https://doi.org/10.2352/ISSN.2470-1173.2019.10.IQSP-321

Subjective Analysis of an End-to-end Streaming System

Christos G. Bampis, Zhi Li, Ioannis Katsavounidis, Te-Yuan Huang, Chaitanya Ekanadham and Alan C. Bovik

***Abstract*—Measuring Quality of Experience (QoE) and integrating these measurements into video streaming algorithms is a multi-faceted problem that fundamentally requires the design of comprehensive subjective QoE databases and metrics. To achieve this goal, we have recently designed the LIVE-NFLX-II database, a highly-realistic database which contains subjective QoE responses to various design dimensions, such as bitrate adaptation algorithms, network conditions and video content. Our database builds on recent advancements in content-adaptive encoding and incorporates actual network traces to capture realistic network variations on the client device. Using our database, we study the effects of multiple streaming dimensions on user experience and evaluate video quality and quality of experience models. We believe that the tools introduced here will help inspire further progress on the development of perceptually-optimized client adaptation and video streaming strategies.**

I. Introduction

HTTP-based adaptive video streaming (HAS) is commonly deployed in modern video streaming services, such as Netflix and YouTube. The main idea behind HAS is to encode multiple video representations with various bitrate and quality levels, and to allow client-driven stream adaptation. As a result, the client device decides the bitrate/quality level of the video chunk to be played next. The client's stream adaptation is usually based on network throughput measurements and the device buffer status [1].

In adaptive video streaming, there are two main types of video distortion: compression/scaling artifacts and rebuffering [2]. When the available bandwidth drops, a client may use a higher compression ratio and/or a lower encoding resolution to reduce the video bitrate, leading to compression and/or scaling artifacts [2], which do not interrupt video playback. In the extreme case where the available throughput drops below a certain value and the device buffer is emptied, the client device must pause the video playback (video rebuffering) and wait until the buffer is filled up with some video data before the playback resumes.

In this work, we focus on how video quality changes and rebuffering affect user Quality of Experience i.e., the overall level of user satisfaction [3] while viewing streaming content. By predicting QoE, we can design better algorithms to optimize streaming QoE while effectively utilizing the available bandwidth. We generate a diverse set of of video streams using different video contents, encoded with different quality levels and streamed using different client bitrate adaptation (ABR) algorithms under various network conditions. Then, we conduct a comprehensive human study to better understand subjective streaming QoE. The ultimate goal is to use the collected data to predict QoE and use these predictions in a feedback manner, to design better encoding and streaming adaptation algorithms.

The unique characteristics of the new database are that we reply on state-of-the-art development in large-scale video encoding and streaming. To create compressed videos, we use the Dynamic Optimizer (DO) [4], an optimization framework that determines the optimal (under certain assumptions) encoding resolution and quantization parameter on a per-shot basis, guided by a perceptual video quality assessment algorithm (VMAF) [2], [4]. On the streaming end, we use actual network measurements and a realistic buffer simulator. Given the diversity of ABR algorithms and network traces, the collected data in the database is able to capture multiple streaming adaptation aspects, such as video quality fluctuations, rebuffering events of varying durations, and content types. Notably, the database is considerably larger than various other public-domain video QoE databases [5]–[7].

Our findings can be summarized as follows. A better bandwidth prediction model can improve most objective streaming metrics, such as the playout bitrate and the number and duration of rebuffers. Start-up is the most challenging part of a session for all ABR algorithms, since ABR algorithms have not built up the video buffer and hence network variations can easily reduce QoE. While this is in line with previous studies [1], we take a step further. Our continuous data analysis shows that humans perceive these differences during start-up, even if they are forgiving and/or forgetful when an overall QoE score is recorded. These observations demonstrate the importance of temporal studies of QoE, especially during start-up, on practical video streaming applications.

II. Related Work

Many databases have been designed towards advancing progress on the general problem of video quality, e.g., [8]–[11] and streaming QoE [5], [6], [12]–[15]. The most important limitation of previous works is that they usually do not always consider continuous-time QoE together with the interplay between rebuffering events and time-varying video quality. Meanwhile, a common approach that most previous efforts have taken is to systematically control and simulate network conditions, e.g., as suddenly decreasing or gradually increasing bandwidth patterns. By varying the position and the length of these events, it is indeed possible to recreate intuitive network patterns. However, real network conditions

C. G. Bampis, Z. Li, TY. Huang and C. Ekanadham are with Netflix Inc. I. Katsavounidis is with Facebook Inc. A. C. Bovik is with the University of Texas (UT) at Austin. The majority of this work was carried out while the first author was with UT Austin.

are far more complex, and hence challenge ABR algorithms to a greater extent. In this work, we have undertaken a more realistic approach, where real network traces have been used to drive the database generation.

Further, in previous studies, fixed bitrate ladders were commonly used, without considering content-aware encoding strategies which are gaining popularity. The main advantage of a content-aware ladder is that not all video contents need the same amount of bits to be encoded; some scenes are easier to encode than others. Therefore, a content-aware ladder achieves bitrate savings for streaming providers and better video quality (or more video data) for consumers.

III. Recreating A Comprehensive End-user Experience

A. Overview of the Streaming System

We designed a new and unique QoE database, whereby perceptual video quality principles are injected into various stages of a modern streaming system: encoding, quality monitoring and client adaptation. To overcome the limitations of previous QoE studies, we built our database using a highly realistic adaptive streaming pipeline model, which comprises four main modules: an encoding module, a video quality module, a network transmission module and a client-based video playout module.

The encoding module constructs a content-driven bitrate ladder which is then fed into the Dynamic Optimizer (DO) [4]: a state-of-the-art encoding optimization approach, which determines the encoding parameters (encoding resolution and Quantization Parameter - QP) to produce compressed videos of optimized quality. The video quality module performs VMAF [2] quality measurements that drive the encoding and client modules. We used the latest VMAF (version 0.6.1), which was trained as described in [16]. The VMAF quality measurements are stored in a chunk map and made available on the client side for client bitrate adaptation. A chunk map contains information (e.g. bitrate and quality in terms of VMAF) about every encoded video segment. The network module incorporates the selected network traces and is responsible for communication between the encoding, video quality and client modules. The client module is responsible for requesting the next chunk to be played.

This streaming model allowed us to recreate a comprehensive end-user experience by focusing on three streaming dimensions: encoding, network throughput and the choice of ABR algorithm. To study each of these dimensions, we incorporated 15 video contents, 7 actual network traces and 4 adaptation algorithms, yielding 420 video streams. Next, we explore the diverse characteristics of each dimension with the overarching goal of recreating a comprehensive end-user experience.

B. Video Contents

To design a diverse encoding space, we considered multiple video contents and encoded them at multiple bitrate values (bitrate ladder). We collected 15 video contents, which span a diverse set of content genres, including action, documentary, sports, animation and video games. The video sequences also contain computer-generated content, such as Blender [17] animation and video games. The videos were shot/rendered under different lighting conditions ranging from bright scenes (Skateboarding) to darker ones (Chimera1102353). There were different types of camera motion, including static (e.g. Asian Fusion and Meridian Conversation) and complex scenes taken with a moving camera, with panning and zooming (e.g. Soccer and Skateboarding). Contents having source resolutions larger than 1920x1080 and/or frame rates larger than 30 fps were downsampled to 1920x1080 and/or 30 fps. We summarize some of the content characteristics in Table I.

C. Video Encoding

A comprehensive encoding space design requires a wide range of encoding bitrates and video quality levels. To this end, we derived a target bitrate ladder, i.e., a set of possible bitrate values, one for each content, using VMAF [2] to generate equally spaced (in terms of VMAF) bitrate points, then fed these bitrate points to DO [4]. The generated encoding bitrates range from about 150 kbps up to almost 6 Mbps. The low bitrate range, i.e., 150 kbps to 1 Mbps is sampled more heavily, which aligns well with our raised interest for challenging network conditions. It should be noted that the encoding bitrate ladder design is orthogonal to the actual network conditions, since the network conditions are not known *a priori*.

TABLE I: Content characteristics of the video contents in LIVE-NFLX-II.

Video Source	ID	Description	Motion
AirShow	AS	blue sky, saliency	medium
AsianFusion	AF	uniform background, zoom-in	low
Chimera1102353	CD	dark background, saliency	medium
Chimera1102347	CF	multiple faces, zoom-in	low
CosmosLaundromat	CL	blender, saliency	low
ElFuenteDance	ED	rich spatial activity, faces	medium
ElFuenteMask	EM	medium spatial activity, saliency	medium
GTA	GTA	gaming content	high
MeridianConversation	MC	low-light, human face	low
MeridianDriving	MD	zoom-in, face close-up	low
SkateBoarding	SB	complex camera, saliency	high
Soccer	SO	rich spatial activity	high
Sparks	SP	face, fire sparks, water	low
TearsOfSteelRobot	TR	multiple objects	high
TearsOfSteelStatic	TS	human close up	low

As already mentioned, the DO framework selects the encoding resolution and QP for each shot, such that the overall quality (as measured by VMAF) is maximized for a given target bitrate. In our implementation of the DO, we used 6 encoding resolutions: 384x216, 480x270, 640x360, 960x540, 1280x720, 1920x1080 and 10 QP values: starting from 43 (worst quality) to 16 (best quality), in steps of 3. However, for display purposes, all compressed videos were upsampled to 1920x1080 to match the display device resolution.

D. Network Simulation

Up to this point, we have only considered the first dimension in the video streaming design space - encoding. Importantly, the number of available bits is not constant in a video streaming session and network resources can vary significantly. To capture the effects of network variability, we manually selected 7 network traces from the HSDPA dataset [18], [19], which contains actual 3G traces collected from multiple travel routes

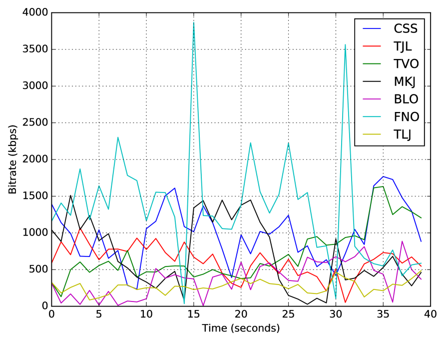


Fig. 1: Network traces used in our streaming pipeline model.

in Norway, using various means of transportation, including car, tram and train, together with different network conditions. This dataset has been widely used to compare adaptation algorithms [20] and is suitable for modeling challenging, low-bandwidth network conditions.

As shown in Fig. 1, the selected traces are approximately 40 seconds long and have varying network behaviors. For example, the TLJ trace has the lowest average bandwidth but does not vary much over time, while the MKJ trace has a much more volatile behavior than TLJ. The network traces densely cover download speeds up to 1Mbps, and there are also samples falling within the 1Mbps-3Mbps range.

E. Client ABR Algorithm

In client-based video streaming, the client is responsible for requesting the next chunk to be played. To decide the appropriate quality representation, the client module is aware of its buffer status, and may estimate future bandwidth (based on past client measurements). The client may also have information regarding the bitrate/quality levels for each video segment. In practice, this can be implemented as part of the manifest exchange between server and client.

The design space of adaptation algorithms is very large [1], [20]–[25], and hence we selected four representative adaptation algorithms. Each focuses on different design aspects, such as preserving buffer status, maximizing download bitrate, or mediating between chunk quality and buffer level.

We implemented the buffer-based (BB) approach from [1], which decides the rate of the next chunk to be played, as a function of the current buffer occupancy. We included this algorithm because it is simple to implement and is commonly evaluated or cited in the ABR literature. A reservoir of $r = 5$ sec. and a cushion of $c = 4.5$ sec. was used. We manually selected these parameters to achieve satisfying performance on a set of tests that we carried out offline. The advantage of the BB approach is that it can reduce the amount of rebuffering by only accessing buffer occupancy.

Viewing adaptation from a different perspective, we also implemented a rate-based (RB) approach which selects the maximum possible bitrate such that, based on the estimated throughput, the downloaded chunk will not deplete the buffer. To estimate future throughput, an average of $w = 5$ past chunks is computed. Selecting w can affect adaptation performance, if the network varies significantly. A low value of w could be insufficient to produce a reliable bandwidth estimate, while a large w might include redundant past samples and have diminishing impact. Another downside of the RB approach is that, when channel bandwidth varies significantly, it may lead to excessive rebuffering and aggressive bitrate/quality switching.

Using video bitrate as a proxy for quality may yield sub-optimal results; a complex shot (rich in spatial textures or motion) requires more bits to be encoded at the same quality compared to a static shot having a uniform background and low motion. Therefore, it is interesting to explore how well a quality-based (QB) adaptation algorithm will correlate against subjective scores. We relied on the dynamic programming consistent-quality adaptation algorithm presented in [26]. We use VMAF measurements (using the video quality module) as a utility function to be maximized within a finite horizon h (in sec.). This was formulated as a dynamic programming (DP) problem solved at each step, which determines the chunk to be played next.

In our QB implementation, the network conditions are estimated similar to our RB implementation. We assume that future throughput (within the horizon h) will be equal to the average throughput over the past $w = 5$ chunks. However, different from RB, QB maximizes visual quality in terms of VMAF, instead of video bitrate. For the QB client, two practical limitations on the buffer size are imposed. To reduce the risk of rebuffering, the QB solution requires that the buffer is never drained below a lower bound B_l (in sec.). Also, due to physical memory limitations, QB never fills the buffer above a threshold B_h. To ensure that the B_l and B_h constraints are satisfied, the QB solution is set to converge to a target buffer $B_t \in (B_l, B_h)$ by imposing in its DP formulation that the buffer at the end of the time horizon has to be equal to B_t. Notably, if the dynamic programming solution fails (when B_l cannot be achieved or B_h is surpassed), the QB algorithm uses a "fallback" mode: if B_l cannot be achieved, then QB selects the lowest quality stream, while if B_h is surpassed, then QB pauses downloading until the buffer frees up and then downloads the highest available stream.

It is impossible for any adaptation strategy to have perfect knowledge of future network conditions. In practice, probabilistic network modeling, or other much simpler estimation techniques can be exploited. For the latter, many adaptation algorithms assume that network conditions are constant over short time scales, and apply filtering using previous network measurements, as in QB. Since accurate knowledge of future bandwidth places an upper bound on the performance of an algorithm, we also included a version of QB which uses the actual network traces, instead of throughput estimates, thereby acting as an "oracle" (OQB).

To demonstrate the diversity of ABR algorithms, Fig. 2 shows the average bitrate (in kbps) and rebuffering time for the 4 adaptation algorithms. We observed bitrate values in the range of 535 to 660 kbps and average rebuffering times from 0.8 to 1.35 seconds. Notably, and since in the start-up phase (first 5-10 seconds) of every stream the video buffer is not filled up yet, we found that most rebuffering events occurred within the first 5-10 seconds of the video stream.

Given the comprehensive nature of the encoding, network

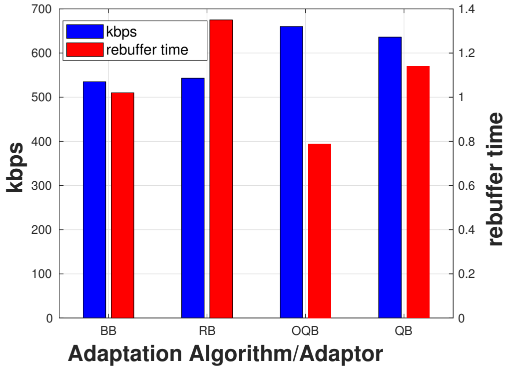


Fig. 2: Average bitrate (in kbps) and rebuffering time (in s.) for the 4 adaptation algorithms in LIVE-NFLX-II.

conditions and ABR designs, we are able to create a rich streaming QoE database by conducting a large subjective test on human perception. In the next Section, we describe the specifics of this test, which led to the creation of the LIVE-NFLX-II database.

IV. Subjective Test on the Recreated Experience

We conducted a single-stimulus continuous quality evaluation study [27] over a period of four weeks at The University of Texas at Austin's LIVE subjective testing lab. We collected overall and continuous-time QoE scores on a 1080p computer monitor from a total of 65 subjects (50 male and 15 female, ages 18-30). Overall QoE scores reflect the final QoE after viewing each video sequence in its entirety, while continuous scores capture the time-varying nature of QoE due to quality changes and rebuffering. To avoid user fatigue, the study was divided into three separate 30-minute viewing sessions of 50 videos each (150 videos per subject). To design the experimental interface, we relied on Psychopy [28] and collected opinion scores (overall and continuous) in the range of [1, 100].

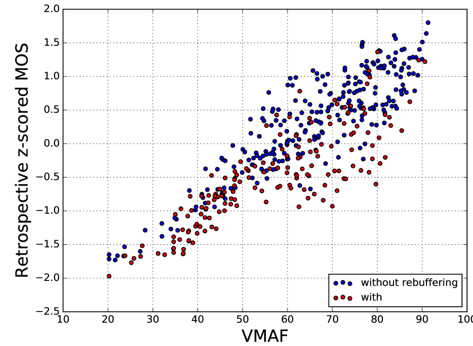
The final database consists of 420 distorted videos with an average of 23.2 scores (overall and continuous) for every distorted video. No video was viewed by less than 22 subjects, ensuring a sufficient number of scores per video. Overall, we gathered 65*150 = 9750 overall scores and 9750 continuous-time waveforms to study subjective QoE. We applied z-score normalization on the overall and continuous-time QoE scores to account for subjective differences when using the rating scale. No subjects had to be rejected for either the final or continuous scores.

V. Subjective Analysis

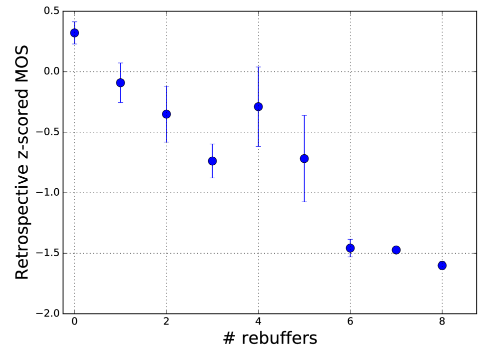
In streaming applications, human opinion scores serve as the ground truth when analyzing streaming video impairments and when evaluating objective models of video quality and QoE prediction. Here we analyze the video database by means of the collected overall and continuous-time subjective scores.

A. Analysis Using Overall Scores

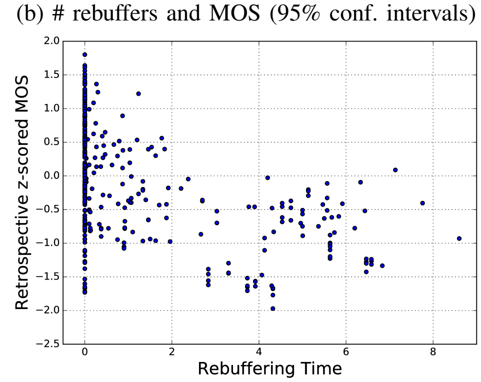
To identify the main QoE factors, Fig. 3 highlights the relationships between overall scores and average VMAF values (calculated on non-rebuffered frames), and the number and duration of rebuffering events respectively. Unsurprisingly, the presence of rebuffering (red points) negatively impacts the



(a) VMAF and MOS



(b) # rebuffers and MOS (95% conf. intervals)



(c) Rebuffer duration (in s.) and MOS

Fig. 3: VMAF measurements, number and duration of rebuffer events against overall QoE scores in LIVE-NFLX-II. Around 40% of the video sequences have at least one rebuffering event.

overall correlation of VMAF with subjective opinion scores, since VMAF does not account for the effects of rebuffering on user experience. Naturally, a larger number of rebuffering events tends to decrease user experience.

As an exception, the points with 3, 4 and 5 rebuffering events are not in decreasing MOS order. We found that the corresponding average rebuffering durations were 4.33, 3.49 and 2.93 sec. respectively, meaning that larger rebuffering occurrence did not necessarily imply larger rebuffering duration. Therefore, Fig. 3b demonstrates that subjects are sensitive to a combined effect of rebuffering occurrence and duration.

In Fig. 3c, we observe that a longer rebuffering time also lowers QoE, but when the rebuffering time is more than 4 seconds, *duration neglect effects* [29] may reduce this effect. According to the duration neglect phenomenon, subjects may recall the duration of an impairment, but they tend to be insensitive to its duration (after a certain cutoff) when making overall QoE evaluations.

We compared the overall QoE scores among different adaptors (Fig. 4). We observed that the opinion scores are not very different across adaptors. This may be due to the fact that most of the rebuffering events occurred early in the video playout, and because, just before the video finishes playing (and the overall score is recorded), the adaptation algorithms have built-up enough buffer to better handle bitrate/quality variations, even if the network is varying significantly. Therefore, it is likely that recency effects [14], [29] led to biases in the overall QoE evaluations, i.e., subjects are forgiving/forgetful when recording overall QoE.

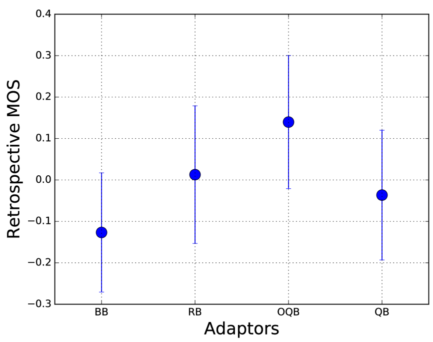


Fig. 4: Overall QoE score distribution for different adaptation algorithms (averaged across traces and contents). The error bars indicate the 95% confidence interval.

To validate this recency phenomenon, we averaged the continuous subjective scores over one second windows and calculated the correlation scores with the final subjective scores, as in [14]. For example, we found that the average continuous scores calculated over the $[4, 5]$ second window correlated weakly with the overall QoE scores (correlation of 0.58). However, by averaging the continuous scores over the $[24, 25]$ second window (20 seconds later), the correlation increased to 0.94. Notably the overall scores were similar across adaptors.

B. Analysis Using Continuous Scores

Figure 5 depicts the continuous-time user experience across adaptation algorithms. We found that, within the first few seconds, the RB aggressive rate strategy initially leads to better QoE, unlike BB, QB and OQB, which opt for buffer build-up. This also means that subjects preferred increased early rebuffering, if it meant better start-up quality, as in the case of RB. Within the first 12 seconds, BB is overly conservative and delivers the lowest QoE among all adaptors, while QB and OQB perform between RB and BB. Nevertheless, after 12 seconds, QB and OQB improve considerably, with OQB tending to produce higher scores for the rest of the session. BB is relatively lower than RB and QB, both of which are statistically close. As before, we note that, after 25 seconds, QoE measurements are decreasing and have larger confidence intervals, since they correspond to videos that rebuffered, and their count decreases over time.

Notably, as in Fig. 4, we found that OQB is not statistically better than QB, even though it has perfect knowledge of the future bandwidth and performs the best in terms of objective metrics. As already explained, for the majority of distorted videos, rebuffering and quality degradations occurred earlier during video playout and this led to smaller differences in the subjective opinions per adaptor and over time. This experimental result does not suggest that better bandwidth prediction is not an important goal, but it does show that better bandwidth prediction does not significantly influence overall QoE scores. Meanwhile, the significant differences in QoE between adaptation strategies in the start-up phase underlines that temporal studies of QoE are highly relevant for adaptive video streaming, given that ABR algorithms are especially challenged during start-up.

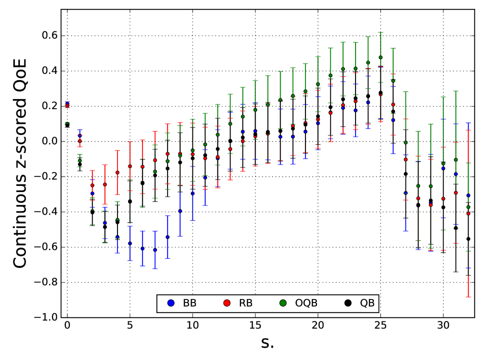


Fig. 5: Continuous-time scores for different adaptation algorithms (averaged across traces and contents). Error bars indicate 95% confidence intervals.

Viewed from the network condition perspective, we found that continuous-time subjective scores are affected by dynamic video quality changes and rebuffering. Figure 6 shows that, for all traces, a few seconds are needed to build up video buffer and hence continuous scores are relatively low. Under better network conditions (e.g. FNO), user experience steadily improves after some time, due to the adaptors switching to higher resolution and lower compression ratio. By contrast, challenging cases such as BLO and TLJ recover slowly or do not recover at all, while very volatile conditions, as in MKJ, may also lead to noticeable drops in QoE much later during video playout.

C. Adaptation Algorithm Performance Discussion

Following our earlier between-adaptor analysis, it is natural to ask which adaptation algorithm performs the best. In terms of overall scores, we were not able to make statistically significant comparisons, in part due to the effects of recency. On a similar note, using continuous-time scores, we found that OQB performed marginally better for a period of time, but the differences were not statistically significant even though OQB has perfect knowledge of the future bandwidth. By contrast, BB was overly conservative during startup and did not select high quality streams.

Comparing RB and QB, we found that they delivered similar QoE over time, except during the start-up phase, where RB

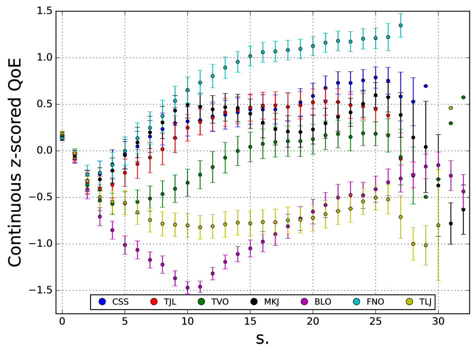


Fig. 6: Continuous-time scores for different network conditions. Error bars indicate 95% confidence intervals.

picked higher quality levels. The similar behavior between QB and RB can be attributed to their inherent properties: RB leads to excessive rebuffering, while QB reduces rebuffering (by taking into account the buffer level in its optimization scheme), but leads to many quality switches. In fact, an important consideration when designing QB is selection of the minimum buffer B_l and target buffer B_t values. When the network changes rapidly, the adaptor may not satisfy these and use its fallback mode, which leads to such large quality switches.

VI. Conclusions

We presented the design of a large, comprehensive subjective video database, which relied on a highly realistic streaming system. The collected data allowed us to analyze overall and continuous-time user experiences under different network conditions, using different adaptation algorithms, and on diverse video contents. We found that start-up is a challenging phase for ABR algorithms, since the video buffer is not sufficient to withstand large network variations. However, human responses were forgetful of negative QoE events during start-up, which underlines the need to better understand continuous streaming QoE. In the future, the collected data can be used to develop better continuous-time QoE predictors that can be injected into the client-adaptation strategy to perceptually optimize video streaming.

VII. Acknowledgement

The authors thank Anush K. Moorthy for discussions regarding content encoding complexity and the content-adaptive bitrate ladder. Also, Anne Aaron and the entire Video Algorithms team at Netflix for supporting this work.

References

[1] 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,” *Computer Communication Review*, vol. 44, no. 4, pp. 187–198, 2015.

[2] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara, “Toward A Practical Perceptual Video Quality Metric.” http://techblog.netflix.com/2016/06/toward-practical-perceptual-video.html.

[3] K. Brunnström, S. A. Beker, K. De Moor, A. Dooms, S. Egger, M.-N. Garcia, T. Hossfeld, S. Jumisko-Pyykkö, C. Keimel, M.-C. Larabi *et al.*, “Qualinet white paper on definitions of quality of experience,” 2013.

[4] I. Katsavounidis, “Dynamic optimizer-a perceptual video encoding optimization framework.” https://medium.com/netflix-techblog/dynamic-optimizer-a-perceptual-video-encoding-optimization-framework-e19f1e3a277f.

[5] C. Chen, L. K. Choi, G. de Veciana, C. Caramanis, R. W. Heath, and A. C. Bovik, “Modeling the time-varying subjective quality of HTTP video streams with rate adaptations,” *IEEE Trans. on Image Proc.*, vol. 23, no. 5, pp. 2206–2221, 2014.

[6] Z. Duanmu, Z. Kai, K. Ma, A. Rehman, and Z. Wang, “A quality-of-experience index for streaming video,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 1, pp. 154–166, 2016.

[7] D. Ghadiyaram, J. Pan, and A. C. Bovik, “A subjective and objective study of stalling events in mobile streaming videos,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[8] V. Hosu, F. Hahn, M. Jenadeleh, H. Lin, H. Men, T. Szirányi, S. Li, and D. Saupe, “The Konstanz natural video database (KoNViD-1k),” in *International Conference on Quality of Multimedia Experience*. IEEE, 2017, pp. 1–6.

[9] S. Winkler, “Analysis of public image and video databases for quality assessment,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 6, pp. 616–625, 2012.

[10] M.-N. Garcia, F. De Simone, S. Tavakoli, N. Staelens, S. Egger, K. Brunnström, and A. Raake, “Quality of experience and HTTP adaptive streaming: A review of subjective studies,” *International Workshop on Quality of Multimedia Experience*, pp. 141–146, 2014.

[11] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hobfeld, and P. Tran-Gia, “A survey on quality of experience of HTTP adaptive streaming,” *IEEE Commun. Surv. & Tutor.*, vol. 17, no. 1, pp. 469–492, 2015.

[12] N. Staelens, J. D. Meulenaere, M. Claeys, G. V. Wallendael, W. V. den Broeck, J. D. Cock, R. V. de Walle, P. Demeester, and F. D. Turck, “Subjective quality assessment of longer duration video sequences delivered over http adaptive streaming to tablet devices,” *IEEE Transactions on Broadcasting*, vol. 60, no. 4, pp. 707–714, Dec 2014.

[13] S. Tavakoli, S. Egger, M. Seufert, R. Schatz, K. Brunnström, and N. García, “Perceptual quality of HTTP adaptive streaming strategies: Cross-experimental analysis of multi-laboratory and crowdsourced subjective studies,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 8, pp. 2141–2153, 2016.

[14] C. G. Bampis, Z. Li, A. K. Moorthy, I. Katsavounidis, A. Aaron, and A. C. Bovik, “Study of Temporal Effects on Subjective Video Quality of Experience,” *IEEE Trans. on Image Process.*, vol. 26, no. 11, pp. 5217–5231, 2017.

[15] J. Søgaard, M. Shahid, J. Pokhrel, and K. Brunnström, “On subjective quality assessment of adaptive video streaming via crowdsourcing and laboratory based experiments,” *Multimedia tools and applications*, vol. 76, no. 15, pp. 16 727–16 748, 2017.

[16] Z. Li, C. G. Bampis, J. Novak, A. Aaron, K. Swanson, A. Moorthy, and J. D. Cock, “VMAF: The Journey Continues.” https://medium.com/netflix-techblog/vmaf-the-journey-continues-44b51ee9ed12.

[17] https://www.blender.org.

[18] H. Riiser, P. Vigmostad, C. Griwodz, and P. Halvorsen, “Commute path bandwidth traces from 3G networks: analysis and applications,” *ACM Multimedia Systems Conference*, pp. 114–118, 2013.

[19] http://skuld.cs.umass.edu/traces/mmsys/2013/pathbandwidth/.

[20] H. Mao, R. Netravali, and M. Alizadeh, “Neural Adaptive Video Streaming with Pensieve,” *Proceedings of SIGCOMM*, pp. 197–210, 2017.

[21] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, “BOLA: Near-optimal bitrate adaptation for online videos,” *IEEE International Conference on Computer Communications*, pp. 1–9, 2016.

[22] A. Beben, P. Wiśniewski, J. M. Batalla, and P. Krawiec, “ABMA+: lightweight and efficient algorithm for HTTP adaptive streaming,” *International Conference on Multimedia Systems*, 2016.

[23] 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.

[24] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, “A control-theoretic approach for dynamic adaptive video streaming over HTTP,” *Computer Communication Review*, vol. 45, no. 4, pp. 325–338, 2015.

[25] J. Kua, G. Armitage, and P. Branch, “A survey of rate adaptation techniques for dynamic adaptive streaming over HTTP,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1842–1866, 2017.

[26] Z. Li, A. C. Begen, J. Gahm, Y. Shan, B. Osler, and D. Oran, “Streaming video over HTTP with consistent quality,” *ACM multimedia systems conference*, pp. 248–258, 2014.

[27] *BT-500-13: Methodology for the Subjective Assessment of the Quality of Television Pictures*, Int. Telecommunication Union Std.

[28] J. W. Peirce, “Psychopy-psychophysics software in python,” *Journal of neuroscience methods*, vol. 162, no. 1, pp. 8–13, 2007.

[29] D. S. Hands and S. Avons, "Recency and duration neglect in subjective assessment of television picture quality," *Applied cognitive psychology*, vol. 15, no. 6, pp. 639–657, 2001.