Public Review for

Understanding Quality-of-Experience of Heuristic-based HTTP Adaptive Bitrate Algorithms

B. Taraghi, A. Bentalb, C. Timmerer, R. Zimmermann,
H. Hellwagner

Most online video streaming services, including major ones such as YouTube and Netflix, use adaptive streaming over HTTP, which dynamically adapts the bitrates of the sessions to maximize the quality of experience (QoE) for different users. Many algorithms have been proposed in the literature, and some deployed in practice, to perform this dynamic adaptation. It is crucial for the research community to occasionally reflect and contrast various adaptation algorithms against each other to discover opportunities for further optimizations.

The authors of this paper contribute to this important issue and offer us a deeper understanding of these adaptation algorithms. They compare seven common rate adaptation algorithms, including buffer-based (Bola, BBA0, and Quetra), throughput-based (Shaka), and Model Predictive Controller (Fast MPC), using objective and subjective studies. The objective study utilizes the ITU-T P.1203 QoE model, which calculates a predicted Mean Opinion Score (MOS) of a video based on the bitrate, frame rate, resolution, frame types, and frame sizes. The subjective study was conducted on a cloud platform with more than 800 real users (Amazon Mechanical Turks). Several real videos were each encoded into 11 representations and used in the subjective and objective studies. The authors also used four network profiles to capture some of the real-life scenarios.

The paper indeed sheds light on understanding the complex problem of quality adaptation in HTTP video streaming. The results should however be considered as the first step in this direction and should be approached with a critical eye. A workshop paper is a way to share exciting results with the community as early as possible. It is also an invitation to further expand the results. Typical future work that can be built upon this paper includes a more comprehensive evaluation, using more network profiles (maybe from actual datasets) and a more diverse catalog of videos (for example high-motion videos). Another promising extension to this paper is the evaluation of the recent learning-based rate adaptation algorithms.

Public review written by
Mohamed Hefeeda
Simon Fraser University, Canada

Understanding Quality of Experience of Heuristic-based HTTP Adaptive Bitrate Algorithms

Babak Taraghi
babak.taraghi@aau.at
Christian Doppler Laboratory ATHENA
Institute of Information Technology (ITEC)
Alpen-Adria-Universität Klagenfurt
Klagenfurt, Austria

Abdelhak Bentaleb
bentaleb@comp.nus.edu.sg
Department of Computer Science
School of Computing (SoC)
National University of Singapore
Singapore

Christian Timmerer
christian.timmerer@aau.at
Christian Doppler Laboratory ATHENA
Institute of Information Technology (ITEC)
Alpen-Adria-Universität Klagenfurt
Klagenfurt, Austria

Roger Zimmermann
rogerz@comp.nus.edu.sg
Department of Computer Science
School of Computing (SoC)
National University of Singapore
Singapore

Hermann Hellwagner
hermann.hellwagner@aau.at
Christian Doppler Laboratory ATHENA
Institute of Information Technology (ITEC)
Alpen-Adria-Universität Klagenfurt
Klagenfurt, Austria

ABSTRACT

Adaptive bitrate (ABR) algorithms play a crucial role in delivering the highest possible viewer's Quality of Experience (QoE) in HTTP Adaptive Streaming (HAS). Online video streaming service providers use HAS – the dominant video streaming technique on the Internet – to deliver the best QoE for their users. A viewer's delight relies heavily on how the ABR of a media player can adapt the stream's quality to the current network conditions. QoE for video streaming sessions has been assessed in many research projects to give better insight into the significant quality metrics such as startup delay and stall events. The ITU Telecommunication Standardization Sector (ITU-T) P.1203 quality evaluation model allows to algorithmically predict a subjective Mean Opinion Score (MOS) by considering various quality metrics. Subjective evaluation is the best assessment method for examining the end-user opinion over a video streaming session's experienced quality. We have conducted subjective evaluations with crowdsourced participants and evaluated the MOS of the sessions using the ITU-T P.1203 quality model. This paper's main contribution is to investigate the correspondence of subjective and objective evaluations for well-known heuristic-based ABRs.

CCS CONCEPTS

- **Information systems** → **Information systems applications**;
- **Multimedia information systems** → *Multimedia streaming*.

KEYWORDS

HTTP Adaptive Streaming; ABR Algorithms; Quality of Experience; Crowdsourcing; Subjective Evaluation; Objective Evaluation; MOS; ITU-T P.1203.

ACM Reference Format:

Babak Taraghi, Abdelhak Bentaleb, Christian Timmerer, Roger Zimmermann, and Hermann Hellwagner. 2021. Understanding Quality of Experience of Heuristic-based HTTP Adaptive Bitrate Algorithms. In *Network and Operating System Support for Digital Audio and Video (NOSSDAV '21)*, September 28–October 1, 2021, Istanbul, Turkey. , 7 pages. <https://doi.org/10.1145/3458306.3458875>

1 INTRODUCTION

Video traffic accounted for 66% of all mobile data traffic in 2020, a share that is forecasted to increase to 77% in 2026 [15]. Moreover, in 2020 the COVID-19 pandemic skyrocketed the number of viewers streaming video and viewing hours [9]. HTTP Adaptive Streaming (HAS) is the dominant technique for video streaming with two most commonly implemented media delivery formats, HTTP Live Streaming (HLS) [2] and Dynamic Adaptive Streaming over HTTP (MPEG-DASH) [23]. HAS-based media delivery formats partition a video file into multiple audio and video segments of the same duration, often between 2 and 10 seconds. Each segment is encoded with various representations (*i.e.*, bitrate levels and resolutions). A playlist file called *manifest* provides the information on available representations and the segments' locations to the media player. The critical decision of choosing the representation for each segment download is the primary job of an adaptive bitrate (ABR) algorithm. There are many metrics that an ABR algorithm could take into account when making such a decision, *e.g.*, buffer size, available bandwidth, network latency, or a combination thereof.

Quality of experience (QoE) is, by definition, the measure of the delight or annoyance of a customer's experiences with a service [10]. An ABR algorithm plays the pivotal role to provide the best experience to the end-user of a video streaming session by choosing a suitable media representation at the right time. Selecting a low-quality representation ideally decreases the number of stall events, but this trade-off is only acceptable if the client is suffering from a terrible network connection. In another scenario, selecting high-quality representations would deliver detailed audio and video to the client. Still, stall events have more chances to happen if the network characteristics are not predicted accurately [7]. Therefore, how delightful or annoyed an end-user of a video streaming service

This work is licensed under a Creative Commons Attribution International 4.0 License.
NOSSDAV '21, September 28–October 1, 2021, Istanbul, Turkey
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8435-3/21/09.
<https://doi.org/10.1145/3458306.3458875>

could be, is directly related to how well the ABR algorithm performs under different network conditions.

There are many studies on the topic of subjective and objective QoE evaluation [13, 16, 34]. Nevertheless, these studies were conducted with a limited number of participants for the subjective evaluations phase. Therefore the results are also narrow and do not present a clear picture of an ABR performance's impact on QoE metrics. In contrast, this paper has studied the QoE and its metrics (e.g., selected bitrate, stall events, bitrate switches, and startup delay) from two different approaches, which gives a comprehensive insight into ABR algorithms' performance. First, we have conducted 224 experiments with various test sequences and varying network conditions using seven well-known heuristic-based ABR algorithms. Second, we have conducted an extensive crowdsourced subjective quality evaluation with 835 participants over the same experiments to provide a comparative analysis of the results.

The remainder of this paper is organized as follows. Section 2 covers a detailed description of the setup we have used to conduct the objective evaluation, the architecture and design of our subjective evaluation portal, the ABR algorithms and media players we studied, and the video sequences' characteristics alongside the network profiles that we have designed to conduct the experiments. Section 3 highlights our findings and results from the evaluations. It mainly presents investigative study results to give a better understanding of the heuristic-based ABR algorithms' performance. Section 4 summarizes related work, and Section 5 concludes the paper with possible future work items.

2 EVALUATION SETUP

We have conducted both objective and subjective evaluations to study various influential QoE metrics such as *startup delay*, *video quality*, and *stall events*. This section describes video sequences' characteristics and network profiles for bandwidth shaping between the client and the server. Next, we briefly describe seven heuristic-based ABR algorithms and the media players we used to execute the assessments. Our testbed architecture will be described in detail, following our objective and subjective evaluation studies' setups.

2.1 Video Sequences

We have selected the following open source movies as used in [27] for the objective and subjective evaluations and used the FFmpeg¹ software to encode and package the DASH content following the ABR ladder in Table 1.

- (1) Sintel, the Durian Open Movie Project²
- (2) Valkaama³
- (3) Big Buck Bunny⁴
- (4) Tears of Steel⁵

The above video sequences are encoded and packaged with the following parameters: AAC for audio coding, AVC/H.264 (x264) for video coding, segment duration of 2 seconds, and group-of-pictures (GoP) length of 24 frames, MP4 segment type, and DASH format.

¹<https://ffmpeg.org>, accessed Nov. 21, 2020.

²<https://durian.blender.org>, accessed Nov. 21, 2020.

³<http://www.valkaama.com>, accessed Nov. 21, 2020.

⁴<https://peach.blender.org>, accessed Nov. 21, 2020.

⁵<https://mango.blender.org>, accessed Nov. 21, 2020.

Table 1: ABR Ladder of Video Sequences.

Representation Index	Resolution	Bitrate (kbps)
1	320x240	235
2	384x288	375
3	512x384	560
4	512x384	750
5	640x480	1050
6	720x480	1750
7	1280x720	2350
8	1280x720	3000
9	1920x1080	4300
10	1920x1080	5800
11	1920x1080	7000

We used two different parts with 2 minutes duration from each selected test sequence to extend the data set varieties. Each video was encoded into eleven representations using the encoding ABR ladder highlighted in Table 1. The choices of bitrate levels and encoding configurations were based on Netflix's recommendation [22] and Apple's recommendation [21] which is presented in [14]. As proposed in [14], we followed Streamroot's encoding configuration recommendation [4] to remove scene cuts and limit the GoP size. Two seconds segments are widely used in the development of ABRs [27, 43].

2.2 ABR Algorithms and Media Players

In the last ten years, there have been several studies in ABR algorithms – at various sophistication levels – that use multiple heuristics such as throughput prediction, buffer occupancy, or hybrid. These algorithms formulate the ABR decisions within mathematical frameworks (e.g., game theory, queuing theory) or use various learning-based techniques to perform ABR decisions. Here, we only investigate a few examples, and the interested readers are encouraged to read more details in [8]. There are seven ABR algorithms and two media players in total used for the evaluations: dash.js v3.1.3 [11] (Dynamic) and Shaka v3.0.4 [17] (Throughput-based) media players default ABR algorithms plus BBA0 [20], BOLA [36], Elastic [12], FastMPC [42], and Quetra [40]. All other ABRs have been integrated with dash.js reference player [11]. In this section, we briefly review each algorithm as follows:

- **dash.js** [11]: dash.js implements a hybrid ABR algorithm called Dynamic as default, a combination of throughput-based and BOLA algorithms. At the beginning of a streaming session, Dynamic starts by invoking throughput-based to select the bitrate based on throughput prediction. Once the current buffer level reaches 10 seconds or above, Dynamic triggers an event to switch the ABR algorithm to BOLA. It switches back again to the throughput-based algorithm when the buffer level falls below 10 seconds.
- **Shaka** [17]: Shaka implements a simple throughput-based algorithm that uses throughput heuristics with an Exponential Weighted Moving Average (EWMA) smoothing function to perform ABR decisions.
- **BBA0** [20]: It is a buffer-based ABR algorithm that uses the current buffer occupancy to select the bitrate for the next segment to be downloaded. This algorithm's key element is the buffer-bitrate map function, which indicates a discrete bijective relation from

buffer level to video bitrate. It uses a reservoir that allows selecting the lowest possible bitrate if the current buffer occupancy goes below the defined reservoir to avoid stall events. At startup, BBA0 selects the lowest video bitrate as default to speed up the rendering. When the buffer occupancy exceeds the reservoir volume, the player will switch to video segments with a higher bitrate. It tries to maintain the buffer at safe levels (between the reservoir and maximum buffer threshold).

- **BOLA** [36]: It is a buffer-based ABR algorithm that formulates ABR decisions as a utility maximization function using Lyapunov optimization.
- **Elastic** [12]: It is a hybrid ABR that uses feedback control theory that generates a long-lived Transport Control Protocol (TCP) flow for more accurate throughput predictions, thus, better ABR decisions.
- **FastMPC** [42]: It is a hybrid ABR algorithm that uses a Model Predictive Control (MPC) approach. The MPC model takes buffer occupancy observations and harmonic mean throughput predictions to select the bitrate, maximizing a given QoE metric over a horizon of five future segments.
- **Quetra** [40]: It is a buffer-based ABR that formulates ABR decisions as a queuing theory model, which calculates the expected buffer occupancy given a bitrate choice, network throughput, and buffer capacity.

2.3 Network Profiles

We have designed four network profiles to shape the total bandwidth capacity between the streaming server and the player. Our network profiles cover the common network connection scenarios. (i) *Ramp Up*, in which the available bandwidth will be increased with an interval of ten seconds starting from 372 kbps up until 7170 kbps for two minutes. (ii) *Ramp Down*, in which the available bandwidth will be decreased with an interval of ten seconds starting from 7170 kbps down to 372 kbps for two minutes. (iii) *Fluctuation*, in which the available bandwidth will be switched continuously between the upper bound (7170 kbps) and the lower bound (372 kbps) with an interval of ten seconds and over a period of two minutes. (iv) *Stable*, in which the available bandwidth will be constant and set to 3770 kbps for two minutes. Figure 1 depicts the time series of the considered network profiles. As proposed in [39], we enforce a network latency of 80 milliseconds between the client and the server to cover a broad range of application scenarios. It corresponds to what can be observed within TCP-based long-distance fixed-line connections or reasonable mobile networks.

2.4 Objective Evaluation

We are using CAdViSE⁶ [38] as our testbed to conduct the objective evaluations. CAdViSE provides a cloud-based platform to evaluate multiple ABR algorithms or media players under various network conditions. For this project, we have added new functionalities to CAdViSE that make the streaming sessions reproducible. By storing the client requests and media player events⁷ as session logs, we can reproduce the same streaming session, including the startup delay and stall events. Each streaming session that we refer to as

⁶<https://github.com/cd-athena/CAdViSE>, accessed Apr. 1, 2021.

⁷<https://html.spec.whatwg.org/multipage/media.html>, accessed Dec. 7, 2020.

Figure 1: Network Profiles Used in the Experiments.

an experiment will store its logs in a *DynamoDB*⁸ table. Once the experiment is finished for a particular ABR algorithm or media player with a specific network shape, we run a script⁹ that fetches the session logs and stitches the segments together. We are using FFmpeg to concatenate the video and audio segments. Using the same software, we can cut the exact required duration of a fake stall video and inject it between the actual video segments. When the audiovisual files are combined, we use ITU-T P.1203 *Standalone Implementation*¹⁰ [32, 33] to extract a JSON file as a feed to the P.1203 model. We then adjust the stall duration and also the defined resolution in the produced JSON file. The JSON file will be passed to the model to retrieve the Mean Opinion Score (MOS). The current implementation allows us to consider all the stall events with a duration of more than 0.001 seconds. Another step in this application is to concatenate the audiovisual files (audio and video files) using FFmpeg to have a final mp4 file ready for the subjective evaluation phase.

2.5 Subjective Evaluation

Using *Serverless Architecture*¹¹ and *AWS Lambda*¹² we have developed a HAS-based subjective evaluation portal. The portal is implemented based on the defined standard in ITU-T P.910 [25] and the best practices proposed in [18]. This subsection describes the architecture, procedures, and measurements we designed to conduct subjective evaluations. We have developed a custom web media player by leveraging the *HTML5 media element*¹³ features.

We use *Amazon Mechanical Turk*¹⁴ (MTurk) to conduct our subjective evaluation. MTurk is a crowdsourcing website to hire remotely located *crowd-workers* to perform discrete on-demand tasks. We share the link to our subjective evaluation portal on the MTurk platform. When the portal's main entry point is invoked by browsing a web address, an AWS lambda function will return the required libraries, HTML, and JavaScript files. After reading the instruction, the user enters an identity number (MTurk worker id), creating a database record. Next, the server will prepare a manifest file specific to the user. Out of 224 test sequences produced in the objective

⁸<https://docs.aws.amazon.com/dynamodb>, accessed Dec. 7, 2020.

⁹<https://github.com/cd-athena/HASClipStitcher>, accessed Mar. 30, 2021.

¹⁰<https://github.com/itu-p1203/itu-p1203>, accessed Dec. 7, 2020.

¹¹<https://www.serverless.com>, accessed Dec. 7, 2020.

¹²<https://docs.aws.amazon.com/lambda>, accessed Dec. 7, 2020.

¹³<https://w3.org/TR/2011/WD-html5-20110113/video.html>, accessed Dec. 7, 2020.

¹⁴<https://www.mturk.com>, accessed Dec. 8, 2020.

Figure 2: Avg. Stall Duration (left) and Avg. Video Startup Delay (right).

evaluation phase and uploaded to an AWS S3¹⁵ bucket, a set of 10 randomized and prioritized test sequences will be selected and locked for 30 minutes in the database and then presented in the manifest file to be consumed by the media player. The prioritization algorithm works based on total votes and the number of not expired requested locks for that specific test sequence. The custom media player parses the manifest file and starts downloading the test sequences from the AWS S3 bucket. When the first test sequence is fully downloaded to the client browser, the playback starts.

To store the test sequences within the subject's device (*i.e.*, web browser), we use *IndexedDB*¹⁶. While the first test sequence is being played, the other 9 test sequences will still be downloaded in the background without interrupting the current playback. When the playback stops and the video played successfully (for two minutes), a dialog box will pop up and ask a video content-related question, *i.e.*, a reliability question, to ensure that we only count the subject's vote if she/he has actually observed the video. Next, the user will be asked to cast a vote on a scale of 1 to 5, where 1 represents the worst experience and 5 stands for an excellent experience. The selected score will be stored in the database via another AWS lambda function by clicking the submit button. Once a subject casts his/her vote for a test sequence, that experiment id will be removed from the locked array and stored in the votes array mapped to the voted score alongside the answer to the reliability question. At the end of the evaluation session, we generate a completion code. Only those crowd-workers who can provide the completion code will be compensated. The subjective evaluation portal will ask the subjects to score 10 test sequences. Each test sequence is two minutes, therefore 20 minutes in total. Plus, 10 minutes for reading the instruction and downloading the first test sequence adds up to 30 minutes. If the users take more than 30 minutes to cast their votes, they cannot be compensated, and also, their provided score will not be counted in the final results.

3 RESULTS AND FINDINGS

This section describes our findings and results from the objective and subjective evaluations and the correlations and contradictions between these two aspects of video streaming assessment.

Objective Evaluations: We have conducted the objective evaluation with the setup described in Section 2.4. Using the ITU-T P.1203 QoE model [32], we calculate a predicted MOS of a video streaming session. The mentioned model provides integral quality

predictions in the range of 1 to 5. Out of four available modes in this QoE model, we used *Mode 1* in our evaluations, which includes *bitrate*, *framerate*, and *resolution*, plus *frame types*, and *frame sizes* in calculating the predicted MOS. Objective MOS in Figures 3, 4, 5, and 6 show the average value from *O46* attributes (representing the overall quality score) of the P.1203 model achieved by different ABRs in various network profiles. The error bars represent the 95% Confidence Intervals (CI)¹⁷ for both subjective and objective MOS. As network profiles are challenging, the results shown in the mentioned figures are not going up to the highest possible QoE score (5). The averages of stall events duration and video startup delay for each ABR algorithm per network profile are shown in Figure 2, which correlates to the objective MOS levels shown in Figures 3, 4, 5, and 6. For instance, in the fluctuation network profile, BBA0 and Elastic ABR algorithms with the highest objective MOS also have the lowest average duration for stall events. As can be observed in Figure 2, the ramp-up network profile presented in Figure 1 has the highest startup delay because the shaped network, in this case, starts from the lowest available bandwidth. In the fluctuation profile, we can see that the FastMPC, dash.js, and Shaka ABR algorithms suffer from wrong bandwidth estimation, leading to sub-optimal ABR decisions and, therefore, high stall duration and bad QoE. In contrast to buffer-based ABRs like BBA0 or Queetra, which use only the buffer level to adapt to network conditions, the Elastic ABR algorithm achieves the lowest average stall duration. It utilizes a robust bandwidth estimator that tracks the available bandwidth efficiently. As a result, Elastic takes suitable ABR decisions that maximize the QoE. We also note that objective MOS values for all the ABRs are pretty low in the fluctuation network profile as the model penalizes much for stall events.

Subjective Evaluations: The setup for subjective evaluations has been described in detail in Section 2.5. Using the MTurk platform, we were able to have 835 participants in our subjective evaluation phase. Out of which, 780 participants had at least one vote with the correct answer to the content-related reliability question and managed to cast their votes before the submission timeout. We have gathered 5723 votes in total, out of which 4704 proved reliable (according to reliability responses and other measurements described in Section 2.5), which counts 21 votes for each experiment. Figures 3, 4, 5, and 6 show the average MOS for each ABR algorithm with a different network profile. To better understand how the votes have been distributed among different ABR algorithms with varying network profiles, we put the number of votes

¹⁵<https://docs.aws.amazon.com/s3>, accessed Dec. 7, 2020.

¹⁶<https://www.w3.org/TR/IndexedDB-2>, accessed Dec. 7, 2020.

¹⁷<http://www.stat.yale.edu/Courses/1997-98/101/confint.htm>, accessed Dec. 14, 2020.

Figure 3: Avg. QoE of ABRs in Ramp Up Network Profile.

Figure 6: Avg. QoE of ABRs with Stable Network Profile.

Figure 4: Avg. QoE of ABRs in Ramp Down Network Profile.

Figure 5: Avg. QoE of ABRs in Fluctuation Network Profile.

for each ABR algorithm score in four diagrams shown in Figure 7. The network profiles' effects on how ABR algorithms make decisions about selecting the representations that directly influence the participants' perceived QoE are shown in Figure 7.

Correlations and Contradictions: Looking at Figure 7 can give a clear insight into how the ABR algorithms perform with different network profiles compared to each other. For example, Shaka media player votes are accumulated around the second score on our scale of 1 to 5, which shows weak performance for this specific media player with a fluctuation network profile. In contrast, the same media player shows more robust performance within

Figure 7: Distribution of Subjective MOS for ABRs.

the ramp-up network profile than its competitors. The mentioned scenario could also be observed in Figure 5. We have calculated the Pearson correlation coefficient¹⁸ values for each network profile between the subjective and objective results, shown in the top right corner of Figures 3, 4, 5, and 6. The higher the Pearson correlation coefficient value, the better these two sets of results are related to each other in terms of strength and direction of linear relationships. Pearson's correlation coefficient has a range of -1 to 1. In all cases, we can see a *strong correlation* with a coefficient value greater than 0.5. A discrepancy observed between the subjective and objective evaluation results can be seen in Figures 3, 4, 5, and 6 where the subjective MOS is almost always higher than the objective MOS by one score. The discrepancy gets more unexpected when we see that the correlation remains perfect even with this difference. The authors assume that this apparent difference could result from having a wide variety of participants from different geographical locations in which Internet connections differ from each other; therefore, their expectations from delightful or annoying video streaming sessions vary compared to what ITU-T P.1203 QoE model has predicted.

¹⁸<https://libguides.library.kent.edu/SPSS/PearsonCorr>, accessed Dec. 13, 2020.

4 RELATED WORK

We divide this section into two parts: QoE evaluation and ABR algorithms.

4.1 QoE Evaluation

There have been several surveys about QoE evaluation and how to conduct subjective and objective tests [1, 3, 31, 35]. Here, we list a few of these techniques. Baraković *et al.* [3] did a survey and investigation over the state-of-the-art research activities covering the field of QoE management. Their study's primary focus is on wireless networks and addressing three management aspects: QoE modeling, monitoring and measurement, and adaptation and optimization. Moreover, they provided an in-depth discussion on the key elements, and challenges researchers deal with in this area. Another comprehensive survey in QoE's area focusing on HAS techniques is presented in [35] by Seufert *et al.* in which authors' main contribution is to investigate works related to QoE from the human-computer interaction and networking domain. Subjective studies that cover the QoE aspect of adaptation dimensions and strategies are studied. Technical influence factors of HAS are identified alongside open issues and conflicting results. Mitra *et al.* [31] did an extensive study on how QoE measurements could help video streaming, Voice over IP (VoIP), and video gaming end-users in availing personalized services from service providers. They have explored the strengths and shortcomings of existing techniques in QoE assessments. Alreshoodi and Woods [1] investigated some current correlation models briefly. They have studied correlation models that attempt to map Quality of Service (QoS) to Quality of Experience (QoE) for multimedia services. Authors claim that most of the existing quality models for multimedia provide only a partial solution for predicting the actual QoE from a given QoS.

4.2 ABR Algorithms

Existing client-driven ABR algorithms [8] can be classified into four main categories based on the heuristics and mechanisms used to perform ABR decisions: (i) throughput-based, (ii) buffer-based, (iii) hybrid, or (iv) learning-based. ABRs in (i to iii) use heuristics such as throughput, buffer, and mixed, respectively, and formulate the ABR decisions as mathematical rules or approximations to maximize QoE. In contrast, ABRs in (iv) try to learn from the system environment and find suitable policies based on past player status, hence adapting to the system dynamics.

Throughput-based ABR. This class of algorithms uses throughput prediction heuristics as an input to perform ABR decisions. Li *et al.* [28] developed the Probe AND Adapt (PANDA) bitrate adaptation solution, which tries to address the bandwidth overestimation problem when multiple DASH players compete at a network bottleneck. This is achieved by probing the network during off periods to measure the fair-share bandwidth accurately. Similarly, Festive [26] tries to address the same problem as PANDA. It uses a harmonic mean algorithm that considers twenty measurement samples to predict the available bandwidth and perform ABR decisions. Mangla *et al.* [29] designed a bandwidth prediction-aware ABR scheme called CrystalBall for mobile DASH. CrystalBall aims to improve the viewer QoE while providing long-term horizon predictions and mitigating prediction errors.

Buffer-based ABR. Using only buffer occupancy heuristics to perform ABR decisions, BBA0 [20], BOLA [36], and Quetra [40] were proposed. BBA0 uses a buffer-rate map function to choose the bitrate, adapt to network resource variations, and avoid stall events. BOLA formulates the bitrate decision as a utility maximization problem based on Lyapunov theory. It derives an online control algorithm that uses only buffer occupancy. Quetra uses a queuing theory model to formulate ABR decisions to keep the buffer within a safe region.

Hybrid ABR. This ABR algorithm class uses various heuristics such as throughput, buffer, and latency in their ABR formulation model. Benefiting from game and consensus theories, Bentaleb *et al.* [5, 6] developed a game theory ABR scheme (GTA) for DASH systems. GTA uses bandwidth prediction (PANDA [28] and CS2P [37]) and buffer level to formulate ABR decisions as a bargaining and consensus mechanism, where the players form an agreement among themselves. De Cicco *et al.* [12] proposed a feedback linearization adaptive streaming controller (Elastic) that leverages feedback control theory and the Harmonic algorithm to provide a high level of fairness and bandwidth utilization. Yin *et al.* [42] developed an MPC algorithm combining available bandwidth measurements (harmonic mean [26]) and buffer occupancy to perform ABR decisions.

Learning-based ABR. This class of ABR algorithms uses one of the learning rules that consider multiple heuristics as input to learning the best bitrate to select. Mao *et al.* [30] developed a deep reinforcement learning ABR scheme, termed Pensieve, that gradually learns the best policy for ABR decisions through observations (*i.e.*, network conditions and player QoE metrics) and experience. In the same context, Yan *et al.* [41] developed Fugu, an ABR that uses supervised learning in situ, with data from real deployment environments, to train a probabilistic predictor of upcoming segment transmission times. This module then informs an MPC control policy to perform ABR decisions. Tianchi *et al.* [19] designed Stick, an ABR algorithm that combines a deep reinforcement learning technique with the traditional buffer-based approach. It aims to train a neural network, which outputs suitable buffer bounds to control a buffer-based ABR to maximize QoE.

5 CONCLUSIONS AND FUTURE WORK

In this study, we asked, "how do heuristic-based ABR algorithms perform under different network conditions?" To explore this question, we have conducted extensive crowdsourced subjective evaluations that aim to provide an understanding of how well a set of heuristics-based ABR algorithms perform with different network profiles. These evaluations then have been compared to objective assessments with the ITU-T P.1203 QoE model. We have studied how well a number of known ABR algorithms make bitrate decisions with different network profiles, ramp-up, ramp-down, fluctuating, and stable. We have examined the correlations and contradictions between the ABR algorithms' subjective and objective evaluations. As future work, we suggest a deeper investigation into the differences or contradictions shown in subjective and objective evaluations in the results and findings in Section 3. Investigating ABR algorithms' performance from subjective and objective evaluation perspectives using the recommendations from ITU-T P.1401 [24] is another suggested direction for future work.

ACKNOWLEDGMENTS

The financial support of the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology and Development, and the Christian Doppler Research Association is gratefully acknowledged. Christian Doppler Laboratory ATHENA: <https://athena.itec.aau.at/>. This research has been supported in part by the Singapore Ministry of Education Academic Research Fund Tier 2 under MOE's official grant number MOE2018-T2-1-103.

REFERENCES

- [1] Mohammed Alreshoodi and John Woods. 2013. Survey on QoE/QoS correlation models for multimedia services. *arXiv preprint arXiv:1306.0221* (2013).
- [2] Apple. 2016. HTTP Live Streaming. <https://developer.apple.com/streaming/>.
- [3] Sabina Baraković and Lea Skorin-Kapov. 2013. Survey and challenges of QoE management issues in wireless networks. *Journal of Computer Networks and Communications* 2013 (2013).
- [4] Erica Beavers. 2014. *How to Encode Multi-Bitrate Videos in MPEG-DASH for MSE Based Media Players*. Retrieved Nov 21, 2020 from <https://blog.streamroot.io/encode-multi-bitrate-videos-mpeg-dash-mse-based-media-players/>.
- [5] Abdelhak Bentaleb, Ali C Begen, Saad Harous, and Roger Zimmermann. 2018. Want to play DASH? A game theoretic approach for adaptive streaming over HTTP. In *Proceedings of the 9th ACM Multimedia Systems Conference*. 13–26.
- [6] Abdelhak Bentaleb, Ali C Begen, Saad Harous, and Roger Zimmermann. 2019. Game of Streaming Players: Is Consensus Viable or an Illusion? *ACM Transactions on Multimedia Computing, Communications, and Applications* 15, 2s (2019), 1–30.
- [7] Abdelhak Bentaleb, Ali C Begen, and Roger Zimmermann. 2016. SDNDASH: Improving QoE of HTTP adaptive streaming using software defined networking. In *Proceedings of the 24th ACM International Conference on Multimedia*. 1296–1305.
- [8] Abdelhak Bentaleb, Bayan Taani, Ali C Begen, Christian Timmerer, and Roger Zimmermann. 2018. A survey on bitrate adaptation schemes for streaming media over HTTP. *IEEE Communications Surveys & Tutorials* 21, 1 (2018), 562–585.
- [9] Bitmovin. 2020. 2020 Video Developer Report. <https://go.bitmovin.com/video-developer-report-2020>.
- [10] Kjell Brunnström, Sergio Ariel Beker, Katrien De Moor, Ann Dooms, Sebastian Egger, Marie-Neige Garcia, Tobias Hossfeld, Satu Jumisko-Pyykkö, Christian Keimel, Mohamed-Chaker Larabi, et al. 2013. Qualinet white paper on definitions of quality of experience. (2013).
- [11] DASH-IF. 2020. dash.js. [Online] Available: <https://reference.dashif.org/dash.js/>.
- [12] Luca De Cicco, Vito Caldralo, Vittorio Palmisano, and Saverio Mascolo. 2013. Elastic: a client-side controller for dynamic adaptive streaming over HTTP (DASH). In *2013 20th International Packet Video Workshop*. IEEE, 1–8.
- [13] Zhengfang Duanmu, Wentao Liu, Zhuoran Li, Diqi Chen, Zhou Wang, Yizhou Wang, and Wen Gao. 2020. Assessing the Quality-of-Experience of Adaptive Bitrate Video Streaming. *arXiv preprint arXiv:2008.08804* (2020).
- [14] Zhengfang Duanmu, Abdul Rehman, and Zhou Wang. 2018. A Quality-of-Experience Database for Adaptive Video Streaming. *IEEE Transactions on Broadcasting* 64, 2 (June 2018), 474–487.
- [15] Ericsson. 2020. Ericsson Mobility Report. <https://www.ericsson.com/4adc87/assets/local/mobility-report/documents/2020/november-2020-ericsson-mobility-report.pdf>.
- [16] Deepti Ghadiyaram, Janice Pan, and Alan C Bovik. 2017. A subjective and objective study of stalling events in mobile streaming videos. *IEEE Transactions on Circuits and Systems for Video Technology* 29, 1 (2017), 183–197.
- [17] Google. 2020. Shaka Player. [Online] Available: <https://github.com/google/shaka-player>.
- [18] Tobias Hossfeld, Christian Keimel, Matthias Hirth, Bruno Gardlo, Julian Habigt, Klaus Diepold, and Phuoc Tran-Gia. 2013. Best practices for QoE crowdsourcing: QoE assessment with crowdsourcing. *IEEE Transactions on Multimedia* 16, 2 (2013), 541–558.
- [19] Tianchi Huang, Chao Zhou, Rui-Xiao Zhang, Chenglei Wu, Xin Yao, and Lifeng Sun. 2020. Stick: A Harmonious Fusion of Buffer-based and Learning-based Approach for Adaptive Streaming. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 1967–1976.
- [20] Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell, and Mark Watson. 2014. A buffer-based approach to rate adaptation: Evidence from a large video streaming service. In *Proceedings of the 2014 ACM Conference on SIGCOMM*.
- [21] Apple Inc. 2016. *Best practices for creating and deploying HTTP live streaming media for apple devices*. Retrieved Nov 21, 2020 from https://developer.apple.com/library/content/technotes/tn2224/_index.html.
- [22] Netflix Inc. 2015. *Per-Title Encode Optimization*. Retrieved Nov 21, 2020 from <http://techblog.netflix.com/2015/12/per-title-encode-optimization.html>.
- [23] ISO/IEC. 2014. 23009-1:2014 – Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats. <https://www.iso.org/standard/65274.html>.
- [24] P ITU-T. 2020. Methods, metrics and procedures for statistical evaluation, qualification and comparison of objective quality prediction models (P.1401). <http://handle.itu.int/11.1002/1000/14159>. *ITU-T Recommendation* (2020).
- [25] P ITU-T Recommendation. 1999. Subjective video quality assessment methods for multimedia applications. *International telecommunication union* (1999).
- [26] Junchen Jiang, Vyas Sekar, and Hui Zhang. 2012. Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive. In *Proceedings of the 8th International Conference on Emerging networking experiments and technologies*. 97–108.
- [27] Stefan Lederer, Christopher Müller, and Christian Timmerer. 2012. Dynamic adaptive streaming over HTTP dataset. In *Proceedings of the 3rd multimedia systems Conference*. 89–94.
- [28] Zhi Li, Xiaoqing Zhu, Joshua Gahm, Rong Pan, Hao Hu, Ali C Begen, and David Oran. 2014. Probe and adapt: Rate adaptation for HTTP video streaming at scale. *IEEE Journal on Selected Areas in Communications* 32, 4 (2014), 719–733.
- [29] Tarun Mangla, Nawanol Theera-Ampornpunt, Mostafa Ammar, Ellen Zegura, and Saurabh Bagchi. 2016. Video through a crystal ball: Effect of bandwidth prediction quality on adaptive streaming in mobile environments. In *Proceedings of the 8th International Workshop on Mobile Video*. 1–6.
- [30] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. 2017. Neural adaptive video streaming with pensieve. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. 197–210.
- [31] Karan Mitra, Arkady Zaslavsky, and Christer Åhlund. 2014. QoE modelling, measurement and prediction: A review. *arXiv preprint arXiv:1410.6952* (2014).
- [32] Alexander Raake, Marie-Neige Garcia, Werner Robitz, Peter List, Steve Göring, and Bernhard Feiten. 2017. A bitrate-based, scalable video-quality model for HTTP adaptive streaming: ITU-T P.1203.1. In *Ninth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, Erfurt. <http://ieeexplore.ieee.org/document/7965631/>.
- [33] Werner Robitz, Steve Göring, Alexander Raake, David Lindegren, Gunnar Heikkilä, Jörgen Gustafsson, Peter List, Bernhard Feiten, Ulf Wüstenhagen, Marie-Neige Garcia, Kazuhisa Yamagishi, and Simon Broom. 2018. HTTP Adaptive Streaming QoE Estimation with ITU-T Rec. P.1203 – Open Databases and Software. In *9th ACM Multimedia Systems Conference*. Amsterdam.
- [34] Kalpana Seshadrinathan, Rajiv Soundararajan, Alan Conrad Bovik, and Lawrence K Cormack. 2010. Study of subjective and objective quality assessment of video. *IEEE Transactions on Image Processing* 19, 6 (2010), 1427–1441.
- [35] Michael Seufert, Sebastian Egger, Martin Slanina, Thomas Zinner, Tobias Hossfeld, and Phuoc Tran-Gia. 2014. A survey on quality of experience of HTTP adaptive streaming. *IEEE Communications Surveys & Tutorials* 17, 1 (2014), 469–492.
- [36] Kevin Spiteri, Rahul Urgaonkar, and Ramesh K Sitaraman. 2020. BOLA: Near-optimal bitrate adaptation for online videos. *IEEE/ACM Transactions on Networking* 28, 4 (2020), 1698–1711.
- [37] Yi Sun, Xiaoqi Yin, Junchen Jiang, Vyas Sekar, Fuyuan Lin, Nanshu Wang, Tao Liu, and Bruno Sinopoli. 2016. CS2P: Improving video bitrate selection and adaptation with data-driven throughput prediction. In *Proceedings of the 2016 ACM SIGCOMM Conference*. 272–285.
- [38] Babak Taraghi, Anatoliy Zabrovskiy, Christian Timmerer, and Hermann Hellwagner. 2020. CAdViSE: cloud-based adaptive video streaming evaluation framework for the automated testing of media players. In *Proceedings of the 11th ACM Multimedia Systems Conference*. 349–352. <https://doi.org/10.1145/3339825.3393581>.
- [39] Christian Timmerer, Matteo Maiero, and Benjamin Rainer. 2016. Which adaptation logic? An objective and subjective performance evaluation of HTTP-based adaptive media streaming systems. *arXiv preprint arXiv:1606.00341* (2016).
- [40] Praveen Kumar Yadav, Arash Shafiei, and Wei Tsang Ooi. 2017. Quetra: A queuing theory approach to DASH rate adaptation. In *Proceedings of the 25th ACM International Conference on Multimedia*. 1130–1138.
- [41] Francis Y Yan, Hudson Ayers, Chenzhi Zhu, Sadjad Fouladi, James Hong, Keyi Zhang, Philip Levis, and Keith Winstein. 2020. Learning in situ: a randomized experiment in video streaming. In *17th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI})* 20. 495–511.
- [42] Xiaoqi Yin, Abhishek Jindal, Vyas Sekar, and Bruno Sinopoli. 2015. A control-theoretic approach for dynamic adaptive video streaming over HTTP. In *Proceedings of the 2015 ACM Conference on SIGCOMM*. 325–338.
- [43] Alex Zambelli. 2009. *Smooth Streaming Technical Overview*. Retrieved Nov 21, 2020 from <https://docs.microsoft.com/en-us/iis/media/on-demand-smooth-streaming/smooth-streaming-technical-overview>.