Title	DASH QoE performance evaluation framework with 5G datasets
Author(s)	Ul Mustafa, Raza; Islam, Md. Tariqul; Rothenberg, Christian E.; Ferlin, Simone; Raca, Darijo; Quinlan, Jason J.
Publication date	2020-11-02
Original citation	Ul Mustafa, R., Islam,M. T., Rothenberg, C. E., Ferlin, S., Raca, D. and Quinlan, J. J. (2020) 'DASH QoE Performance Evaluation Framework with 5G Datasets', 2020 16th International Conference on Network and Service Management (CNSM), Izmir, Turkey [online], 2-6 Nov. doi: 10.23919/CNSM50824.2020.9269111
Type of publication	Conference item
Link to publisher's version	https://ieeexplore.ieee.org/document/9269111 http://dx.doi.org/10.23919/CNSM50824.2020.9269111 Access to the full text of the published version may require a subscription.
Rights	**© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works**
Item downloaded from	http://hdl.handle.net/10468/10672

DASH QoE Performance Evaluation Framework with 5G Datasets

Raza Ul Mustafa
University of Campinas (UNICAMP)
Brazil
razaul@dca.fee.unicamp.br

Md Tariqul Islam
University of Campinas (UNICAMP)
Brazil
tariqsaj@dca.fee.unicamp.br

Christian Rothenberg
University of Campinas (UNICAMP)
Brazil
chesteve@dca.fee.unicamp.br

Simone Ferlin
Ericsson AB
Sweden
simone.ferlin@ericsson.com

Darijo Raca
University of Sarajevo, Sarajevo, BiH
draca@etf.unsa.ba

Jason J. Quinlan
University College Cork, Ireland
j.quinlan@cs.ucc.ie

***Abstract*—Fifth Generation (5G) networks provide high throughput and low delay, contributing to enhanced Quality of Experience (QoE) expectations. The exponential growth of multimedia traffic pose dichotomic challenges to simultaneously satisfy network operators, service providers, and end-user expectations. Building QoE-aware networks that provide run-time mechanisms to satisfy end-users' expectations while the end-to-end network Quality of Service (QoS) varies is challenging, and motivates many ongoing research efforts. The contribution of this work is twofold. Firstly, we present a reproducible data-driven framework with a series of pre-installed Dynamic Adaptive Streaming over HTTP (DASH) tools to analyse state-of-art Adaptive Bitrate Streaming (ABS) algorithms by varying key QoS parameters in static and mobility scenarios. Secondly, we introduce an interactive Binder notebook providing a live analytical environment which processes the output dataset of the framework and compares the relationship of five QoE models, three QoS parameters (RTT, throughput, packets), and seven video KPIs.**

***Index Terms*—5G, QoE , QoS, ABS algorithm, DASH**

I. Introduction

5G is expected to support significantly high bandwidth content with speeds in excess of 10 GB/s, very low (i.e. 1-millisecond) end-to-end over-the-air latency, real-time information transmission, and lower network management operation complexity [1]. The key challenge of streaming video in 5G is soothing the juxtaposition of the increased growth of multimedia traffic and user satisfaction. On average, multimedia users spend six hours a day watching different streaming content[1]. Furthermore, the recent coronavirus (COVID-19) pandemic has dramatically increased the amount of video streaming in 2020 [2].

The impact of end-user QoE for multimedia traffic ultimately depends on underlying network-level Quality of Service (QoS) performance. QoE represents the user perception on the quality of a provided service whereas QoS relates to network quality indicatores (e.g., latency, packet loss). In HTTP Adaptive Streaming (HAS), the choice of the Adaptive Bitrate Streaming (ABS) algorithm plays a significant role in end-user satisfaction [3]. In recent years, the goal of many ABS algorithms is to provide interrupt-free videos and hence provide maximum achievable video quality. These ABS algorithms works on the principal by calculating network condition and utilize the maximum resources thus provide better video quality during a video session. Comparing different ABS algorithms is a non-trivial task, some algorithms focus on smooth streaming, resulting in lower bitrate and fewer quality switching occurs. Other algorithms aim to provide high quality content, utilizing more network resources, irrespective to the number of stalls (freezing). Ultimately, the main goal of all ABS algorithms is to provide best the QoE to end-users.

The exponential growth of mobile data and smart devices, the investigation of 5G QoE in terms of video quality assessment has become a research focus both in industry and academia. Video perceived quality in 5G network is critical and various methods have been used to optimize video delivery over 5G networks such as video compression and better resource utilization [4], [5]. In 5G/future networks QoE is management is crucial as the estimation and resource allocation for better video quality should be completed quickly. Although 5G networks are still at conceptual stage, it is necessary to understand the correlation between ABS behaviour, its metrics for QoE and network-level QoS.

The contributions of the framework presented in this paper are divided into two phases:

Phase 1 presents a multi-user reproducible framework containing (i) goDASH - an ABS video player [6], (ii) Caddy - a WSGI web server hosting DASH video content, (iii) Mininet-Wifi - a wireless network emulation environment [7], (iv) Scripts - Bash scripts to apply the 5G bandwidth values sampled from the 5G traces at run-time; and Python scripts to process the per segment QoE/QoS logs created during experimentation and generate the associated datasets.

[1]https://www.nielsen.com/us/en/insights/report/2018/q2-2018-total-audience-report/

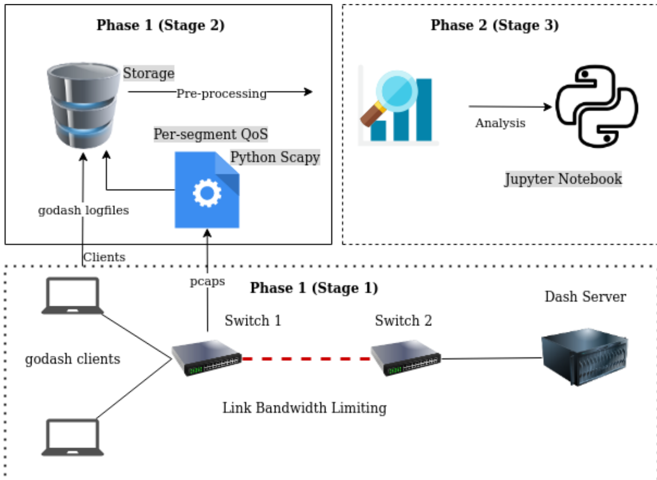


Fig. 1: Phase 1 (Stage 1), DASH streaming environment, Phase 2 (Stage 2), godash logfiles and per-segment QoS processing, Phase 2 (Stage 3), Jupyter notebook interacting with the processed dataset

Phase 2 receives as input the processed QoE/QoS dataset from the first phase and demonstrates an interactive Binder notebook to analyse the ABS algorithm with specific objective QoE KPIs, per-segment QoS features and the output of five QoE models Claey [8], Dunamu [9], Yin [10], and Yu [11] and ITU-T Rec. P.1203 standard [12], [13] (mode 0 considering metadata only, bitrate, frame rate, and resolution).The framework uses the pre-installed ABS algorithms provided by godash [6]. The available algorithms are categorised as: Rate-based — Conventional [14] and Exponential, Buffer-based — Logistic [15] and BBA [16], and Hybrid — Arbiter+ [17] and Elastic [18].

The rest of the paper is structured as follows: Section II presents background and related work. Section III describes the proposed framework in Phase 1 followed by the dynamic analysis of Phase 2 in Section IV. The experimental use case is presented in Section V. Section VI concludes our paper and discuses some future work.

II. BACKGROUND AND RELATED WORK

In adaptive streaming, a video content is split into multiple segments, with each segment having the duration of 2 to 20 seconds. Each segment is then encoded with a different video bitrate. The ABS algorithm decides about the quality of the segments to be downloaded based on the network's available resources. The structure of each media file is described in a MPD (media presentation description) file that has the information of available representations along with path to each segment available on server. When a video client wants to play a movie it first downloads a corresponding MPD file and then the client's ABS algorithm is responsible for requesting the most appropriate representation for each segment throughout the playback.

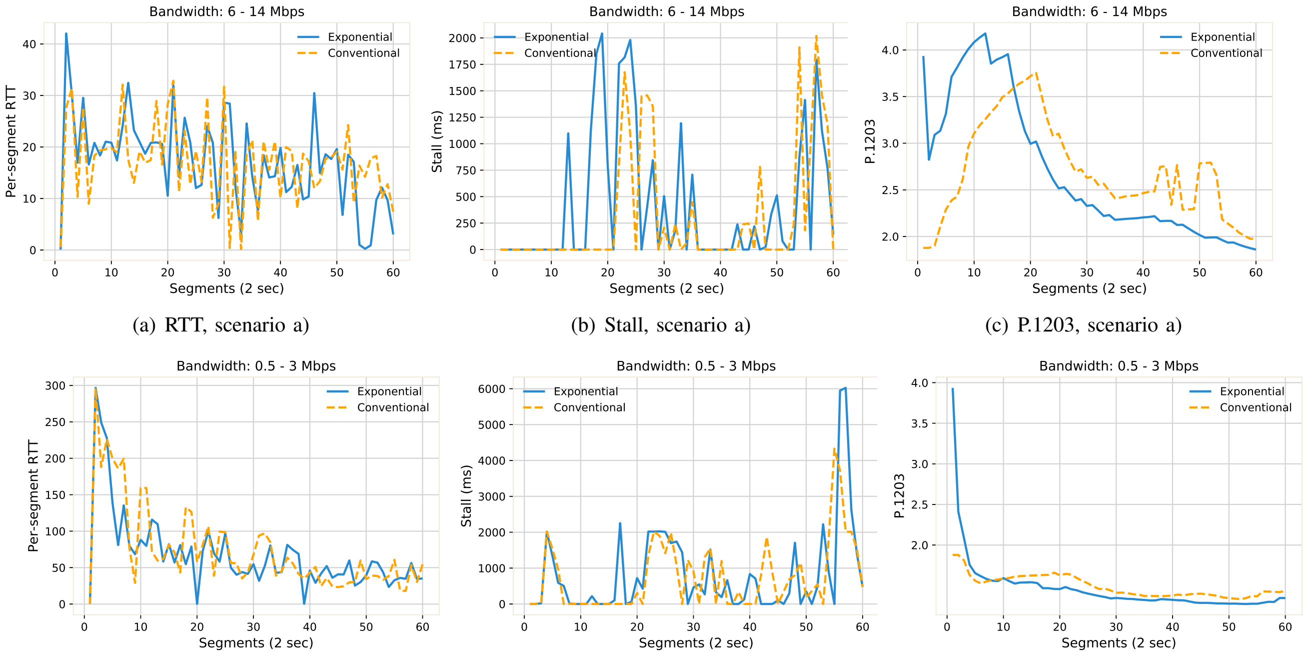
The ABS algorithms are divided into three major categories i) rate-based [19], buffer-based [3] and hybrid-based [20]. In rate-based algorithms a decision is made on the delivery rate of the previously downloaded segments. Buffer-based algorithm monitors the state of the playback buffer while in hybrid-based algorithms both playback buffer and delivery rate are considered for next segment.

Many studies have been carried out to find the key indicators for better video quality such as TCP slow-start [21] and "ON-OFF" status of HAS players [22]. Similarly, Saamer et.al, [23] evaluated two major commercial players for their findings (Smooth Streaming, Netflix) and one open source player (OSMF). Several QoE key factors have been identified such as how long a video streaming player take to converge to maximum bitrate, what happen when two adaptive video players compete for available bandwidth on a bottleneck link. Authors also point out how does adaptive streaming perform with live content. In [24] provides comprehensive comparative study of state-of-art ABS algorithms. Authors have concluded that buffer-based ABS shows better QoE as compared to rate- and hybrid-based algorithms. In another study authors evaluated both objective and subjective QoE, but they only consider throughput based algorithms [25].

We have found that many studies exist in the literature lack a comprehensive comparison of HAS algorithms. Also, many ABS algorithms are limited in their functionality as the authors have not released their framework for reproducibility. Additionally, comparison to previous studies, much attention has been given to QoE evaluation's rather than state-of-art per-segment QoS to QoE mapping. The QoS to QoE mapping is necessary to deliver more evidence-based higher quality video content through understanding how limited network resources can impact quality of experience of end-users. To fill this gap, we provide a flexible framework for analysis of DASH videos considering many combinations of real 5G traces. The framework is equipped with many other options such as the ability to change the video content for streaming, a range of different ABS algorithms to compare QoS to QoE metrics, and a rich set of different evaluations scenarios.

III. PHASE 1 - STREAMING FRAMEWORK

We present a re-producible DASH framework supporting the evaluation of six state-of-art ABS algorithms through the emulation of ten different real 5G traces to stream DASH videos. The provided tools process seven objective Key Performance Indicators (KPIs), five QoE models output (P.1203, Yin, Yu, Duanmu, Clae), and three per-segment QoS features extracted from trace files (RTT, throughput, packets). The framework encompasses a DASH streaming environment and the pre-processing of network and video client logs and associated scripts. For ease of use, the framework includes a Virtual Machine (VM) [26] with all software and dependencies installed as shown in Figure 1. The VM provides all tools and the environment needed to stream DASH content in a multi-user realistic 5G network. Currently, the VM showcases a single combination of mobility, host competition, and link bandwidth parameters to run the `Mininet-WiFi` [7] emulated topology, collect `goDASH` log(s), pcap file(s), and



(a) RTT, scenario a) (b) Stall, scenario a) (c) P.1203, scenario a)

(d) RTT, scenario b) (e) Stall, scenario b) (f) P.1203, scenario b)

Fig. 2: Rate-based, scenarios a) and b) (1st video client): RTT, stalls, and P.1203 score per video segment for 60 video segments

TABLE I: *godash* log file, First 5 video segments of 2s for case (6-14) Mbps using Conventional ABS algorithm

Seg_#	Algorithm	Seg_Dur	Codec	Width	Height	FPS	Play_Pos	RTT	P.1203	Clae	Duanmu	Yin	Yu
1	conventional	2000	H264	320	180	24	0	25.025	1.878	0.000	51.077	-5760.485	0.240
2	conventional	2000	H264	320	180	24	2000	78.83	1.878	0.480	46.477	-11520.970	0.24
3	conventional	2000	H264	384	216	24	4000	12.09	1.9	0.417	46.898	718.545	0.286
4	conventional	2000	H264	512	288	24	6000	16.86	2.106	0.314	47.826	1097.122	0.404
5	conventional	2000	H264	640	360	24	8000	74.93	2.287	0.302	48.77	1863.42	0.54

TABLE II: Processed dataset first 5 video segments of 2s for case (6-14) Mbps using Conventional ABS algorithm

Host	Stall	Bitrate	Segment	Total_Users	Buffer	RTT	Throughput	Packets	P.1203	Clae	Duanmu	Yin	Yu
1	0	8	1	2	2000	0.14	7443037.97	2	1.87	0	51.07	-5760.48	0.24
1	0	329	2	2	4000	27.65	240702.88	30	1.87	0.48	46.47	-11520.97	0.24
1	0	720	3	2	4643	31.39	280181.47	64	1.9	0.41	46.89	718.54	0.28
1	0	1408	4	2	5212	10.33	465851.21	117	2.10	0.31	47.82	1097.12	0.40
1	0	1191	5	2	5277	27.68	325186.14	104	2.28	0.30	48.77	1863.42	0.54

process the raw video logs and network data. However, the framework is versatile and can be easily modified to accommodate additional DASH algorithms, 5G traces, etc. The proposed framework provides a convenient mechanism to generate multimedia traffic processed data. Video instructions on the framework's use within the VM are available online [27].

IV. Phase 2 - Dynamic Analysis Demo

In Phase 2, we provide the processed dataset collected by running ten different combinations of real 5G traces across six state-of-art ABS algorithms. We assess the impact of concurrent video streaming clients (1,...,2), (1,...,3), (1,...,5); with all of the clients streaming from the same server.

The processed dataset generated in Phase 1 is imported and examined using a `Jupyter` notebook, a well-known Web-based interactive environment for data analyses. For ease of use, the framework integrates the provided VM with `JupyterLab`[2]. For each DASH video-segment, the `Jupyter` notebook analyses the impact of five QoE models (P.1203, Clae, Duanmu, Yin and Yu), seven video client objective KPIs (arrival time (ms), delivery time (ms), stall (ms), delivery rate of network (Kbps), segment size (bytes), bitrate (Kbps) and buffer level (s) after the segment was just downloaded, and three QoS features (derived from packet captures).

Note that the data and the notebook hosted on `Github` are in read-only (static) mode. However, a live `Binder`[3] service is available in our `GitHub` repository [28], allowing inter-

[2]https://jupyterlab.readthedocs.io
[3]https://mybinder.org

action with the read-only notebook in an executable dynamic environment. In addition, we provide a video demonstration on how to use the environment [27]. In the video, we showcase how to modify the VM generated `Jupyter` static notebook as an interactive notebook with the Binder service and how to visualize changes in the data.

V. Experimental Use Case

Figure 1 gives an overview of the two phases, divided into three stages: Phase 1 (Stage 1): data acquisition from the network interface and `godash` player; Phase 1 (Stage 2): data pre-processing from `godash` and the network and Phase 2 (Stage 3): where we use a `Jupyter` notebook to analyse and visualise the processed dataset, as shown in Figure 3.

We begin in the VM with Phase 1 (Stage 1) - logs generation. In this stage, each experiment is performed using the `Mininet-WiFi` [29] network emulator as shown in Figure 1, using the setup detailed in Figure 3. To emulate the HTTP streaming video, we use a lightweight DASH compatible video streaming tool called `godash` [6] at the host node(s) and `Caddy`, a `WSGI` web server, hosting a popular 2-second segment duration x264 animated video titled *Sintel*[4], sourced from a publicly available 4K DASH video dataset [30].

To simplify dataset generation for the article, we asses the impact of 2,3 and 5 concurrent clients streaming from the same server. The network bandwidth values are based on the 5G trace parameters [31], where we select ten different combinations[5] (in Mbps) across two scenarios: Mobility (driving) — (38.26 to 10.33), (29.33 to 10.55), (0.5 to 3) and (6 to 14), and Static — (72.42 to 9), (70 to 20), (52.06 to 0.5), (4.19 to 8), (0.5 to 6) and (8.29 to 57.15). These bandwidth combination values consist of different variations of (static and mobility) network throughput so that the video clients stream from very high bandwidth to low and moderate and vice versa. Note that the bandwidth during each experiment is changed in real-time between Switch 1 and Switch 2 link after every 4 seconds as shown in Figure 1 using Linux Traffic Control (TC) and Hierarchical Token Bucket (HTB) [32]. i.e., in each 4s sampling interval, two video segments can be downloaded before a new bandwidth value is sampled from the 5G trace files. We terminate each video session after 120 seconds thus we have 60 segments of 2 seconds each. A parametrized python script is used to collect per-run pcap by `tcpdump`[6] and corresponding godash logfiles.

Table I illustrates an example of a `godash` log file for a single client in the Mobility (driving) scenario using (6 to 14) bandwidth, with each line representing per segment metrics for the conventional ABS algorithm. Note that we also run the Exponential algorithm in these tests. Detailed information on each feature and ABS algorithms is available in `goDASH` [33].

In Phase 1 (Stage 2), a Python script is used to fetch per segment QoS metrics (RTT, Throughput and Packets) from the pcap files, merging the QoS metrics and godash logfiles output as a single CSV dataset (example presented in Table II). The features of the processed dataset are: (Host, Stall, Bitrate, Segment) are indicated as (host number, stall and bitrate (during the video segment), and segment number) followed by (Total_Users and Buffer) as (total user during the experiment and buffer level on the corresponding segment). The QoS collected from the pcap traces of each segment is indicated as (RTT, throughput, number of packets) for each segment and the output of five QoE models (P.1203, Clae, Duanmu, Yin and Yu).

Figure 2 takes information from both Table I and Table II, and depicts the RTT, stalls and P.1203 score per video segment for 60 video segments, i.e., 60×2s = 120s or 2 minutes of video, for both the conventional and exponential algorithms in the Mobility (driving) scenario using (6 to 14 Mbps) and (0.5 to 3 Mbps) bandwidth. Note that in the evaluation of the QoS impact on QoE in Figure 2, the length of the video file in the experiment is 2 minutes, 2 hosts competing for video stream considering 5G dynamic cases, the bandwidth combinations we select gradually increases from lower to upper limit[7] and 1st user experience is shown in Figure 2. It is important to note in Figure 2 that both rate-based algorithms select wrong segments to stream, which causes the stalls to appear frequently and ultimately lowering the Mean Opinion Score (MOS) as in our case P.1203. We can also observe that conventional has slightly better QoE and QoS features when compared to exponential. The QoS feature (RTT) for 60 segments is similar in both cases (6-14), (0.5-3) Mbps, as presented in Figure 2 (a) and (d). Exponential experiences more peaks of stalls see Figure 2 (b) from (10 to 35) segments ultimately causing P.1203 score to converge to lower values as per Figure 2 (c). However, in less performing networks (0.5-3) Mbps, both algorithms experience similar stall ratio, as shown in Figure 2 (e), as well as P.1203 scores with fewer jumps to higher scores, as shown in Figure 2 (f).

Moving now to Phase 2, we use the CSV dataset and create a `Jupyter` notebook. The `Jupyter` notebook and csv dataset are uploaded from the VM to `GitHub` and through a live dynamic `Binder` service, we can interact, analyse and visualise the input dataset. To visualise your own data, the easiest option is to fork our repository [28] and upload your data to the forked version of it. Figure 4 highlights the outline and design of the Binder service, while Figure 5 illustrates some of the features that can be selected to update and revise the output plots. Note that we will open source all remaining code used for processing the dataset for reproducibility after the acceptance of this paper, with all computational scripts and utilities embedded in the VM.

[4]http://cs1dev.ucc.ie/misl/4K_non_copyright_dataset/2_sec/x264/sintel/DASH_Files/full/sintel_enc_x264_dash.mpd

[5]Value combinations taken from: https://github.com/uccmisl/5Gdataset

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

[7]10 5G real cases: https://github.com/razaulmustafa852/dashframework/blob/master/5G_TC_Cases.csv

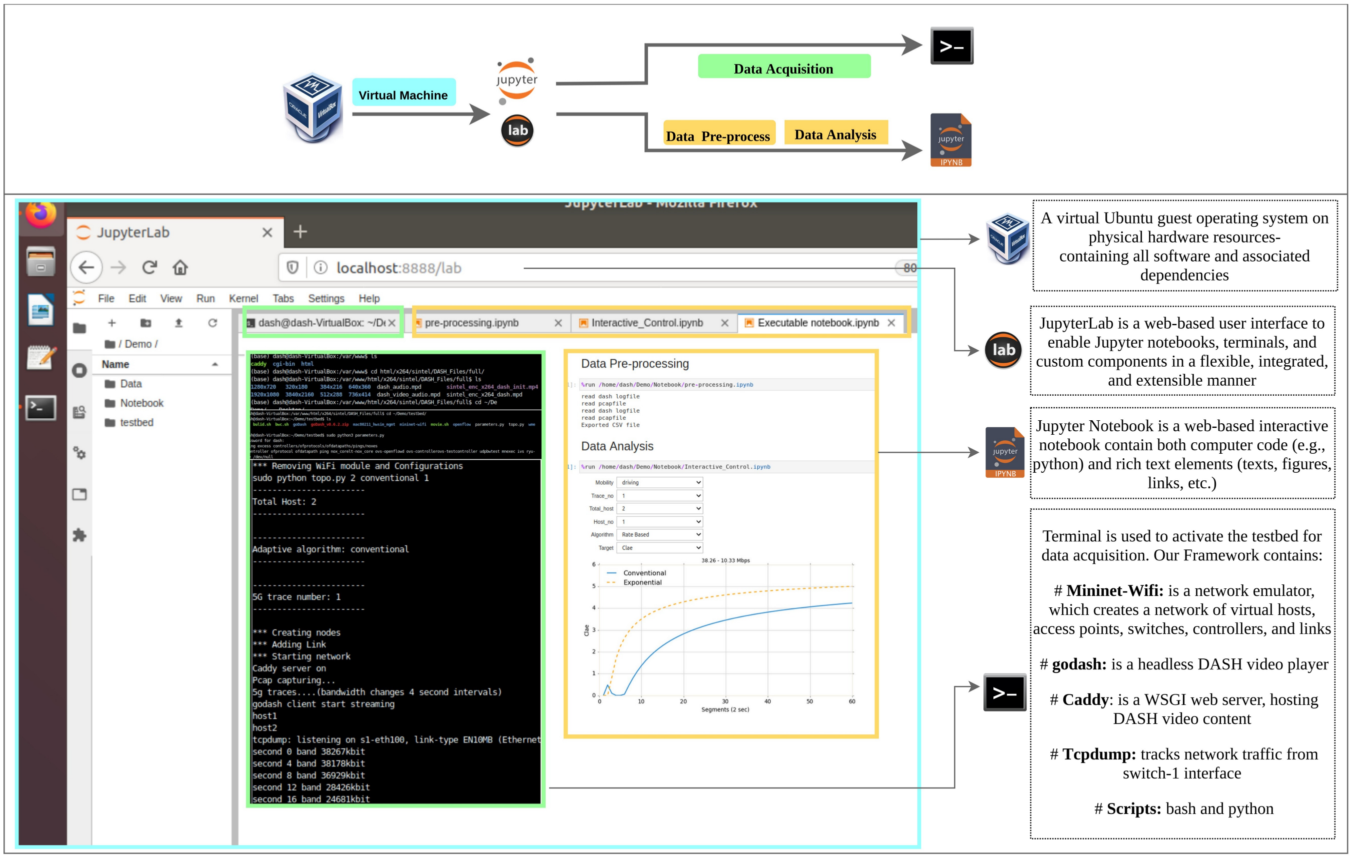


Fig. 3: An Ubuntu 18.04 VM including a DASH streaming environment containing: `Mininet-Wifi`, `Jupyter` lab and notebook, `godash` player, `Caddy` server and DASH content, `tcpdump`, and scripts

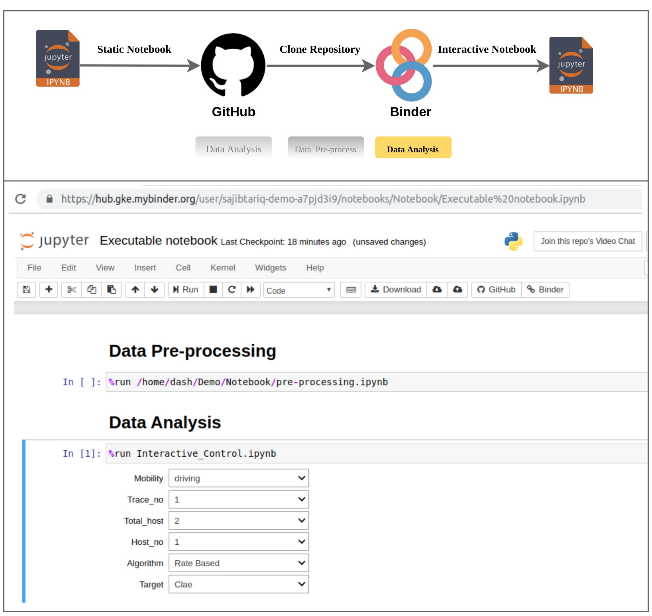


Fig. 4: `Binder`, turns the `Github` notebook into an interactive notebook in an executable environment for data analysis

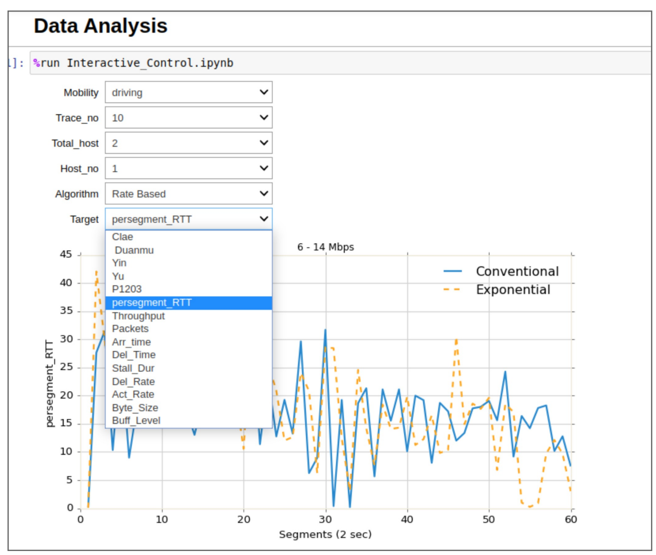


Fig. 5: First user experience (mobility) with Conventional and Exponential ABS algorithms over (6-14) Mbps. Per-segment QoS RTT on (y-axis), 60 segments on (x-axis)

VI. Conclusions

This work presents a reproducible framework which generates a QoS and QoE metric dataset for DASH experiments using different state-of-art adaptive bitrate streaming algorithms. A convenient and interactive Binder notebook is used to demonstrate live analytical environment processing the dataset output of the framework. We observe the impact of bandwidth variation on the adaptation strategies of each category of ABS algorithm (Rate-based, Buffer-based and Hybrid), discerning the relationship between network QoS metrics, video QoE models and DASH streaming KPI values. Future work includes further analysis of the impact of other QoS metrics (e.g., delay and packet loss) on HAS performance and the support of machine learning research to correlate and predict QoE based on the observed QoS.

References

[1] N. Panwar, S. Sharma, and A. K. Singh, "A survey on 5g: The next generation of mobile communication," *Physical Communication*.

[2] Conviva, "Conviva's state of streaming q1 2020," 2020.

[3] 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," in *ACM SIGCOMM Computer Communication Review*, vol. 44, pp. 187–198, ACM, 2014.

[4] J. Qiao, X. S. Shen, J. W. Mark, and L. Lei, "Video quality provisioning for millimeter wave 5g cellular networks with link outage," *IEEE Transactions on Wireless Communications*, vol. 14, no. 10, pp. 5692–5703, 2015.

[5] M. Ismail and W. Zhuang, "Statistical qos guarantee for wireless multihoming video transmission," in *2013 IEEE Global Communications Conference (GLOBECOM)*, pp. 4615–4620, IEEE, 2013.

[6] D. Raca, M. Manifacier, and J. J. Quinlan, "goDASH - GO accelerated HAS framework for rapid prototyping," in *Proceedings of the 12th International Conference on Quality of Multimedia Experience*, 2020.

[7] R. Fontes, S. Afzal, S. Brito, M. Santos, and C. Esteve Rothenberg, "Mininet-WiFi: emulating Software-Defined wireless networks," in *2nd International Workshop on Management of SDN and NFV Systems, 2015(ManSDN/NFV 2015)*, (Barcelona, Spain), Nov. 2015.

[8] S. Petrangeli *et al.*, "QoE-Driven Rate Adaptation Heuristic for Fair Adaptive Video Streaming," *ACM Trans. Multimedia Comput. Commun. Appl.*, Oct. 2015.

[9] Z. Duanmu *et al.*, "A Quality-of-Experience Database for Adaptive Video Streaming," *IEEE Transactions on Broadcasting*, vol. 64, pp. 474–487, June 2018.

[10] X. Yin *et al.*, "A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP," in *2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, 2015.

[11] L. Yu *et al.*, "QoE-Driven Dynamic Adaptive Video Streaming Strategy With Future Information," *IEEE Transactions on Broadcasting*, vol. 63, pp. 523–534, Sep. 2017.

[12] W. Robitza *et al.*, "HTTP Adaptive Streaming QoE Estimation with ITU-T Rec. P. 1203: Open Databases and Software," in *9th ACM Multimedia Systems Conference*, MMSys '18, pp. 466–471, 2018.

[13] A. Raake *et al.*, "A bitstream-based, scalable video-quality model for HTTP adaptive streaming: ITU-T P.1203.1," in *Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, May 2017.

[14] Z. Li *et al.*, "Probe and Adapt: Rate Adaptation for HTTP Video Streaming At Scale," *IEEE Journal on Selected Areas in Communications*, vol. 32, pp. 719–733, April 2014.

[15] Y. Sani *et al.*, "Modelling Video Rate Evolution in Adaptive Bitrate Selection," in *2015 IEEE International Symposium on Multimedia (ISM)*, pp. 89–94, Dec 2015.

[16] T. Huang *et al.*, "A Buffer-based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service," in *Proceedings of the 2014 ACM Conference on SIGCOMM*, SIGCOMM '14, (New York, NY, USA), pp. 187–198, ACM, 2014.

[17] A. H. Zahran *et al.*, "ARBITER+: Adaptive Rate-Based InTElligent HTTP StReaming Algorithm for Mobile Networks," *IEEE Transactions on Mobile Computing*.

[18] L. D. Cicco *et al.*, "ELASTIC: A Client-Side Controller for Dynamic Adaptive Streaming over HTTP (DASH)," in *2013 20th International Packet Video Workshop*.

[19] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive," in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, pp. 97–108, 2012.

[20] A. H. Zahran, D. Raca, and C. J. Sreenan, "Arbiter+: Adaptive rate-based intelligent http streaming algorithm for mobile networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 12, pp. 2716–2728, 2018.

[21] T.-Y. Huang, N. Handigol, B. Heller, N. McKeown, and R. Johari, "Confused, timid, and unstable: picking a video streaming rate is hard," in *Proceedings of the 2012 internet measurement conference*, pp. 225–238, 2012.

[22] S. Akhshabi, L. Anantakrishnan, A. C. Begen, and C. Dovrolis, "What happens when http adaptive streaming players compete for bandwidth?," in *Proceedings of the 22nd international workshop on Network and Operating System Support for Digital Audio and Video*, pp. 9–14, 2012.

[23] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http," in *Proceedings of the second annual ACM conference on Multimedia systems*, pp. 157–168, 2011.

[24] T. Karagkioules, C. Concolato, D. Tsilimantos, and S. Valentin, "A comparative case study of http adaptive streaming algorithms in mobile networks," in *Proceedings of the 27th Workshop on Network and Operating Systems Support for Digital Audio and Video*, pp. 1–6, 2017.

[25] C. Timmerer, M. Maiero, and B. Rainer, "Which adaptation logic? an objective and subjective performance evaluation of http-based adaptive media streaming systems," *arXiv preprint arXiv:1606.00341*, 2016.

[26] P. 1-VM, "Download link (be available soon)." https://drive.google.com/drive/folders/1y4HZ7sYxzCi__yXTpAnZwMQlQy5na04b?usp=sharing, 2020.

[27] Dashframework-video, "Demonstration videos." https://drive.google.com/drive/folders/1JayDnKF8NLneIFj1nc2CLKD7UZ0AnYWM?usp=sharing, 2020.

[28] Dashframework, "Github repository." https://github.com/sajibtariq/demo, 2020.

[29] R. R. Fontes, S. Afzal, S. H. Brito, M. A. Santos, and C. E. Rothenberg, "Mininet-wifi: Emulating software-defined wireless networks," in *2015 11th International Conference on Network and Service Management (CNSM)*, pp. 384–389, IEEE, 2015.

[30] J. J. Quinlan *et al.*, "Multi-profile Ultra High Definition (UHD) AVC and HEVC 4K DASH Datasets," in *9th ACM MMSys Conference*.

[31] D. Raca, D. Leahy, C. J. Sreenan, and J. J. Quinlan, "Beyond throughput, the next generation: a 5g dataset with channel and context metrics," in *Proceedings of the 11th ACM Multimedia Systems Conference*, pp. 303–308, 2020.

[32] D. G. Balan and D. A. Potorac, "Linux htb queuing discipline implementations," in *2009 First International Conference on Networked Digital Technologies*, pp. 122–126, IEEE, 2009.

[33] D. Raca, M. Manifacier, and J. J. Quinlan, "godash-go accelerated has framework for rapid prototyping," 2020.