Linking QoE and Performance Models for DASH-based Video Streaming

Susanna Schwarzmann* and Thomas Zinner§
*TU Berlin, 10587 Berlin, Germany, susanna@inet.tu-berlin.de
§NTNU - Norwegian University of Science and Technology, 7041 Trondheim, Norge, thomas.zinner@ntnu.no

***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. Recently a couple of analytical models for video streaming have been proposed allowing to compare these input parameters and derive their impact on QoE-relevant input parameters for HAS-based video delivery. The outcome of these models are typically asymptotic probabilities, distribution functions or centralized and standardized moments. This contradicts to QoE prediction models like P.1203 which compute the QoE based on the chronological sequence of a specific video playback. So far, it is unclear how and to which extend the generalized results of the analytical models can be utilized to derive sequence-based QoE values or the QoE distribution for a set of sequences for similar input parameters with stochastic variations. To address this problem, we compare measurements with the output of a GI/GI/1 model with pq-policy and buffer-based quality switching capability and conclude to which extend the results still allow to approximate the video QoE.**

***Index Terms*—Adaptive video streaming, Modeling, QoE estimation, DASH**

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 properties of the segments are summarized in an XML-based media presentation description (MPD) file. The DASH client requests the MPD file and afterwards downloads the segments in a quality dictated by the client's internal quality adaptation strategy.

The adaptation strategy considers a combination of parameters to decide about the next segment's quality, so 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]–[8]. They differ with regard to their quality selection process, which allows them to improve the played back video quality, while reducing video stallings. Alongside the adaptation strategies, thresholds for the initial buffer time or the segment duration have a high impact on the QoE [9].

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, and the output is typically assessed using QoE models for video streaming like P.1203 [10].

Recently a couple of queueing-based models [11]–[13] have been developed. These models are based on certain assumptions regarding the adaptation strategy and other relevant parameters, but allow to easily compute QoE-relevant metrics like the stalling probability for a large set of different network scenarios and parameter settings. The outcome of these models are typically asymptotic probabilities, distribution functions or centralized and standardized moments. This contradicts to the aforementioned QoE prediction models which compute the QoE based on the chronological sequence of a specific video playback. So far, it is unclear how and to which extend the generalized results of the analytical models can be utilized to derive sequence-based QoE values or the QoE distribution for a set of sequences for similar input parameters with stochastic variations.

To address this problem, we implement the GI/GI/1 model with pq-policy and buffer-based quality switching capability presented in [13] and derive chronological video playback sequences using a Monte Carlo approach for a specific video clip and different networking scenarios. For these sequences we compute the perceived QoE distributions using the open source implementation of P.1203 [14]. Similarly, we conduct measurements for the same network scenarios using the the Bola ABR controller implemented in the dash.js framework and further compute the corresponding QoE distribution. Our evaluations show that the output of the performance model allows a good QoE estimation.

II. Related Work and Background

A. HTTP Adaptive Streaming

HTTP Adaptive Streaming (HAS) allows to adapt the play back video quality to current network conditions. To do so, the video is split into segments of equal duration, typically within a range of 2 and 10 seconds. Each of these segments is encoded several times using different bitrates or scaled to different resolutions, so to obtain several quality representations for each segment. The Media Presentation Description (MPD) is downloaded by the client and lists the required meta data, such as available qualities, the segment duration, and the URL to the specific video segments. A DASH heuristic running at the client selects the quality representation to download next. The decision is either based on the current video buffer level, the estimated throughput, or both. Accordingly, one distinguishes between buffer-based, throughput-based, and hybrid HAS heuristics. The heuristics decide so to maximize the playback quality, whilst simultaneously avoiding rebuffer events, i.e. video stallings.

B. HAS Performance Models

HAS performance models allow to compute QoE influence factors (QoE-IF) based on certain assumptions. $M/M/1/\infty$ models, for example, work on a high level of abstraction on the one hand, but allow to easily compute relevant metrics. Such a model, applying a pq-policy, is presented in [12]. The threshold q denotes the buffer value which triggers the client enter idle state, i.e. to pause requesting segments. The idle state is left as soon as the buffer level is below threshold p, i.e. the client resumes requesting video portions. The proposed model is applied to investigate the impact of user profiles on the QoE of adaptive streaming. The authors 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 video versus browsing videos.

As part of presenting a HAS adaptation heuristic, De Cicco et al. [15] formalizes the behavior of an Akamai video streaming session. The authors model the system as a hybrid automaton. To do so, they use upon others the quality level, the current throughput, and the playout buffer as state variables. With their model, the authors show that rebuffer events can be prevented when the quality switching thresholds are properly tuned. Furthermore, they motivate to properly set the ratio between idle state and segment downloading to avoid large buffering, which would result in high network resource wasting in case the user aborts the video.

Burger et al. [16] applies discrete time-analysis to model the video buffer of an HAS client. From the computed video buffer distribution, conclusions about the probability of stalling events as well as their durations can be drawn. The video buffer is modeled as a $GI/GI/1$ queue with pq-policy. The buffered play time at the client is considered as the amount of unfinished work in the system. Playing back the video corresponds the service time, i.e. draining the buffer. The work of [13] builds upon the work of Burger et al. [16] by extending the model so to allow modeling the DASH quality switching behavior.

As shown in the evaluations of [16], those models are capable to compute QoE influence factors, such as stallings, to a decent degree of accuracy. There is quite a number of parameters influencing HAS performance, such as buffer thresholds, bandwidth variability, or video characteristics. To optimize all adjustable and non-adjustable parameters in this large problem space by doing measurements or simulation will quickly hit the wall. The advantage of those models, compared to measurements or simulation, is the cost-efficient computation. On the other hand, however, those models work on a certain level of abstraction and omit parts of real systems. For example, when considering a bandwidth distribution as input, temporal aspects and correlations are commonly neglected. Furthermore, the results from those models are asymptotic and thus involve a loss of information, e.g. how stalling probabilities evolve over time. So far, it is not clear if the output of HAS performance models can be used as an input for existing QoE models.

C. Modeling QoE

Yin et al. [17] proposes to model video QoE as a weighted sum of average video quality, average quality variation, the total rebuffer time, and the start-up delay. The model omits temporal dependencies, i.e. when do quality switches or rebuffering events occur. However, it was shown that the timings of those events affect the user's perceived quality, this fact is often referred to as recency effects [18].

The standardized model ITU-T P.1203 [14], which has specifically been designed for HAS, takes the temporal characteristics of quality switches and rebuffer events into account. For its development and validation, 30 subjective test databases and over 1000 audiovisual sequences have been used. The ITU-T standard gains attention in the research community. Robitza et al. [19] applied the model to measure YouTube QoE under constrained bandwidth conditions. Seufert et al. [20] studied the impact of various application-level performance indicators on the QoE computed by the model.

III. Methodology

A comprehensive overview on the applied methodology is illustrated in Figure 1. It illustrates the whole process with respect to the measurement runs and model-based evaluations, the time series depicting the video playback behavior, and the QoE assessment using ITU-T P.1203.

The investigation of a specific video clip for different network traces using testbed measurements and the mathematical model are shown on the top left. As network configurations we consider a fixed bandwidth scenario and three scenarios based on a real bandwidth trace [21]. We pick the trace *Car*, depicting a bandwidth time series during a car ride on a time scale of one second. For the latter, we further distinguish between fixed order of the trace, random starting point with turnaround and shuffle. Segment sizes and the

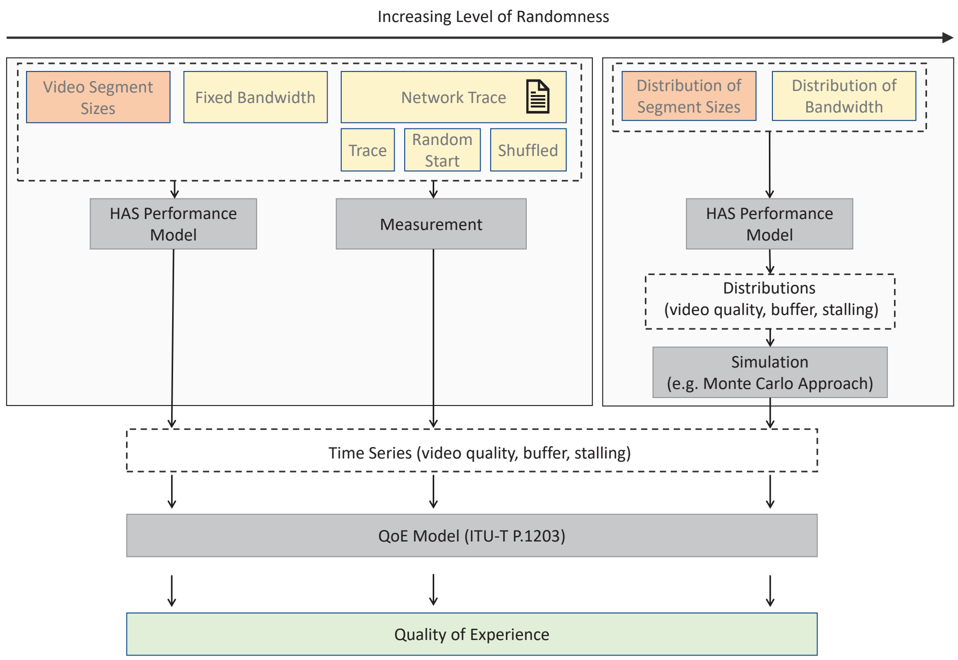

Fig. 1. Methodology overview

different bandwidth configurations are used as input parameters for the analytical HAS performance model (III-B)and for the measurements (III-A) to generate time series of HAS parameters, such as quality or buffered play time.

On the right side distributions for segment sizes and the available bandwidth are considered. The distributions are computed with respect to the utilized video clip and network traces and thus are an abstraction of the previously mentioned time series.These distributions are taken as input for the performance model, which return steady state results of relevant adaptive streaming metrics like stalling probability or video buffer distribution. In order to generate video play back sequences from these distributions, we perform Monte Carlo [**?**] simulations (III-C).

Hence, we constantly increase the level of randomness of the input data starting with fixed traces and time series to probability distributions. To study the impact of disorder and abstraction on the reliability of estimating QoE, we use the time series obtained by the different methods as input for the ITU-T P.1203 QoE model (III-D). The used scripts and configuration files will be published on GitHub[1].

A. Testbed Measurements

Our measurement setup consists of three virtual machines. The first VM acts as a network emulator (netem) and connects the server VM with the client VM. By throttling the outgoing traffic on the netem interface towards the client, we can emulate different bandwidth characteristics. We consider on the one hand a bandwidth which does not change throughout the experiments, which we refer to as *static*. On the other hand, we use a bandwidth trace [21], which we scale so to achieve similar average values as for the static case. The bandwidth characteristics are summarized in Table I, with c_{var}. Table I summarizes the bandwidth characteristics, i.e., average bandwidth, standard deviation and the corresponding coefficient of variation c_{var}.

TABLE I
CHARACTERISTICS OF APPLIED BANDWIDTH CONFIGURATIONS

Static	Trace			
	Name	Average	Standard deviation	c_{var}
500	trace0	496	260	0.52
1000	trace1	992	520	0.52
2000	trace2	1984	1040	0.52

The trace is scaled for different average bandwidth values and used in the measurements in three different ways. Firstly, we update the available bandwidth according to the order in the trace, which we refer to as *trace*. Secondly, we start the trace on a random point and preserve the timely order. In case the end of the trace is reached, we loop the trace and proceed with the first value in the trace. We refer to this mode as *random start*. Lastly, we shuffle the trace, i.e. we do not preserve any timely order. This mode is called *shuffled*. The server VM runs a customized version of the dash.js [2] player using the Bola [8] adaptation heuristic. As a test video, the server provides a snippet of 240 seconds of the Big Buck Bunny clip [3]. We use segments of 4 seconds duration and provide each segment in 4 quality layers with the bitrate characteristics as shown in

[1]repository not included due to double-blind review process

[2]https://github.com/Dash-Industry-Forum/dash.js

[3]https://peach.blender.org/

Table II. Measurements are repeated 20 times to take statistical variations into consideration.

TABLE II
QUALITY LEVELS AND BITRATES OF THE PROVIDED VIDEO

Level	Average bitrate [kbps]	Standard deviation [kbps]
1	510	154
2	1016	313
3	1524	474
4	2034	634

The client initiates the video stream using google-chrome browser with disabled cache.

B. HAS Performance Model

The discrete-time GI/GI/1 queue [22] has recently been applied to performance analysis of adaptive video streaming systems given certain degrees of freedom with respect to the utilized inter-arrival time and the service time distributions, as well as the flexibility with respect to the composition of the outlined distributions. It is possible to model the random variables following an independent and identical distributions (iid), but also following correlated or time-dependent distributions. To better understand this feature let's consider a simple video streaming example with a video clip transmitted via a network with a fixed throughput. The video clips is segmented within i segments of fixed segment length, e.g., 4 seconds. The content and playback order of a video clip are fixed resulting in the same segment order and size $C_1, C_2, C_3, ...C_i$ to be transmitted, independent of the current time and possible repetitions. Since we also assume a fixed throughput, a fixed number D of data can be transmitted per second via the network. The corresponding transmission durations for the segments can easily be computed as $A_1 = \frac{C_1}{D}, ..., A_i = \frac{C_i}{D}$. Hence, this process is not random at all, but nevertheless the arrival process can be modeled using a sequence of i deterministic distributions with dirac delta at $A_1, A_2, ...A_i$. After each arrival the video buffer at the client is then increased by the segment length, and the next segment is downloaded, and asymptotic or steady-state performance metrics can be computed using the discrete-time model presented in [11].

A more complex example would be that only random distributions for segment size and the available throughput are known. In this case the corresponding random processes C and D would be modeled iid. Hence, all possible combinations of segment sizes and network throughput can happen, with some being more likely than others depending on the specific distributions. Hence, the transmission durations can be computed as ratio distribution, as outlined in [11]. After each arrival, the video buffer at the client is then increased by the segment length, and the next segment is downloaded, i.e., these transmission times are a reasonable approximation of the segment inter-arrival time at the client. Using segment length and inter-arrival distributions, it possible to compute steady-state performance metrics using the discrete-time model.

In both cases, the model will compute the video buffer client distribution at the points in time when a segment download has ended and a new video segment is downloaded. This method can be utilized for buffer-based ABR mechanisms, as shown in [13], where the decision which quality to download next is taken after a segment is downloaded and the playtime of the segment is added to the video buffer. In this case, the segment size distributions for different qualities and the throughput distribution are used to compute the inter-arrival time distribution for each quality. In case of iid inter-arrival and service time distributions the outlined download and playback process continuous independent from past system states, only depending from the current system states. The corresponding quality to download, and therewith the segment size depends on the current video buffer level and predefined switching thresholds. Hence, it is possible to embed a markov chain at this "renewal" points and to describe the process using a transition matrix P, i.e., a matrix summarizing the transition probabilities between all possible video buffer levels at the embedding points. The steady state probabilities of the video buffer can then be computed by finding the eigenvectors of P for the eigenvector 1. If the transition matrix cannot be computed due to its size it is possible to compute the video buffer distribution using an iterative approach as outlined in [22]. Further metrics like the stalling probability, switching probabilities or probabilities for switching amplitudes can be derived using the video buffer distribution, as outlined in [13].

When applying the model, we consider the same video quality levels and characteristics as in the testbed measurements. The buffer threshold to switch from level 1 to level 2 is set to 10 seconds. 15 seconds of buffered ply time are required to download level 3, and 20 seconds for requesting the highest quality.

C. Video Playback Sequence Generation

. In this subsection we describe how to generate video time sequences based on the output of the HAS performance model. To generate a couple of different video playback sequences we use a Monte Carlo simulation approach, namely a random walk. We start with the empty system and compute the transmission time for the first segment. This is done by drawing a random inter-arrival time from the corresponding segment inter-arrival time distribution for the selected quality defined by the current buffer level and the switching thresholds. The video playback buffer depletes until the next segment arrives at the client resulting in an increase of the video playback buffer by a random variable following the segment length distribution. Relevant QoE metrics like the downloaded video quality and stalling time and duration are logged. Then, the random walk is continued by drawing further random variables for inter-arrival times depending on the developing video buffer state, as well as random variables for the segment length distributions. This is continued until the desired number of segments are downloaded. In total, we randomly generate 50 video playback sequences using the outlined method.

D. QoE Estimation using P.1203

The P.1203 model can be applied in various modes, which differ in terms of the required input data granularity and the goodness of the QoE prediction. We apply mode 0, which firstly considers the audio visual quality as an input parameter. As our test video does not contain an audio track, the model assumes a high constant audio quality throughout the video. Concerning stallings, the model considers their temporal placement as well as the respective durations. Finally, the visual quality is passed as the video bitrate on a per-segment scale. Hence, the actual bitrate of each video segment has to be included. This can be done in the case of measurements and in the case of applying the HAS performance model with the actual segment sizes and bandwidth values, i.e. fixed bandwidth or traces. However, if the playback sequence is generated based on distributions for bandwidth and segment sizes, the bitrate per video segment is not available. In this case, we use the average bitrate of the respective quality level for all segments. The resulting output of P.1203 is the mean opinion score (MOS) for the specified video playback sequence.

IV. Evaluation

We first evaluate the obtained QoE values for different bandwidth modes with an average available bandwidth of 500 kbps. The results are depicted in Figure 2. The plot on the left side shows the results for the two bandwidth modes without randomness, i.e. the *static* bandwidth and the bandwidth *trace*. In case of *static*, the HAS performance model (pm) returns a MOS value of 1.94 and a MOS value of 1.98 when the bandwidth trace is applied. As the model computes stallings and the next segment's quality based fixed inputs, we obtain only one MOS value for *trace* and one MOS value*static*. The testbed measurements (tb), however, return slightly varying QoE values. In the case of *static*, the values are in a range between 1.95 and 2.04. If testbed measurements are performed using *trace*, the variability increases slightly, returning MOS values between 1.85 and 2.0. These variations in the measurements are due to several aspects, including the behavior of TCP and software. When applying the HAS performance model with distributions for available bandwidth and segment sizes, the returned MOS is between 1.8 and 2.4.

The results when using random start points in the bandwidth trace or shuffling the bandwidth trace are shown on the right side of Figure 2. While the computation using the performance model resulted in a single MOS value for *static* and *trace*, we can now see different QoE values for different computation runs, due to the induced randomness. The retrieved MOS lies between 1.84 and 2.3 for *random start* and between 1.85 and 2.0 for *shuffled*. The QoE as measured in the testbed ranges from 1.84 to 2.0 for *random start* and 1.84 and 2.1 for *shuffled*.

The cumulative distribution functions for the modes with an average available bandwidth of 1000 kbps are shown in Figure 3. Once again, we obtain one QoE value when applying the HAS performance model with the modes *static* and *trace*,

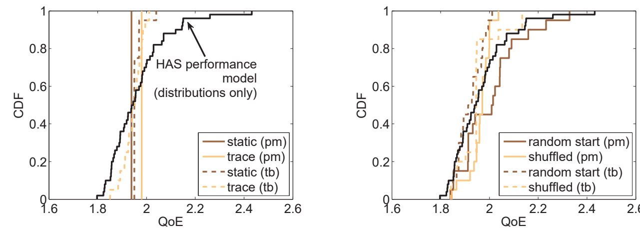

Fig. 2. CDF of the QoE values resulting from the different bandwidth modes with an average bandwidth of 500 kbps. Black solid lines indicate the HAS performance model's outcome on distributions. The remaining solid lines indicate QoE values computed by the HAS performance model (pm) relying on time series as input. Dashed lines represent the results from measurements in the testbed (tb).

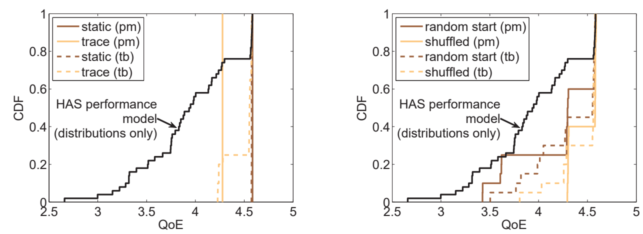

Fig. 3. CDF of the QoE values resulting from the different bandwidth modes with an average bandwidth of 1000 kbps. Black solid lines indicate the HAS performance model's outcome on distributions. The remaining solid lines indicate QoE values computed by the HAS performance model relying on time series as input. Dashed lines represent the results from measurements in the testbed.

giving a QoE of 4.58 and 4.28, respectively. The variation of the QoE values obtained from testbed measurements in *static* mode is negligible. This is due to the fact the provided bandwidth allows the client to smoothly stream on the lowest quality. It rarely selects segments on the second quality level or suffers stalling. On average, a MOS of 4.58 is achieved. If the bandwidth mode *trace* is applied, there occur periods where the available bandwidth is not sufficient to download the segment in time, but also periods where the available bandwidth is sufficient to build up a buffer that allows a higher quality. Hence, the MOS shows a slight variability between 4.2 and 4.6.

The variability of the MOS values further increases if we increase the randomness of available bandwidth by applying the modes *random start* and *shuffled*. In these cases, the values obtained from the testbed measurements and the performance model do better reflect the results from applying the performance model with distributions and generating video sequences from the resulting buffer distribution.

Finally, we show the evaluation results when considering an available bandwidth of 2000 kbps on average in Figure 4. Once again, the HAS performance model returns a single MOS value, which is 4.585 in case of *static* bandwidth and 4.584 when we use the bandwidth trace. Similar as with a static bandwidth of 1000 kbps, the testbed measurement yields

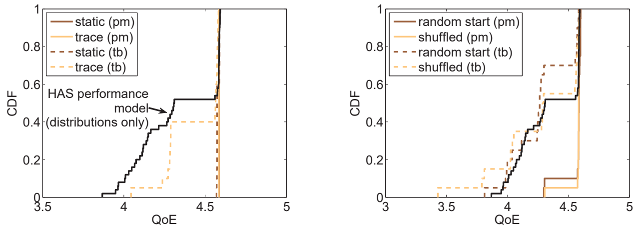

Fig. 4. CDF of the QoE values resulting from the different bandwidth modes with an average bandwidth of 2000 kbps. Black solid lines indicate the HAS performance model's outcome on distributions. The remaining solid lines indicate QoE values computed by the HAS performance model relying on time series as input. Dashed lines represent the results from measurements in the testbed.

QoE values with negligible variation. On average, a MOS of 4.58 is achieved. The testbed measurement using mode *trace* yields values that range from 4.04 to 4.58. Again, it holds that the variability of the obtained MOS is the highest for the HAS performance model with distributions as input parameters. When increasing the randomness in the 2000 kbps scenario, as shown on the right side of Figure 4, the MOS retrieved from the HAS performance model deviates from the prior static value of 4.58 in a few cases. Furthermore, the testbed measurements approximate to the results obtained from applying the HAS performance model with input distributions.

To summarize, the estimated QoE for the model-based results is slightly smaller than the results obtained by the measurements. This stems from factors like protocol header overhead or TCP behavior, which influences the measurements but is not included in the model computation. Besides, the variations tends to be higher when utilizing the model, which comes from the higher abstraction level of the model input parameters. Last, we note that the results of measurements and model remain consistent between the different investigated scenarios.

V. Conclusion

Recently, a couple of analytical models for video streaming have been proposed. They allow to understand the impact of parameters like switching thresholds or network variations on the streaming behavior and relevant application metrics like steady-state stalling or switching probabilities. Due to the abstract nature of these metrics, the computation of the video QoE using time-dependent models like P.1203 cannot be achieved in a straight forward manner. This paper investigates if and to which extent it is possible to compute the QoE based on such analytical models. To achieve this goal, we generate synthetic video playback sequences from the output of one of these models using monte carlo simulation techniques and compute the QoE using P.1203. We further conduct measurements in a local testbed using the same configurations and compute the video QoE accordingly.

For our study we considered a different network scenarios with respect to average bandwidth, its standard deviation, and its temporal correlation. The results show that the output of abstract analytical models still allows a good estimation of the QoE.

References

[1] C. V. N. Index, "The zettabyte era–trends and analysis," *Cisco white paper*, 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] 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.

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

[6] 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.

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

[8] 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.

[9] 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.

[10] "Itu-t recommendation p. 1203: Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport," ITU-T, Geneva, CH, Standard ITU-T Recommendation P. 1203, 2016. [Online]. Available: https://www.itu.int/rec/T-REC-p.1203/en

[11] 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.

[12] 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.

[13] S. Schwarzmann, P. Breitbach, and T. Zinner, "Computing qoe-relevant adaptive video streaming metrics using discrete-time analysis," in *22nd Conference on Innovation in Clouds, Internet and Networks*. IEEE, 2019, pp. 1–6.

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

[15] 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)*, 2018, preprint on webpage goo.gl/UB6XL8.

[16] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over http," in *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4. ACM, 2015, pp. 325–338.

[17] T. Hoßfeld, M. Seufert, C. Sieber, and T. Zinner, "Assessing effect sizes of influence factors towards a qoe model for http adaptive streaming," in *Quality of Multimedia Experience (QoMEX), 2014 Sixth International Workshop on*. IEEE, 2014, pp. 111–116.

[18] A. Raake, M.-N. Garcia, W. Robitza, P. List, S. Göring, and B. Feiten, "A bitstream-based, scalable video-quality model for http adaptive streaming: Itu-t p. 1203.1," in *Quality of Multimedia Experience (QoMEX), 2017 Ninth International Conference on*. IEEE, 2017, pp. 1–6.

[19] W. Robitza, S. Göring, A. Raake, D. Lindegren, G. Heikkilä, J. Gustafsson, P. List, B. Feiten, U. Wüstenhagen, M.-N. Garcia *et al.*, "Http adaptive streaming qoe estimation with itu-t rec. p. 1203: open databases and software," in *Proceedings of the 9th ACM Multimedia Systems Conference*. ACM, 2018, pp. 466–471.

[20] M. Seufert, N. Wehner, and P. Casas, "Studying the impact of has qoe factors on the standardized qoe model p. 1203," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2018, pp. 1636–1641.

[21] H. Riiser, P. Vigmostad, C. Griwodz, and P. Halvorsen, "Commute path bandwidth traces from 3g networks: analysis and applications," in *Proceedings of the 4th ACM Multimedia Systems Conference*. ACM, 2013, pp. 114–118.

[22] P. Tran-Gia, *Discrete-time analysis technique and application to usage parameter control modelling in ATM systems*. Inst. für Informatik, 1993.