DashReStreamer: Framework for Creation of Impaired Video Clips under Realistic Network Conditions

KERIM HODŽIĆ, Faculty of Electrical Engineering, University of Sarajevo, BiH
MIRSAD COSOVIC, Faculty of Electrical Engineering, University of Sarajevo, BiH
SASA MRDOVIC, Faculty of Electrical Engineering, University of Sarajevo, BiH
JASON J. QUINLAN, School of Computer Science & Information Technology, University College Cork, Ireland
DARIJO RACA, Faculty of Electrical Engineering, University of Sarajevo, BiH

The continuous rise of multimedia entertainment has led to an increased demand for delivering outstanding user experience of multimedia content. However, modelling user-perceived Quality of Experience (QoE) is a challenging task, resulting in efforts for better understanding and measurement of user-perceived QoE. To evaluate user QoE, subjective quality assessment, where people watch and grade videos, and objective quality assessment in which videos are graded using one or many objective metrics are conducted. While there is a plethora of video databases available for subjective and objective video quality assessment, these videos are artificially infused with various temporal and spatial impairments. Videos being assessed are artificially distorted with startup delay, bitrate changes, and stalls due to rebuffering events. To conduct a more credible quality assessment, a reproduction of original user experiences while watching different types of streams on different types and quality of networks is needed. To aid current efforts in bridging the gap between the mapping of objective video QoE metrics to user experience, we developed DashReStreamer, an open-source framework for re-creating adaptively streamed video in real networks. The framework takes inputs in the form of video logs captured by the client in a non-regulated setting, along with an .mpd file or a YouTube URL. The ultimate result is a video sequence that encompasses all the data extracted from the video log. DashReStreamer also calculates popular video quality metrics like PSNR, SSIM, MS-SSIM and VMAF. Finally, DashReStreamer allows creating impaired video sequences from the popular streaming platform, YouTube. As a demonstration of framework usage we created a database of 332 realistic video clips, based on video logs collected from real mobile and wireless networks. Every video clip is supplemented with bandwidth trace and video logs used in its creation and also with objective metrics calculation reports. In addition to dataset, we performed subjective evaluation of video content, assessing its effect on overall user QoE. We believe that this dataset and framework will allow the research community to better understand the impacts of video QoE dynamics.

CCS Concepts: • **Information systems → Multimedia streaming**; • **Networks → Public Internet**; **Wireless access networks**.

Additional Key Words and Phrases: QoE, Dataset, Mobility, throughput, context information, adaptive video streaming, 3G, 4G, WiFi

ACM Reference Format:
Kerim Hodžić, Mirsad Cosovic, Sasa Mrdovic, Jason J. Quinlan, and Darijo Raca. 2018. DashReStreamer: Framework for Creation of Impaired Video Clips under Realistic Network Conditions. 1, 1 (January 2018), 25 pages. https://doi.org/XXXXXXX.XXXXXXX

Authors' addresses: Kerim Hodžić, kerim.hodzic@etf.unsa.ba, Faculty of Electrical Engineering, University of Sarajevo, Zmaja od Bosne bb, Sarajevo, BiH; Mirsad Cosovic, mcosovic@etf.unsa.ba, Faculty of Electrical Engineering, University of Sarajevo, Zmaja od Bosne bb, Sarajevo, BiH; Sasa Mrdovic, Faculty of Electrical Engineering, University of Sarajevo, Zmaja od Bosne bb, Sarajevo, BiH, smrdovic@etf.unsa.ba; Jason J. Quinlan, School of Computer Science & Information Technology, University College Cork, Ireland, j.quinlan@cs.ucc.ie; Darijo Raca, Faculty of Electrical Engineering, University of Sarajevo, Zmaja od Bosne bb, Sarajevo, BiH, draca@etf.unsa.ba.

1 INTRODUCTION

Multimedia entertainment represents the dominant type of traffic carried on today's networks. Video streaming dominates the Internet, accounting for almost 66% of all Internet traffic in 2022 [65]. Streamed content can vary from live events, (e.g., big sports events, video games or cultural events) and on-demand content (e.g., movies and TV shows), with applications such as YouTube, Netflix, Amazon Prime, Disney+, Tik Tok and Apple+ dominating overall traffic share [65].

The main streaming approach for content delivery is the HTTP adaptive streaming (HAS) technique. HAS allows seamless content quality adaptation to the varying network conditions by splitting the video content into multiple fixed-duration segments. Each segment is encoded in multiple qualities (e.g., 200, 500, 1000, 4000 kbps). At the client side, the player stores downloaded segments in the playback buffer for decoding. Typically, all the intelligence for segment quality selection is at the client side. Player estimates the available network bandwidth and requests the segment with maximum quality minimising stalling or rebuffering events. Over the years, many HAS algorithms have been developed [8]. Traditionally, these algorithms can be classified based on the methods available network resources are estimated: rate-based, buffer-based, and hybrid-based. The rate-based algorithms estimate available resources by measurement of available throughput for segment quality decision [33]. The buffer-based algorithms track changes in the playback buffer levels and map them to segment quality [66]. However, most state-of-the-art algorithms combine both approaches when making the decision [15, 32, 74, 95]. Also, many authors employ different approaches when designing algorithm's adaption logic, including machine learning [46], control theory [15], and optimisation [91].

With the increasing popularity of streaming services, user demand for high Quality of Experience (QoE) has become a cornerstone in design of HAS system. By definition, QoE represents the magnitude of annoyance or the delight of a user's experience with an application or service [10]. Due to its subjective intrinsic component, measuring and modelling user QoE is a formidable task. The overall QoE in HAS comprises of impairments including initial delay, average quality, stall events, switching frequency, and video duration [42]. Minimising and finding optimal combination of these impairments represents a challenging task. Typical approach consists of performing subjective studies devising weights for each of the impairments [17, 42, 56]. The derived QoE models become an objective function in designing adaptation logic of adaptive algorithms [90, 91]. On the network side, vendors usually rely on network metrics, such as packet loss and utilisation, to map to user QoE.

There are two main approaches in the evaluation of QoE: subjective and objective evaluation of QoE. In the objective video evaluation the video sequence is graded automatically without user interaction. Further, the objective evaluation can be classified into three groups: **No Reference (NR)** approach, where original video sequence is not available for comparison with the distorted one. Some popular NR models are Video-BLIINDS [64] and no-reference edge-based blur metric [47]; **Reduced Reference (RR)** models, where the original video sequence is partly available. Some popular RR models are SRR [35], ST-RRED [73] and LOW BANDWIDTH RR VQ [54]; **Full Reference (FR))** approach where the original video sequence is fully available for comparison with the distorted video sequence. The most popular FR models are Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) [84], multi-scale SSIM (MS-SSIM) [85] and Multimethod Assessment Fusion (VMAF) [1]. However the subjective evaluation of QoE represents a foundation for better understanding and modelling user experience. To estimate subjective experience, researchers design a few test sequences containing video impairments. Typically, these impairments are added artificially to the video sequence [42, 76]. Few studies perform both subjective and objective QoE evaluation [17, 20, 24, 42, 69].

The main limitation with subjective evaluation of QoE represents the artificial design of video impairments, which does not reflect realistic network conditions. There is a diverse set of distortions that can occur in streaming session and streamed video sequences are very diverse in terms of content type and quality. It is very important for researchers to create and use realistic-based impaired video sequences datasets in order to improve the adaptation algorithm's logic. In the literature there are many datasets with bandwidth traces collected in various mobile environments under different wireless technologies [38, 60]. These datasets can be used to obtain objective performance of adaptation algorithms, including rate distribution, stall duration, and stall occurrence. Generating test video sequences based on realistic video logs complements the current literature on QoE. To the best of our knowledge, there are only few datasets generated based on real traffic patterns available to the research community [5, 21, 22]. However, only some of them are supplemented with network traces, and none of them is supplemented with video logs and framework that can reproduce or expand them.

Motivated by this observation, we offer a framework for creating video sequences based on video logs collected either in real network or based on realistic bandwidth traces. This work is an extension to our previously published work [28]. Furthermore, we provide 234 video sequences based on video logs analysed over different bandwidth profiles collected from various wireless networks [67]. Video logs were generated by HAS streaming algorithms under bandwidth profiles from different networks, resulting in a realistic snapshot of decisions algorithms made, including bitrate decisions (giving us rate distribution) and stall events (number and duration of stalls). Our contributions are summarised as follows:

- We present DashReStreamer [1], a framework for generating test video sequences with encoded stall and rate changes. The framework supports different Media Presentation description (MPD) or manifest profiles, making it suitable for various types of HAS video content.
- In addition to the generated video sequences, the framework provides objective FR metrics calculation for the distorted video. These metrics include PSNR, SSIM, MS-SSIM and VMAF allowing the design of QoE models with both subjective and objective metrics.
- The framework supports the creation of impaired videos from YouTube links. This contribution allows for creating a more diverse set of video sequences from different genres and user generated content. Content type and user preference for video content can have high relevance in video quality assessment tests as authors discovered in [62]. This can be utilised to additionally investigate it and use it to improve visual quality assessment studies.
- We provide an extensive dataset containing video sequences created over 3G, 4G and WiFi networks. In total, 324 video sequences were generated with a duration of 1 to 5 minutes[2]. The dataset contains video logs and bandwidth traces used for the generation of video sequences with audio included. These video sequences are suitable for subjective QoE evaluation and can aid in the better understanding of user experience in different scenarios. To the best of our knowledge, our QoE dataset is the first publicly available dataset that contains video sequence, logs, FR metrics, Spatial/Temporal information, bandwidth traces, and subjective testing results.
- We performed subjective evaluation with 28 participants, quantifying impact of 196 impaired video content on overall user QoE. Our key findings include the importance of user engagement and abandonment rate on perceived user experience.

[1]https://github.com/khodzic2/DashReStreamer
[2]https://shorturl.at/dtISV

The remainder of this paper is organised as follows. Section 2 describes related work regarding similar datasets and QoE-related video metrics. The overview and key features of proposed framework are explained in Section 3, while Section 4 provides an overview of the dataset generated by DashReStreamer. Section 5 outlines performed subjective evaluation. In Section 6 we layout future work, while Section 7 outlines our conclusion.

2 BACKGROUND AND RELATED WORK

The main goal of HAS algorithms is maximising user perceived QoE. This daunting task relies on accurate representation of the subjective impact of video impairments on the user QoE through mapping objective QoE metrics at client side (e.g., initial delay, average bitrate, rebuffering events, and switching frequency) or metrics measured at the network such as utilisation and packet-loss rate. Also, the majority of the proposed HAS algorithms in the literature relies on using QoE models to quantitatively compare their performance to existing state-of-the art HAS algorithms. Furthermore, QoE models expressed as linear combination of impairments, as depicted in (1), represent a suitable candidate for designing a HAS algorithm that maximises a given QoE model. A typical approach includes the modeling of the QoE model as the utility function of the optimisation problem [7, 90, 92].

A typical template equation used for deriving QoE model is [17, 42, 56]:

$$\mathrm{QoE}_s = w_o \cdot \mathrm{QoE}_m - (w_t \cdot I_t + w_v \cdot I_v) + f(I_t, I_v), \tag{1}$$

where I_t represents temporal impairment factor, and w_t represents its weight. Temporal quality impairments indicate degradation due to initial delay and stall performance (stall number and stall duration). While initial delay has a minor negative effect on QoE (up to 16 seconds), stall events have the highest negative impact on overall user experience [70]. I_v, and w_v represent visual quality impairment factors and its weight, respectively. The average bitrate and switching behaviour represent visual quality impairments. Similar to stall performance, bitrate quality amplitude has a significant effect on QoE [31], unlike switching between different qualities while retaining the same resolution [31]. However, switching between different resolutions can influence user experience [2]. QoE_m depicts the maximum (initial) value (score) for QoE or growth factor depending on the QoE model, and w_o denotes a weight for the QoE_m score. Some QoE models take into account impairments that occur simultaneously. In these scenarios, aggregate subjective effect is not a direct sum of each impairment [42]. The role of function $f(I_t, I_v)$ is to compensate for this effect. However, these impairments (i.e., metrics) are mutually contradictory. High bitrate increases the chance of buffer underflow resulting in stall events, while streaming at low bitrate quality has a severe negative impact on perceived user experience.

To capture the mapping between user perceived experience and objective metrics, many studies use subjective evaluation. This evaluation relies on the assessment of the video quality by participants in a controlled lab environment [13, 42, 56, 71]. Each participant rates a video sequence on a 100-point scale (denoted as R, where some studies use 5 or 10-point scale). The procedure is repeated for a series of test sequences. Each test sequence is embellished with one or more impairments. Finally, for each test sequence and given score R, the impairment impact is calculated as 100-R. Subjective evaluation is an expensive, time-consuming process performed with a limited number of human subjects (usually around 30) restricting the statistical validity of collected results. Alternatively, some studies opt for a crowd-sourcing approach, where a large number of users rate video sequences online in an uncontrolled environment [17, 36, 76]. Subjective studies published up to 2014 are reviewed in [23]. The most recent subjective studies are depicted in Table 1, with details on datasets used, type of subjective testing, number of participants and whether subjective testing is supplemented with objective assessment.

Table 1. Subjective Video Quality Assessment (VQA) overview

Year/Paper	Dataset	Type	Participants	Objective VQA
2014 [49]	LIVE Mobile Stall	Laboratory	54	no
2016 [6]	LIVE-NETFLIX	Laboratory	56	yes
2017 [24]	LIVE Mobile Stall II	Laboratory	54	yes
2017 [97]	BVI-HD	Laboratory	86	yes
2017 [29]	KoNVid-1k	Crowdsourcing	642	no
2018 [49]	LIVE-NFLX-II	Laboratory	55	yes
2019 [39]	Waterloo IVC 4K	Laboratory	66	yes
2020 [94]	LIVE Wild Compressed	Laboratory	40	yes
2020 [93]	LSVQ Database	Crowdsourcing	6300	yes
2020 [48]	KoSMo-1k	Crowdsourcing	1800+	yes
2020 [44]	LIVE-YT-HFR	Laboratory	84	yes
2020 [49]	LIVE-SJTU (A/V-QA)	Laboratory	35	yes
2020 [22]	Waterloo SQoE-III	Laboratory	34	yes
2020 [71]	LIVE-APV Livestream	Laboratory	40	yes
2022 [72]	LIVE HDR	Laboratory	66	no
2022 [94]	ETRI-LIVE STSVQ	Laboratory	34	yes
2022 [20]	Waterloo SQoE-IV	Laboratory	97	yes

The main challenge for subjective evaluation is the augmentation of the test video sequences with particular impairments. Typically, these impairments are artificially created and added to video clips. However, artificially created impairments do not necessarily reflect impairments observed in real network conditions, either their frequency (e.g., number of rate switches, number of stalls), or duration (e.g., stall duration). There are plethora of video quality assessment datasets in the literature. We provide details for the most recent datasets, as shown in Table 2.

Other researchers conducted large-scale studies on the impact of stalling and bitrate switches on user QoE. Unfortunately, their datasets are not available for public use. In [36], the authors use an analytic plugin on the client side to collect more than 23 million video playbacks from 6.7 million unique users. A similar approach with client-side instrumentation is used in [19] to collect information from more than 2 million unique views from over 1 million viewers [41] where the authors measured startup delays and buffering ration from more than 200 million video sessions. In [58] crowdsourcing campaign was run to determine the QoE of each implementation in order to determine the current state-of-the-art for MPEG-DASH systems within real-world environments.

Datasets from Table 2 modelling adaptively streamed videos are: LIVE Mobile Stall, LIVE Mobile Stall II, LIVE-NETFLIX where distortions are synthetically inserted using predefined patterns and, LIVE-NFLX-II, Waterloo SQoE-III and Waterloo SQoE-IV with authentically obtained distortions as stalling and bitrate changes. In [77], authors use predefined stall event patterns; however, the dataset is not publicly available.

Motivated by the lack of video sequences with the impairments based on real network conditions, and a plethora of bandwidth datasets collected in real networks availability in literature [38, 57, 60] that can reflects real conditions observed in networks, we designed a tool for creating video sequences with impairments collected from video sessions collected over realistic bandwidth traces.

There are many tools and frameworks found in the literature that are designed for subjective video QoE assessment. Some of these frameworks are limited to the creation of testing scenarios when deriving QoE models [30]. These

Table 2. VQA datasets overview

Dataset	Year	Original/distorted	Resolutions/duration	Distortions
EPFL-PoliMI [16]	2010	6/78	360p/10s	H.264/AVC, error-prone network sim
ECVQ, EVVQ [80]	2012	8/90	360,480p/10s	H.264/AVC,MPEG-4 compression
MCL-V [40]	2014	12/96	1080p/6s	compression and image size scaling
ReTRiEVED [83]	2014	8/184	570p/7-9s	Packet loss, jitter, delay, throughput
BVI-HFR [43]	2015	22/88	1080p/10s	same video different fps
LIVE Mobile Stall [49]	2016	24/176	360-720p/29-134s	simulated stalls and startup delays
MCL-JCV Dataset [83]	2016	30/1650	1080p/5s	H.264/AVC compression
LIVE-NETFLIX [6]	2016	14/112	1080p/60s+	compression and rebuffering
LIVE Mobile Stall II [24]	2017	24/174	360-720p/29-134s	systematically inserted stall patterns
BVI-HD [97]	2017	32/384	1080p/5s	HEVC compression
KoNVid-1k [29]	2017	1200	720-1080p/8-30s	in the wild distortions
LIVE-NFLX-II [5]	2018	15/420	adaptive/25s	rebuffering, resolution changes
Waterloo IVC 4K [39]	2019	20/1200	540-2160p/10s	AVC, HEVC, VP9, AVS2, AV1
LIVE WC [94]	2020	55/3740	360-1080p/10s	in-capture, compression
LSVQ Database [93]	2020	39095	92%1080p/5-12s	in the wild distortions
KoSMo-1k [48]	2020	30/1350	1440p−>540p/8s	frame interpolation
LIVE-YT-HFR [44]	2020	16/480	1080,2160p/10s	compression , frame rate adjustment
LIVE-SJTU (A/V-QA) [49]	2020	14/336	1080p/8s	audio, video compression, scaling
Waterloo SQoE-III [22]	2020	20/450	240-1080p/13s	stalls, resolution changes
KonVid-150k [27]	2021	150k	720p/5s	in the wild distortions
LIVE-APV Livestream [71]	2021	33/315	1080,2160p/7s	aliasing, judder, flicker, framedrops
AVT-VQDB-UHD-1 [59]	2019/22	16/300	360p-2160p/8-10s	H.264 , HEVC, VP9 compression
LIVE HDR [72]	2022	31/310	540p-2160p/7-10s	compression and aliasing
ETRI-LIVE STSVQ [94]	2022	15/437	540p-2160p/5-7s	space-time subsampling, compression
Waterloo SQoE-IV [21]	2022	5/1350	180p-2160p/34s	stalls, resolution changes

frameworks include Amazon Mechanical Turk[3], Microtask [4], Microworkers [5], and Quadrant of Euphoria [11]. The main drawback of proposed frameworks is their limitation to web-based assessment, excluding mobile and PC implementations.

Other frameworks employ a more active approach, collecting various objective Quality of Service (QoS) metrics at client side (e.g., initial delay, average bitrate, rebuffering events, and switching frequency) for the QoE model derivation [18, 25]. Bitstream-based Quality Prediction Software (BiQPS) is a machine-learning based framework proposed for prediction of the overall quality of the HAS sessions [79]. Nam et al. [50] propose YouSlow, a Chrome plug-in designed to detect various playback events (start-up latency, rebuffering, bitrate changes, video-loaded fraction, and location) while a video is being played. The authors used the proposed framework to collect more than 400,000 YouTube views to evaluate various QoE metrics by analysing video abandonment rates on YouTube. Similarly, Chen et al. [12] proposed QoE Doctor, a tool that runs on the Android mobile device and uses UI control techniques to drive Android apps to automatically replay user behaviour traces, while collecting the corresponding QoE data for offline analysis. Another Android-based application, YoMoApp (YouTube Performance Monitoring Application) [82], passively monitors various metrics (i.e., player state/events, buffer, and video quality level) while streaming YouTube video on end-user smartphones. The authors extended YoMoApp with a cloud dashboard to openly share the full raw measurements retrieved by YoMoApp on registered devices [86]. Unlike client-based, some researchers propose

[3]https://www.mturk.com/
[4]https://microtask.com/
[5]https://www.microworkers.com/

server-based solutions to maximise user QoE by recommending the best encoding scheme depending on the time and user location [34]. Similar, cloud-based framework is proposed for evaluating HAS performance under various network conditions, followed by derivation of Mean opinion score (MOS) score from the P.1203 model [76, 78]. Recently, there have been efforts to design a conceptual generic and extensible framework for model training, model deployment, and re-evaluation in encrypted video streaming [52, 63]. We conclude that, although there are many tools and frameworks used in the field of adaptive streaming and VQA, there is no similar framework published in the literature as the one we present in this paper.

Next, we utilised the fact that our tool can provide every original and impaired video sequence to implement some FR objective models that compare the original video sequence with the distorted video sequence [75]. Some popular FR models that are automatically calculated are PSNR, one of the oldest metrics for image comparison in decibel signal scale that is commonly used as reference for other video quality assessment methods. PSNR is later upgraded with SSIM [84], multi-scale structural similarity index MS-SSIM [85] and VMAF [1]. Some of the earlier works covering surveys of objective quality video assessment methods are published by: Olsson et. al. [51], Winkler et. al. [87, 88], Wu et. al. [89], S. Chikkerur et. al. [14], and Zhou et. al. [98]. Other popular objective VQA methods are Motion-based Video Integrity Evaluation (MOVIE) index [68] that evaluates dynamic video fidelity of spatial and temporal aspects of distortion assessment, and MOSp [9], the perceptual metric based on the spatial texture content and cognition-based factors to identify parts of a video attracting users attention. In [53], authors suggested Full-Reference Video Quality Assessment (FR-VQA) method that analyses the “worst” scores along the spatial and temporal dimensions of a video. In [81], authors explained a an adaptive spatial/temporal pooling strategy based on the observed distribution which is an extension of the most apparent distortion (MAD) index implemented and explained in [37]. Flow similarity index [45] is FR-VQA metric based on distortions in local optical flow statistics. In [3] authors described FR-VQA metric that predicts distortion visibility taking into account models of luminance adaptation, spatiotemporal contrast sensitivity and visual masking. In [96], authors presented a VQA perception-based hybrid model that simulates the human visual system perception process by adaptively combining distortion and blurring artifacts using an enhanced nonlinear model. Bampis et. al. [4] suggested two improvements to the VMAF metric mentioned earlier, called spatio-temporal VMAF and ensemble VMAF, based on perceptually-motivated space-time features calculated at multiple scales.

The key challenge in subjective evaluation lies in augmenting test video sequences with particular impairments. These impairments are typically artificially generated and then incorporated into the video clips. However, these artificially generated impairments may not always accurately represent the impairments experienced in real network conditions. This discrepancy can apply to factors like frequency (such as the number of rate switches or stalls) and duration (for example, the duration of stalls). Recognizing the lack of video sequences that replicate impairments based on real network conditions and the plethora of bandwidth data available in existing literature from real network scenarios, we developed a tool for creating video sequences with impairments gathered from video sessions conducted under realistic bandwidth traces. While there are numerous tools and frameworks available in the literature for subjective video Quality of Experience (QoE) assessment, it is worth noting that no similar framework to the one presented in this paper has been published. Furthermore, we leveraged the capability of our tool to provide both the original and impaired video sequences. This enabled us to implement particular Full-Reference (FR) objective models that compare the original video sequence with the distorted video sequence. We believe this framework and dataset are unique in the existing literature and that they will aid in ongoing research to better understand factors affecting user experience.

3 DASHRESTREAMER OVERVIEW

DashReStreamer is an open-source multiplatform framework that enables the reproduction of network effects on video player performance by creating video clips that include all resolution and bitrate changes and rebuffering events. It can be used on different types of content including the content that is stored on some server and prepared for adaptive streaming by splitting into chunks of a different resolutions and described in regular or byterange .mpd files. Besides of that, the framework can be also used to re-create YouTube videos, by providing only a video Uniform Resource Locator (URL) instead of an .mpd. This functionality opens up many possibilities for a researchers to investigate an impact of all distortions caused by adaptively streaming algorithms on a different video genres, including a wide spectrum of user generated content. As content type and user preference for video content can have high relevance in video quality assessment tests as authors discovered in [62], it can be utilised to additionally investigate it and use it to improve visual quality assessment studies. Objective VQA metrics are very important in evaluating and improving the quality of video content. For example, metrics like PSNR, SSIM, MS-SSIM or VMAF can be used for creation of a more complex video quality prediction models or for the development or improvements of video codecs and streaming protocols. We took advantage of our framework functionality to gather original videos for reference and implemented automatic full reference objective metrics calculation per segment.

The implementation of DashReStreamer is done using the Python programming language and the FFmpeg[6] library. FFmpeg is a cross-platform multimedia framework that can be used to perform various operations on a wide range of media formats including video and image. These operations include transforming, e.g., encoding, decoding, transcoding, multiplexing, demultiplexing, streaming, and filtering.

DashReStreamer main functionality is achieved through the use of video logs generated by the client during the original content stream in an uncontrolled environment (i.e., a real production network). These logs contain information related to HAS QoS metrics, such as segment bitrate, resolution, duration, and stall information. A video log format is shown in Table 3.

Table 3. Sample output from the video log

Type	Description	Unit
Seg_#	Streamed segment number	int
Seg_Dur	Segment duration	ms
Arr_Time	Arrival time	ms
Del_Time	Time taken to receive the segment	ms
Seg_fps	Segment FPS	int
Stall_Dur	Stall duration	ms
Rep_Level	Representation Quality	kbps
Del_Rate	Delivery rate	kbps
Act_Rate	Actual rate	kbps
Byte_Size	Size of segment	byte
Buffer_Level	Buffer level	ms

DashReStreamer requires three key pieces of information in order to generate video clips:

[6]https://www.ffmpeg.org/

- Segment number: index representing the position of each segment.
- Segment bitrate: this is used to map specific subsets of segments used during playback with their representations described in an .mpd file.
- Stall events: the occurrence and duration of stall events are used to add stalls, which involve duplicating the last frame of a segment, at the end of segments that have been affected by rebuffering events.

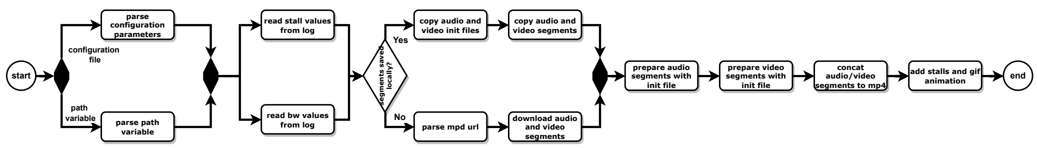


Fig. 1. DashReStreamer workflow

Figure 1 depicts DashReStreamer workflow in creating video impairment content. DashReStreamer starts by parsing video log file, which can be stored in .csv (comma-separated values) or tabular format, where all relevant information about streamed segments are identified for further processing. Logs are parsed and results are stored in python dictionaries. Every segment's index and bitrate are stored in one dictionary. Position and duration of each stall are stored in the other dictionary. The next step includes filtering only a subset of streamed segments. Segments can be stored locally or remotely on a web server. In the first case, all needed audio and video segments and their initialization segment files are copied from a location where they are locally stored to an output location. The initialization segment file contains information required to initialize the video decoder. In the latter case, an .mpd file or a *youtube* link is used for downloading the streamed segments from the server to the local machine. This procedure is similar to the behaviour of traditional HAS client (without actual decoding of the data). If a *youtube* link is passed as a parameter instead of an *.mpd* url then *youtube-dl* library[7] is used to download the different video representations (identified when parsing video logs) of a video clip from a given url. The video clips are then split into segments of a given duration, followed by transfer of all the segments needed to the output location for further processing. *Python-mpegdash*[8] library is used for parsing *.mpd* files. When the regular *.mpd* file is recognised, it is parsed and the urls of the necessary audio and video segments are saved into a dictionary and then downloaded to the local destination. For a byterange .mpd representation type, file is simultaneously parsed and byte ranges of needed segments are downloaded to the local destination. The pseoudocode od these functions is shown in Algorithm 1.

After initialization, audio and video segments are prepared, DashReStreamer proceeds with combining segments with init file (originally segments are in an .m4s format). The output of this operation are new audio and video segments (in an .avi[9] and an .mkv[10] format respectively) which can be played independently. These functions pseudocode is shown in Algorithm 2.

Next, if objective metrics calculation is required and they need to be calculated for a YouTube movie, highest resolution video representation is downloaded and splitted into a segments of a needed duration. Those segments are then later used as reference segments for objective metrics calculation. If objective metrics calculation is calculated for regular/byterange .mpd described video, then maximum parsed resolution saved in dictionary is used to download

[7] https://youtube-dl.org/
[8] https://github.com/sangwonl/python-mpegdash
[9] Audio Video Interleave
[10] Matroska Multimedia Container

Algorithm 1 Video log parsing and local/youtube/mpd segments preparation

```
procedure READ_REPLEVELS_STALLS_LOG(path, log_column_names, delimiter)
    parse_file(csv or tab)
    dictionary ← index, bitrate, duration, stall,
end procedure
procedure PREPARE_LOCAL_SEGMENTS(path_to_segments, destination, dictionaries)
    find_and_copy_audio_init_file()
    find_and_copy_video_init_file()
    copy_video_segments()
    copy_audio_segments()
end procedure
procedure DOWNLOAD_YOUTUBE_MOVIES(path, url, dictionary)
    for resolution in dictionary
        youtube-dl(movie)
        yt_dictionary ← video_name
end procedure
procedure YOUTUBE_SPLIT(path, segment_duration, dictionary)
    for movie in yt_dictionary
        ffmpeg_split(movie,segment_duration)
    copy_needed_segments(dictionary)
end procedure
procedure PARSE_MPD(mpd_url, destination)
    MPEGDASHParser.parse(mpd_url)
    audio_dictionary ← audio_urls
    video_dictionary ← video_urls
    for audio_url in audio_dictionary
        download_segment(audio_url,destination)
    for video_url in video_dictionary
        download_segment(video_url,destination)
end procedure
procedure PARSE_BYTERANGE(mpd_url, destination)
    calculate_byterange(mpd_url)        ▷ For all needed audio and video segments and init files
    download_byterange(destination)     ▷ Download specific bytes of a byterange for every file
end procedure
```

Algorithm 2 Prepare audio and video files with init

```
procedure PREPARE_INIT(path)
    find_init(destination)              ▷ Finds audio and video init files in the destination
    for audio_segment in pathaudio_dictionary
        combine_with_init()
    for video_segment in pathaudio_dictionary
        combine_with_init()
end procedure
```

maximum resolution segments, and combine them with initialization file, in order to be used as reference segments in metrics calculation. When all reference segments are prepared, then every original segment is scaled to a maximum resolution in order to be able to achieve different metrics calculation. After that we use Netflix libvmaf [11] library using

[11] https://github.com/Netflix/vmaf/tree/master

ffmpeg for objective metrics calculation to calculate PSNR, SSIM, MS-SSIM and VMAF metrics. For VMAF metric, the user can send a url file where different pretrained models are stored. By default, currently last model version (in the time of writing this paper that was v0.6.1) is used if no other is provided. The default VMAF model is trained to predict the quality of videos displayed on a 1080p HDTV in a living-room-like environment. VMAF is also calculated with 4k and phone model versions. The subjective experiment used to train phone model uses similar video sequences as the default 1080p HDTV model, except that they were watched on a cellular phone screen. 4k model predicts the subjective quality of video displayed on a 4KTV and viewed from the distance of 1.5 times the height of the display device. In total 4 .csv files (1080p - tv, 4k - tv, 1080p - mobile, and 4k -mobile) with different VMAF version in addition to PSNR, SSIM and MS-SSIM are calculated for every streamed video segment. These functions pseudocode is shown in Algorithm 3.

Algorithm 3 Prepare and scale segments then calculate objective metrics

```
procedure DOWNLOAD_YT_SEGMENTS(path, url, segment_duration)
    youtube_dl(path, url)                      ▷ Download max resolution video representation
    ffmpeg_split(path, segment_duration)       ▷ Split video into a segments of a given duration
end procedure
procedure DOWNLOAD_MAXRES_SEGMENTS(path, dictionary, destination)
     getmaxres_helper(destination)             ▷ Find maximum resolution segment and save it as a variable
    max_resolution ← maxres
     for segment in dictionary
        download_segment(segment, max_resolution, destination)
        init_segment(segment, destination)
end procedure
procedure SCALE_SEGMENTS(path, dictionary)
    getmaxres_helper(destination)              ▷ Find maximum resolution segment and save it as a variable
    resolution ← maxres
     for inited_segment in destination
        ffmpeg_scale(inited_segment,resolution)
end procedure
procedure CALCULATE_METRICS(path, modelpath, modelpath4k)
     for segment, reference_segment in path
        calculate_psnr(segment, reference_segment)
        calculate_ssim(segment, reference_segment)
        calculate_msssim(segment, reference_segment)
        calculate_vmaf(segment, reference_segment, modelpath, modelpath4k)
        save_to_csv(psnr, ssim, mssim, vmaf_normal, vmaf_normal_phone, vmaf4k, vmaf4k − phone)
end procedure
```

Next, if video segments merging is required, we combine the individual pairs of audio and video segments, using the FFmpeg library. In this step, combined segments can also be rescaled to a different resolution if that is indicated by a parameter. For YouTube video, this step is skipped as YouTube segments are already combined with audio. These functions pseudocode is shown in Algorithm 4.

To produce a video sequence that includes all bitrate/resolution changes and stall events, we follow a specific process. Initially, we generate stall-induced segments, which involves using the duration of the stall and the segment just before it starts. We then take the last frame of the identified segment and append it to the end of the segment for the duration of the stall. Next, we overlay a .gif[12] showing stalling event on top of the stall-induced segments. Once all segments are

[12] Graphics Interchange Format

Algorithm 4 Create final merged video

procedure CONCAT_SEGMENTS_FINAL($path, gifpath, stalls\_dictionary, segments\_dictionarypath\_final$)
  for $stall$ in $stalls\_dictionary$
    ffmpeg_sseof($merged\_segment, stall.duration$) ▷ create jpg picture from the stalled segments last frame
    ffprobe($merged\_segment$) ▷ use ffprobe to get segment audio and video details
    $segment\_info \leftarrow width, height, fps, duration, sample\_rate, channel\_layout, codec\_name$
    $stalled\_jpg \leftarrow ffmpeg\_loop$(segment_info,stall.duration) ▷ create new segment of a stalled part - jpg of the stall duration
    $stalled\_segment \leftarrow ffmpeg\_filter\_complex$(merged_segment,stalled_jpg, gifpath) ▷ concat original segment and stalled part + add stalling gif animation
    segment_dictionary.add($stalled\_segment$
    ffmpeg_merge($segment\_dictionary, path\_final$) ▷ merge all segments including stalled ones into one movie and save it to a required path
end procedure

ready, we merge them into a final .mkv video file. Finally, if it is indicated by an input parameter, all intermediate files that are created in a process are deleted except the final .mkv video.

The main limitation of the proposed framework lies in its reliance on external video logs to create impaired video sequences. While this approach is suitable for the 'offline' generation of impaired video sequences, it could be extended to an "online" approach where video sequences are streamed directly to end-users over a real network. In this case, video logs and impaired video sequences would be generated on the fly. This approach eliminates the need for a data-driven testbed. Furthermore, the framework currently only supports locally stored video content or the YouTube platform. While proposed framework relies on the creation of video sequences with realistic impairments, the addition of artificial impairments would be beneficial for conducting fine-grained subjective studies aimed at assessing the impact of particular impairments.

4 QOE DATASET OVERVIEW

In this section, we provide a brief summary of the dataset[13] that was utilized to generate a variety of video sequences under different wireless conditions. The majority of the video sequences within this dataset includes at least one instance of rebuffering, as these cases are particularly relevant for modelling adaptive streaming QoE.

4.1 Video Logs Generation

For the creation of the video sequences, we rely on video logs produced by experiments described in [67]. The video logs are generated based on bandwidth traces collected from real operational networks. Figure 2 illustrates a generalised testbed used for producing video logs.

The experimental setup involves a server machine, an intermediate device (such as a wireless access point), and one or more wireless-enabled end devices (like mobile devices). The server machine serves as both a web server for video content and a traffic shaper for the connection between the server and intermediate device. To simulate different network conditions, the traffic shaping process uses tools like Linux traffic control (tc) and bandwidth logs. Values are extracted from the bandwidth log and applied to a bottleneck link (i.e., constrained link) using tc tool. Following a specific time interval, determined by the granularity of the bandwidth log, this value is replaced with the next value

[13] https://shorturl.at/dtISV

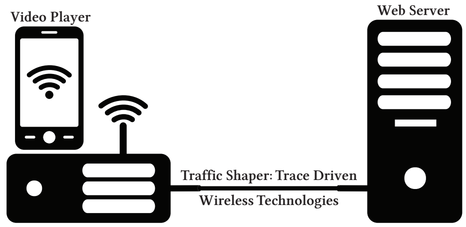


Fig. 2. The data-driven generation testbed.

from the log. The intermediate device connects to the end devices via a WiFi channel. The end devices stream content from the server through a constrained link, resulting in the creation of a video log once the streaming is complete.

The video content stored on the server is an animation clip encoded in 4K resolution using the H.264/AVC codec. The clip is encoded at thirteen different bitrates, ranging from 235 Kbps to 40 Mbps, and across eight different resolutions.

To shape the traffic, bandwidth logs were collected from three wireless technologies: 3G, 4G, and WiFi. The logs included various mobility patterns such as static, pedestrian, car, bus, and tram.

Table 4 depicts a summary of the statistics, including the average and standard deviation of measured bandwidth, for the 3G, 4G, and WiFi logs [67]. 3G logs exhibit the lowest average bitrate when compared to 4G and WiFi. The relatively high standard deviation in 3G bandwidth logs negatively affects the video QoE metrics. To illustrate, Table 7 reveals that video logs based on 3G bandwidth data exhibit a higher number of quality switches, stalls, longer stall durations, and lower average quality bitrates compared to the other two technologies. Conversely, with the highest average bitrate, WiFi logs have the least detrimental impact on video QoE metrics, as demonstrated in Table 7.

Table 4. Throughput Statistics for collected bandwidth logs

Technology	**Average (Mbps)**	**Standard Deviation (Mbps)**
3G	1.26	0.97
4G	11.32	13.17
WiFi	18.71	17.73

4.2 Video Sequences Generation

For the first part of our dataset, we used the video logs described in section 4.2 and our proposed tool (discussed in Section 3) to create 234 impaired video clips. As for the video content, we selected three open-source clips from paper [55]. Video information is shown in Table 5.

Table 5. Dataset I info

Video clip	Type	Duration	Max resolution / fps	avg si/ti metric
Big Buck Bunny (BBB)[14]	animated characters with a simple background	10m:34s	3840x2160 / 60	30 / 5.6
Sintel[15]	complex animated characters and scenery	4m:48 s	3840x2160s / 24	28.3 / 9.6
Tears of Steel (TOS)[16]	real actors with superimposed digital effects	12m:14s	3840x2160s / 24	29.8 / 9.6

Each of the selected clips is encoded at thirteen different bitrates and eight different resolutions, as shown in Table 6 and sourced from paper [55]. Additionally, all clips include audio for a duration of five minutes plus the total stall duration. We chose 27, 25, and 26 video logs generated from 3G, 4G, and WiFi network traces, respectively. Table 7 provides the video quality-of-service metric statistics for the selected logs.

Table 6. Ladder for the average encoding rate, and resolution for the used dataset

No.	Bitrate	Resolution
13	40 Mbps	3840x2160
12	25 Mbps	3840x2160
11	15 Mbps	3840x2160
10	4.3 Mbps	1920x1080
9	3.85 Mbps	1920x1080
8	3 Mbps	1280x582
7	2.35 Mbps	1280x582
6	1.75 Mbps	720x328
5	1.05 Mbps	640x292
4	750 kbps	512x234
3	560 kbps	512x234
2	375 kbps	384x174
1	235 kbps	320x146

Table 7. Average QoS metrics for selected video logs

Network	Bitrate (Mbps)	Num. Switches	Num. Stalls	Stall Dur. (s)
3G	1.6	19.6	3.4	53.9
4G	5.8	18.8	0.96	14.3
WiFi	6.3	12.5	0.77	1.95

For the second part of our dataset we have chosen 2 popular YouTube videos per category (movie, animated, documentary, gaming, sport, music and news). We made sure that the videos were uploaded using CC (Creative Commons) YouTube license type. Information about videos is given in Table 8

Figure 3 depicts boxplot of measured throughput for WiFi, 3G, and 4G technologies. On average WiFi logs shows highest throughput values compared to 3G and 4G. This result is intuitive as WiFi logs are collected in static environment. Also, 4G exhibits highest variation in measured throughput with values ranging up to 50000 Kbps. Overall 3G depicts lowest measured throughput. This leads to highest number of stalls and stall duration for 3G traces, followed by 4G and WiFi network traces. This result is intuitive and aligns with the throughput statistics presented in Table 4.

High throughput values for WiFi and 4G result in a low number of stall events compared to 3G, as outlined in Table 4. The time when stall occurs is shown in Figure 4. According to the Figure 3, most of the stall events for 4G and WiFi occur in the first 200 seconds of the video sessions. Further analysis of stall events is depicted in Figure 5a and 5c, showing the time when stall occurs and its duration. Most of the stalls for 4G happen at the beginning of the session, with very few stalls taking place toward the end of the session. However, for the WiFi, stall events are more evenly spread over the duration of the session. We believe this observation is due to nature of collection of WiFi logs. WiFi logs are collected in static environment, limiting fluctuation in wireless channel thus having less negative impact on video QoE metrics.

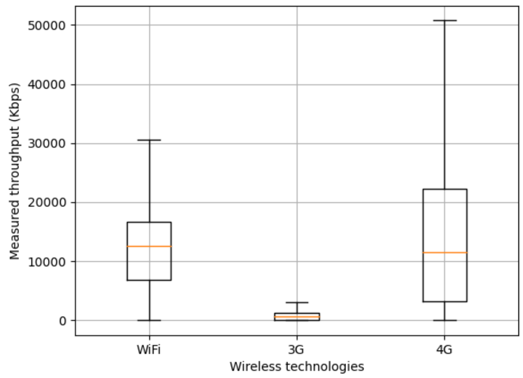


Fig. 3. Throughput distribution across different wireless technologies

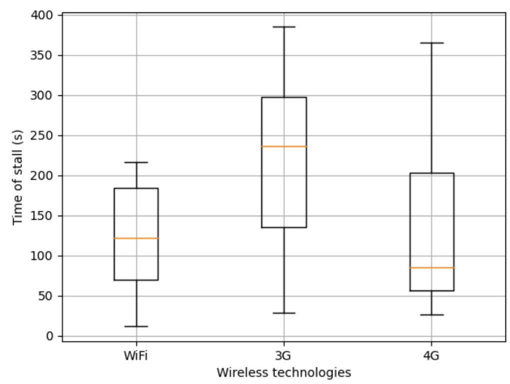
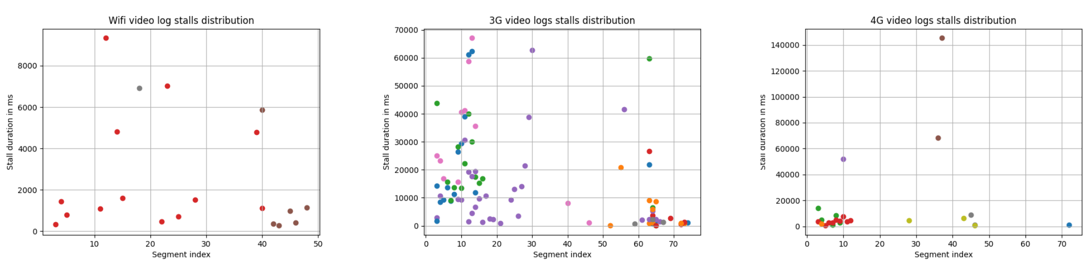


Fig. 4. Distribution of stall occurrence across different wireless technologies



(a) WiFi (b) 3G (c) 4G

Fig. 5. Distribution of stall events and stall duration across video sessions

Table 8. YouTube dataset information

Video clip	Category	Duration	Number of views	avg si/ti metric
HIGHLIGHTS | Valencia 1-3 Real Madrid | Spanish Super Cup [17]	Sport	2m:29s	4.2M+	26.4 / 11.4
THANK YOU, C. RONALDO | Real Madrid Official Video[18]	Sport	6m:36s	59M+	32.9 / 12.5
LES TWINS | World of Dance CHAMPIONS | WE MADE IT[19]	Music	5m:18s	3.9M+	20.5 / 3.7
Real Madrid official music video | If You Create The Noise[20]	Music	2m:59s	6.5M+	15.8 / 10.2
EVERTON STADIUM UPDATE | Stadium Being Brought To Life[21]	News	5m:30s	27k+	31 / 4.7
Vivek Ramaswamy on Fox News 6.29.23[22]	News	5m:30s	44k+	68.1 / 3.3
Dark Souls III - Opening Cinematic Trailer | PS4, XB1, PC[23]	Gaming	3m:35s	6.1M+	16.6 / 2
Pocket Champs | Official Trailer | 2022 [24]	Gaming	0m:52s	68M+	12.9 / 6
TERMINATOR 7: End Of War (2022) Official Trailer Teaser[25]	Movie	1m:21s	16M+	10.9 / 5.7
Wrong Number Mr Bean! | Classic Mr Bean[26]	Movie	10m:45s	21M+	19.7 / 7.4
F-35B in action[27]	Documentary	9m:49s	7.3M+	15.6 / 3.9
SUPERSPREADER - Documentary Trailer - Faith Forward[28]	Documentary	1m:0s	1M+	29.7 / 11.1
Teen Titans Go! to the Movies -Alan Walker - Spectre [29]	Animated	0m:52s	82M+	33.3 / 7.3
Turn That Crown Upside Down - Pencilmation [30]	Animated	4m:38s	68M+	29.6 / 5.6

For the 3G logs, most of the stall events occur after 150 seconds of video session, as depicted in Figure 4. Figure 5b depicts that most of the stalls occur at the beginning and end of the video sessions. One reason for this observation can be attributed to the presence of diverse mobility patterns within the collected 3G logs. These logs include routes such as metros, ferries, and trains, where the bandwidth values tend to decrease as users move farther away from the base station. Previous analysis shows limitations of modelling stall events arbitrarily, as the distribution of stall events is heavily dependent on environment in which users stream video content.

Finally our dataset consists of the following features:

(1) Video sequences encoded with the impairments based on a real bandwidth logs collected in 3G, 4G and WiFi environments.
(2) Objective video metrics ((VMAF, SSIM, MS-SSIM, PSNR)) and metrics related to compression difficulty [61] (i.e., Spatial Information and Temporal Information) calculated for each video sequence.
(3) Bandwidth logs containing measured throughput captured in 3G, 4G, and WiFi networks under different mobility patterns.
(4) Video log files containing information for each segment bitrate, bitrate switching behavior and stall events obtained based on collected bandwidth logs.

5 SUBJECTIVE EVALUATION OF IMPAIRED VIDEO SEQUENCES

We complement our dataset with a subjective study conducted in a controlled laboratory environment. We use YouTube videos for subjective testing. The video dataset consists of a total of 98 videos, with 14 videos representing each of the seven categories: sports, music, news, gaming, movies, documentaries, and animated content, as depicted in Table 8. Video lengths vary from 50 seconds to 330 seconds, depending on the video genre.

Subjective testing was conducted in a controlled environment using the modified AVRate Voyager, an open-source online testing platform [26]. The entire experiment took place on the same PC in a controlled laboratory environment, using a 32” 4K monitor for viewing.

Before each session, users received a brief explanation of how the testing would be conducted, including the meanings of the terms listed in the questionnaire. During the training stage, each user was provided with an example of a short video and a questionnaire. Following the training stage, a demographic form was administered, followed by seven videos randomly chosen from each category.

After watching each video, users rate quality on a scale 1-5, as depicted in Table 9. In addition to the overall rating, users can mark all degradations that have negatively affected the rating. These degradations include stalling, resolution switches, low quality (including artifacts and pixelation), and uninteresting content. Finally, users can indicate whether they would normally stop watching the video. In total, 28 subjects participated with 189 video sequence rankings.

Table 9. Evaluation criteria for video sequences

Quality Evaluation	Description
5	Excellent experience
4	Minor impairments
3	Noticeable impairments
2	Clearly impairments
1	Annoying experience

The average quality rating is 3.18 across all videos, with distribution for each rating depicted in Table 10. Majority of the user felted that videos had minor or major visual impairments.

Table 10. Throughput Statistics for collected bandwidth logs

Rating	**Percentage (%)**
1	5.3
2	27
3	23.8
4	32.1
5	11.6

The leading impairments for video with annoying experience (1) were low quality and switching frequency (45.2%) followed by the stalling events. Similarly, low quality impairment was leading factor for videos rated as 2. For the remaining ratings, switches were the dominant impairment in deciding overall user experience.

Observation #1: Overall, switching frequency has a significant effect on user experience across all ratings. While previous studies show that stalling events and low quality are the main driver for the user experience [31, 70], the use of 4K large screen exaggerates the impact of switches on the user QoE.

Out of 98 video sequences, 22% were encoded with the stall events. In 95% of them, user ranked stalling events as the dominant factor negatively affecting the overall user QoE.

In our analysis we introduced the possibility of users to mark if they find content interesting. Our hypothesis is that the lack of interest in the video content, would result in lower overall user QoE. The average rating for content that users found interesting is 3.28. However, for uninteresting content, average rating drops to 2.88.

Observation #2: User engagement plays a key role in user QoE, resulting in 13% decrease of average user QoE for the uninteresting content. Furthermore, 25% of the user would stop watching the video due to low user engagement.

Another important aspect is the analysis of the abandonment rate. The abandonment rate represents percentage of video content which user would stop watching. In our study 36% of video content would be abandoned. Dominant factor for abandoning the content is low quality (33.8%), followed by frequency of switches (28%) and stall events (12.5%).

Observation #3: The abandonment rate represents one of the key factors for overall user QoE. However, the majority of QoE models only predicts overall QoE score of the content, they omit modelling of the abandonment rate.

Similar to previous studies, our study shows that quality, switching frequency and stall events play a dominant role in overall user QoE. However, engagement of the user is a factor that needs to be included when deriving objective QoE model. Finally, modelling of the abandonment rate and its effect on QoE model represents exciting open venue for future research.

6 FUTURE WORK

Future work will include extended subjective testing evaluation of a dataset created with DashReStreamer and quantifying impact of user engagement and abandonment rate on overall user QoE. Based on the subjective evaluation, future work will focus on deriving novel objective QoE model that incorporates probability of user abandoning content due to different impairments.

For the DashReStreamer, future work will include extending the framework with arbitrary addition of impairments to analyse in detail effect on each impairment. As for the bandwidth traces, we plan to collect a set of a 5G mobile network traces and use them to complement existing dataset.

7 CONCLUSIONS

The paper describes a framework called DashReStreamer, which is an open-source and cross-platform tool for reproducing adaptively streamed video from real operational networks. With DashReStreamer, it is possible to recreate video clips with all the bitrate/quality changes and stall events that occur in the network. The tool employs video logs generated by adaptive streaming algorithms to mimic their behaviour and selects bitrates based on realistic time-varying conditions observed in the network. It also supports different full reference objective metrics calculation automatically, and also scaling video clips to a required resolution. We supplement the framework with 332 video clips that mimic the behaviour of various adaptive streaming algorithms under different wireless technologies (3G, 4G, and WiFi), creating a dataset with realistic bitrate changes and stall events. We believe that the dataset and subjective evaluation results will be useful for researchers to better understand the factors affecting user experience for adaptive streaming multimedia technologies and to aid in both objective and subjective quality of experience evaluations.

ACKNOWLEDGMENTS

The authors acknowledge the support of the Ministry of Education, Science and Youth of Sarajevo Canton, while also acknowledging the support of CSIT, University College Cork, Ireland. We would also want to express our gratitude for all of the participants who took their time to participate in our subjective analysis research

REFERENCES

[1] 2016. Toward A Practical Perceptual Video Quality Metric. (2016). https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652

[2] A. Asan, W. Robitza, I. h. Mkwawa, L. Sun, E. Ifeachor, and A. Raake. 2017. Impact of video resolution changes on QoE for adaptive video streaming. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*. 499–504. https://doi.org/10.1109/ICME.2017.8019297

[3] Tunç Ozan Aydin, Martin Čadík, Karol Myszkowski, and Hans-Peter Seidel. 2010. Video Quality Assessment for Computer Graphics Applications. *ACM Trans. Graph.* 29, 6, Article 161 (dec 2010), 12 pages. https://doi.org/10.1145/1882261.1866187

[4] Christos G. Bampis, Zhi Li, and Alan C. Bovik. 2019. Spatiotemporal Feature Integration and Model Fusion for Full Reference Video Quality Assessment. *IEEE Transactions on Circuits and Systems for Video Technology* 29, 8 (2019), 2256–2270. https://doi.org/10.1109/TCSVT.2018.2868262

[5] Christos G. Bampis, Zhi Li, Ioannis Katsavounidis, Te-Yuan Huang, Chaitanya Ekanadham, and Alan C. Bovik. 2018. Towards Perceptually Optimized End-to-end Adaptive Video Streaming. arXiv:1808.03898 [eess.IV]

[6] Christos George Bampis, Zhi Li, Anush Krishna Moorthy, Ioannis Katsavounidis, Anne Aaron, and Alan Conrad Bovik. 2017. Study of Temporal Effects on Subjective Video Quality of Experience. *IEEE Transactions on Image Processing* 26, 11 (2017), 5217–5231. https://doi.org/10.1109/TIP.2017.2729891

[7] 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* (Amsterdam, Netherlands) *(MMSys '18)*. Association for Computing Machinery, New York, NY, USA, 13–26. https://doi.org/10.1145/3204949.3204961

[8] Abdelhak Bentaleb, Bayan Taani, Ali C. Begen, Christian Timmerer, and Roger Zimmermann. 2019. A Survey on Bitrate Adaptation Schemes for Streaming Media Over HTTP. *IEEE Communications Surveys & Tutorials* 21, 1 (2019), 562–585. https://doi.org/10.1109/COMST.2018.2862938

[9] Abharana Bhat, Sampath Kannangara, Yafan Zhao, and Iain Richardson. 2012. A Full Reference Quality Metric for Compressed Video Based on Mean Squared Error and Video Content. *IEEE Transactions on Circuits and Systems for Video Technology* 22, 2 (2012), 165–173. https://doi.org/10.1109/TCSVT.2011.2158465

[10] Kjell Brunnström et. al. 2013. Qualinet White Paper on Definitions of Quality of Experience. https://hal.archives-ouvertes.fr/hal-00977812 Qualinet White Paper on Definitions of Quality of Experience Output from the fifth Qualinet meeting, Novi Sad, March 12, 2013.

[11] Kuan-Ta Chen, Chi-Jui Chang, Chen-Chi Wu, Yu-Chun Chang, and Chin-Laung Lei. 2010. Quadrant of euphoria: a crowdsourcing platform for QoE assessment. *IEEE Network* 24, 2 (2010), 28–35. https://doi.org/10.1109/MNET.2010.5430141

[12] Qi Alfred Chen, Haokun Luo, Sanae Rosen, Z. Morley Mao, Karthik Iyer, Jie Hui, Kranthi Sontineni, and Kevin Lau. 2014. QoE Doctor: Diagnosing Mobile App QoE with Automated UI Control and Cross-Layer Analysis. In *Proceedings of the 2014 Conference on Internet Measurement Conference* (Vancouver, BC, Canada) *(IMC '14)*. Association for Computing Machinery, New York, NY, USA, 151–164. https://doi.org/10.1145/2663716.2663726

[13] Manri Cheon and Jong-Seok Lee. 2018. Subjective and Objective Quality Assessment of Compressed 4K UHD Videos for Immersive Experience. *IEEE Transactions on Circuits and Systems for Video Technology* 28, 7 (2018), 1467–1480. https://doi.org/10.1109/TCSVT.2017.2683504

[14] Shyamprasad Chikkerur, Vijay Sundaram, Martin Reisslein, and Lina J. Karam. 2011. Objective Video Quality Assessment Methods: A Classification, Review, and Performance Comparison. *IEEE Transactions on Broadcasting* 57, 2 (2011), 165–182. https://doi.org/10.1109/TBC.2011.2104671

[15] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo. 2013. ELASTIC: A Client-Side Controller for Dynamic Adaptive Streaming over HTTP (DASH). In *2013 20th International Packet Video Workshop*. IEEE, 1–8. https://doi.org/10.1109/PV.2013.6691442

[16] F. De Simone, M. Tagliasacchi, M. Naccari, S. Tubaro, and T. Ebrahimi. 2010. A H.264/AVC video database for the evaluation of quality metrics. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2430–2433. https://doi.org/10.1109/ICASSP.2010.5496296

[17] J. De Vriendt, D. De Vleeschauwer, and D. Robinson. 2013. Model for estimating QoE of video delivered using HTTP adaptive streaming. In *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*. 1288–1293.

[18] Lam Dinh-Xuan, Michael Seufert, Florian Wamser, and Phuoc Tran-Gia. 2017. Study on the accuracy of QoE monitoring for HTTP adaptive video streaming using VNF. In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. 999–1004. https://doi.org/10.23919/INM.2017.7987425

[19] Florin Dobrian, Vyas Sekar, Asad Awan, Ion Stoica, Dilip Joseph, Aditya Ganjam, Jibin Zhan, and Hui Zhang. 2011. Understanding the Impact of Video Quality on User Engagement. *SIGCOMM Comput. Commun. Rev.* 41, 4 (aug 2011), 362–373. https://doi.org/10.1145/2043164.2018478

[20] 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:2008.08804 [eess.IV]

[21] Zhengfang Duanmu, Wentao Liu, Zhuoran Li, Diqi Chen, Zhou Wang, Yizhou Wang, and Wen Gao. 2020. The Waterloo Streaming Quality-of-Experience Database-IV. https://doi.org/10.21227/j15a-8r35

[22] Zhengfang Duanmu, Abdul Rehman, and Zhou Wang. 2018. A Quality-of-Experience Database for Adaptive Video Streaming. *IEEE Transactions on Broadcasting* 64, 2 (2018), 474–487. https://doi.org/10.1109/TBC.2018.2822870

[23] M.-N Garcia, F. De Simone, S. Tavakoli, N. Staelens, S. Egger, K. Brunnström, and A. Raake. 2014. Quality of experience and HTTP adaptive streaming: A review of subjective studies. In *2014 Sixth International Workshop on Quality of Multimedia Experience (QoMEX)*. 141–146. https://doi.org/10.1109/QoMEX.2014.6982310

[24] Deepti Ghadiyaram, Janice Pan, and Alan C. Bovik. 2019. A Subjective and Objective Study of Stalling Events in Mobile Streaming Videos. *IEEE Transactions on Circuits and Systems for Video Technology* 29, 1 (2019), 183–197. https://doi.org/10.1109/TCSVT.2017.2768542

[25] Gerardo Gómez, Lorenzo Hortiguela, Quiliano Perez, Javier Lorca, Raquel Garcia, and Mari Aguayo-Torres. 2014. YouTube QoE Evaluation Tool for Android Wireless Terminals. *EURASIP Journal on Wireless Communications and Networking* 2014 (05 2014). https://doi.org/10.1186/1687-1499-2014-164

[26] Steve Göring, Rakesh Rao Ramachandra Rao, Stephan Fremerey, and Alexander Raake. 2021. AVRate Voyager: an open source online testing platform. In *2021 IEEE 23st International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 1–6.

[27] Franz Götz-Hahn, Vlad Hosu, Hanhe Lin, and Dietmar Saupe. 2021. KonVid-150k: A Dataset for No-Reference Video Quality Assessment of Videos in-the-Wild. *IEEE Access* 9 (05 2021). https://doi.org/10.1109/ACCESS.2021.3077642

[28] Kerim Hodzic, Mirsad Cosovic, Sasa Mrdovic, Jason J. Quinlan, and Darijo Raca. 2022. Realistic Video Sequences for Subjective QoE Analysis. In *Proceedings of the 13th ACM Multimedia Systems Conference* (Athlone, Ireland) *(MMSys '22)*. Association for Computing Machinery, New York, NY, USA, 246–251. https://doi.org/10.1145/3524273.3532894

[29] Vlad Hosu, Franz Hahn, Mohsen Jenadeleh, Hanhe Lin, Hui Men, Tamás Szirányi, Shujun Li, and Dietmar Saupe. 2017. The Konstanz natural video database (KoNViD-1k). In *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*. 1–6. https://doi.org/10.1109/QoMEX.2017.7965673

[30] Tobias Hoßfeld, Matthias Hirth, Pavel Korshunov, Philippe Hanhart, Bruno Gardlo, Christian Keimel, and Christian Timmerer. 2014. Survey of web-based crowdsourcing frameworks for subjective quality assessment. In *2014 IEEE 16th International Workshop on Multimedia Signal Processing (MMSP)*. 1–6. https://doi.org/10.1109/MMSP.2014.6958831

[31] T. Hoßfeld, M. Seufert, C. Sieber, and T. Zinner. 2014. Assessing effect sizes of influence factors towards a QoE model for HTTP adaptive streaming. In *2014 Sixth International Workshop on Quality of Multimedia Experience (QoMEX)*. 111–116. https://doi.org/10.1109/QoMEX.2014.6982305

[32] 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* (Chicago, Illinois, USA) *(SIGCOMM '14)*. ACM, New York, NY, USA, 187–198. https://doi.org/10.1145/2619239.2626296

[33] J. Jiang, V. Sekar, and H. Zhang. 2014. Improving Fairness, Efficiency, and Stability in HTTP-Based Adaptive Video Streaming With Festive. *IEEE/ACM Transactions on Networking* 22, 1 (Feb 2014), 326–340. https://doi.org/10.1109/TNET.2013.2291681

[34] Takuto Kimura, Masahiro Yokota, Arifumi Matsumoto, Kei Takeshita, Taichi Kawano, Kazumichi Sato, Hiroshi Yamamoto, Takanori Hayashi, Kohei Shiomoto, and Kenichi Miyazaki. 2017. QUVE: QoE Maximizing Framework for Video-Streaming. *IEEE Journal of Selected Topics in Signal Processing* 11, 1 (2017), 138–153. https://doi.org/10.1109/JSTSP.2016.2632060

[35] Michail-Alexandros Kourtis, Harilaos G. Koumaras, and Fidel Liberal. 2016. Reduced-reference video quality assessment using a static video pattern. *Journal of Electronic Imaging* 25, 4 (2016), 043011. https://doi.org/10.1117/1.JEI.25.4.043011

[36] S. Shunmuga Krishnan and Ramesh K. Sitaraman. 2012. Video Stream Quality Impacts Viewer Behavior: Inferring Causality Using Quasi-Experimental Designs. In *Proceedings of the 2012 Internet Measurement Conference* (Boston, Massachusetts, USA) *(IMC '12)*. Association for Computing Machinery, New York, NY, USA, 211–224. https://doi.org/10.1145/2398776.2398799

[37] Eric Larson and Damon Chandler. 2010. Most apparent distortion: Full-reference image quality assessment and the role of strategy. *J. Electronic Imaging* 19 (01 2010), 011006. https://doi.org/10.1117/1.3267105

[38] Li Li, Ke Xu, Dan Wang, Chunyi Peng, Qingyang Xiao, and Rashid Mijumbi. 2015. A measurement study on TCP behaviors in HSPA+ networks on high-speed rails. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. 2731–2739. https://doi.org/10.1109/INFOCOM.2015.7218665

[39] Zhuoran Li, Zhengfang Duanmu, Wentao Liu, and Zhou Wang. 2019. AVC, HEVC, VP9, AVS2 or AV1? — A Comparative Study of State-of-the-Art Video Encoders on 4K Videos. In *Image Analysis and Recognition: 16th International Conference, ICIAR 2019, Waterloo, ON, Canada, August 27–29, 2019, Proceedings, Part I* (Waterloo, ON, Canada). Springer-Verlag, Berlin, Heidelberg, 162–173. https://doi.org/10.1007/978-3-030-27202-9_14

[40] Joe Yuchieh Lin, Rui Song, Chi-Hao Wu, TsungJung Liu, Haiqiang Wang, and C.-C. Jay Kuo. 2015. MCL-V: A streaming video quality assessment database. *Journal of Visual Communication and Image Representation* 30 (2015), 1–9. https://doi.org/10.1016/j.jvcir.2015.02.012

[41] Xi Liu, Florin Dobrian, Henry Milner, Junchen Jiang, Vyas Sekar, Ion Stoica, and Hui Zhang. 2012. A Case for a Coordinated Internet Video Control Plane. *SIGCOMM Comput. Commun. Rev.* 42, 4 (aug 2012), 359–370. https://doi.org/10.1145/2377677.2377752

[42] Y. Liu, S. Dey, F. Ulupinar, M. Luby, and Y. Mao. 2015. Deriving and Validating User Experience Model for DASH Video Streaming. *IEEE Transactions on Broadcasting* 61, 4 (Dec 2015), 651–665.

[43] Alex Mackin, Fan Zhang, and David R. Bull. 2015. A study of subjective video quality at various frame rates. In *2015 IEEE International Conference on Image Processing (ICIP)*. 3407–3411. https://doi.org/10.1109/ICIP.2015.7351436

[44] Pavan C. Madhusudana, Xiangxu Yu, Neil Birkbeck, Yilin Wang, Balu Adsumilli, and Alan C. Bovik. 2021. Subjective and Objective Quality Assessment of High Frame Rate Videos. *IEEE Access* 9 (2021), 108069–108082. https://doi.org/10.1109/access.2021.3100462

[45] K. Manasa and Sumohana S. Channappayya. 2016. An Optical Flow-Based Full Reference Video Quality Assessment Algorithm. *IEEE Transactions on Image Processing* 25, 6 (2016), 2480–2492. https://doi.org/10.1109/TIP.2016.2548247

[46] 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* (Los Angeles, CA, USA) *(SIGCOMM '17)*. Association for Computing Machinery, New York, NY, USA, 197–210. https://doi.org/10.1145/3098822.3098843

[47] P. Marziliano, F. Dufaux, S. Winkler, and T. Ebrahimi. 2002. A no-reference perceptual blur metric. In *Proceedings. International Conference on Image Processing*, Vol. 3. III–III. https://doi.org/10.1109/ICIP.2002.1038902

[48] Hui Men, Vlad Hosu, Hanhe Lin, Andrés Bruhn, and Dietmar Saupe. 2020. Visual Quality Assessment for Interpolated Slow-Motion Videos Based on a Novel Database. In *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. 1–6. https://doi.org/10.1109/QoMEX48832.2020.9123096

[49] Xiongkuo Min, Guangtao Zhai, Jiantao Zhou, Mylène C. Q. Farias, and Alan Conrad Bovik. 2020. Study of Subjective and Objective Quality Assessment of Audio-Visual Signals. *IEEE Transactions on Image Processing* 29 (2020), 6054–6068. https://doi.org/10.1109/TIP.2020.2988148

[50] Hyunwoo Nam, Kyung-Hwa Kim, and Henning Schulzrinne. 2016. QoE matters more than QoS: Why people stop watching cat videos. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*. 1–9. https://doi.org/10.1109/INFOCOM.2016.7524426

[51] Sofie Olsson, Mario Stroppiana, and Jamal Baina. 1997. Objective methods for assessment of video quality: state of the art. *IEEE transactions on broadcasting* 43, 4 (1997), 487–495.

[52] Irena Orsolic and Lea Skorin-Kapov. 2020. A Framework for in-Network QoE Monitoring of Encrypted Video Streaming. *IEEE Access* 8 (2020), 74691–74706. https://doi.org/10.1109/ACCESS.2020.2988735

[53] Jincheol Park, Kalpana Seshadrinathan, Sanghoon Lee, and Alan Conrad Bovik. 2013. Video Quality Pooling Adaptive to Perceptual Distortion Severity. *IEEE Transactions on Image Processing* 22, 2 (2013), 610–620. https://doi.org/10.1109/TIP.2012.2219551

[54] Margaret H. Pinson. 2005. Low Bandwidth Reduced Reference Video Quality Monitoring System.

[55] Jason J. Quinlan and Cormac J. Sreenan. 2018. Multi-Profile Ultra High Definition (UHD) AVC and HEVC 4K DASH Datasets. In *Proceedings of the 9th ACM Multimedia Systems Conference* (Amsterdam, Netherlands) *(MMSys '18)*. Association for Computing Machinery, New York, NY, USA, 375–380. https://doi.org/10.1145/3204949.3208130

[56] A. Raake, M. Garcia, W. Robitza, P. List, S. Göring, and B. Feiten. 2017. A bitstream-based, scalable video-quality model for HTTP adaptive streaming: ITU-T P.1203.1. In *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*. 1–6. https://doi.org/10.1109/QoMEX.2017.7965631

[57] Darijo Raca, Jason J. Quinlan, Ahmed H. Zahran, and Cormac J. Sreenan. 2018. Beyond Throughput: A 4G LTE Dataset with Channel and Context Metrics. In *Proceedings of the 9th ACM Multimedia Systems Conference* (Amsterdam, Netherlands) *(MMSys '18)*. Association for Computing Machinery, New York, NY, USA, 460–465. https://doi.org/10.1145/3204949.3208123

[58] Benjamin Rainer and Christian Timmerer. 2014. Quality of Experience of Web-Based Adaptive HTTP Streaming Clients in Real-World Environments Using Crowdsourcing. In *Proceedings of the 2014 Workshop on Design, Quality and Deployment of Adaptive Video Streaming* (Sydney, Australia) *(VideoNext '14)*. Association for Computing Machinery, New York, NY, USA, 19–24. https://doi.org/10.1145/2676652.2676656

[59] Rakesh Rao Ramachandra Rao, Steve Göring, Werner Robitza, Bernhard Feiten, and Alexander Raake. 2019. AVT-VQDB-UHD-1: A Large Scale Video Quality Database for UHD-1. In *2019 IEEE International Symposium on Multimedia (ISM)*. 17–177. https://doi.org/10.1109/ISM46123.2019.00012

[60] Haakon Riiser, Paul Vigmostad, Carsten Griwodz, and Pål Halvorsen. 2013. Commute Path Bandwidth Traces from 3G Networks: Analysis and Applications. In *Proceedings of the 4th ACM Multimedia Systems Conference* (Oslo, Norway) *(MMSys '13)*. Association for Computing Machinery, New York, NY, USA, 114–118. https://doi.org/10.1145/2483977.2483991

[61] Werner Robitza, Rakesh Rao Ramachandra Rao, Steve Göring, and Alexer Raake. 2021. Impact of Spatial and Temporal Information on Video Quality and Compressibility. In *2021 13th International Conference on Quality of Multimedia Experience (QoMEX)*. 65–68. https://doi.org/10.1109/QoMEX51781.2021.9465452

[62] Demóstenes Zegarra Rodríguez, Renata Lopes Rosa, and Graça Bressan. 2014. Video quality assessment in video streaming services considering user preference for video content. In *2014 IEEE International Conference on Consumer Electronics (ICCE)*. 570–571. https://doi.org/10.1109/ICCE.2014.6776137

[63] Piotr Romaniak, Mu Muy, Andreas Mauthe, Salvatore D'Antonio, and Mikołaj Leszczuk. 2011. Framework for the Integrated Video Quality Assessment. *Multimedia Tools and Applications* 61 (12 2011). https://doi.org/10.1007/s11042-011-0946-3

[64] Michele A. Saad, Alan C. Bovik, and Christophe Charrier. 2014. Blind Prediction of Natural Video Quality. *IEEE Transactions on Image Processing* 23, 3 (2014), 1352–1365. https://doi.org/10.1109/TIP.2014.2299154

[65] Sandvine. 2023. *The Global Internet Phenomena Report*. Technical Report.

[66] Y. Sani, A. Mauthe, and C. Edwards. 2015. Modelling Video Rate Evolution in Adaptive Bitrate Selection. In *2015 IEEE International Symposium on Multimedia (ISM)*. 89–94. https://doi.org/10.1109/ISM.2015.65

[67] Yusuf Sani, Darijo Raca, Jason J. Quinlan, and Cormac J. Sreenan. 2020. SMASH: A Supervised Machine Learning Approach to Adaptive Video Streaming over HTTP. In *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. 1–6. https://doi.org/10.1109/QoMEX48832.2020.9123139

[68] Kalpana Seshadrinathan and Alan Conrad Bovik. 2010. Motion Tuned Spatio-Temporal Quality Assessment of Natural Videos. *IEEE Transactions on Image Processing* 19, 2 (2010), 335–350. https://doi.org/10.1109/TIP.2009.2034992

[69] 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. https://doi.org/10.1109/TIP.2010.2042111

[70] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hobfeld, and P. Tran-Gia. 2015. A Survey on Quality of Experience of HTTP Adaptive Streaming. *Communications Surveys Tutorials, IEEE* 17, 1 (Firstquarter 2015), 469–492. https://doi.org/10.1109/COMST.2014.2360940

[71] Zaixi Shang, Joshua P. Ebenezer, Alan C. Bovik, Yongjun Wu, Hai Wei, and Sriram Sethuraman. 2021. Assessment of Subjective and Objective Quality of Live Streaming Sports Videos. arXiv:2106.08431 [eess.IV]

[72] Zaixi Shang, Joshua P. Ebenezer, Alan C. Bovik, Yongjun Wu, Hai Wei, and Sriram Sethuraman. 2022. Subjective Assessment Of High Dynamic Range Videos Under Different Ambient Conditions. In *2022 IEEE International Conference on Image Processing (ICIP)*. 786–790. https://doi.org/10.1109/ICIP46576.2022.9897940

[73] Rajiv Soundararajan and Alan C. Bovik. 2013. Video Quality Assessment by Reduced Reference Spatio-Temporal Entropic Differencing. *IEEE Transactions on Circuits and Systems for Video Technology* 23, 4 (2013), 684–694. https://doi.org/10.1109/TCSVT.2012.2214933

[74] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman. 2016. BOLA: Near-optimal bitrate adaptation for online videos. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*. 1–9. https://doi.org/10.1109/INFOCOM.2016.7524428

[75] Akira Takahashi, David Hands, and Vincent Barriac. 2008. Standardization activities in the ITU for a QoE assessment of IPTV. *IEEE Communications Magazine* 46, 2 (2008), 78–84. https://doi.org/10.1109/MCOM.2008.4473087

[76] Babak Taraghi, Abdelhak Bentaleb, Christian Timmerer, Roger Zimmermann, and Hermann Hellwagner. 2021. Understanding Quality of Experience of Heuristic-Based HTTP Adaptive Bitrate Algorithms. In *Proceedings of the 31st ACM Workshop on Network and Operating Systems Support for Digital Audio and Video* (Istanbul, Turkey) *(NOSSDAV '21)*. Association for Computing Machinery, New York, NY, USA, 82–89. https://doi.org/10.

1145/3458306.3458875

[77] Babak Taraghi, Minh Nguyen, Hadi Amirpour, and Christian Timmerer. 2021. Intense: In-Depth Studies on Stall Events and Quality Switches and Their Impact on the Quality of Experience in HTTP Adaptive Streaming. *IEEE Access* 9 (2021), 118087–118098. https://doi.org/10.1109/ACCESS.2021.3107619

[78] 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* (Istanbul, Turkey) *(MMSys '20)*. Association for Computing Machinery, New York, NY, USA, 349–352. https://doi.org/10.1145/3339825.3393581

[79] Huyen T. T. Tran, Duc Nguyen, and Truong Cong Thang. 2020. An Open Software for Bitstream-Based Quality Prediction in Adaptive Video Streaming. In *Proceedings of the 11th ACM Multimedia Systems Conference* (Istanbul, Turkey) *(MMSys '20)*. Association for Computing Machinery, New York, NY, USA, 225–230. https://doi.org/10.1145/3339825.3394925

[80] Mario Vranješ, Snježana Rimac-Drlje, and Krešimir Grgić. 2013. Review of objective video quality metrics and performance comparison using different databases. *Signal Processing: Image Communication* 28, 1 (2013), 1–19. https://doi.org/10.1016/j.image.2012.10.003

[81] Phong V. Vu, Cuong T. Vu, and Damon M. Chandler. 2011. A spatiotemporal most-apparent-distortion model for video quality assessment. In *2011 18th IEEE International Conference on Image Processing*. 2505–2508. https://doi.org/10.1109/ICIP.2011.6116171

[82] Florian Wamser, Michael Seufert, Pedro Casas, Ralf Irmer, Phuoc Tran-Gia, and Raimund Schatz. 2015. YoMoApp: A tool for analyzing QoE of YouTube HTTP adaptive streaming in mobile networks. In *2015 European Conference on Networks and Communications (EuCNC)*. 239–243. https://doi.org/10.1109/EuCNC.2015.7194076

[83] Haiqiang Wang, Weihao Gan, Sudeng Hu, Joe Yuchieh Lin, Lina Jin, Longguang Song, Ping Wang, Ioannis Katsavounidis, Anne Aaron, and C.-C. Jay Kuo. 2016. MCL-JCV: A JND-based H.264/AVC video quality assessment dataset. In *2016 IEEE International Conference on Image Processing (ICIP)*. 1509–1513. https://doi.org/10.1109/ICIP.2016.7532610

[84] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612. https://doi.org/10.1109/TIP.2003.819861

[85] Z. Wang, E.P. Simoncelli, and A.C. Bovik. 2003. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, Vol. 2. 1398–1402 Vol.2. https://doi.org/10.1109/ACSSC.2003.1292216

[86] Sarah Wassermann, Pedro Casas, Michael Seufert, and Florian Wamser. 2019. On the Analysis of YouTube QoE in Cellular Networks through in-Smartphone Measurements. In *2019 12th IFIP Wireless and Mobile Networking Conference (WMNC)*. 71–78. https://doi.org/10.23919/WMNC.2019.8881828

[87] Stefan Winkler. 2005. *Digital Video Quality – Vision Models and Metrics*. https://doi.org/10.1002/9780470024065

[88] Stefan Winkler. 2009. Video quality measurement standards — Current status and trends. In *2009 7th International Conference on Information, Communications and Signal Processing (ICICS)*. 1–5. https://doi.org/10.1109/ICICS.2009.5397585

[89] Hong Ren Wu, K. Rao, and Ashraf Kassim. 2007. Digital Video Image Quality and Perceptual Coding. *Journal of Electronic Imaging - J ELECTRON IMAGING* 16 (01 2007). https://doi.org/10.1117/1.2778686

[90] Praveen Kumar Yadav, Abdelhak Bentaleb, May Lim, Junyi Huang, Wei Tsang Ooi, and Roger Zimmermann. 2021. *Playing Chunk-Transferred DASH Segments at Low Latency with QLive*. Association for Computing Machinery, New York, NY, USA, 51–64. https://doi.org/10.1145/3458305.3463376

[91] 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 Special Interest Group on Data Communication* (London, United Kingdom) *(SIGCOMM '15)*. ACM, New York, NY, USA, 325–338. https://doi.org/10.1145/2785956.2787486

[92] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli. 2015. A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP. *SIGCOMM Comput. Commun. Rev.* 45, 4 (Aug. 2015), 14 pages.

[93] Zhenqiang Ying, Maniratnam Mandal, Deepti Ghadiyaram, and Alan Bovik. 2021. Patch-VQ: 'Patching Up' the Video Quality Problem. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. https://doi.org/10.1109/cvpr46437.2021.01380

[94] Xiangxu Yu, Neil Birkbeck, Yilin Wang, Christos G. Bampis, Balu Adsumilli, and Alan C. Bovik. 2021. Predicting the Quality of Compressed Videos With Pre-Existing Distortions. *IEEE Transactions on Image Processing* 30 (2021), 7511–7526. https://doi.org/10.1109/TIP.2021.3107213

[95] A. H. Zahran, D. Raca, and C. Sreenan. 2018. ARBITER+: Adaptive Rate-Based InTElligent HTTP StReaming Algorithm for Mobile Networks. *IEEE Transactions on Mobile Computing* (2018), 1–1. https://doi.org/10.1109/TMC.2018.2825384

[96] Fan Zhang and David R. Bull. 2016. A Perception-Based Hybrid Model for Video Quality Assessment. *IEEE Transactions on Circuits and Systems for Video Technology* 26, 6 (2016), 1017–1028. https://doi.org/10.1109/TCSVT.2015.2428551

[97] Fan Zhang, Felix Mercer Moss, Roland Baddeley, and David R. Bull. 2018. BVI-HD: A Video Quality Database for HEVC Compressed and Texture Synthesized Content. *IEEE Transactions on Multimedia* 20, 10 (2018), 2620–2630. https://doi.org/10.1109/TMM.2018.2817070

[98] Wei Zhou, Xiongkuo Min, Hong Li, and Qiuping Jiang. 2022. A Brief Survey on Adaptive Video Streaming Quality Assessment. *J. Vis. Comun. Image Represent.* 86, C (jul 2022), 7 pages. https://doi.org/10.1016/j.jvcir.2022.103526

A DASHRESTREAMER: EXAMPLE OF USE

There are several options available to run DashReStreamer, either directly through the command line or using a configuration file. For command line use, Table 11 depicts the supported options for running the framework.

Table 11. Options for running QoE framework

Parameter	Description
parameter_type	Flag indicating use of command line arguments or config file
path_to_log	Location of video log
rep_lvl_col	Column name used in video log for bitrate
seg_index_col	Column name used in video log for segment index
stall_dur_col	Column name used in video log for stall duration
chunk_dur_col	Column name used in video log for segment duration
height_col	Column name used in video log for resolution height
log_separator	Separator used in video log (example: tab)
config_path	Location of config file
path_video	Location of video segments
path_audio	Location of audio segments
gif_path	Location of gif file
log_location	Flag indicating location of segments (local or remote)
dest_video	Location where to save intermediate files during processing (segments)
final_path	Location where final concated video is saved
auto_scale	Options for enabling auto-scaling of segment resolution
scale_res	Rescaling segments to predetermined resolution (example: 1080p)
calculate_metrics	True or false flag indicating whether objective metrics should be calculated
merge_video	True or false flag indicating whether separate segments should be merged into final video
cleanup	True or false flag indicating removal of intermediate files (segments)

Case #1: For segment files stored locally, the command outlined in Listing 1 produces a video file based on the video log file, calculates objective metrics and deletes all intermediate files.

The depicted example in Listing 1 utilises the open-source movie Sintel, filters segment qualities used by adaptation algorithm outlined by video log file (video_log.log file), re-creates video sequence adding stall events (with the rebuffering image) and saves the output to the final folder. This command retains native resolution for each segment causing a visual change in the aspect ratio when the segments of the video switch from one resolution to another. Alternatively, we can mandate that all segments have the same output resolution through the option of autoscaling. We support two types of autoscaling: scaling to the highest resolution observed in the log file, or scaling to predetermined resolution given by parameter *scale_res*. The Listing 2 example shows how to create an output video file with a fixed 1080p resolution for all segments, where objective metrics are not calculated and intermediate segments are not deleted.

```
# python video_log_merger.py --path_to_log video_log.log
```

```
--rep_lvl_col Rep_Level
--seg_index_col Chunk_Index
--log_separator tab
--stall_dur_col Stall_Dur
--path_video ./sintel/DASH_Files/full/
--dest_video ./tmp_files/
--path_audio ./sintel/DASH_Files/audio/full/
--gif_path ./gif.gif
--final_path ./final/ --parameter_type path
--merge_video True
--calculate_metrics True
--cleanup True
```

Listing 1. Example of creating video from local segments

Case #2: Creating video file with same predetermined resolution is depicted in Listing 2.

```
# python video_log_merger.py --path_to_log video_log.log
--rep_lvl_col Rep_Level
--seg_index_col Chunk_Index
--log_separator tab
--stall_dur_col Stall_Dur
--path_video ./sintel/DASH_Files/full/
--dest_video ./tmp_files/
--path_audio ./sintel/DASH_Files/audio/full/
--gif_path ./gif.gif
--final_path ./final/ --parameter_type path
--scale_resolution 1080p
--auto_scale 2
--merge_video True
--calculate_metrics False
--cleanup False
```

Listing 2. Example of creating video with same resolution for all segments

Similar to Listing 1, we recreate an output video clip from the video log file, with the difference that we scale each segment to a Full HD resolution. This option is achieved by setting auto_scale to 2 (where we have three supported values 0, 1, 2), and setting scale_res to 1080p.

The DashReStreamer framework also supports the use of a configuration file as input to the python script. Listing 3 illustrates an example of a configuration file. Note that all the input parameters are the same as the parameters used for the command-line input.

```
[parameters]
parameters = config
path_to_log = <path>
rep_lvl_column = Rep_Level
```

```
chunk_index_column = Chunk_Index
stall_dur_column = Stall_Dur
height_col = Height
log_separator = tab
path_audio = <path to audio segments>
path_video = <path to video segments>
dest_video = <where to save/download segments>
gif_path = <path to gif file>
final_path = <where to save final video>
mpd_path = <url for mpd file>
auto_scale = 0
calculate_metrics = True
merge_video = True
log_location = local
```

Listing 3. Example of config file

When calculate_metrics parameter is set to true, then supported objective metrics are calculated for every video segment separately. Table 12 shows an example of a table exported as a result of this calculation.

Table 12. Objective metrics calculation example

Segment	**PSNR**	**SSIM**	**MSSSIM**	**VMAF_norm**	**VMAF_4k**	**VMAF_norm_phone**	**VMAF_4k_phone**
0	27.625	0.880	0.870	65.87	63.87	64.81	61.47
1	24.132	0.816	0.847	4.96	4.56	3.96	5.42
2	22.532	0.868	0.854	6.88	6.13	5.82	5.55