance metrics. The estimation of the stalling probability (Figure 7a) is quite reliable, especially for cases where $c_v = 0.6$ and in cases where no stallings occur.

Again, there are large differences in terms of stalling duration (cf. Figure 5b), which can be attributed to the low number of stalling events and the short measurement runs.

Figure 7c and Figure 7d show that the analytical computation is more conservative, i.e., underestimates both metrics. This holds for all investigated parameter combinations.

3) Summary: To sum up, the comparisons between model and measurements indicate that the model enables a very good prediction of all metrics despite the stalling durations for the buffer-based ABR method. For rate-based ABR, the model is accurate for specific parameter combinations and also captures trends when modifying the input parameters. It should be noted that the measurements were conducted using a relatively short video, which in turn leads to a small sample size when evaluating metrics like the stalling duration. Here, a higher statistical significance in the measured results could be obtained when using a longer video clip. Concerning the discrepancy between measurements and analytical results one should also note that due to the on-off behavior of adaptive video streaming, the ABR mechanisms tend to inaccurately estimate the available bandwidth [21]. This is not captured by the model so far.

V. Conclusion

HTTP adaptive video streaming has become the prevalent way of video consumption in the Internet. In order to optimize the user-perceived quality many improvements, e.g., new ABR schemes, and best practices, e.g., with respect to the number of quality levels or segment duration, have been proposed. The impact of these adjustments is typically evaluated by employing measurements for specific use-cases, which limits the generalization of the gained results. This results in a limited understanding of the influence that various factors have on the performance metrics of adaptive video streaming across various operational settings.

This paper addresses this limited understanding by generalizing previous models to obtain a computationally tractable discrete-time model suitable for ABR systems. The proposed model allows to study all relevant HAS performance metrics and thus enables a fundamental understanding of HAS yielding generalizable results. By implementing the analytical model and comparing the results to testbed measurements for a vast number of different bandwidth configurations, the practical applicability of the presented model is shown.

Acknowledgment

The authors would like to thank Leonie Reichert for her continuous support during the course of this work. This work has been supported by the DFG Grant "SDN-enabled Application-aware Network Control Architectures and their Performance Assessment" (ZI1334/2-1).

References

[1] "Cisco Visual Networking Index: Forecast and Trends, 2017–2022," White Paper, Cisco Syst., Inc., 02 2018.

[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] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. C. Begen, and D. Oran, "Probe and adapt: Rate adaptation for http video streaming at scale," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 719–733, 2014.

[4] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "Bola: Near-optimal bitrate adaptation for online videos," in *IEEE INFOCOM*, 2016.

[5] 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.

[6] 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.

[7] S. Schwarzmann, P. Breitbach, and T. Zinner, "Computing qoe–relevant adaptive video streaming metrics using discrete–time analysis," in *The Third International Workshop on Quality of Experience Management*. IEEE, 2019. [Online]. Available: https://tinyurl.com/y5juvopb

[8] C. Zhou, C.-W. Lin, X. Zhang, and Z. Guo, "Buffer-based smooth rate adaptation for dynamic http streaming," in *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*. IEEE, 2013, pp. 1–9.

[9] H. T. Le, D. V. Nguyen, N. P. Ngoc, A. T. Pham, and T. C. Thang, "Buffer-based bitrate adaptation for adaptive http streaming," in *2013 International Conference on Advanced Technologies for Communications (ATC 2013)*. IEEE, 2013, pp. 33–38.

[10] C. Mueller, S. Lederer, R. Grandl, and C. Timmerer, "Oscillation compensating dynamic adaptive streaming over http," in *2015 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2015, pp. 1–6.

[11] C. Liu, I. Bouazizi, and M. Gabbouj, "Rate adaptation for adaptive http streaming," in *Proceedings of the second annual ACM conference on Multimedia systems*. ACM, 2011, pp. 169–174.

[12] C. Liu, I. Bouazizi, M. M. Hannuksela, and M. Gabbouj, "Rate adaptation for dynamic adaptive streaming over http in content distribution network," *Signal Processing: Image Communication*, vol. 27, no. 4, pp. 288–311, 2012.

[13] B. Rainer, S. Lederer, C. Müller, and C. Timmerer, "A seamless web integration of adaptive http streaming," in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*. IEEE, 2012, pp. 1519–1523.

[14] A. Bentaleb, B. Taani, A. C. Begen, C. Timmerer, and R. Zimmermann, "A survey on bitrate adaptation schemes for streaming media over http," *IEEE Communications Surveys & Tutorials*, 2018.

[15] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 187–198, 2015.

[16] 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.

[17] Z. Li, A. C. Begen, J. Gahm, Y. Shan, B. Osler, and D. Oran, "Streaming video over http with consistent quality," in *Proceedings of the 5th ACM Multimedia Systems Conference*. ACM, 2014, pp. 248–258.

[18] L. De Cicco, G. Cofano, and S. Mascolo, "A hybrid model of the akamai adaptive streaming control system," *IFAC Proceedings Volumes*, 2014.

[19] L. Kleinrock, *Queueing Systems: Volume 1: Theory, 1975*. Wiley-Interscience, 1975.

[20] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo, "Tapas: a tool for rapid prototyping of adaptive streaming algorithms," in *Proceedings of the 2014 Workshop on Design, Quality and Deployment of Adaptive Video Streaming*. ACM, 2014, pp. 1–6.

[21] C. Wang, A. Rizk, and M. Zink, "Squad: A spectrum-based quality adaptation for dynamic adaptive streaming over http," in *Proceedings of the 7th International Conference on Multimedia Systems*. ACM, 2016.