Machine Learning Approach to Estimate Video QoE of Encrypted DASH Traffic in 5G Networks

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

David Moura
Centro Tecnológico do Exército (CTEx)
Brazil
david.moura@eb.mil.br

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

***Abstract*—5G communication technologies promise reduced latency and increased throughput, among other features. The so-called enhanced Mobile Broadband (eMBB) type of services will contribute to further adoption of video streaming services. In this work, we use a realistic emulation environment based on 5G traces to investigate how Dynamic Adaptive Streaming over HTTP (DASH) video content using three state-of-art Adaptive Bitrate Streaming (ABS) algorithms is impacted in static and mobility scenarios. Given the wide adoption of end-to-end encryption, we use machine learning (ML) models to estimate multiple key video Quality of Experience (QoE) indicators (KQIs) taking network-level Quality of Service (QoS) metrics as input features. The proposed feature extraction method does not require chunk-detection, significantly reducing the complexity of the monitoring approach and providing new means for QoE evaluation of HAS protocols. We show that our ML classifiers achieve a QoE prediction accuracy above 91%.**

***Index Terms*—5G, QoE, TLS, machine learning, QoS, HTTPS, DASH, HAS**

I. Introduction

Video content providers such as YouTube, Netflix, Amazon Prime, and Hulu use HTTP adaptive streaming (HAS) with HTTPS to deliver end-to-end encrypted video streaming services [2]. Cisco predicts that the HAS traffic will be at the top of traffic load with 82% from all Internet traffic by 2022 [1]. This increase in HAS traffic has opened many research dimensions. While encryption is necessary, network providers face many challenges in monitoring and managing their network resources because encryption limits their visibility to Quality of Experience (QoE) and Quality of Service (QoS) metrics. In addition, traditional network monitoring techniques rely on Deep Packet Inspection (DPI) to assess Key Performance Indicators (KPIs). However, network operators have limited power for traffic inspection in current network scenarios due to Transport Layer Security (TLS) encryption mechanisms, which provide a secure and private connection with end-to-end encryption [1].

Recently, QoS to QoE mapping has received a lot of attention [3, 13, 14]. To this end, methods based on machine learning offer promising avenues for QoE inference based on network-level QoS metrics. In this work, we examine encrypted QoS features derived from real 5G network traces to estimate QoE indicators. We select three state-of-the-art Adaptive Bitrate Streaming (ABS) algorithms for video quality adaptation, namely: (*i*) Hybrid – Elastic, (*ii*) Buffered – BBA, and (*iii*) Rate-based – Conventional [11]. The main contributions of this work can be therefore summarized as:

- QoE assessment in 500 ms time window with varying bandwidth in static and mobile 5G scenarios. This is the smallest granularity proposed so far for the detection of anomaly and troubleshooting approaches. The analysis is undertaken through objective QoE models such as the P.1203 QoE standard [8, 6].
- A proposal of a machine learning classifier to estimate QoE based on packets length distribution into (10-90) percentile in 0.5 s intervals. Moreover, the classifiers are unaware of the specific ABS algorithm and 5G scenarios, using only network QoS metrics (throughput and packets) and not requiring any chunk detection.

II. Related Work

Previous works support that stalls, resolutions and bitrate are the main reasons that affect end users QoE [10]. However, other factors cannot be ignored as well, such as ABS adaptation mechanisms. Similarly, it has also been observed that continuous quality switching is also a relevant QoE factor [4].

In a recent study, a research based on YouTube QoE prediction used packet level information such as packet size, arrival time, and packet length [14]. Another research conducted on cellular networks considering same sorts of packet level information and KPIs such as stall and bitrate etc [3]. In another study authors used objective QoE metrics of YouTube to map user level QoE such as number of stalling events, total stalling time, initial delay [13]. Most of research used YouTube as a reference to conduct their respective studies ignoring different adaptation algorithms for different types of services. Thus, in our previous work we find QoS features such as per-segment (RTT, throughput, packets) that can be mapped to objective QoE metrics [11, 9]. but, many techniques become challenging when traffic is encrypted with HTTPS.

In this work, we consider only packets level statistics to find QoE class from encrypted video stream. In contrast to our preliminary work, here, we present a QoS feature extraction method and video-QoE prediction with varying bandwidth in static and mobile 5G scenarios.

III. Approach

Our primary purpose is to introduce methods that make it easier to estimate QoE from encrypted QoS features. In Algorithm 1, we present a method to extract QoS features from packet captures (pcap traces). We first initialize three $Arrays$, A, C and QoS, to store packets length, packets time and QoS features in a 0.5 s stream. We iterated from $Time$ 0 to entire video session S with a step size of 0.5 s. We saved packets length and packets time in $Arrays$ A and C. Next ($lines$ $9-14$), we extract packets from $Array$ A, and check if packets length greater than 100, then increment the counter variable gt_100. In addition to that, on $line$ 13, we convert packets length to bits/s. We compute all these features until $Array$ A total values. On line 15, we compute throughput in 0.5 s. Next ($lines$ $16-18$), we convert packets length A into the distribution of $(10-90)$ $percentile$ for 0.5-time. On $line$ 19, we compute the average time between packets followed by average time of packets length greater than 100 as AT_gt100. Finally we have total packets in $Array$ A on $line$ 21. On $line$ 22, the resulting fourteen QoS features stored in $Array$ QoS and used as inputs to machine learning classifiers to estimate QoE into three classes (Bad, Average, Good). For QoE labeling, we leverage video player (*goDASH*) logs at 0.5 s time slots and take aggregated values of P.1203, stall, bitrate, resolutions which were running at that moment using arrival and delivery of segment feature available in the *goDASH* log files. Then, we define output values based on QoE model P.1203 classes, namely, `Poor` if the output value is between 0 and 2, `Average`, if it fits between 2 and 3, and `Good` if the observed value ranges between 3 and 5.

Algorithm 1 Features extraction method

1: $S = 120s$
2: $A = Array\ ()$
3: $C = Array\ ()$
4: $QoS = Array\ ()$
5: $gt\_100 = 0$ ▷ Packet length greater than 100
6: **for** $Time = 0$ To S STEP + $0.5seconds$ **do**
7: $A() = Array\ of\ (Packets\ length\ in\ 0.5s)$
8: $C() = Array\ of\ (Packets\ time\ in\ 0.5s)$
9: **for** $E = 0$ To $Count\ of\ A$ **do**
10: **if** $Length\ in\ A[E] > 100$ **then** ▷ Packet length
11: gt_100 = gt_100 +1
12: **end if**
13: $TP = TP + (A[E] * 8)$ ▷ Throughput in 0.5s as bits/s
14: **end for**
15: $TP = TP/0.5s$ ▷ Throughput in 0.5s
16: **for** P = 10 To 90 **do**
17: P percentile of A
18: **end for**
19: AT = Average-time of packets in A
20: AT_gt100 = Average-time of packets in A greater than 100
21: T = Total packets as $Count\ of\ array\ A$
22: $QoS = (10-90)P, gt\_100, T, AT, TP, AT\_gt100$
23: **end for**

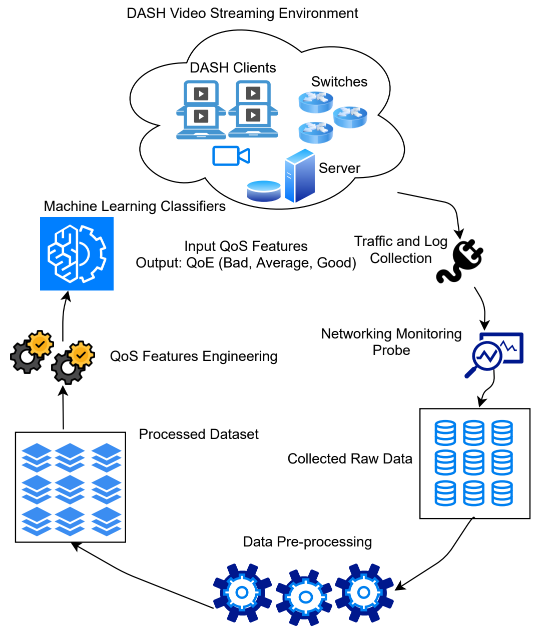


Fig. 1: QoS features extraction and ML steps.

IV. Experimental Setup

The complete setup consists of topology with 2 Open vSwitches (switch 1 and switch 2). A DASH client streaming from the server (Caddy) [1] hosting DASH video named *Sintel* encoded in eight different resolutions. For emulation, we used *Mininet* [2] and *goDASH* [7] an-open source DASH video player. To change bandwidth on the link between switch 1 and switch 2 we use Linux traffic control (TC) [3] - a traffic controller in the Linux kernel. Figure 1 illustrates all of the methodological steps used during experiments.

We use 2-second segment duration x264 animated video titled *Sintel*, sourced from a publicly available 4K DASH video dataset [5]. Total video length is 14 minutes, however we stream 2 minutes i.e, 2x60 (segments)=120s. We use three different state-of-art ABS algorithms (*i*) Hybrid – Elastic, (*ii*) Buffered – BBA, and (*iii*) Rate-based – Conventional, representing the different categories of ABS for QoE evaluation and prediction. The network bandwidth values are based on the 5G trace parameters [11]. We design a bash script that read 5G trace value from excel sheet and changes downlink bandwidth parameter after every 4 seconds using Linux TC (Hierarchical Token Bucket) between switch 1 and switch 2. We design two cases of 5G mobility, i) case with low to high bandwidth, ii) a combination of high and low bandwidth traces.[4]. In static cases we keep bandwidth a combination of both high and low bandwidth traces. The motivation to select 4s to change bandwidth is to analyze the behavior of two

[1] https://caddyserver.com/
[2] http://mininet.org/
[3] https://linux.die.net/man/8/tc
[4] https://github.com/razaulmustafa852/encrypted

consecutive segments downloaded. Moreover, the mobility case with low bandwidth traces are used to observe stall, quality switching and other objective KPIs.

V. QoE Analysis

In this section, we present objective QoE (resolutions) and QoS metrics (throughput in bits per second) analyses. it is worthwhile noting that the throughput presented in Figure 3 is calculated by each time slot (0.5s) in a sequential manner.

In scenario (a) (smaller QoS), *Conventional* is a combination of different resolution fluctuations, as presented in Figure 2 (a). In *Elastic* and *BBA* cases, we see a consistent behaviour and less quality switching, as shown in Figure 2 (b, c). For all three ABS types in scenario (a) , *Conventional* presents a better throughput. However, *Elastic* and *BBA* present an opposite effect, i.e., less resources availability driving to less throughput, while a higher resources level allows better throughput, as we can see in Figure 3 (a, b).

For the other hand, in scenario (b), focused in higher QoS, segments in *Conventional* prefer low resolution at the beginning, leading into resolution increase over time. Additionally, we have identified *Conventional* preference on higher resolution settings, as presented in Figure 2 (d), and high throughput, as shown in Figure 3 (b).

Finally, we can see in Figure 2 (d) that all segments remained in 1920x1080 mode after (7-8 segments), while the resolution pattern remained as the same in *Elastic* case, as shown in Figure 2 (e). Accordingly, we see a lot of consistency in the resolution for the *BBA* case, as we can see in Figure 2 (f).

VI. Results

For a comprehensive analysis of the dataset, requiring less computational overhead during the pre-processing phases, such as scaling and normalization, we selected Random Forests (RF), k-nearest neighbors (KNN) and Artificial Neural Network (ANN) for classification. We show 5-folds — ANN (best classifier) results in Table I for case static, mobility and for case when we input both static and mobility QoS features to model presented as All.

TABLE I: 5-folds accuracy for case Mobility and Static

K-folds	Mobility %	Static %	All %
1^{st}	0.96	0.91	0.91
2^{nd}	0.96	0.88	0.91
3^{rd}	0.95	0.91	0.91
4^{th}	0.96	0.87	0.91
5^{th}	0.95	0.89	0.92

VII. Conclusion

In this paper, we present a methodology for building a prediction model on encrypted DASH video replaying 5G traces. We consider state-of-art Adaptive Bitrate Streaming (ABS) algorithms commonly found in HTTP Adaptive Streaming (HAS) and stream video content by varying the network quality-of-service (bandwidth) sampled from real 5G traces. We applied different machine-learning classifiers on the dataset that can accurately estimate QoE classes (Poor, Average, Good) derived from the ITU-T Rec. P.1203 QoE standard mode 0 considering metadata only, bitrate, frame rate, and resolution. We find that packet level statistics can be effectively used to estimate QoE from QoS metrics. In future work, we are interested to investigate stall prediction as well as the impact of both bandwidth and throughput, i.e. when both values change simultaneously. In addition, we would like to investigate the performance of multiple DASH client streaming from the same server. Finally, we intend to update our publicly available DASH QoE evaluation framework [12] featuring an interactive Jupyter notebook and Binder service to reproduce the experiments presented in this work among other related research results.

References

[1] Abdelhak Bentaleb, Saad Harous, et al. "Inferring Quality of Experience for Adaptive Video Streaming over HTTPS and QUIC". In: *2020 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE. 2020, pp. 81–87.

[2] Abdelhak Bentaleb et al. "A survey on bitrate adaptation schemes for streaming media over HTTP". In: *IEEE Communications Surveys & Tutorials* 21.1 (2018), pp. 562–585.

[3] Giorgos Dimopoulos et al. "Measuring video QoE from encrypted traffic". In: *Proceedings of the 2016 Internet Measurement Conference*. 2016, pp. 513–526.

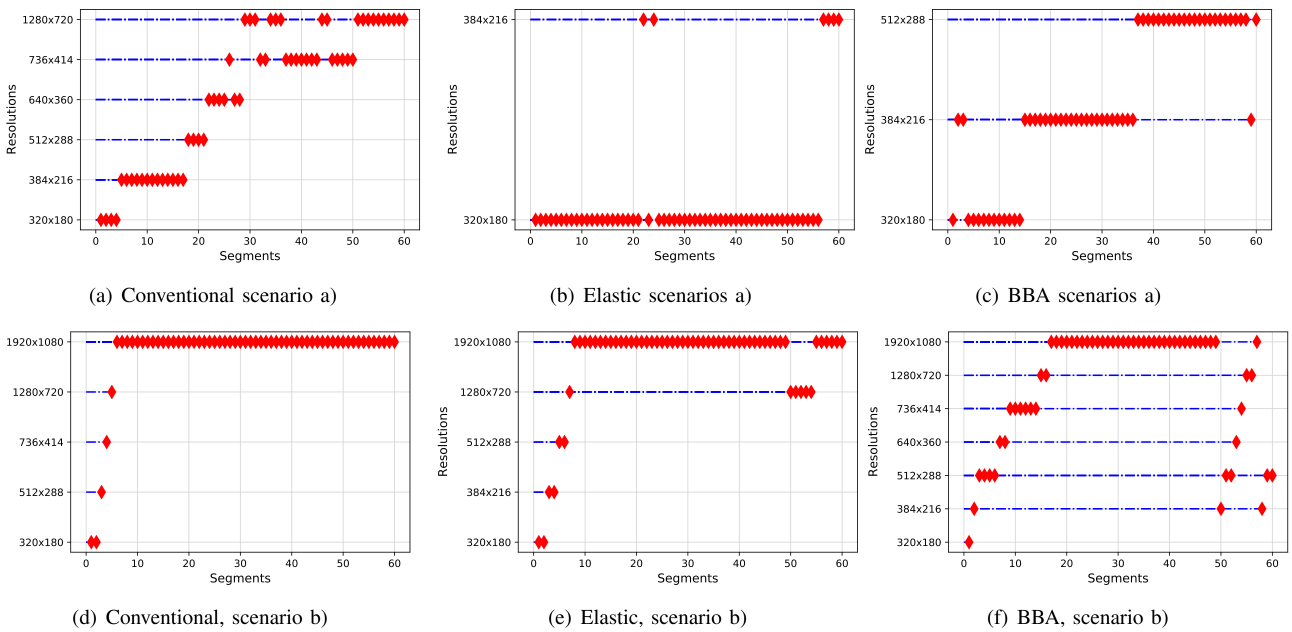
[4] Blazej Lewcio et al. "Video quality in next generation mobile networks—Perception of time-varying transmission". In: *2011 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR)*. IEEE. 2011, pp. 1–6.

[5] Jason J. Quinlan et al. "Multi-profile Ultra High Definition (UHD) AVC and HEVC 4K DASH Datasets". In: *9th ACM MMSys Conference*. Amsterdam, Netherlands. ISBN: 978-1-4503-5192-8. DOI: 10 . 1145 / 3204949.3208130.

[6] Alexander 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. ISBN: 978-1-5386-4024-1. DOI: 10 . 1109 / QoMEX.2017.7965631.

[7] Darijo Raca, Maëlle Manifacier, and Jason J Quinlan. "goDASH-GO accelerated HAS framework for rapid prototyping". In: (2020).

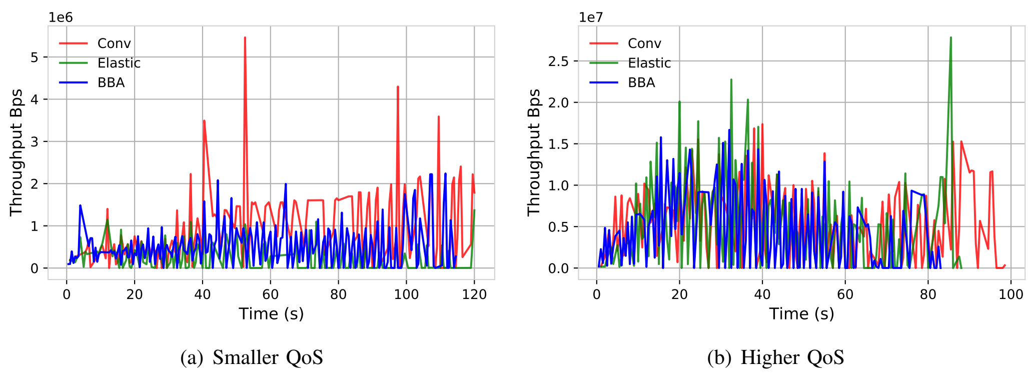
[8] Werner Robitza et al. "HTTP Adaptive Streaming QoE Estimation with ITU-T Rec. P. 1203: Open Databases and Software". In: *9th ACM Multimedia Systems Conference*. MMSys '18. Amsterdam, Netherlands, 2018, pp. 466–471. ISBN: 978-1-4503-5192-8. DOI: 10.1145/3204949.3208124.



(a) Conventional scenario a) (b) Elastic scenarios a) (c) BBA scenarios a)

(d) Conventional, scenario b) (e) Elastic, scenario b) (f) BBA, scenario b)

Fig. 2: Bandwidth, scenarios a) (smaller) and b) (higher) QoS bandwidth: Resolutions change for 60 video segments



(a) Smaller QoS (b) Higher QoS

Fig. 3: Throughput of 60 segments stream for both (smaller, higher) bandwidth scenario using three ABS

[9] C. E. Rothenberg et al. "Intent-based Control Loop for DASH Video Service Assurance using ML-based Edge QoE Estimation". In: *2020 6th IEEE Conference on Network Softwarization (NetSoft)*. 2020, pp. 353–355.

[10] Michael Seufert et al. "A survey on quality of experience of HTTP adaptive streaming". In: *IEEE Communications Surveys & Tutorials* 17.1 (2014), pp. 469–492.

[11] Raza Ul Mustafa et al. "A Supervised Machine Learning Approach for DASH Video QoE Prediction in 5G Networks". In: *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*. 2020, pp. 57–64.

[12] Raza Ul Mustafa et al. "DASH QoE Performance Evaluation Framework with 5G Datasets". In: *2020 16th International Conference on Network and Service Management (CNSM)*. 2020, pp. 1–6. DOI: 10.23919/CNSM50824.2020.9269111.

[13] Sarah Wassermann, Nikolas Wehner, and Pedro Casas. "Machine learning models for YouTube QoE and user engagement prediction in smartphones". In: *ACM SIGMETRICS Performance Evaluation Review* 46.3 (2019), pp. 155–158.

[14] Sarah Wassermann et al. "ViCrypt to the Rescue: Real-time, Machine-Learning-driven Video-QoE Monitoring for Encrypted Streaming Traffic". In: *IEEE Transactions on Network and Service Management* (2020).