UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Raza Ul Mustafa

Machine Learning Assisted DASH Video QoE Inference Through Network QoS Features in 4G and 5G Scenarios

Inferência de QoE de Vídeo DASH assistida por aprendizado de máquina usando métricas de QoS de rede em cenários 4G e 5G

Campinas
2022

UNIVERSIDADE ESTADUAL DE CAMPINAS
Faculdade de Engenharia Elétrica e de Computação

Raza Ul Mustafa

Machine Learning Assisted DASH Video QoE Inference Through Network QoS Features in 4G and 5G Scenarios

Inferência de QoE de Vídeo DASH assistida por aprendizado de máquina usando métricas de QoS de rede em cenários 4G e 5G

Thesis presented to the School of Electrical and Computer Engineering of the University of Campinas in partial fulfillment of the requirements for obtaining Doctor degree in Electrical Engineering, in the area of Computer Engineering.

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutor em Engenharia Elétrica, na área de Engenharia de Computação.

Supervisor/*Orientador*: Prof. Dr. Christian Rodolfo Esteve Rothenberg

Este trabalho corresponde à versão final da tese defendida pelo aluno Raza Ul Mustafa, e orientada pelo Prof. Dr. Christian Rodolfo Esteve Rothenberg

____

Campinas
2022

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

	Mustafa, Raza Ul, 1990-
M978m	Machine learning assisted DASH video QoE inference through network QoS features in 4G and 5G scenarios / Raza Ul Mustafa. – Campinas, SP : [s.n.], 2022.
	Orientador: Christian Rodolfo Esteve Rothenberg.
	Tese (doutorado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.
	1. Aprendizado de máquina. 2. Roteadores (Redes de computador). 3. Sistemas de comunicação móvel 4G. 4. Sistemas de comunicação móvel 5G. 5. Arquitetura de redes de computador. I. Esteve Rothenberg, Christian Rodolfo, 1982-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações Complementares

Título em outro idioma: Inferência de QoE de vídeo DASH assistida por aprendizado de máquina usando gerenciamento de QoS de rede em cenários 4G e 5G
Palavras-chave em inglês:
Machine learning
Routers (Computer networks)
4G mobile communication systems
5G mobile communication systems
Architecture of computer networks
Área de concentração: Engenharia de Computação
Titulação: Doutor em Engenharia Elétrica
Banca examinadora:
Christian Rodolfo Esteve Rothenberg [Orientador]
Aldebaro Barreto da Rocha Klautau Júnior
Fuad Mousse Abinader Junior
Paulo Cardieri
Rafael Pasquini
Data de defesa: 15-12-2022
Programa de Pós-Graduação: Engenharia Elétrica

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0003-1436-9127
- Currículo Lattes do autor: https://lattes.cnpq.br/9856256255657450

COMISSÃO JULGADORA - TESE DE DOUTORADO

Candidato: Raza Ul Mustafa **RA:** 211580
Data da Defesa: 15 de dezembro de 2022

Título da Tese: Machine Learning Assisted DASH Video QoE Inference Through Network QoS Features in 4G and 5G Scenarios (*Inferência de QoE de Vídeo DASH assistida por aprendizado de máquina usando métricas de QoS de rede em cenários 4G e 5G*)

Prof. Dr. Christian Rodolfo Esteve Rothenberg - Presidente
Prof. Dr. Aldebaro Barreto da Rocha Klautau Júnior - Membro Titular
Prof. Dr. Fuad Mousse Abinader Junior - Membro Titular
Prof. Dr. Paulo Cardieri - Membro Titular
Prof. Dr. Rafael Pasquini - Membro Titular

A Ata de Defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação/Tese) e na Secretaria de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação.

I dedicate this thesis to family of Muhammad Nazim Khokhar, who have always supported and encouraged me in all of my endeavors. Their unwavering love and belief in me have been the foundation of my journey and I am forever grateful. Specifically, I want to express my deep appreciation to Raheel Nazim, who has supported me and offered encouragement along the way. Finally, to my partner, who has been my constant source of love, support, and encouragement throughout this journey. This thesis was only possible with all of you. Thank you.

Acknowledgments

First, I would like to thank my advisor, Prof. Dr. Christian Esteve Rothenberg. Without his innovative ideas, thoughtful encouragement, and careful supervision, this thesis would never have taken shape. I want to express my gratitude for being a great instructor and for the confidence to work closely with his stellar group of students and collaborators. At the University of Campinas (Unicamp), I have had the opportunity to learn from the best professors. I want to acknowledge Dr. Sandra Avila, Dr. Nelson Fonseca, and Dr. Tiago Fernandes Tavares for sharing their knowledge and experience, which was a constant motivation throughout this work. I am also grateful to Dr. Jason Quinlan from Ireland, who helped me a lot to ease my experimental setup. I would also like to thank Dr. Chadi Barakat (Research Director, Inria, Université Côte d'Azur, France), who helped me during the final phase of my Ph.D. research work.

I would like to thank the committee members Prof. Dr. Aldebaro Klautau (UFPA), Dr. Fuad Mousse Abinader Junior (CPQD), Prof. Dr. Paulo Cardieri (FEEC/UNICAMP) and Prof. Dr. Rafael Pasquini (UFU) for the recommendations and insightful criticism they offered me in order to improve the final version of my thesis.

I had the good fortune to meet very talented people within the INTRIG groups. My special thanks go to Tariq, Nathan, Dany, David, Samira, Gyanesh, Raphael, and Suneet for their friendship and support throughout these years. I would like to express special gratitude for the financial and technical support received from the Ericsson Innovation Center (Brazil) and INRIA (France) which allowed me to carry out this research.

Abstract

With the recent rise of video traffic, ensuring Quality of Experience (QoE) becomes a challenge. The increasing adoption of end-to-end encryption hampers any payload inspection method for QoE assessments. This poses an additional obstacle for network operators to monitor the video QoE of a user, which by itself is tricky due to the adaptive behavior of Dynamic Adaptive Streaming over HTTP (DASH) mechanisms. This thesis presents a series of contributions on this topic. We present a DASH QoE Evaluation Framework named `EFFECTOR` equipped with all features for experimental evaluation in 4G and 5G use cases. `EFFECTOR` is reproducible and leverages real 4G and 5G datasets collected from a commercial network with multiple use case scenarios. Moreover, this thesis proposes novel machine learning based techniques as lightweight and fine-grained approaches to estimate the end-user QoE for DASH video service by passively monitoring the encrypted network traffic. We first consider i) Packet Time and ii) Packet Size to derive QoS metrics that are highly interrelated to the QoE of encrypted video streaming. We then propose novel metrics based on Inter Packet Gap (IPG) analytics in the form of time windows. Time windows are a technique to map QoE from QoS features without relying on chunk-level statistics, which are unfeasible in encrypted traffic. Furthermore, we investigate the DASH video performance over traditional TCP (HTTPS) and QUIC transport protocols. For this purpose, we experimentally evaluate 5G cellular network traces in our high-fidelity emulated testbed environment comparing the QoE KPIs of state-of-the-art Adaptive Bitrate Streaming (ABS) algorithms over HTTPS and QUIC. We provide Interactive Jupyter Notebooks with all the datasets produced during the experimental phase of this work for open science purposes. Finally, we run a notable endeavor of 6+ months of real 4G and 5G trace collection focused on YouTube streaming performance. The intuition behind this effort is multifold. Related to our work on `EFFECTOR`, the traces provide additional realistic 4G and 5G experimental capabilities under multiple scenarios. We turn public the obtained datasets consisting of a massive number of YouTube QoE KPIs and 100+ Channel Level Metrics (CLM) that unlock new research opportunities. We conclude by finding CLM relationship with YouTube stalling events and propose a machine learning technique to predict QoE KPIs (Stall) using metrics derived from CLM.

Key-words: Machine Learning; DASH; ABS; 5G; QoE; QoS; Youtube.

Resumo

Com o recente aumento do tráfego de vídeo, garantir a Qualidade de Experiência (QoE) torna-se um desafio. A crescente adoção de criptografia ponta a ponta dificulta qualquer método de inspeção de carga útil dos pacotes para avaliações de QoE. Isso representa um obstáculo adicional para as operadoras de rede monitorarem a QoE de vídeo de um usuário, o que é complicado, por si só, devido ao comportamento adaptativo dos mecanismos do tipo Dynamic Adaptive Streaming over HTTP (DASH). Esta tese apresenta uma série de contribuições sobre este tema. Apresentamos o `EFFECTOR` como framework para avaliação de QoE de vídeo DASH. O `EFFECTOR` oferece um conjunto de recursos para avaliação experimental em casos de uso de redes 4G e 5G. O `EFFECTOR` é reprodutível e aproveita capturas de tráfego 4G e 5G coletadas em redes operacionais, considerando vários cenários de casos de uso. Além disso, esta tese propõe novas técnicas baseadas em aprendizado de máquina como métodos leves e refinados para estimar a QoE do usuário final para o serviço de vídeo DASH, monitorando passivamente o tráfego de rede criptografado. Primeiramente consideramos o Tempo (i) e o Tamanho (ii) do pacote para derivar métricas de QoS que são altamente interligadas com a QoE do streaming de vídeo criptografado. Em seguida, propomos novas métricas baseadas na análise do Inter Packet Gap (IPG) na forma de janelas de tempo. As janelas de tempo representam uma técnica para mapear QoE a partir de recursos de QoS, sem depender de estatísticas em nível de bloco, que são inviáveis em tráfego criptografado. Além disso, investigamos o desempenho do vídeo DASH sobre os protocolos de transporte tradicionais TCP (HTTPS) e QUIC. Para isso, avaliamos experimentalmente o tráfego de rede celular 5G em nosso ambiente de teste emulado de alta fidelidade, comparando os Indicadores Chave de Desempenho (KPIs) de QoE dos principais algoritmos de Adaptive Bitrate Streaming (ABS) de última geração sobre HTTPS e QUIC. Fornecemos Interactive Jupyter Notebooks com todos os conjuntos de dados produzidos durante a fase experimental deste trabalho para fins de pesquisa baseada em ciência aberta. Por fim, realizou-se um esforço considerável ao coletar tráfego real 4G e 5G, por mais de 6 meses, focando no desempenho de streaming do YouTube. O intuito por trás desse esforço é múltiplo. Relacionado ao `EFFECTOR`, fruto deste trabalho, o tráfego capturado fornece recursos experimentais 4G e 5G adicionais, realistas em vários cenários. Os conjuntos de dados obtidos foram disponibilizados de forma pública e consistem em um grande número de KPIs de QoE do YouTube e mais de 100 métricas de nível de canal (CLM) que abrem novas oportunidades de pesquisa. Por fim, concluímos encontrando a relação do CLM com os eventos de paralisação do YouTube e propomos uma técnica de aprendizado de máquina para prever KPIs de QoE (Stall) usando métricas derivadas do CLM.

Palavras-chave: Aprendizado de Máquina; DASH; ABS; 5G; QoE; QoS; Youtube.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

1.1 Motivation

The steady growth of Internet data services drove the development of the third (3G) and fourth (4G) generations of the mobile communications standard. Now, the technology is evolving towards its fifth-generation (5G) [2], motivated by increasing demands, including significantly higher throughput (10 Gbps), 1-millisecond end-to-end over-the-air latency, real-time information processing, and transmission, and lower network management operation complexity.

Moreover, 5G and Beyond networks are expected to equip with the Edge Computing (e.g., Multi-Access Edge Computing) paradigm, which brings the computing of traffic and services from a centralized cloud to the edge of the network and closer to the end-users. The main goal of edge computing is to reduce latency, provide real-time response and network conditions, lower network bandwidth load, and provide better performance to end-users in real-time.

Video streaming is expected to be one of the main services of new generation cellular networks. In fact, the share of video streams in the overall internet traffic has recently reached 70% and continues to grow. Most of this type of traffic is generated by popular video hosting services, such as YouTube, Netflix, and others, which use the Dynamic Adaptive Streaming over HTTP (DASH) technology, whereas mobile video traffic is supposed to increase by 73% in 2023 [3]. Therefore, one of the main challenges in video delivery will be the ability to effectively manage the increased growth in traffic while ensuring the expected Quality of Experience (QoE) to the end-users.

Therefore, it has become a critical issue for Mobile Network Operator (MNO) stakeholders to fulfill such traffic demand and provide a satisfactory experience to their end-users, known as QoE. For MNOs to ensure better QoE, understanding and monitoring the Key Performance Indicators (KPIs) that impact users' perceived experience and service quality has become a challenging and trending topic.

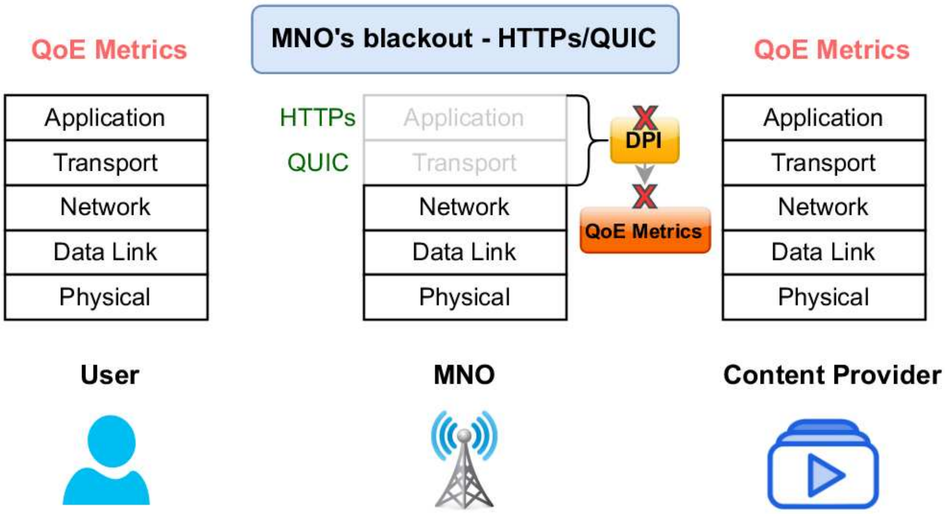


Figure 1.1: End-to-End encryption over HTTPs and QUIC. Source: [1]

Traditional network monitoring relied on Deep Packet Inspection (DPI) to understand the KPIs, which are heavyweight and costly. Moreover, these days, such information, i.e., chunk information is also unavailable because of TLS encryption [4, 1, 5]. TLS establishes a more secure and private connection with end-to-end encryption [6]. Network operators no longer have access to extract the application-level QoE metrics from HTTP packet flow, as shown in Figure 1.1. Video content providers such as YouTube, Netflix, Amazon Prime, Hulu, etc., are already using HTTP adaptive streaming (HAS) with HTTPS [7]. Moreover, compared to the traditional way of managing network resources, Machine Learning (ML) provides a more secure, cost-effective, efficient, and fast solution to manage current network needs at the edge closer to the DASH client [8, 9].

In the field of video streaming technologies, the Moving Pictures Expert Group (MPEG), an international authority on digital audio and video standards, developed Dynamic Adaptive Streaming over HTTP (DASH) as an industry standard [10]. DASH is now the de-facto choice of popular services such as YouTube and Netflix for Internet video distribution. In adaptive streaming, video content is split into multiple segments, typically with an individual segment duration of between 2 to 10 seconds. Each segment is then encoded with a different video bitrate and resolution. In DASH, each segments' structure describes in Media Presentation Description (MPD) file. The Adaptive Bitrate Streaming (ABS) algorithm decides the quality of the segments to be downloaded based on the network's available resources see Figure 1.2.

To ease access to video content on the associated web server, a MPD file is created. This MPD file contains general information such as clip length, segment duration, and DASH video profile, but more importantly, the MPD file contains metrics specific to each of the video representations available. Each representation represents a different quality level determined by video resolution and average encoding bitrate, thus offering an easy mechanism to enable the player to adapt video quality for the user. Once the player

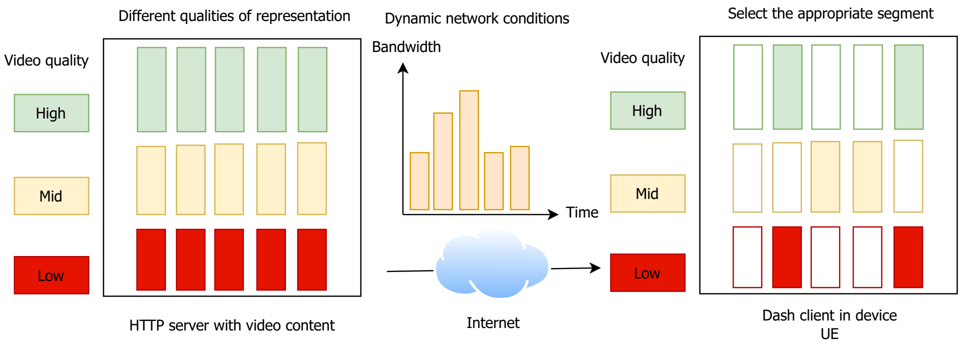


Figure 1.2: The role of Adaptive Bitrate Streaming (ABS) algorithm.

downloads the MPD file, the ABS algorithm can adjust quality by selecting the most appropriate segment for each video over a time period. Thus, there is a need to be aware of network conditions, i.e., traffic patterns, to optimize video delivery quality by selecting the most suitable ABS algorithm, which is ideal for MNOs.

In the past, several approaches have been proposed to measure the KPIs aimed at delivering acceptable video service quality. Most solutions require end-user awareness, which is not viable from the MNOs' perspective. Therefore we can only rely on the information available in the IP packets header, such as Packet Time and Packet Size, to extract the Quality of Service (QoS) KPIs and their pattern to infer QoE using different machine learning techniques [1]. In this case, the edge computing facility can be an appealing location to analyze target end-users perceived video service quality (i.e., QoE) [11]. Thus QoS KPIs are used to infer objective QoE KPIs to increase end user satisfaction.

1.2 Research Questions and Contributions

The exponential growth of video traffic over the 5G networks poses many challenges to meet end user satisfaction. Therefore, we need to understand the performance footprint of the current state-of-the-art ABS under 5G and how it compares with 4G. Despite anticipated QoS improvements in 5G networks, video traffic has opened many research dimensions. One challenge for 5G networks is to manage the exponential growth of multimedia traffic while maximizing the end-user perceived QoE for a given service. With the increasing adoption of end-to-end encryption [6], MNOs face additional challenges in monitoring and managing their network resources because encryption limits their visibility to service QoS metrics to map QoE.

Considering the presented problem scope, we identify the following research questions:

- RQ1: Which QoS features can be effectively used from the network level of encrypted and unencrypted DASH traffic to estimate the QoE of adaptive video streaming?

- RQ2: How to conduct large-scale experiments using real 4G and 5G use cases, which requires low system requirements and at the same time, provide necessary data for analysis of the streaming sessions?

- RQ3: What is the performance footprint of the current state-of-art ABS algorithms under 5G, and how does it compare with 4G? Which ABS algorithm is more similar to the most popular streaming platform YouTube?
- RQ4: How can ML techniques effectively predict QoE using network-level QoS features?
- RQ5: How can we correlate 4G and 5G Channel Level Metrics (CLM) to the QoE KPIs of real YouTube Traffic?

Research Question #1: We discuss QoS feature extraction methods both for *Encrypted* and *Unencrypted* traffic. In Unencrypted traffic, we find segment level (Chunks) information from network traces. We find that per-segment features (throughput, per-segment RTT, total packets) can estimate QoE by using machine learning techniques. Thus we use different regression models, i.e., (Linear Regression with Multiple Variables, Decision, and Random Regressor) to predict QoE (P.1203) [12] continuous scores ranging from 1-5. The main contributions of this work are listed in **Publication A** [13]. In short,

- An in-depth analysis of six state-of-the-art ABS algorithms streaming with varying bandwidth in static and mobile 5G scenarios. The analysis is undertaken through the assessment of associated QoE models such as the P.1203 QoE standard.
- A proposal for a Machine Learning regression model to estimate QoE based on per-segment RTT, number of packets and throughput.

However, due to the increased adoption of end-to-end encryption, network operators are hampered in using payload information about the video traffic because encryption limits their visibility for traditional deep packet inspection. Therefore, an appealing solution is to find the relation between QoE and QoS by relying on network-level statistics.

The only available network-level information is Packet Size and Packet Time. Different statistical methods have been used to derive more meaningful features from these two basic metrics. Then machine learning deep network models are widely used to find complex relationships [1, 5]. We used these two basic features and derived 30+ QoS features that would estimate QoE in a different time slot in a sequential manner. We also reduced the traditional complexity of finding segment information, thus improving the performance and efficiency of the proposed solution.

We propose a solution to estimate QoE using a novel metric named Inter Packet Gap (IPG), where we further derive a large number of QoS features from it. We use regression techniques for unencrypted traffic to predict QoE on a scale (1-5), and we rely on classification techniques to classify the QoE as Poor, Good, and Excellent for all types of ABS and both protocols TCP and QUIC.

Such an approach is ideal for deploying QoE estimators at Multi-access Edge Computing (MEC) hosts. The MEC presents an opportunity to enhance the user experience by adapting the video quality at the edge instead of a centralized mechanism. Moreover, the concept best provides fair network resources to all clients by knowing their current player information. Moreover, the MEC-assisted network provides the best bitrate to clients by efficiently utilizing bandwidth hence providing maximum QoE. The proposed approach can be used as per the needs of the streaming service providers. However, network traffic generally comprises parallel traffic flows for several services and a network interface containing raw network packets. Moreover, application-level header information remains inaccessible due to the end-to-end encryption over TCP (HTTPs) and QUIC transport protocols. Therefore, to infer video flows the following few options are ideal, i) a unique 5-tuple (IP source, IP destination, port source, port destination, protocol), ii) the IP address of the video content providers, iii) inspect DNS responses (matching the DNS lookups against the known signature of video service), iv) use of machine learning techniques such as clustering to classify the video flows [14, 15].
Publications **C, D, E, H** [16, 17, 18] summarize the contributions for encrypted video streaming sessions.

- QoS and QoE assessment using a time window of (0.5, 1, 2, 3, 4, 5)-seconds. The analysis is undertaken through objective QoE models P.1203 [19, 20].
- A proposal of a machine learning classifier to estimate QoE based on novel metric inter packet gap IPG, packets length and throughput distribution into (10-90) percentile, metrics derived from IPG, i.e., Exponentially Weighted Moving Average (EMA), Double Exponentially Weighted Moving Average (DEMA), Cumulative Sum (CUSUM). Usually, EMA and DEMA are technical indicators used to identify a potential uptrend or downtrend in time series data. We use these metrics to find where the continuity of data packets changes from the mean value. Moreover, the classifiers are unaware of the specific ABS algorithm and use cases, using only network QoS metrics (throughput and packets) and not requiring any chunk detection.

Research Question #2: To conduct large-scale 4G and 5G experiments, which provides necessary data for analysis of video streaming session in both technologies, we provide flexible frameworks with massive 4G and 5G datasets for analyzing QoS to QoE in DASH videos.

The framework is equipped with many other options, such as changing the DASH videos, a range of different ABS algorithms to compare QoS to QoE metrics, i.e., Throughput, Buffered, and Hybrid. We provide processed QoS features in the form of Interactive Jupyter Notebook (IJN) to see the impact of QoS and QoE in 4G and 5G with both protocols, TCP and QUIC. With IJN, users can visualize and control changes in the data realtime with the help of drop down options.

QUIC, a relatively new protocol with the promise of performance improvement over the widely used TCP, motivated us to reconsider an ideal transport protocol for adaptive video streaming service. Thus, the frameworks are ideal for investigating TCP over QUIC in 4G and 5G. Moreover, we generated a massive dataset for 4G and 5G use cases using both the protocols TCP and QUIC and provided IJN.

Publication **B and H** [21, 18], we provide two reproducible DASH QoS to QoE evaluation framework for `unencrypted` and `encrypted` traffic.

- We provide reproducible DASH QoS and QoE evaluation frameworks with open 4G and 5G datasets. Frameworks provide flexibility to change many settings, i.e., Protocols – TCP, QUIC, Videos, Technology – 4G, 5G, QoS – Encrypted, Unencrypted, ABS – Buffered, Hybrid, Throughput, etc.
- Frameworks are equipped with all the dependencies to run large experiments using real 4G and 5G datasets collected in the wild.

Research Question #3: We discuss the performance of 4G and 5G using different Adaptive Bitrate Streaming (ABS) algorithms. ABS algorithms are responsible for selecting the bitrate of the next segment. However, Adaptive algorithms follow different strategies to select the next segment, keeping in mind the network condition. We select three state-of-the-art ABS, i) Convention – Throughput based, ii) Buffered – Buffer based, iii) Elastic – Hybrid. We show how 5G performs under different network scenarios compared to 4G.

In Publications **A, B, C, H** [13, 21, 16, 18]:

- We compare the performance of 4G and 5G using different ABS, i.e., Buffered, Hybrid, and Throughput in an emulation environment using real 4G and 5G datasets collected in the wild.
- We provide a comparative analysis of YouTube streaming sessions with different ABS algorithms. We conclude that TCP and QUIC using ABS - Conventional are similar to YouTube player (Chapter 3).
- We also provide Interactive Jupyter Notebook to play with the dataset generated during the experimentation phase. Where various options facilitate getting an in-depth understanding of Protocols – TCP, QUIC.

Research Question #4: Machine Learning techniques have been widely used for video quality enhancement [1, 22]. ML classifier works by finding the key pattern in the dataset and then predicting the target class (Y). In the case of video quality, it predicts video quality ranging from 1-5. We initially used regressors ML models to predict MOS because of the continuous nature of the target variable P.1203 score. We observe QoS features derived from chunks have a linear relation with the QoE score. Thus, we used regression algorithms such as i) Linear Regression with multiple features and ii) Decision and Random Regressor, which are used both for regression and classification.

Next, we find QoS features of encrypted DASH traffic. We use 3 classes of video quality named i) Poor, ii) Good, iii) Excellent using P.1203 scores ranging from 1-5. We use state-of-the-art machine learning classifiers that require less computational cost and produce efficient results, such as Decision Tree (DT), Random Forests (RF), k-nearest neighbor (KNN), and Artificial Neural Network (ANN). More explanation is in Chapters 3 and 4 of this thesis.

In Publications **A, C** [13, 16]:

- We use various supervised machine learning benchmarks to provide classifiers that can predict QoE based on QoS features.
- We also find a relationship between QoS and QoE features from encrypted video sessions in a time based QoS feature extraction approach, i.e., IPG, Throughput and Packet Size distributions, along with shifts QoS metrics derived from IPG.
- Deep (Artificial) Neural Networks and Random Forests provide high accuracy as compared to Decision Tree and KNN. Moreover, various time windows, i.e., (1-5)-seconds, have different accuracy for different types of adaptive algorithms.

Research Question #5: We provide commercial 4G and 5G datasets covering a period of six months in different regions (Brazil, France, USA) [23]. We use different use cases i) Mobility, ii) Pedestrian – less mobility, iii) Indoor, iv) Outdoor. The dataset we provide is also equipped with CLM features, i.e., CQI, RSRQ, RSRP, SNR, Download Bitrate, Handover Events, etc. Moreover, we also provide a comprehensive footprint of 5G performance compared to 4G using YouTube as a baseline under different use cases. We looked at the Objective QoE KPIs – Stalls, Quality shifts, and Handover events. Furthermore, we provide an alternative solution to predict Objective QoE KPI - Stall in real YouTube traffic by using only CLM metrics and the features derived from them.

In Publication **F, G** [23] the contributions are:

- We provide a web-based Framework to run a large number of experiments using YouTube as a baseline to see the impact of 4G and 5G commercial networks in

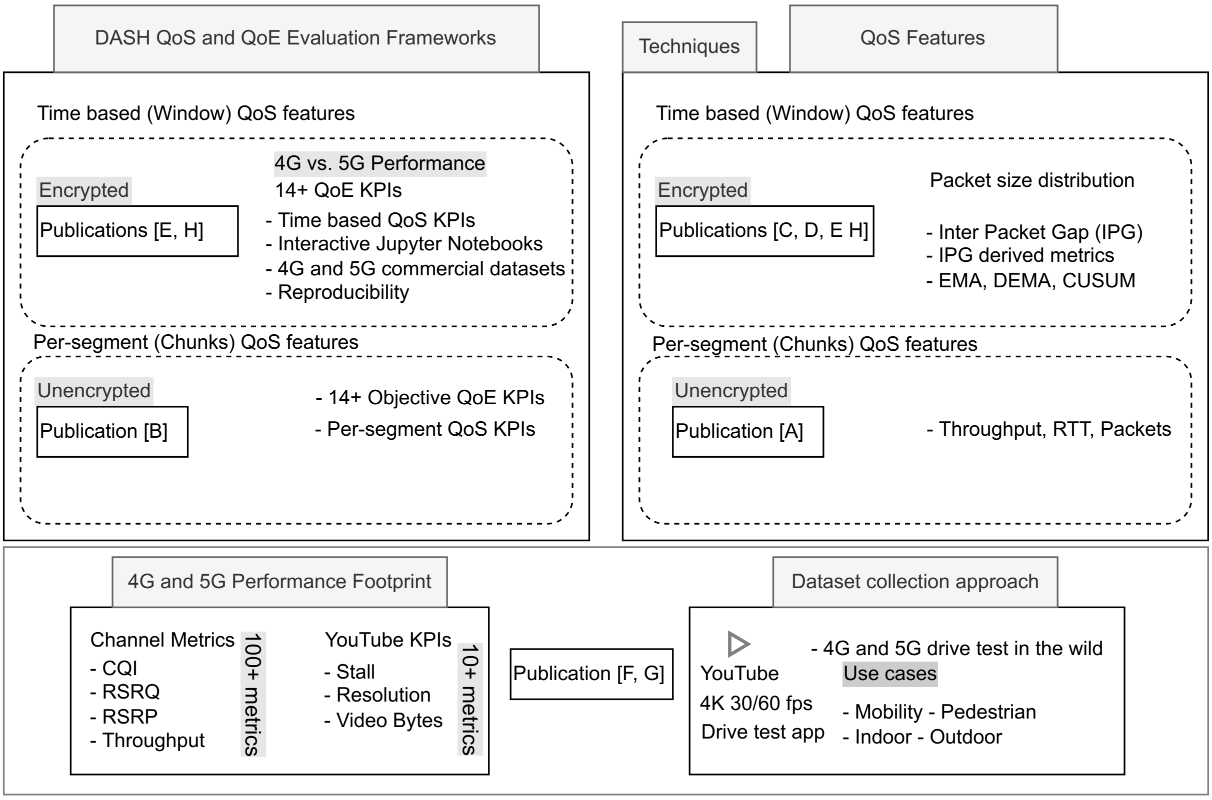


Figure 1.3: Overview of the main topics and publications of the thesis.

the wild. We provide YouTube player information after every 1-second interval, i.e., i) Current Quality, ii) Video Bytes Downloaded, iii) Resolutions, iv) Time Sequence along with Player events, i) Stall, ii) Change in Quality (Change in Resolutions), iii) Video Paused, iv) Video Ended, v) Time Sequence.

– We use Channel Level Metrics to provide a classifier to predict stalling events. We used various time sequences, i.e., (1, 3, 5, 7, 9) seconds. We find a 7-second time window has a strong interrelationship with Interruptions.

– We provide a massive dataset covering over six months for further studies with a 1-second granularity of both YouTube KPIs and CLM KPIs.

1.2.1 Publications

Figure 1.3 gives an overview of how the publications can be mapped to the different areas of the contributions.

A. **Raza Ul Mustafa**, Simone Ferlin, Christian Esteve Rothenberg, Darijo Raca, Jason J. Quinlan, "A Supervised Machine Learning Approach for DASH Video QoE Prediction in 5G Networks" Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks, ACM, 2020.

B. **Raza Ul Mustafa**, Md Tariqul Islam, Christian Rothenberg, Darijo Raca, Jason J. Quinlan, "DASH QoE performance evaluation framework with 5G datasets" 2020 16th International Conference on Network and Service Management (CNSM) AnServApp Workshop. IEEE, 2020.

C. **Raza Ul Mustafa**, David Moura, Christian Esteve Rothenberg "Machine Learning Approach to Estimate Video QoE of Encrypted DASH Traffic in 5G Networks". In IEEE Signal Processing (SSP2021) Workshop, 2021.

D. **Raza Ul Mustafa**, Christian Esteve Rothenberg "Machine Learning Assisted Real-time DASH Video QoE Estimation Technique for Encrypted Traffic" In ACM MHV, 2022.

E. **Raza Ul Mustafa**, MD. Tariqul Islam, Christian Esteve Rothenberg, Pedro Henrique Gomes "A Framework For QoS and QoE Assessment of Encrypted Video Traffic With 4G and 5G Open Datasets" In IEEE Globecom Demo Session, 2022.

F. **Raza Ul Mustafa**, Chadi Barakat, Christian Esteve Rothenberg, "YouTube Goes 5G: Benchmarking YouTube in 4G vs 5G Through Open Datasets", In IEEE Globecom Demo Session, 2022.

G. **Raza Ul Mustafa**, Christian Esteve Rothenberg, Chadi Barakat, "YouTube goes 5G: Benchmarking YouTube in 4G vs. 5G", IEEE Dataport, https://dx.doi.org/10.21227/h00h-ew92, 2022.

H. **Raza Ul Mustafa**, Md Tariqul Islam, Christian Rothenberg, Pedro Henrique Gomes. "EFFECTOR: DASH QoE and QoS Evaluation Framework For EnCrypTed videO tRaffic." IEEE/IFIP Network Operations and Management Symposium (NOMS), 2023.

1.3 Further Contributions, Results & Collaborative Activities

The complete effort around this work incorporates a number of collaborative activities referenced to the corresponding scientific article, blog post and/or demo indicating the co-authors. The list of publications is shown below:

1. **Framework Github Repository**, "Dash Quality of Experience Open Source Evaluation Framework", https://github.com/razaulmustafa852/dashframework-1, 2020.

2. Christian Esteve Rothenberg, Danny Alex Lachos Perez„ Nathan F. Saraiva de Sousa, Raphael Rosa, **Raza Ul MUSTAFA**, Md Tariqul Islam, Pedro Henrique

Gomes. "Intent-based Control Loop for DASH Video Service Assurance using ML-based Edge QoE Estimation". In 6th IEEE International Conference on Network Softwarization (NetSoft'20) - Demo Session, Ghent, Belgium. Jun 2020.

3. Nathan F. Saraiva de Sousa, Md Tariqul Islam, **Raza Ul Mustafa**, Danny Alex Lachos Perez, Christian Esteve Rothenberg, Pedro Henrique Gomes, "Machine Learning-Assisted Closed-Control Loops for Beyond 5G Multi-Domain Zero-Touch Networks", JNSM, 2022.

1.4 Outline of the Thesis

The remainder of this thesis is organized as follows. Chapter 2, discusses the background and literature work. We highlight the state-of-the-art work and comprehensive comparison. Chapter 3, "DASH QoS to QoE Evaluation Frameworks for 4G & 5G", provides DASH QoE evaluation frameworks. The frameworks are equipped with all the dependencies to run 4G and 5G experiments with 4G and 5G datasets. Thus providing the research community to explore QoS metrics and their relation with the QoE KPIs. In this chapter, we also provide a rich dataset generated during the experimental phase of the frameworks followed by Chapter 4, "DASH Video QoE Prediction Using Machine Learning Techniques", where we propose a window based QoS features extraction approach, a novel metrics IPG and the features derived from these novel metrics to feed into Machine Learning Classifiers to Predict QoE. Chapter 5, "YouTube goes 5G: Benchmarking YouTube in 4G vs 5G Through Open Datasets", proposes a technique to use only Channel Level Metrics to predict stalling events in YouTube Traffic. We also provide a rich dataset for both the technologies 4G and 5G collected in the wild in France. We provide a fair comparison of different objective KPIs in 4G and 5G along with future work in the Chapter. Finally, we present our conclusions with remarks for future goals and activities in Chapter 6, "Conclusion & Future Work".

Chapter 2

Background

This chapter discusses recent work and a fair comparison of 4G vs. 5G technology and Quality of Experience (QoE) KPIs for DASH video.

2.1 Why is DASH popular?

The large diversity of content of today's video content providers made DASH a very popular mechanism for video delivery. A video is present on the server with different resolutions and bitrates; thus, the same video provides different QoE on different network conditions. Video content providers store video in different encoding formats. Among them, Variable Bitrate (VBR) encoding technique is very popular in which complex scenes of the video are allocated with higher bitrates, whereas less complex scenes with lower bitrates. Thus higher bitrates video contents require higher network QoS resources for smooth video playout.

This makes DASH a popular mechanism where network resources are directly proportional to video bitrates delivering higher resolutions and improved QoE. Therefore, understanding ABS behavior in different network conditions and choosing the best ABS has gained tremendous attention from the research community. ABS algorithms dynamically select the appropriate segments based on network conditions. The purpose of this dynamic segment selection is to adapt to changes in network conditions and provide an interrupt-free (e.g., stall) service. In 4G and 5G networks, understanding QoS and QoE for different state-of-the-art ABS is still challenging, because different Over-the-Top (OTT) platforms use different adaptation algorithms to deal with video traffic demand and provide end user maximum QoE. Furthermore, finding the right QoS features to adequately estimate QoE in encrypted network traffic is also challenging.

2.2 4G vs. 5G Networks

5G cellular technology uses high data rates (around 1Gbps) and low latency (1ms), which is a 10x increase compared to 4G LTE [24]. 5G has much higher radio frequencies (28 GHz compared to 2.5 GHz for 4G) to transfer more data with faster speeds, reduced congestion and lower latency [25]. 5G promises these three core values:

1. Latency of less than one second
2. Increased data rates of at least 1 Gbps for tens of thousands of users simultaneously
3. Increased energy efficiency

Certainly, reduced latency is the core of 5G networks, where 5G network users can upload and download files very quickly [26, 27]. That means downloading a two-hour video on a 4G connection will typically take six minutes, while the same download on 5G takes about four to five seconds. Moreover, 4G can support about 4,000 devices per Km^2, whereas 5G will support around one million [25]. This means more Netflix streaming, voice calls and YouTube carried, without interruption, over the limited air space. In addition to that, Applications utilizing social media, gaming, and recent advances in Augmented/Virtual Reality, have accelerated the demands for 5G.

The two most significant factors that have accelerated the demand for the 5G include the rapid increase in the number of connected devices along with the exponential increase in multimedia traffic; thus, more throughput demands. While 5G can sustain these high throughput demands, it is yet to be proven what is the performance footprint of multimedia traffic such as Video on Demand (VoD) and live streaming over 4G and 5G.

2.3 Quality of Experience Metrics of Video Streaming

The end-user QoE depends on various factors (see Figure 2.1), which can be categorized as follows:

- **Content level** features are related to video, such as encoding rate, encoding format, resolutions, video duration, quality of the video, and finally, the popularity of the video.
- **Environment** includes the environment while watching the video content.
- **User** considers the psychological factors such as end-user expectations, mood, background and previous history. It also includes the purpose of watching the video, such as the video watched for educational purposes or entertainment.

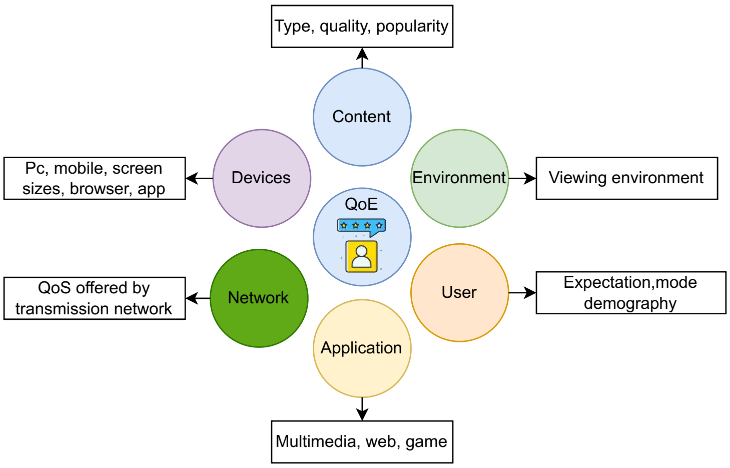


Figure 2.1: Factors affecting the video QoE.

- **Application** includes applications, is the end user playing a game, watching multimedia content, or using a web browser for different type of surfing.

- **Network** QoS offered by transmission network, in this research work, we focused on network level QoS features.

- **Devices** User device type (screen size, mobile/pc), browser (Chrome, Firefox, Safari, etc.).

2.4 Methods for Gauging Video QoE

2.4.1 Subjective Video QoE

Subjective QoE assessment utilizes end-users who grade video quality at the end of a video session using perceiving video quality so-called Mean Opinion Score (MOS). The ITU-T recommendations [28] for subjective quality evaluation follow strict setup and testing conditions. However, subjective QoE is expensive, time-consuming, and doesn't scale very well. Moreover, there are many other factors, such as psychological or psycho-physiological, e.g., age, mood, time of day, gender, and socio-economic status [29] that may influence the results.

2.4.2 Objective Video QoE

Due to subjective QoE limitations, objective QoE assessment has gained more popularity [30], with some models that directly map objective QoE to well-known metrics such as MOS, Peak Signal to Noise Ratio (PSNR), and Structural Similarity Index Metrics (SSIM). Variance in the results of these metrics can be tied directly to the quality of the original video stream, and as such more ground truth is needed to improve objective QoE values. The objective QoE involves various KPIs such as the initial delay, stall duration and stall events, quality switching, and the duration of video played in higher resolutions. We discuss each KPI in detail below.

2.4.2.1 Startup Delay

The startup delay is the time taken for the video to start playing after the client requests the video. Usually, when a client requests the desired video, clients first download video chunks and the video starts playing when there are enough chunks downloaded in the buffer for smooth video transmission. Startup delay plays a significant role in end-user QoE as a large startup delay can ultimately cause a user to abandon video. This means startup delay has to be minimal for maximum QoE expectations [31].

2.4.2.2 Stall Events

Stalling events occur when the playout buffer gets empty. The first study that investigates video quality, freezing stall events, and stall duration is proposed in [32] and was adopted by the ITU [33]. Stalling events have a huge impact on end-user QoE. The authors in [34, 35] concluded that frequent stall events cause users to give less QoE as compared to a single long stalling event. In addition to that, stall position has an impact on QoE. Stall events, in the beginning, have less influence as compared to stall events in the middle of the stream. Authors also find that users can only tolerate one stall event. However, frequent periodical events have a worse effect on MOS.

2.4.2.3 Quality Switches/Adaptation

Video quality switches from one resolution to another depending on the network conditions. However, these days a client can switch to resolutions of his own desire; if the network throughput is very low, then the best option is to play a video to a lower resolution and vice-versa. However, too many quality switches have a negative impact on QoE [36]. Authors find that step-wise decrease in quality, has less impact on QoE as compared to a sudden jump to a lower resolution. In another study authors also find that video played in lower resolution has less impact on QoE as compared to video played in higher resolutions with frequent quality switching and stall events [37]. Overall, a video played in a resolution with nice perceived quality is better than repeated switching.

2.5 Methodologies for Modeling Video QoE

The video QoE depends on QoS features, which in turn is dependent on network quality of service. QoS to QoE mapping is required to find the complex relationship between these two metrics in the dataset. For any QoE prediction model, data is required to find such a relation. Data is either collected in the wild or built by controlled experimentation [38].

2.5.1 QoE Modeling Using Data Collected in the Wild

In this approach, the data is generated by network service providers. They consider real users streaming video content using famous video content providers such as YouTube or Netflix. For instance, authors in [1] consider packet Inter Arrival Time (IAT) and packet size and used state-of-the-art ML models to find the complex relationships between QoE and QoS of encrypted YouTube Traffic. The authors used six resolutions as a target variable and considered continuous bitrate and stall as binary events. The dataset was collected using 15,000 + video sessions with a duration of 9 months in 2018/2019.

The dataset collection approach and machine learning work very close to the above work is Requet [5]. In Requet, the target KPIs are stall, resolutions, and bitrate. Similar to ViCrypt by [1], the authors consider 127 QoS features and the same methods to find QoE. The difference is the total number of video sessions, features computational efficiency, devices (Laptop only), and time granularity of 5-10 seconds. In another research, YoMoApp was used to find QoE features such as player state, buffer, buffer-event time, and video quality level for YouTube traffic [39]. Similarly, in another research, authors developed a tool to crawl YouTube different videos and collect additional information related to the network to map QoE [40].

Additional works such as [41, 42] examine to find which objective QoE affects MOS. These works consider Stall (re-buffering), rebuffering ratio and average video bitrate. Their main findings rely on exploring the issue that causes quality degradation. They find that network throughput, CDN performance and ISP are the main cause of less MOS. Motivated by [1, 5], in this research work, we investigated both online and offline. We collected real 4G and 5G datasets along with channel and context and showed a relationship of CLM metrics with objective QoE stall. Next, we use the emulation environment and find the QoS factors that affect the video quality of experience.

2.5.1.1 5G Dataset Collection in the Wild

Studies [24, 43, 44, 45, 46] collected a 5G dataset in the wild with various key findings, i.e, the interconnection of CLM – CQI with throughput. In [24], authors collected a 5G dataset using 5G download, Netflix and Streaming Amazon Videos. Next, in [43] collected dataset using online gaming in a 5G environment. Followed by controlled video streaming, i.e., custom videos, network resources, etc., in [44, 45, 46]. Authors consider different use cases for the collection of datasets in the wild. Compared to 5G dataset collection in the wild, we consider YouTube as a baseline for video streaming and collected Channel Metrics such as CQI, SNR, RSRQ, RSRP among 100+ features with the smallest granularity of 1-second. We also provide YouTube QoE logs for all the experiments with 1-second intervals, i.e., Quality, Video Bytes Downloaded, Available Quality, along with Player Events (Stall, Pause, Playing, Ended, etc.). Moreover, we find CLM patterns to predict the objective QoE – Stall (Interruptions) of YouTube. More explanation of the whole work and methodology is discussed in Chapter 5 of this thesis.

2.5.2 QoE Modeling by Controlled Experimentation

In a controlled environment, the dataset is generated by controlled experimentation. Researchers build their own datasets for QoE evaluations. Here the QoS metrics are accurately controlled to build the relationship between the QoS and the QoE. Thus, we generated our own datasets for offline experiments with various network use cases (Chapter 3). Moreover, during the emulation of different network use cases, we also consider ABS algorithm. The choice of ABS algorithm affects the video quality MOS score [47]. It has also been observed that continuous quality switching is also a big factor influencing QoE [36].

Authors in [48] create an HTTP adaptive streaming platform where QoS metrics are artificially varied to see their impact on QoE. Furthermore, the authors used ten users simultaneously to see the effect. Similarly, in another work [49], the approach was different where more than 100+ volunteers were asked to rate QoE with different QoS settings using controlled experiments. Both studies conclude that bitrate, startup delay, bitrate switching are the main factors that influence QoE.

Authors in [22] consider startup delay, total number of stalling events, the spatial resolutions, and quality switches as objective QoE and infer them from QoS features of encrypted YouTube traffic. Similarly, another work by [38], provided a framework of controlled experimentation based on active learning, that allows the collection of rich datasets covering the experimental space intelligently.

In another work [31], the authors perform controlled experiments and build a dataset of 5488 and 5375 video sessions for QUIC and TCP respectively. They use machine learning methods to estimate QoE such as startup delay, the stalling events and the video resolution from QoS metrics packet inter arrival time, packet sizes and throughput.

2.5.3 QoE Prediction Using Machine Learning Techniques

Machine learning has been widely used for mapping QoS and QoE. This led to the development of QoE prediction model that estimates linear, exponential, or logarithm relationships between QoS features to find end-user QoE.

Dimopoulos et al. [50] leverage machine learning to evaluate the interrelation between QoS and QoE to overcome the challenge of measuring end-user satisfaction. Another area where ML techniques were recently shown a lot of promise is improving and designing ML-backed ABR schemes. Authors in [51, 52] use ML to compute parameters of the existing ABR scheme to adapt to dynamically changing network conditions. Unlike these approaches, several approaches train ML model as the replacement for ABR algorithm [53, 54, 55].

In a recent study, research based on YouTube QoE prediction used Network Level information such as packet size, arrival time and packet length [1]. Another research conducted on cellular networks considers the same sorts of Network Level information and KPIs such as stall and bitrate etc [50]. Authors also used ML using packet loss, chunk size, round-trip times, etc., as input features to predict QoE. In another study, authors used objective QoE metrics of YouTube to map user level QoE, such as the number of stalling events, total stalling time, and initial delay [56].

Given the importance of video QoE estimation, the International Telecommunications Union (ITU) has developed a QoE prediction model named ITU-T Rec. P.1203 QoE standard [19, 20]. In our study, we consider (mode 0 considering metadata only, bitrate, frame rate, and resolution) for QoE evaluation and assessment. The conventional approach uses Deep Packet Inspection (DPI) to infer (Predict) QoE. However, due to TLS encryption, such an approach is no longer valuable. Thus we can only rely on the limited information available such as i) Packet Size, and ii) Packet Time. In this approach, the network QoS consists of features such as Throughput, Packet Inter Arrival Time and Packet Size where further QoS features can be derived from these metrics of encrypted video traffic to estimate the video QoE.

Next, using these features various Machine Learning classifiers are used to map QoS and QoE such as DT, RF, KNN and ANN, etc [4]. DT utilizes a tree-like structure in its models and is popular in both regression and classification problems. Departing from the root (parent) node, child nodes are decided by the metrics named Information Gain (IG) [57], and the iterative process terminates when the leaves are so-called *pure*.

RF is used for classification and regression by building multiple decision trees. RF is also commonly known as Bagging, because it trains each DT on different data samples and, finally, instead of relying on a single tree, it considers all decisions (e.g., voting)together before taking the final decision [58].

KNN is also a supervised ML algorithm that assigns a target class based on similarity. It is mostly used to perform data point classification based on how its neighbors are classified. Where K in KNN stands for the nearest neighbors during the voting target class process for a given new instance [59].

ANN works in a layered manner. There are three types of layers i) input, ii) hidden, and iii) output. Patterns or predictive features are presented to the network, which communicates to one or more hidden layers. Each hidden layer processes its input features and forwards its outputs to the next layer. Layers are connected to each other with processing units named *neurons*. Initially, small randomized weights are assigned to input features to reach target output Y. In the end, the last hidden layer is connected to the output layer, where the target class is shown. The calculation of the target class in each layer is done by different activation functions such as RELU, sigmoid, Tanh, and softplus, among other options. [60, 61].

In this thesis, we devise methodologies and models using controlled experimentation and machine learning techniques for estimating video QoE from QoS features derived from packet size and packet time [13, 16]. We used RF, DT, ANN and k-NN for the prediction video QoE on a scale of 1-5. The complete work is reproducible and we also provide a framework for conducting large scale experiments using real 4G and 5G traces. The complete explanation of how to use the framework `EFFECTOR` is discussed in Chapter 3 of this thesis, whereas, the QoS features and machine learning methods are described in Chapter 4 of the thesis.

Overall, to the best of our knowledge, we make the following novel contributions:

- `EFFECTOR` is a framework to conduct large scale 4G and 5G experiments using real 4G and 5G traces. The framework supports multiple options such as i) Protocols - TCP, QUIC, ii) ABS algorithms - Conventional, Hybrid, Buffered, iii) Video and iv) Use cases. A complete explanation of how to use the framework is discussed in Chapter 3.

- We provide a fair comparison of 4G and 5G using multiple use cases. We show the performance footprint of both technologies by different ABS algorithms and protocols using a well-known QoE model P.1203 [19, 20]. We provide a massive dataset in the form of Interactive Jupyter Notebooks to see the impact of network use cases on QoE of the video streaming sessions.

- We propose a QoS features extraction approach in the form of time windows, i.e., (0.5, 1,...,5) seconds. However, the time window is flexible, i.e., we can find QoS features at any time.

- We develop supervised machine learning based models to predict the QoE of video streaming sessions using QoS features (Chapter 4). QoS lightweight features consist of Inter Packet Gap (IPG) as a novel metric and the metrics derived from it. We show each ABS algorithm's accuracy and QoS to QoE relationship in Chapters 3 and 4.

- We also provide a framework to collect real 4G and 5G traces using YouTube as a baseline for video streaming. The framework collects YouTube QoE KPIs, i) Stall, ii) Current Quality, iii) Video Bytes Downloaded with Player Events (Chapter 5). We collect the commercial 4G and 5G datasets over different regions (France, Brazil and USA) with different use cases (Mobility, Pedestrian, Indoor, Outdoor). We infer objective QoE KPIs - Stall of YouTube using Channel Level Metrics (CLM), i.e., CQI, RSRQ, RSRP, and the features derived from them. The 4G and 5G datasets with context, CLM, and QoE logs of the most popular streaming platform YouTube are ideal for the research community. Moreover, the dataset can be used in `EFFECTOR` to emulate different use cases of 4G and 5G, while knowing the context, CLM, and QoE.

Chapter 3

DASH QoS to QoE Evaluation Frameworks for 4G & 5G

3.1 Introduction

5G supports significantly high bandwidth content with speeds over 10 GB/s, ultra low (i.e., 1-millisecond) end-to-end over-the-air latency, real-time information transmission, and lower network management operation complexity [2] compared to 4G. The key challenge of streaming video 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 [62].

The impact of end-user QoE for multimedia traffic ultimately depends on underlying network-level QoS performance. QoE represents the user perception of the quality of a provided service, whereas QoS relates to network quality indicators (e.g., latency, packet loss).

In HTTP Adaptive Streaming (HAS), the choice of the ABS algorithm plays a significant role in end-user satisfaction [68]. HAS allows graceful adaptation of video quality during the playback through the segmentation of video content. Different types of adaptation algorithms are listed in Table 3.1. In recent years, many ABS algorithms have aimed to provide interrupt-free videos and the maximum achievable video quality. These ABS algorithms work on the principle of calculating network conditions and utilizing the maximum resources, thus providing 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 less quality switching. Other algorithms aim to provide high-quality content, utilizing more network resources, irrespective of the number of stalls (freezing). Ultimately, the main goal of all ABS algorithms is to provide the best

[1] https://bit.ly/3RefeQf

Table 3.1: ABS algorithm adaptation mechanisms.

Algorithm	Type	Mechanism
Arbiter + [63]	Hybrid	The quality selection policy includes targeting a reduction of the stall risk by performing smooth representation switches.
Elastic [64]	Hybrid	Is designed to ensure application-level fairness in the case of sharing a bottleneck.
BBA [65]	Buffer-based	Represents the class of algorithms that solely depends on buffer-level in the adaptation decision.
Logistic [66]	Buffer-based	This model is able to find the optimal buffer required for any given set of video quality levels.
Conventional [67]	Rate-based	The TCP download throughput observed by a client is directly taken as its fair share of the network bandwidth.
Exponential	Rate-based	Exponential growth of past throughput.

QoE to end users.

With the exponential growth of mobile data and smart devices, the investigation of 4G and 5G QoE in video quality assessment has become a research focus in both industry and academia. Video perceived quality in the network is critical; thus, various methods have been used to optimize video delivery over the networks, such as video compression and better resource utilization [69, 70].

In 5G networks, QoE management is crucial as the estimation and resource allocation for better video quality should be completed quickly. Therefore, it is necessary to understand the relationship between different ABS behavior, its metrics for QoE, and network-level QoS compared to 4G.

Moreover, TLS encryption establishes a more secure and private connection, where classic DPI techniques no longer provide valuable information [16, 1, 5, 22]. Due to the limited information available to inspect video flows, it is incredibly difficult to find novel lightweight QoS patterns affecting the video QoE that requires less computations and processing.

Thus we provide DASH QoS to QoE evaluation frameworks equipped with all the dependencies to run real 4G and 5G experiments both for encrypted and unencrypted video traffic with different approaches to find QoS features. We also provide real 4G and 5G datasets collected in France and USA to emulate the scenarios in the frameworks. We collected the commercial 4G and 5G datasets with different use cases, i.e., i) Mobility, ii) Pedestrian, iii) Static, iv) Static Terminals – Bus and Railway, v) Static Outdoor – Crowded. The dataset is also equipped with Channel Level metrics, e.g., CQI, RSRQ, RSRP, SNR, Download Bitrate, and Upload Bitrate, among 100 + Channel Level metrics, which we explain in Chapter 5 of this thesis.

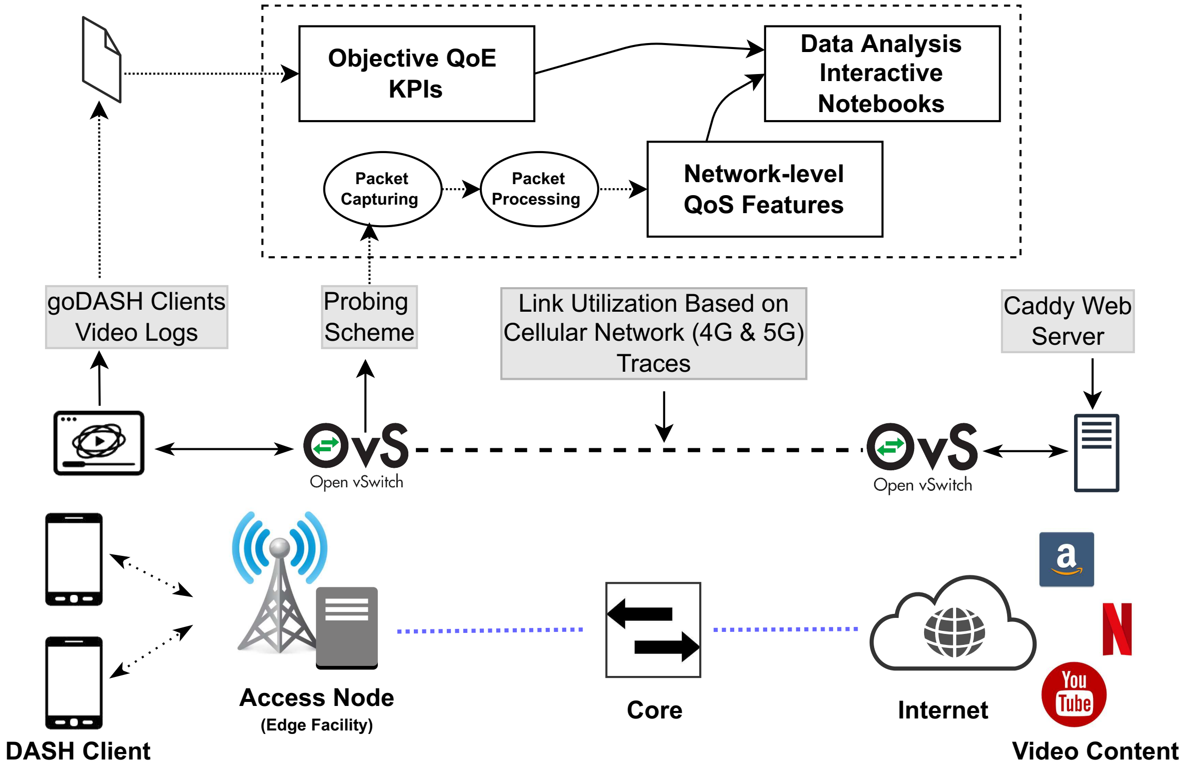


Figure 3.1: DASH QoS and QoE evaluation framework architecture.

3.2 Framework Design and Implementation

Figure 3.1 shows a synopsis of the frameworks. We present multi-user reproducible frameworks containing (i) godash - an ABS video player [71], (ii) Caddy - A Web Server Gateway Interface (WSGI) web server hosting DASH video content, (iii) Mininet-Wifi - a wireless network emulation environment [72], (iv) Scripts - Bash scripts to apply the 4G and 5G bandwidth values at run-time, bandwidth values are taken from 4G and 5G datasets [24, 23], where we use download bitrate of the streaming sessions as a bandwidth and Python scripts to process the QoE/QoS logs created during experimentation.

3.2.1 Per-segment: DASH QoE Performance Evaluation Framework

In the first step, we provide a Framework to investigate unencrypted traffic with different 4G and 5G use cases. We provide an in-depth analysis of state-of-the-art ABS algorithms, such as Rate-based — Conventional [67] and Exponential, Buffer-based — Logistic [66] and BBA [65], and Hybrid — Arbiter+ [63] and Elastic [64]. The adaptation mechanism is listed in Table 3.1. The analysis is undertaken by assessing associated QoE models such as the P.1203 QoE standard and four other QoE models named Clay, Duanmu, Yin and Yu. We consider this work a first step to developing a framework to emulate different use cases with current state-of-the-art network scenarios, e.g., without encryption. The intuition behind the work is to understand different ABS algorithms under different network use cases.

To emulate the HTTP video streaming session, we use a popular 2-second segment duration x264 animated video titled *Sintel*,[2] sourced from a publicly available 4K DASH video dataset [73]. We select a 2-second segment because short segments allow quicker bandwidth changes. When longer segments are used, the client is not able to adjust as flexibly and quickly as it would be possible with shorter segments. We assess the impact of 2 and 3 concurrent clients streaming from the same server. We select ten combinations of Mobility and Static (in Mbps); Mobility — (0.5 - 3), (6 - 14), (38 - 10) and (29 - 10); Static — (0.5 - 6), (8 - 57), (4 - 7.6), (52 - 0.5), (70 - 20) and (72 - 9).[3] We choose different use cases to see the QoS impact on video, i.e., i) Case with very Low bandwidth, ii) Case with Medium level QoS resource – Bandwidth, iii) Case with High Bandwidth. Link bandwidth is in increasing order, i.e., if the range is 0.5-3 Mbps, then the bandwidth, which we change every 4-second is > 0.5. Similarly, the values are in decreasing order for the use cases, which are 38-10 Mbps. The intuition here is to see the impact of QoS – bandwidth on the QoE for all ABS algorithms and how they respond to increasing and decreasing QoS – bandwidth. Note: QoS – bandwidth or sometimes QoS both refer to link utilization based on cellular network 4G and 5G traces between two switches in the setup of frameworks.

The bandwidth during each experiment is changed in real-time between Switch 1 and Switch 2 link as shown in Figure 3.1 using Linux Traffic Control (TC) and Hierarchical Token Bucket (HTB) [74]. A python script is used to collect per-run pcap by `tcpdump`.[4] Later, the python Scapy package is used to get per-segment QoS features from pcap.

Table 3.2 illustrates an example of a `godash` log file for a single client in the Mobility (driving) scenario using (6 to 14) Mbps, with each line representing per segment metrics for the conventional ABS algorithm. Detailed information on each feature and ABS algorithms are available in `godash` [75].

We fetch per segment QoS metrics (RTT, Throughput and Packets) from the pcap files. We merged the QoS metrics and godash logfiles output as a single CSV dataset (example presented in Table 3.3). The first two columns present the network context for each experiment, i.e., (Total_Users, Host) indicated as total users competing for the video stream and host number. The next column has Segments followed by three video KPIs (Stall, Bitrate and Buffer level) of each corresponding segment. The QoS features extracted from pcap traces of each segment are indicated as (RTT, Throughput, Packets) and finally, the five QoE models provided by `godash`.

The framework encompasses a DASH streaming environment, the pre-processing of the network, video client logs and associated scripts. For ease of use, the framework

[2] https://bit.ly/3eTkLxE
[3] 10 5G real cases: https://github.com/sajibtariq/dashframework/tree/master/Testbed/5g_traces
[4] https://www.tcpdump.org

Table 3.2: *Conventional*: goDASH log file of first 5 video segments, Case *Mobility* (6-14) Mbps.

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

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

includes a Virtual Machine (VM) [76] with all software and dependencies installed. The VM provides all tools and the environment needed to stream DASH content in a multi-user realistic 4G and 5G network. Currently, the VM showcases a single combination of mobility, host competition, and link bandwidth parameters to run the `Mininet-WiFi` emulated topology, collect `godash` log(s), pcap file(s), and process the raw video logs and network data as shown in Figure 3.2. The framework offers to emulate network links with Mininet-WiFi, enabling reproducing sophisticated network scenarios. Since Mininet-WiFi is forked of Mininet, thus, using the framework, we can use customized network topology that supports wired connection or wireless (e.g., WiFi) connection.

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 [77]. Note that we have released all remaining code used for processing the dataset for reproducibility.[5] The computational scripts and utilities are already available in the VM.

The `Jupyter` notebook and CSV dataset are uploaded from the VM to `GitHub` and through a live dynamic `Binder` service, we can interact, analyze and visualize the input dataset. To visualize your own data, the easiest option is to fork our repository [78] and upload your data to the forked version of it. Figure 3.3 highlights the outline and design of the Binder service, while Figure 3.4 illustrates some of the features that can be selected to update and revise the output plots.

We now show QoS features RTT, Objective QoE Stall and QoE model P.1203. We consider two use cases to present results, scenario a) – 6-14 Mbps, scenario b) 0.5 - 3 Mbps. These use cases among others are used in DASH framework for the generation of datasets to see the impact of QoS on QoE along with ABS behavior. Figure 3.5 shows RTT,

[5] https://github.com/sajibtariq/dashframework

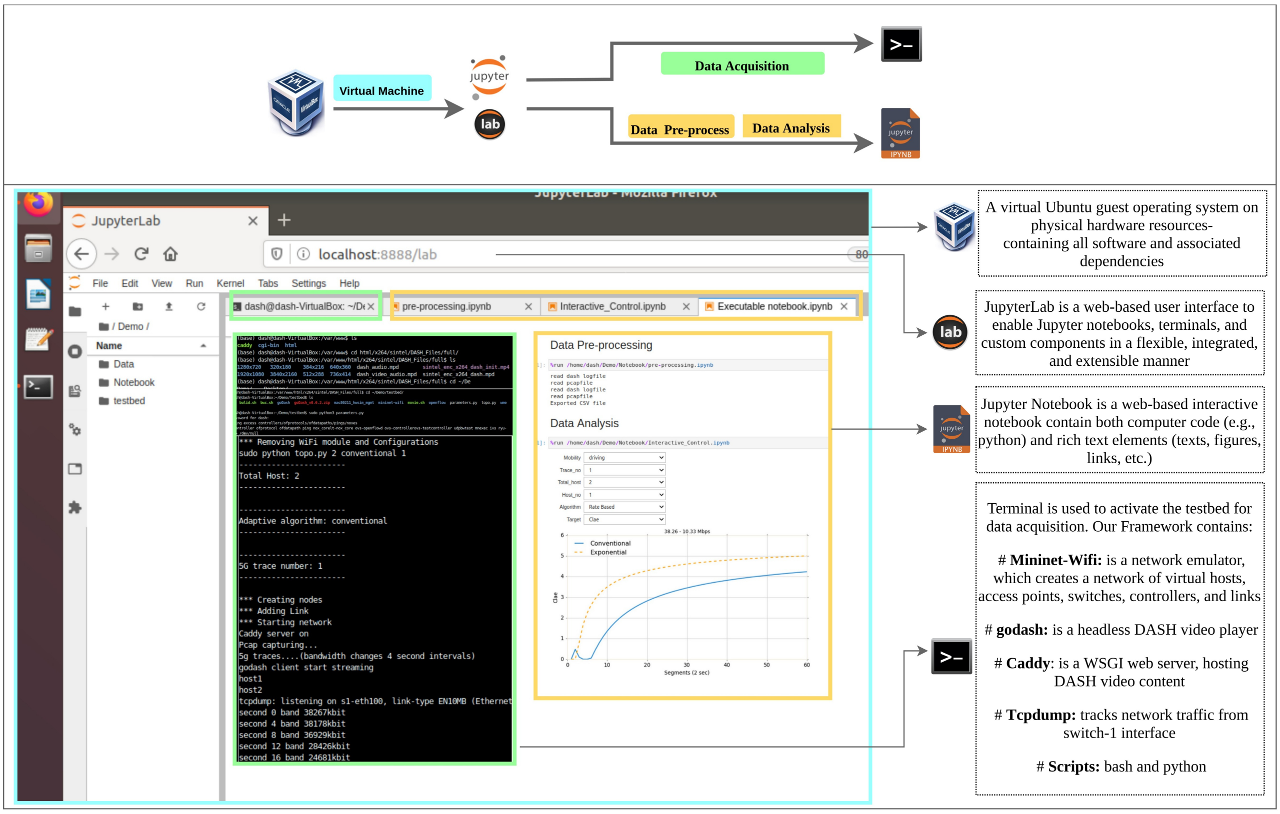


Figure 3.2: An Ubuntu 18.04 VM including a DASH streaming environment containing: `Mininet-Wifi`, `Jupyter` lab and notebook, `godash` player, `Caddy` server and DASH content, `tcpdump`, and scripts.

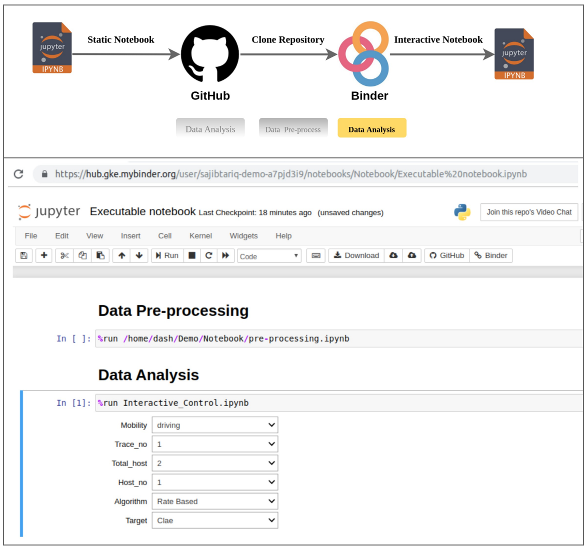


Figure 3.3: `Binder`, turns the `Github` notebook into an interactive notebook in an executable environment for data analysis.

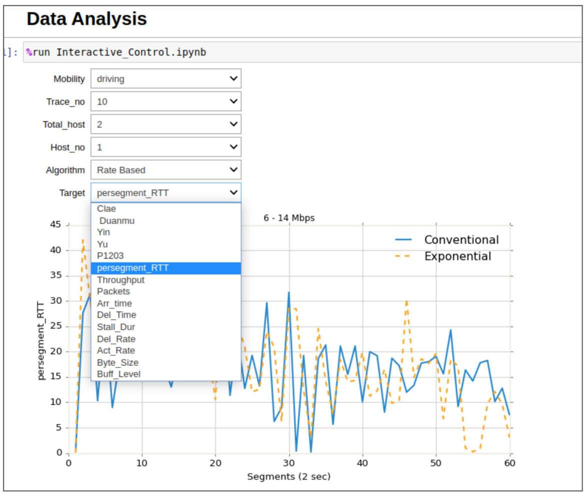
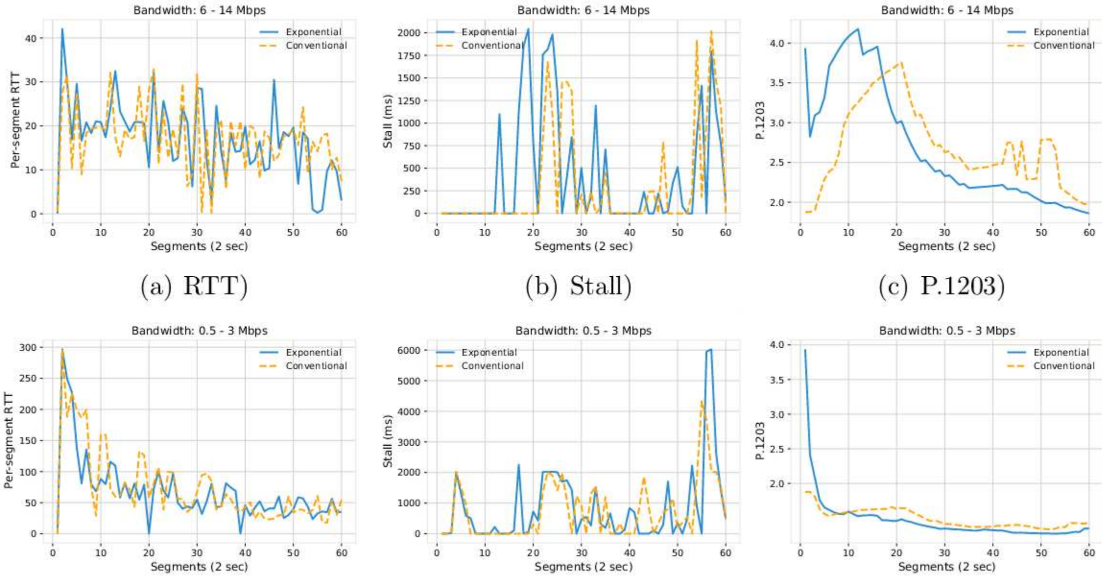


Figure 3.4: First user experience (mobility) with Conventional and Exponential ABS algorithms over (6-14) Mbps. Per-segment QoS RTT on (y-axis), 60 segments on (x-axis).

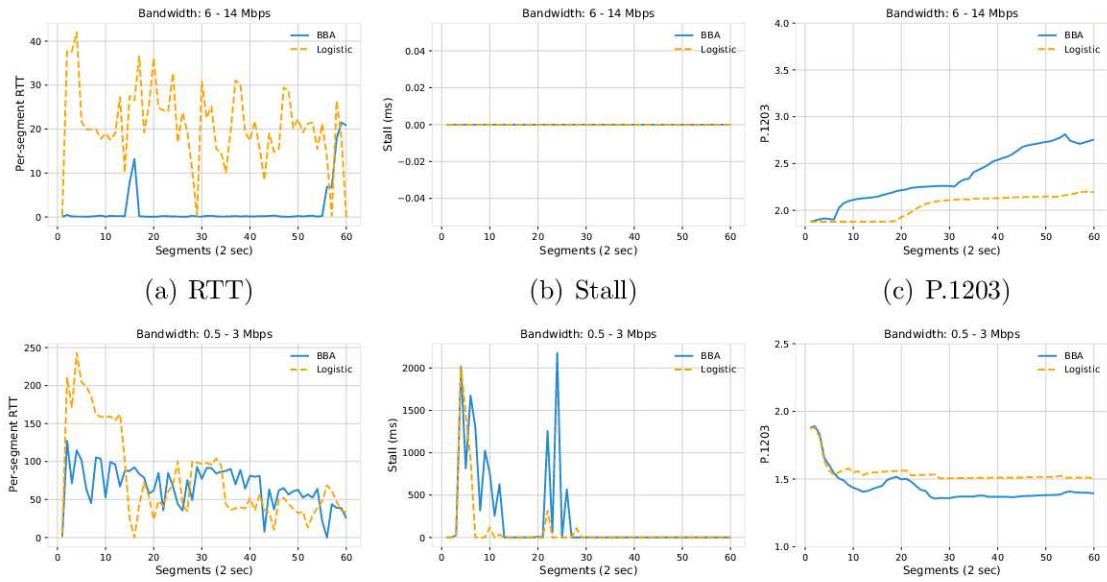
number of stalls, and P.1203 score, with an overall slight advantage for Conventional in scenario a). In the Moderate scenario, Exponential achieves a small advantage in overall segment bitrate, i.e., achieving higher visual video quality. It is important to note that in



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

(d) RTT) (e) Stall) (f) P.1203)

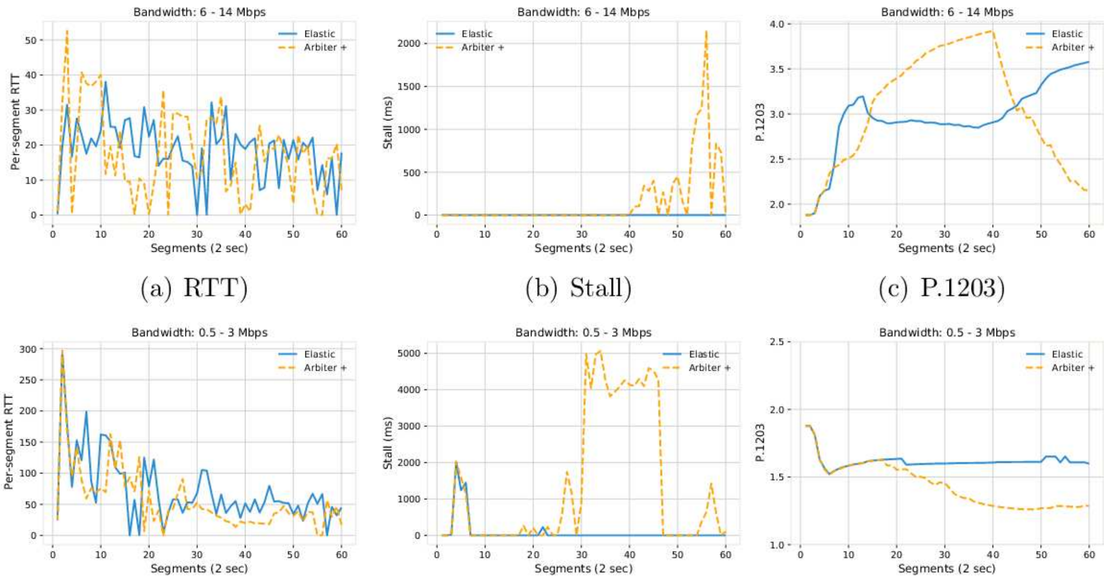
Figure 3.5: Rate-based – RTT, stalls, and P.1203 score per video segment for 60 video segments.



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

(d) RTT) (e) Stall) (f) P.1203)

Figure 3.6: Buffer-based – RTT, stalls, and P.1203 score per video segment for 60 segments.



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

(d) RTT) (e) Stall) (f) P.1203)

Figure 3.7: Hybrid – RTT, stalls, and P.1203 score per video segment for 60 segments.

Figure 3.5, both algorithms constantly make the wrong choice of representation bitrate (achieving an overall higher representation level of video quality), which can be seen in the levels of stalls in both test scenarios.

Figure 3.6 shows that although Logistic does not report any stall in scenario a), video segments have considerably higher RTT and overall lower P.1203 score. However, scenario b) reports that both algorithms are comparably similar, with BBA reporting overall lower P.1203 scores thus translated into lower video resolution. It can also be seen in Figure 3.6, that in moderate bandwidth scenarios, fluctuations in the bandwidth can cause the buffer-based algorithms to make the wrong choice. Highlighting the need for a predictive model for adaptive algorithms in cellular networks.

However, looking at Figure 3.7 higher segment bitrate comes at the cost of more stalls that also result in lower P.1203 scores translated into lower video quality. In other words, one can conclude in both scenarios, a) and b), that Arbiter+ tries to maintain higher segment bitrate, even though it experiences more stall events and lower P.1203 scores as a direct consequence.

The main difference between buffered and throughput based adaptation is that buffered ABS algorithms mostly rely on buffer size and future segment sizes and is quite conservative – if the buffer is low, it will request the low quality. However, throughput based ABS algorithms, on the other side, are riskier. If it sees that the delivery rate is high and the variance of delivery rate in the past is low, it will take advantage of this situation and will request a higher video quality [47].

Limitations: The intuition behind the per-segment QoS features of an unencrypted video stream, i.e.,(Throughput, Packets, Per-segment RTT) is to provide an in-depth analysis of different state-of-the-art ABS algorithms such as (BBA – Buffered, Conventional – Throughput, Elastic – Hybrid) using 4G and 5G datasets. However, TLS encryption and Deep Packet Inspection (DPI) is the main limitation of the framework. Therefore, next, we jump to provide a framework with recent technologies.

3.3 EFFECTOR: DASH QoE and QoS Evaluation Framework For EnCrypTed videO tRaffic

Deep packet inspection, in spite of being a widely used solution to estimate the KPIs directly from network traffic, is not a convenient solution anymore due to the adoption of end-to-end video streaming-encryption over TCP (HTTPs) and QUIC transport protocols [22, 16, 1, 5]. Therefore, we can only rely on the information available in the IP packets header, such as Packet Time and Packet Size, to extract the QoS KPIs and their pattern to infer QoE. Therefore, in EFFECTOR, we consider encrypted video streaming sessions, where we extract QoS KPIs, and showed QoS interconnection with objective QoE KPIs such as stall and resolutions. In EFFECTOR, we consider the aforementioned design and implementation see Figure 3.1, by supporting end-to-end encrypted video traffic and providing a different lightweight, fine-grained QoS feature extraction.

However, we use three videos of 2-seconds segment duration x264 videos named Sintel, Tears of Steel and Big Buck Bunny sourced from a publicly available 4K DASH video dataset [73] in EFFECTOR. We use 4G and 5G drive test traces [24, 23] to generate a rich dataset. The trace is generated with 1-second granularity (window) in different scenarios such as i) Bus – Mobility, ii) Pedestrian – Low Mobility, iii) Static Download – Downloading a large file continuously for a fixed time period. During the trace and data collection campaign 1-user is considered for streaming and downloading the content. We consider static download scenarios to emulate them in the framework, which are synonyms for the maximum capacity of the network, i.e., (4G and 5G). Inside the framework, link capacity as per 4G and 5G traces during each experiment are changed in real-time using Linux Traffic Control (TC) and Hierarchical Token Bucket (HTB) [74]. We use the time granularity of 1 second for the generation of the QoE and QoS dataset in the framework by using real 4G & 5G traces [24, 23]. The aforementioned time granularity is also ideal for the detection of anomalies, troubleshooting approaches, as well as proactive traffic management [1]. The contributions of EFFECTOR can be summarized as follows:

- EFFECTOR provides a total of 30+ QoS features captured in each 1-second time window. We also show the interdependence of QoS features derived from EFFECTOR with QoE in Section 3.3.2 by taking real 4G and 5G use cases. An explanation of time window and QoS features is discussed in Section 3.3.1. However, the framework

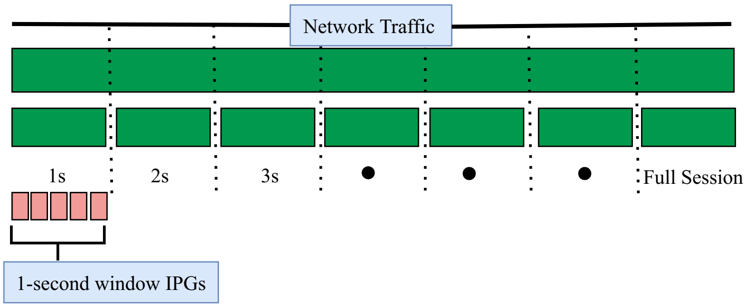


Figure 3.8: IPGs of 1s window.

is flexible, using different time windows for QoS, video contents, drive test traces, and host numbers to stream the same content from a specific server.

- We also provide a total of 14 QoE features generated from video QoE logs of each experiment, which include different objective QoE metrics and QoE model outputs. In Section 3.3.1.4 we explain each metric.

- Moreover, we derived QoS features from inter packet gap named EMA, DEMA and CUSUM. These QoS features are used to check the continuity of data packets from the mean value. Therefore looking only at packet times provides new features to estimate QoE.

- Finally, this work offers a massive generated dataset that includes QoS and QoE KPIs from the framework yielded video traffic using 4G and 5G drive test traces. We provide IJN to analyze the pre-process datasets obtained over TCP (HTTPS) and QUIC transport. These analysis Notebooks contain QoS features fetched from encrypted network traffic with a time granularity (window) of 1-second and per-segment QoE metrics from the video log files.

3.3.1 QoS Features Extraction Approach

3.3.1.1 Inter Packet Gap – IPG

We consider a given set of network flows $f \in \mathcal{F} = \{f_1, f_2, ..., f_M\}$, where M represents the total number of flows. The IPG metric is updated every time a network packet gets into the switch pipeline. When a packet enters the switch, the last seen ingress timestamp ($TS_f^l \in \mathbb{N}^+$) is subtracted from the current timestamp ($TS_c \in \mathbb{N}^+$) to calculate the current IPG_f^c estimator of network flow f, [79] (equation 3.1). We extracted QoS features in the form of time windows, i.e., (0.5, 1,...,5) seconds. Thus IPG of a single window is used for various other QoS metrics see Figure 3.8.

$$IPG_f^c = TS_c - TS_f^l, \qquad f \in \mathcal{F} \tag{3.1}$$

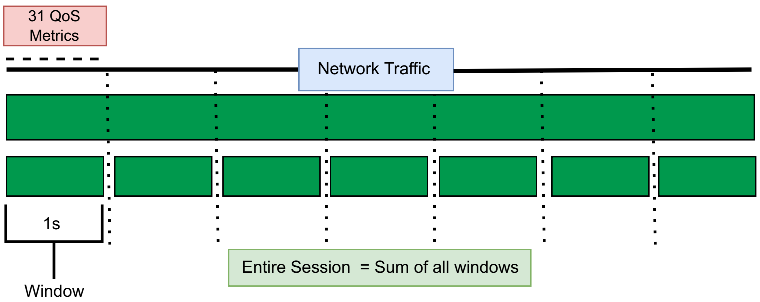


Figure 3.9: Time windows (temporal resolution, QoS window).

3.3.1.2 Time Window

EFFECTOR continuously extracts features from the encrypted stream of packets in a stream-like manner, as shown in Figure 3.9. Features are based on packet-level statistics and their computation does not require a chunk-detection mechanism. Finding chunk-level features in encrypted network traffic is a research challenge. Moreover, video content providers change their delivery methods over time, making chunk-level statistics prediction a non-trivial task [47, 1]. Therefore, we consider a window based QoS features extraction method where we split the entire QoS metrics collection approach into small windows. As shown in Figure 3.9 the concatenation of all window's QoS is equal to the QoS of the entire DASH video streaming session.

3.3.1.3 QoS Features

In Table 3.4, we provide QoS features extracted from encrypted network traffic. IPG presents Inter Packet Gap, IAT as Inter Arrival Time of window, i.e., (1-second), which means the difference between the last and first packet in the window. Next, we have IPG_avg and IPG_avg > 100, which is the IPG average of a window for all packets and the IPG average of those packets whose length is greater than 100B (Bytes), followed by total packets and packets whose length is > 100B. We are differentiating acknowledgment (control) packets from data packets using the method of packet size, such as size > 100B in TCP, however, we also used the same threshold for QUIC. Next, we have the standard deviation of packets sizes and IPG of a single window. Then we have throughput and throughput distribution into (10-90)-percentile in window, i.e., 1-second, followed by packet size distribution into (10-90)-percentile.

In Algorithm 1, the first five lines initialize variables such as the video streaming session – `Session (n-minutes)`, current time – `Ctime`, QoS features extraction time – `Step`, Total number of packets – `Tpackets`, and Total packets with size greater than 100B – `Tpackets_GT100`.

Next, in `While` loop, we store `Packet_size` and `Packet_time` in two separate `arrays` for later use. Next, on line 9, we convert `Packet_size` into bits and assign all the bits to variable `Bits`. From lines 10-13 we check `Packet_size`, If it is greater than 100B, we store size and time in arrays named `Array_GT100_Size`, `Array_GT100_Time` respectively. Finally, on line 14 we increment the `Step` of a time window.

Next, On lines 16-18, we extract (10-90)-percentile of packet size from the array that contains packet sizes `Array_Size`. Then we have (10-90)-percentile throughput distribution in time intervals on lines 19-21. On line 22, we divide all numbers of bits by window size. Next, we find the total number of packets on line 23 by taking `Count` of array `Array_Size` followed by taking `Count` of packet sizes greater than 100 on line 24. We take Inter Arrival Time (IAT) of a window on line 25, which is the difference between the last packet and the first packet of a window. Next, on line 26, we find Inter Packet Gap (IPG) average of packets followed by the IPG average of packets whose length is greater than 100B on line 27. Next, on lines 28-29, we save the standard deviation of the packet sizes followed by the standard deviation of IPG on lines 30-31. At this point, we have 29 QoS features. We show all the QoS features extracted in Table 3.4.

Table 3.4: QoS and QoE features.

QoS	**QoE**
IPG	Arrival
IAT	Delivery
IPG_avg	Stall
IPG_avg > 100	Representation rate
Std_IPG	Delivery rate
Std_IPG > 100	Actual bitrate
Total Packets	Segment size
Total Packets > 100	Buffer
Std of Packet Size	Resolutions
Std of Pacaket Size > 100	P.1203
Throughput	Yin
TP (10-90)P	Yu
PS (10-90)P	Duanmu
EMA	Clae
DEMA	-
CUSUM	-

3.3.1.4 QoE Features

In Table 3.4, we show QoE features as well, where each feature represents information of a single segment from logs i.e., the arrival time of a segment in ms, time spent for delivery of current segment in ms, stall in ms followed by representation rate of the downloaded segment in Kbit/s (taken from MPD file). Next, the delivery rate of the

Algorithm 1 QoS features extraction approach for a time window of a video streaming session.

```
Session ← n_minutes                                   ▷ Streaming session
Ctime ← 0                                             ▷ Current time – Begin from 0
Step ← window                                         ▷ Step (1,2,3,4,5)-seconds
Tpackets ← 0                                          ▷ Total packets
Tpackets_GT100 ← 0                                    ▷ Total packets size > 100
while Ctime ≤ Session do
    Array_Size () ← Packet_Size
    Array_Time () ← Packet_Time
    BITS ← BITS + len(p[IP]) * 8)
    if Packet_Size > 100 then
        Array_GT100_Size () ← Packet_Size
        Array_GT100_Time () ← Packet_Time
    end if
    Ctime ← Ctime + Step
end while
while P = 10 to 90 do
    P percentile of Array_Size
end while
while P = 10 to 90 do
    P percentile of throughput in a time window
end while
Throughput ←BITS/window
Tpackets ← Count (Array_Size)
Tpackets_GT100 ← Count (Array_GT100_Size)
IAT_w ← Last − First                                  ▷ Last Packet Time - First Packet Time in a window
IPG_avg ← Average (IPG (Array_Time))
IPG_avg_gt100 ← Average (IPG (Array_GT100_Time))
Std ← STD (Array_Size)
Std_gt100 ← STD (Array_GT100_Size)
Std_IPG ← STD (IPG (Array_Time))
Std_IPG_gt100 ← STD (IPG (Array_GT100_Time))
```

network in Kbit/s (segment size divided by the time for delivery), the actual bitrate of this segment (segment size divided by the segment duration) in Kbit/s, and segment size in bytes. Next, the buffer level after the current segment was just downloaded, in seconds followed by resolutions of the segment. Finally, we have five QoE models named P.1203, Yin, Yu, Clae, and Duanmu `godash` [47], [75].

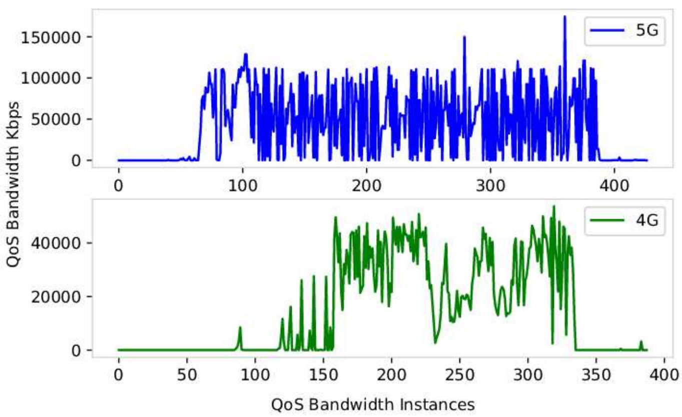


Figure 3.10: 4G and 5G use case used from commercial 4G and 5G dataset to emulate in EFFECTOR.

3.3.2 QoS Features Interdependence with QoE

This section explains the interrelation of QoS features derived from EFFECTOR with the objective QoE stall and quality shifts. To do that, we use a use case of commercial

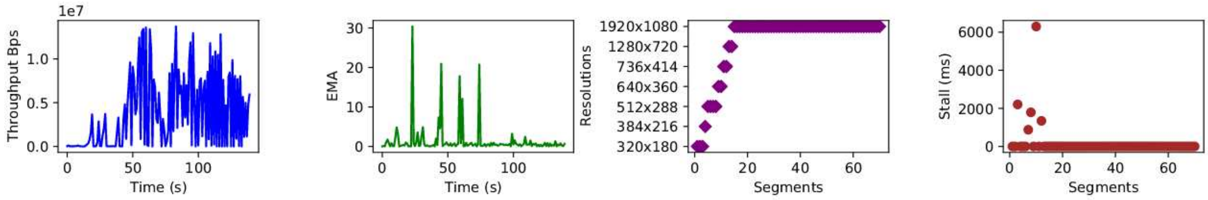


Figure 3.11: 4G: QoS interdependence with objective QoE – stall and quality shifts using BBA – ABS and a window size of 1-second.

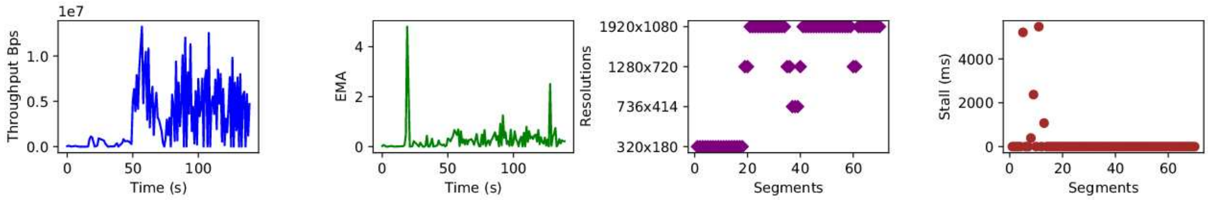


Figure 3.12: 4G: QoS interdependence with objective QoE – stall and quality shifts using Elastic – ABS and a window size of 1-second.

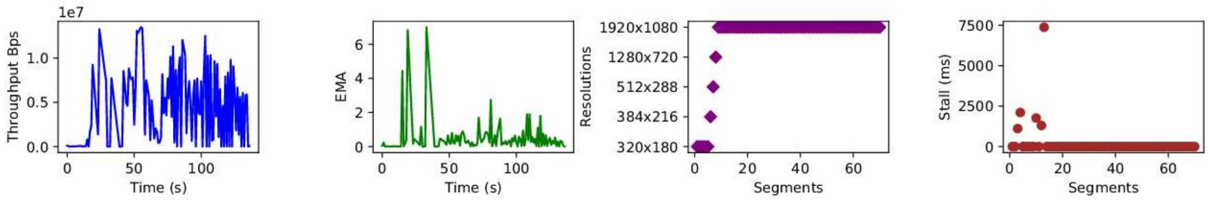


Figure 3.13: 4G: QoS interdependence with objective QoE – stall and quality shifts using Conventional – ABS and a window size of 1-second.

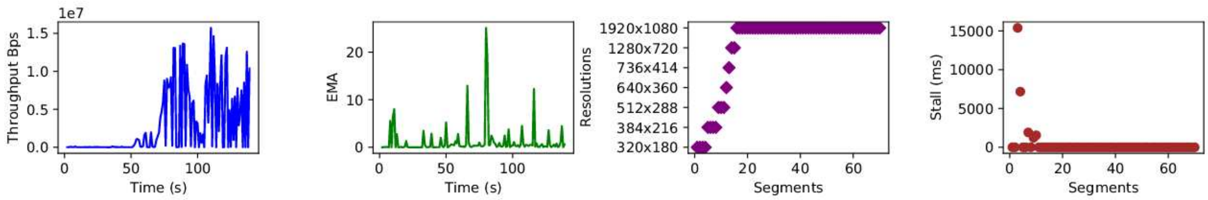


Figure 3.14: 5G: QoS interdependence with objective QoE – stall and quality shifts using BBA – ABS and a window size of 1-second.

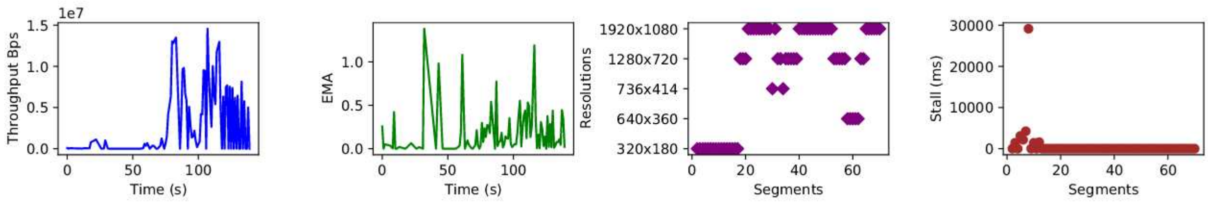


Figure 3.15: 5G: QoS interdependence with objective QoE – stall and quality shifts using Elastic – ABS and a window size of 1-second.

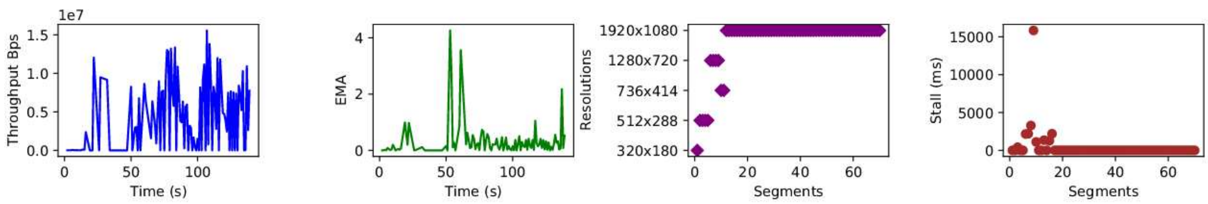


Figure 3.16: 5G: QoS interdependence with objective QoE – stall and quality shifts using Conventional – ABS and a window size of 1-second.

4G and 5G datasets collected in the wild with CLM and emulate them in EFFECTOR see Figure 3.10. The dataset is collected in France over a period of six months with YouTube as a baseline. We collected 100+ CLM and objective QoE of YouTube with the smallest granularity of 1 second. We choose different use cases i) Pedestrian, ii) Mobility, iii) Indoor, and iv) Outdoor. We use G-NetTrack Pro and YouTube IFRAME API.[6] [7] We saved both QoE logs from the player such as i) Stall, ii) Video Bytes Downloaded, iii) Current Quality – Resolution, and iv) Time, along with Player Events such as 3 – Stall, 1 – Playing, 0 – Ended, 2 – Paused, 5 – Cued, -1 – Unstarted. The complete explanation of each feature and broad objectives are explained in Chapter 5.

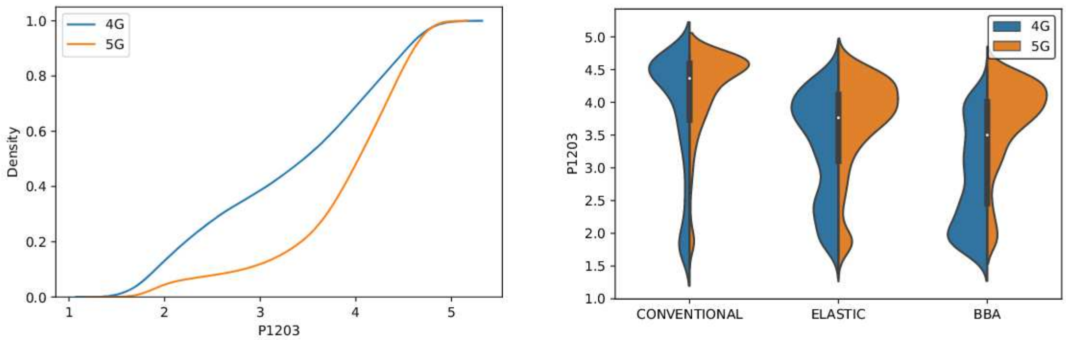
We select two use cases from 4G and 5G, where we experience real stalling events and quality shifts (Change in Resolutions). Use cases are embedded in the framework and are also available on GitHub with QoE of the real YouTube traffic, 100+ CLM, and QoS and QoE of the emulation in the framework.

In Figures 3.11, 3.12, 3.13, we show two QoS metrics Throughput and EMA with two QoE metrics Quality shifts score and Stalls using protocol – TCP, video – Tears of Steel and Segment – 2s. On the x-axis of two QoS features, there is time in seconds. We show a 1-second window QoS pattern. Whereas on the x-axis of two QoE metrics, there are video segments. Y-axis on each case shows the QoS and QoE values. We observe during each case of ABS, throughput and EMA values show a positive relationship with shifts and objective QoE KPI stall. The first 50 seconds of the streaming session suffer from low QoS resources, thus, causing low throughput and less QoE. Low throughput has a strong relationship with shifts and thus causes stalling events during the first 20 segments of the streaming session. We consider a 2-second segment, thus 20x2=40-second of the streaming session. We can also see a few peaks of EMA values in the first 45-50 seconds of the streaming session, which shows the interrelation of the QoS feature derived from IPG with quality shifts and stalls. Similarly, in Figures 3.14, 3.15, 3.16, we show 5G use case results when we emulate it in "EFFECTOR". Overall we observe ABS `Conventional` shows more stability with quality shifts.

3.3.3 4G vs. 5G Performance Footprint Through Open Datasets

We start by comparing the P.1203 score collected over the 4G and 5G technologies. We select Video – Sintel and Protocol – TCP to compare 4G and 5G. We show three QoE models i) P.1203, ii) Clae, and iii) Duanmu. We show the Cumulative Distribution Function (CDF) of the QoE models mentioned earlier for both technologies. We observe 5G outclass 4G see Figure 3.17 (a). 80 % of the QoE P.1203 score for 4G remains 4.0; however, in the case of 5G, we observe a much higher QoE of 4.5. We notice a similar 5G

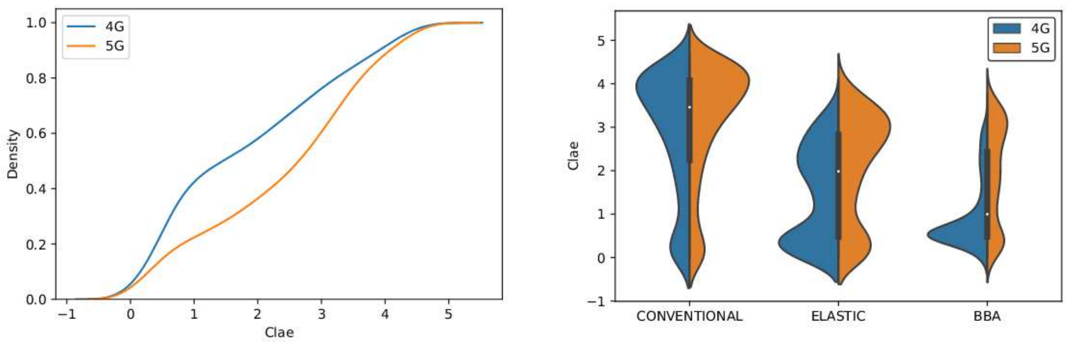
[6] https://bit.ly/3MU0Rj0
[7] https://bit.ly/3DiAuQD



(a) QoE P.1203

(b) QoE P.1203 by ABS

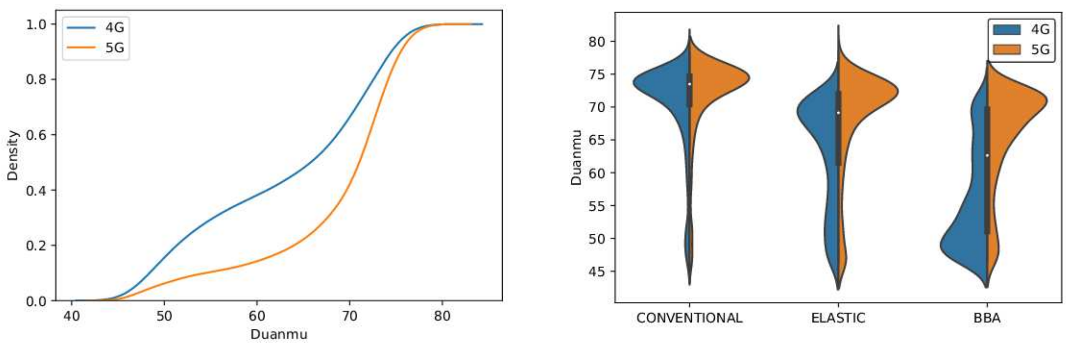
Figure 3.17: QoE model P.1203 in 4G vs. 5G.



(a) QoE Clae

(b) QoE Clae by ABS

Figure 3.18: QoE model Clae in 4G vs. 5G.



(a) QoE Duanmu

(b) QoE Duanmu by ABS

Figure 3.19: QoE model Duanmu in 4G vs. 5G.

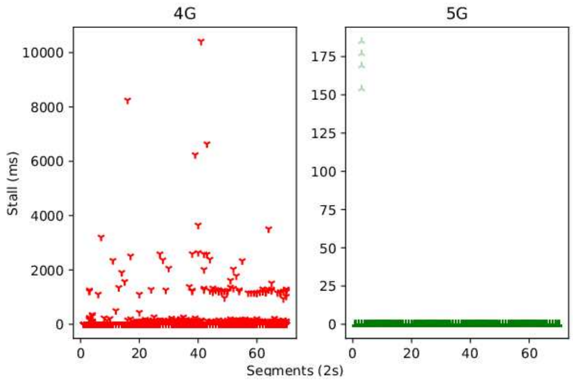


Figure 3.20: Objective QoE stall in 4G and 5G.

dominance in all cases of ABS see Figure 3.17 (b). Similarly, in Clae, we experience that 5G has a better overall QoE score than 4G see Figure 3.18 (a) and QoE model score by ABS see Figure 3.18 (b). We show the QoE model Duanmu in Figure 3.19 for both the technologies 4G and 5G in Figure 3.19 (a) and by ABS in Figure 3.19 (b).

Moreover, In Table 3.5 and Table 3.6 we show the Mean, Standard Deviation, Min, Max, 25 %, 50 %, 75 % and Max value of throughput in Mbps for both the technologies 4G and 5G for the protocols TCP and QUIC respectively. 5G provides much better QoS than 4G for all the combinations of use cases.

Table 3.5: Throughput (Mbps) across 4G and 5G on different use-cases (video – Sintel) TCP.

Technology	**Mean**	**STD**	**Min**	**25 %**	**50 %**	**75 %**	**Max**
4G	3.26	3.90	0.000416	0.17	1.41	5.60	19.39
5G	5.10	4.40	0.000416	0.625736	4.90	7.88	19.76

Table 3.6: Throughput (Mbps) across 4G and 5G on different use-cases (video – Tears) QUIC.

Technology	**Mean**	**STD**	**Min**	**25 %**	**50 %**	**75 %**	**Max**
4G	1.75	2.17	0.000416	0.39	1.05	2.20	15.08
5G	6.32	6.03	0.000416	0.08	5.02	11.99	25.40

Table 3.7: Percentage of segments resolutions in 4G & 5G.

Technology	**320x180**	**384x216**	**512x288**	**640x360**	**736x414**	**1280x720**	**1920x1080**
4G	5.93	4.12	8.14	7.75	5.93	11.54	56.55
5G	4.21	1.05	2.10	1.16	1.15	2.66	87.63

In Table 3.7, we also show the percentage of a single quality – resolution for the 4G and 5G. We consider video – Sintel, Protocol – TCP using all ABS for a fair comparison. However, to see the ABS impact on the resolutions, we also provide Interactive Jupyter Notebook to visualize quality shifts during each video streaming session. We can see in

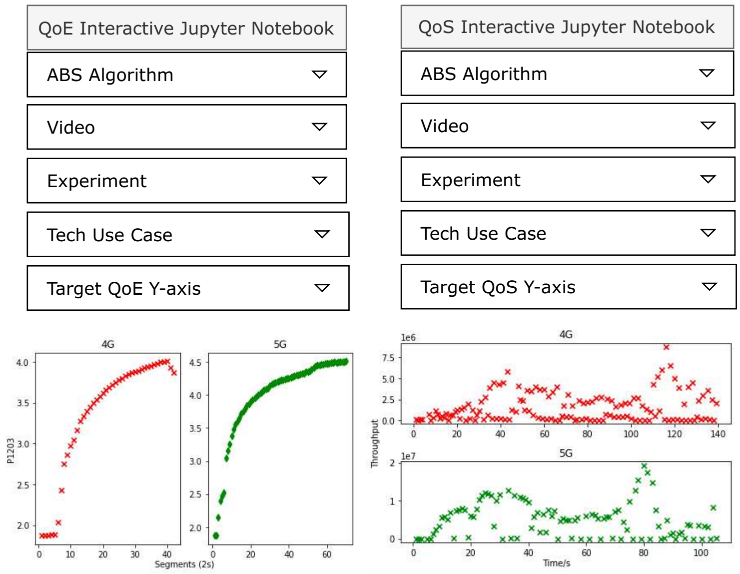


Figure 3.21: Interactive Jupyter Notebooks for the QoE and QoS.

5G that 87 % of the segments remain in 1920x1080 at higher resolutions, with very few segments in lower resolutions, whereas in 4G, 56 % of segments remain in 1920x1080, and there are frequent quality shifts during the session. 4G suffers from massive stalling events compared to 5G see Figure 3.20.

3.4 Real YouTube vs. Emulation Based QoE Experiments

In this section, we provide a comparison of DASH QoE experiments using `EFFECTOR` and experiments using real YouTube traffic in the wild. The methodology consists of collecting YouTube QoE KPIs in the wild and emulating them in the framework to draw a fair comparison. YouTube QoE and Channel metrics KPIs collection process is discussed in Chapter 5 of this thesis. The intuition is to see which ABS algorithm is close to YouTube streaming. Therefore, instead of doing experiments in the wild to draw a complex relationship between QoE and QoS, research can be done using emulation based experiments [30].

We show the comparison of different ABS algorithms, i.e., i) Buffered – BBA, ii) Conventional – Throughput, iii) Elastic – Hybrid with YouTube player. Our findings show that Conventional shows more similarity with a YouTube player in terms of quality shift and dominant resolution throughout the video streaming session. However, we are using the maximum bitrate of YouTube streaming session as a capacity of the experiments with `EFFECTOR`.

We select `Mobility` as a use case and emulate it in the `EFFECTOR`. We run the same video on YouTube and in "EFFECTOR" and show QoE KPI – Quality switching. Other use cases experience very less QoE events; therefore, our focus is on mobility. In 4G and 5G (Figure 3.22, 3.23), ABS `Conventional` is very close to YouTube player adaptation.

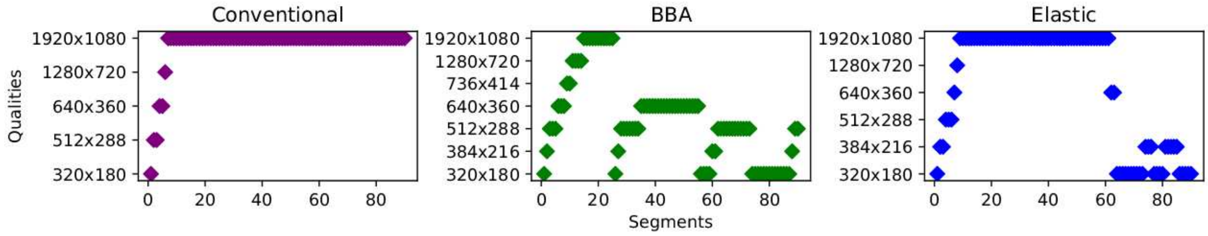


Figure 3.22: TCP: Quality shifts in all ABS with 4G mobility use case.

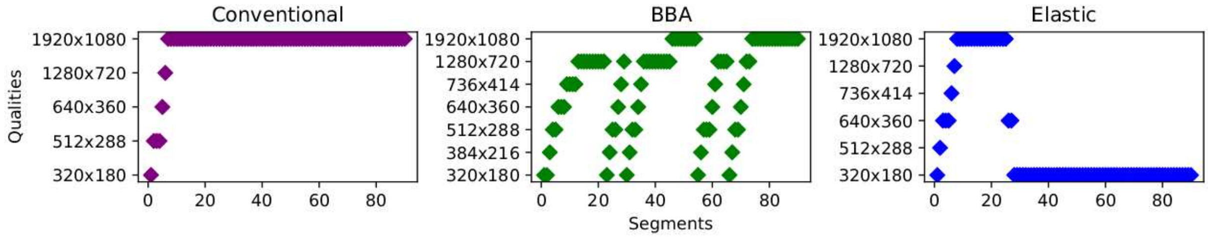


Figure 3.23: TCP: Quality shifts in all ABS with 5G mobility use case.

We observe few switching during the start of the streaming session, then a continuous quality throughout the session.

In 4G, 76 % of the time, YouTube streaming session remains in higher resolutions, i.e., (hd1080, hd1440, hd2160). In emulation, we experience conventional showing similar trends followed by elastic. However, unlike YouTube, BBA experiences massive shifting under such a use case. In 5G, 94 % of the dominant resolution is the maximum resolution in emulation, whereas we observe a similar pattern in a YouTube streaming session. YouTube streaming sessions use cases and video QoE after every 1 second is available on our GitHub Repository. [8] For TCP (4G and 5G), we use case. [9], [10] Similarly, for QUIC (4G and 5G), we use. [11], [12] The quality of experience of YouTube streaming sessions can be accessed from QoE logs file available on GitHub.

Next, in Figures 3.24, 3.25, we show the shift patterns (quality shifting) for protocol QUIC. We observe `Conventional` provides similarity with YouTube using QUIC as well. The ABS algorithms in `EFFECTOR` starts from the lowest resolution and keep updating the selection of the next segment based on current network conditions. We observe that `Elastic` and `BBA` experience frequent quality shifts. Compared to TCP, in QUIC these algorithms also remain at lower quality with very few segments in maximum resolutions.

In conclusion, `Conventional`, a throughput based adaptive algorithm, provides high similarity with the most popular streaming platform YouTube. Therefore, `EFFECTOR` an open-source DASH QoE and QoS evaluation framework is ideal for drawing a complex relationship between QoE and QoS using both protocols.

[8] https://github.com/razaulmustafa852/youtubegoes5g/
[9] https://bit.ly/3gNfKrB
[10] https://bit.ly/3gNLf4Y
[11] https://bit.ly/3GVcll3
[12] https://bit.ly/3ueBnE8

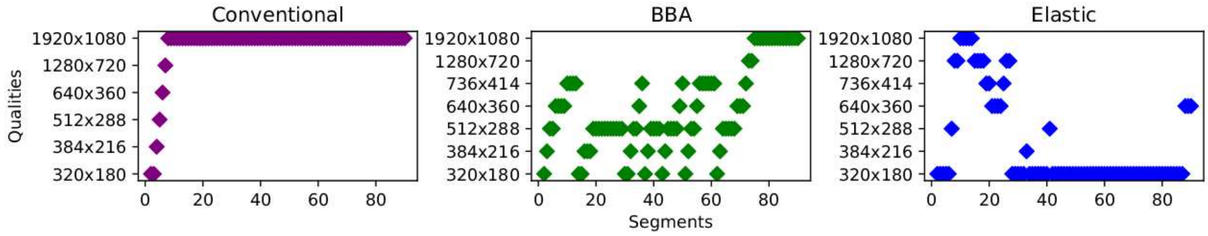


Figure 3.24: QUIC: Quality shifts in all ABS with 4G mobility use case.

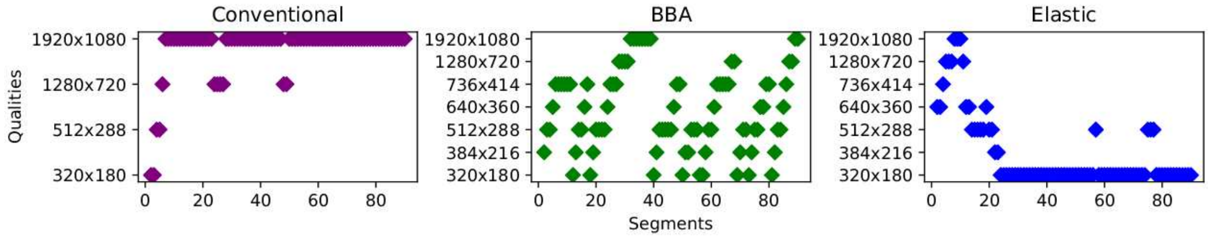


Figure 3.25: QUIC: Quality shifts in all ABS with 5G mobility use case.

In Section 3.3.2, we use use cases with interruptions for 4G and 5G. Not all the time 4G and 5G datasets collection campaign experience stalling events. However, 4G use-case selected in Section 3.3.2 experiences stalling events and quality shifts. Therefore we also compare a few other metrics such as i) Time spent with QoE degradation, ii) Percentage of maximum time of streaming in higher, iii) Percentage of maximum time in lower resolutions, iv) Stalling events, and v) Stall in maximum resolution during the streaming session. Table 3.8 shows QoE KPIs for Emulation and YouTube.

During a video streaming session, 88.4 % remains in a maximum resolution of `go-dash` player, whereas a YouTube streaming session remains 90 %. Therefore, `EFFECTOR` emulates the video streaming session with ± 2 % tradeoff between the QoE, such as resolutions. Moreover, we experience stalling events both in YouTube and Emulation based experiments.

Table 3.8: Conventional: goDash player vs. YouTube QoE KPIs.

Parameters	Emulation	YouTube
QoE Degradation	20s	20s
Max Resolutions	88.4 %	90.4 %
Low Resolutions	7.1 %	8.6 %
Stalls	Yes	Yes
Stalls at Max Resolutions	78 %	100 %

3.5 Virtual VM and Interactive Notebooks

3.5.1 Virtual Machine VM

For ease of use, we provide a Virtual Machine (VM) that is equipped with all these dependencies to run DASH video streaming experiments [80, 18]. We also provide interactive notebooks to play with the dataset. There are two types of notebooks available i) To see the impact of different 4G & 5G scenarios on different ABS algorithms performance, ii) Per-segment QoE logs and per-second QoS KPIs of a video see Figure 3.21. The `Jupyter` notebook and CSV dataset are uploaded to `GitHub`.[13]

3.5.2 QoE Interactive Jupyter Notebook

In order to visualize and see the impact of QoE and QoS in 4G and 5G technologies, we provide Interactive Jupyter Notebooks to see massive datasets with different use-cases.[14] For the evaluation of QoE, we provide Interactive Jupyter Notebook with nine objective QoE and 5 QoE Models, as mentioned in Table 3.4, and see Figure 3.21. In the first dropdown, select ABS algorithm – i) bba, ii) elastic, iii) conventional, followed by Video and Experiment. Note: We repeated each experiment 3-times with five different combinations of 4G and 5G use-cases. Next, select the 4G and 5G technology and the final selection is the target, which is nine objective QoE with 5 QoE Models (P.1203, Yin, Yu, Duanmu, Clae). The interactive notebook provides a subplot with 4G on the left and 5G on the right to compare both technologies. On the x-axis, we show segments which are a total of 70 and 70x2=140-seconds a duration of video streaming sessions. Whereas on the y-axis, we show the objective QoE and QoE Models.

3.5.3 QoS Interactive Jupyter Notebook

In this section, we provide Interactive Jupyter Notebook to visualize the QoS metrics of the dataset generated with different combinations of 4G and 5G technologies. We provide 29 QoS features derived with a window of 1-second granularity. The QoS features are mentioned in Table 3.4, and the Interactive Jupyter Notebook layout is shown in Figure 3.21. The QoS Notebook dropdown selection follows the same pattern as QoE Notebook. On the x-axis, we provide 1-second time granularity with a sequential increase up to the video streaming session time, which is 140 seconds. On the y-axis, we show the target QoS features.[15]

[13] https://github.com/razaulmustafa852/EFFECTOR
[14] https://github.com/razaulmustafa852/EFFECTOR/blob/main/iBooks/QoEInter.ipynb
[15] https://github.com/razaulmustafa852/EFFECTOR/blob/main/iBooks/QoSInter.ipynb

3.6 Frameworks Configuration

Table 3.9 illustrate a different type of use-cases to run DASH streaming session. The framework mainly operates on six combinations, as mentioned in Table 3.9. First, select the technology, i.e., 4G, 5G, to run the DASH video session that invokes a `Mininet`[16] python-based network topology (topo.py) consisting of two Open vSwitch (OvS) where first OvS is considered as edge point to capture raw network traffic using `Tcpdump`. Next, the second argument is the number of DASH clients followed by the ABS algorithm, i.e., (bba, conventional, elastic) that invokes the configuration of the `godash` and selects the ABS algorithm. Next, Trace selects the link capacity (e.g., throughput) values from the CSV file and invokes a bash script to change the link capacity after every 1-second between two OvS virtual interfaces. In the next step, select the protocol, i.e., (TCP, QUIC), which sets the available protocol for streaming. Finally, select the number of repetitions to run the experiment, e.g., [1,2,3,4,5] generates different folders with all experiment results.

To initiate execution, open a terminal in the directory of scripts and start (run.py). The DASH video streaming sessions will start and continue until the different combinations are set above. Each experiment will generate a folder in the current directory with the name 1_4g_Case_1_1_bba_tcp, which is Experiment – 1, Technology – 4G, Trace-Case – 1, DASH Clients – 1, ABS algorithm – bba and protocol – TCP, including video log and captured network traffic. Moreover, we provide two scripts to generate QoS logs by extracting QoS features from captured raw TCP and QUIC-based network traffic inside the VM.

Table 3.9: `EFFECTOR` configurations.

Parameter	**Example**
Technology	[4G, 5G]
Host	[1]
Algorithm	[BBA, Elastic, Conv]
Trace	[Use Cases]
Protocol	[TCP, QUIC]
Experiment Repetitions	[1,2,3,...,n]

Limitations DASH QoS to QoE evaluation frameworks are equipped with all the dependencies to run 4G and 5G use cases with commercial 4G and 5G datasets collected in the wild. However, we consider a few limitations of the frameworks. For instance, we consider three popular videos, Sintel, Tears of Steel, and Big Bug Bunny, more videos can be used to generalize the QoS features extraction approach. Moreover, we use a headless goDASH player, however; in reality, users are streaming video content from different OTT

[16] http://mininet.org/

Platforms, e.g., YouTube, Amazon, and Netflix. Moreover, devices also impact QoE, i.e., Mobile, PC, and Tablets. QoS is also impacted by various other factors, which include streaming using official apps, or by using Browsers – Chrome, Mozilla, etc.

3.7 Summary

In order to provide a better adaptive streaming service experience (e.g., QoE), network and service operators are required to assess the DASH user's perceived performance. However, the assessment approach on end-to-end encrypted network traffic yields challenges for network operators.

We presented adaptive streaming compatible frameworks with a QoS and QoE evaluation method to assess the user's perception of the DASH content when streamed through the encrypted and unencrypted network.

Our proposed in-band QoS feature engineering method in `EFFECTOR` is based on monitoring network traffic at edge nodes in near real-time, which does not require chunk-level inspection but rather observing the pattern of network packet arriving time and volume. As a result, this work produces a rich dataset of network-level extracted QoS, and application-level QoE features with a heterogeneous combination that is presented in the form of Interactive Jupyter Notebooks to visualize the trend of QoS and QoE. Such interrelation is highly relevant to network operators to review the service-level agreement for proactive network capacity planning and reactive QoE-aware network traffic management.

Chapter 4

DASH Video QoE Prediction Using Machine Learning Techniques

4.1 Introduction

In this chapter, we present Machine Learning solutions to estimate QoE with two types of solutions to consider. In the first step, we use per-segment QoS to QoE prediction using only (RTT, Throughput, and Total Packets) [13]. The proposed solution is ideal when you know the chunk level statistics of video streaming sessions. However, due to the increased end-to-end encryption, we need alternative solutions to use machine learning techniques when only a piece of limited information exists to derive QoS features.

Thus we provide a window-based QoS features extraction approach to deal with QoE. We find a novel metric named Inter Packet Gap (IPG), which we used to derive more features from it, i.e., EMA, DEMA and CUSUM. Usually, EMA and DEMA are technical indicators used to identify a potential uptrend or downtrend in time series data. We use these metrics to find where the continuity of data packets changes from the mean value. We used different statistical methods to reach up to 30+ downlink QoS features of encrypted video streaming.

Typically, ML problems can be classified as i) Supervised, and ii) Unsupervised. Supervised can be further divided into two types, i) regression, and ii) classification. During a typical classification problem, the ML models try to predict certain categorical classes, but in regression, the model outputs real values, such as integers or floating-point numbers, based on input variables.

4.2 Related Work

Several approaches were applied to QoE measurements in the literature, such as client-level, network-level for encrypted and unencrypted traffic, hybrid, and edge-centric measurement. Moreover, a couple of works were carried out for QoE performance evaluation over TCP and QUIC transport.

In Table 4.1 and Table 4.2, we show recent work using controlled experimentation. We highlight the difference between our work and the recent proposed work using window based QoS features. We also show the QoS features extraction method. In the past, studies used QoS features that were extracted in different time windows. In recent work [56, 31, 81, 82, 83] authors used three types of windows to extract QoS features usually named i) Current Window, ii) Session Window, iii) Trend Window. That requires more computational cost to store QoS features. However, we only rely on the current window to estimate QoE, which is comparatively faster than storing the QoS logs in three windows. Literature contributions are limited to a specific platform such as YouTube, which consider a specific algorithm for streaming video content. Models generated by previous approaches differ by different video content providers but also over time for the same video content provider. Therefore, the QoS features extraction approach in the form of time-windows is ideal for anomaly detection, troubleshooting, and proactive traffic management. It is also equally beneficial for MNOs to select among time windows. Moreover, we rely on QoS metrics named IPG and features derived from them such as EMA, DEMA, and CUSUM. We find the QoS flow features with other basic features such as Throughput and Packet Count can estimate QoE with less computational cost.

Apart from all the QoS feature extraction methods and techniques, we released DASH QoE evaluation frameworks, where the research community and industry can understand the QoS patterns and objective QoE for both the technologies 4G and 5G. We also provide a methodology to collect commercial 4G and 5G traces which we used in our frameworks. The real 4G and 5G datasets are not only ideal in the emulation environment but also provide channel level metrics, i.e., CQI, SNR, RSRQ, RSRP, among other 100 + features with 1-second granularity.

Table 4.1: Comparison to state-of-the-art work in time granularity.

Reference	Chunk Prediction	QoS Metrics	QoE Metrics	Reproducible	ABS	Streaming	Transport	Time Granularity
[56]	No	Network Level	Initial delay Stalling Visual quality Video bitrate	No	No	YouTube	TCP & UDP	1s
[31]	No	Network Level	Startup delay Rebuffering events	No	No	YouTube	TCP & UDP	10s
[81]	No	Network Level	Stall events	No	No	YouTube	TCP & UDP	1s
[82]	No	Network Level	Resolutions Bitrate	No	No	YouTube	TCP	1s
[83]	Yes	Application Level	Buffer warning Stall Resolutions	No	No	YouTube	TCP & QUIC	5s
This Work	No	Network Level	Stalls Bitrate Resolutions	Yes	Yes	goDASH Player	TCP & QUIC	0.5, (1-5)-sec

Table 4.2: A comparison between this work and ML-based real-time QoE estimation state-of-the-art works.

		[81]	[84]	[82]	**This Work**
Target QoE	**Real-time**	Stall Event (BC)	Resolution (MC)	Resolution (BC) Bit-rate (BC)	QoE Class- Poor, Good, Excellent
	Per-session	Startup (initial) Delay Stall Event Number Stall Event Duration Stall Ratio	-	Resolution (BC) Bit-rate (BC) ITU-T MOS (MC)	ITU-T MOS
Granularity	-	1 second	1 second	1 second	0.5, (1-5)seconds
Temporal Time Window	-	Current Trend Session	Current Trend Session	Current Trend -	Current
Basic Features	-	Packet Size Packet Count Packet IAT Inter Arrival Time -	Packet Size Packet Count Packet IAT -	Packet Size Packet Count Packet IAT Throughput	Packet Size Packet Count IPG Throughput
Total Features	**Real-time**	208 (TCP and QUIC)	208 (TCP and QUIC)	218 (TCP)	30+ (TCP) 30+ (QUIC) 3 - RTT , TP, Packets
	Per-session	208 (TCP and QUIC)	208 (TCP and QUIC)	62 (TCP)	
Window Used for Session QoS Aggregation	-	All (Current, Trend and Session)	All (Current, Trend and Session)	All (Current and Trend)	
Streaming Service	-	YouTube	YouTube	YouTube	goDASH Player

BC – Binary Classification, MC – Multiclass

4.3 QoE Prediction Using Multi Linear Regression

We first looked at the linear relation of QoE with its corresponding QoS parameters. The proposed work can be utilized in such scenarios where an ML technique finds chunk-level statistics. We can estimate the QoE of video sessions by using chunk-level QoS features such as Throughput, RTT, and Packets. The setup and the use cases are discussed in Chapter 3.

4.3.1 Regression Techniques

We now introduce the various Machine Learning models we use in our work. For a comprehensive analysis of the dataset that requires less pre-processing, such as scaling and normalization, we selected Decision Tree Regression (DTR), Multi-linear Regression (MLR), and Random Forest Regression (RFR).

DTR utilizes a tree-like structure in its models and is popular in both regression and classification problems. Departing from the root (parent) node, child nodes are decided by the largest Information Gain (IG) [57], and the iterative process terminates when the leaves are enough *pure*. MLR is a statistical technique used to predict a correlation between variables from independent predictors. MLR, or simply Multiple Regression (MR), is used to explain the relationship between one continuous dependent variable and two or more independent variables [85]. The relationship between variables can tell the change in the target value, by fitting a line through the observations. In our experiments, we have three main input variables (network QoS parameters) to the MLR classifier: RTT, number of packets for each video segment, and throughput.

Finally, we also apply RFR, which is used for classification and regression by building multiple DTs. In our experiments using different regression classifiers, we find the strength of the independent variable on dependent variable Y (P.1203) [19, 20], in other words, how RTT, number of packets per video segment and throughput have impact P.1203's score.

4.3.2 Pre-processing and Features Engineering

By inspecting the pattern changes in RTT and throughput, we split 60 video segments from the log file into 4 equal parts, i.e, 30s each or a 30s window. Together with this processed data, we take all comprehensive information available in *godash* log files, i.e., aggregated RTT, throughput, number of packets per video segment, and P.1203 score as shown in Table 4.3. The three columns in the middle of Table 4.3 are used as input for ML classifiers to predict P.1203 scores.

Table 4.3: Processed dataset used in the ML classifiers.

Column	User	Algorithm	RTT	Throughput	Packets	P1203
2	1	Arbiter	3.76	2584825.53	126.46	3.12
5	2	Elastic	0.23	7682269.18	65	3.02
2	2	BBA	0.58	2866549.71	64	2.94
5	1	Logistic	0.16	7212008.60	17	1.87
4	1	Conventional	0.66	6377796.25	87.73	3.56
4	2	Exponential	8.65	1077560.73	291.86	4.84

The first three columns (Column, User, Algorithm) are used to differentiate each trace fed into the ML model separately. For instance, (Column=1) means the first scenario from the 5G trace parameters. To train a single model for static and mobility scenarios, we use *pandas.get_dummies* to convert categorical algorithm names into dummy, or indicator, variables. The proposed ML methods, i.e., DTR, MLR and RFR using Python's *scikit-learn* library were trained on 80 % of data, while the remaining 20 % was used for testing trained ML models.

4.3.3 Results and Discussion

In our analysis, we use a total of 13,547 observations (approximately 225 client runs - across 2-client, 3-client and 5-client experiments) as input and evaluate three regression models, namely, DTR, MLR, and RFR. The input dataset used in our experiments describes a static and mobility (driving) scenario.

Table 4.4: Static scenario: Models' accuracy with MAE.

Algorithm	**Classifiers**	**MAE [%]**
Arbiter, Elastic	DTR	0.20
	RFR	0.17
	MLR	0.55
BBA, Logistic	DTR	0.12
	RFR	0.07
	MLR	0.12
Conventional, Exponential	DTR	0.23
	RFR	0.10
	MLR	1.03

Table 4.5: Mobility scenario: Models' accuracy with MAE.

Algorithm	**Classifiers**	**MAE [%]**
Arbiter, Elastic	DTR	0.31
	RFR	0.31
	MLR	0.55
BBA, Logistic	DTR	0.01
	RFR	0.01
	MLR	0.19
Conventional, Exponential	DTR	0.13
	RFR	0.07
	MLR	0.70

Table 4.6: Models' accuracy in predicting P.1203 for all 5G combinations of static and mobility.

Case	**Classifiers**	**Accuracy**
Static	DTR	78.68 %
	RFR	87.63 %
	MLR	40.01 %
Driving	DTR	72.37 %
	RFR	79.00 %
	MLR	58.67 %

Initially, each classifier is separately trained with each of the ABS algorithm categories, namely: Rate-based, buffer-based and hybrid. In other words, each classifier is trained with data from both Arbiter+ and Elastic for the Hybrid category, and the same is done for rate- and buffer-based ABS algorithms. Table 4.4 shows for the static scenario that RFR achieves much higher accuracy compared to DTR and MLR.

To quantify the predicted error of the P.1203 values to the ground-truth P.1203 scores, we use the Mean Absolute Error (MAE):

$$MAE = (\frac{1}{n})\sum_{i=1}^{n}|y_i - \overline{x}_i| \tag{4.1}$$

MAE is a metric used to find the similarity between two sequences. The absolute error is the absolute difference, whereas the error is the difference between two numbers. In order to find MAE, we first need to find the absolute error between two values and then find the mean of these values. In Equation (4.1), y and $\overline{x}$ correspond to the actual and predicted value, respectively.

Similarly, Table 4.5 shows the models' results for the mobility (driving) scenario, where RFR has a considerably lower MAE and much better accuracy see Table 4.6. In the static scenario, the same regressor RFR has an accuracy of 87.63 %, whereas DTR has 78.68 % and 72.37 % in the static and mobility scenarios, respectively.

4.4 QoE Prediction Using Classification Techniques

In this section, we talk about the classification techniques used for the prediction of QoE from the QoS features derived from an encrypted video stream. Finding a linear relation from QoS features in a non-trivial task. However, based on the MOS score, which is on a scale of 1-5, we can classify the QoE of video sessions into a different level of user score. We first talk about QoS features derived from the IPG followed by setup, use cases, and then analysis. The analysis includes the QoS impact on QoE under different network scenarios. Finally, we present the results of different ML classifiers.

4.4.1 Features Derived From IPG

We introduce a QoS features collection approach in a window-based manner that continuously extracts meaningful information from packet-level statistics to estimate QoE.

The proposed QoS features collection approach mainly works on two key matrices (packet time, and packet size). Then we derive three more QoS features from these metrics in a time interval such as (IPG, Total packets, Throughput) followed by flow features from IPG (EMA, DEMA, DEMA), then packet size and throughput distribution in (10-90)-percentile. We explain QoS features extraction approach in Chapter 3 of this thesis. Here, we will briefly explain the QoS features derived from IPG named i) EMA, ii) DEMA, and iii) CUSUM. We show three recursive QoS features pseudocode in Algorithm 2, 3, 4.

EMA is a quantitative or statistical measure used to find shifts or trends. In EMA, we place greater weight and significance on the most recent data points. As mentioned in Algorithm 1, we store each packet's time in an `array` named `array_packet_time`. Next, we derive IPG from the packet's time, which is the difference in time to process from one packet to another in a time window. We forwarded all IPGs to a function that recursively computes EMA value and returns to the function. Algorithm 2 demonstrate EMA calculation. We initialize $\alpha = 0.99$, since α value is between 0 and 1. The parameter α decides how important the current observation is in the calculation of the EMA. Next, in Algorithm 3 we present DEMA calculation. The function takes an input of IPG as an array and returns a value of DEMA for each time interval. Finally, we have CUSUM calculation mentioned in Algorithm 4. CUSUM is a Cumulative Sum of IPGs for a time interval.

Algorithm 2 EMA calculation, function takes IPG values of a window as an array.

```
FUNCTION ← START
α ← 0.99
EMA_array () ← IPG[0]
I ← 0
while I < count (IPG) do
    EMA_array () ← IPG[I] ∗ (1 − α) + IPG[I − 1] ∗ α
    I + +
end while
return array_sum(EMA_array)
FUNCTION ← END
```

Algorithm 3 DEMA calculation, function takes IPG values of a window as an array.

```
FUNCTION ← START
α ← 0.99
EMA_array () ← IPG[0]
I ← 0
while I < count(IPG) do
    EMA_array() ← (α ∗ IPG[I]) + ((1 − α) + EMA_array[I − 1])
    I + +
end while
return array_sum(EMA_array)
FUNCTION ← END
```

Algorithm 4 CUSUM calculation, function takes IPG values of a window as an array.

```
FUNCTION ← START
I ← 0
CUSUM ← 0
while I < count(IPG) do
    CUSUM+ = I
    I + +
end while
return CUSUM
FUNCTION ← END
```

4.4.2 Experimentation and Results

The experimental setup consists of "`EFFECTOR`". We stream BBB and Sintel popular videos from the publicly available 4K DASH video dataset for Advanced Video Coding

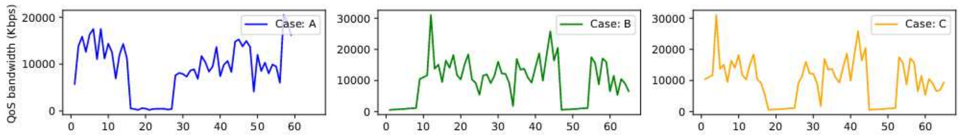


Figure 4.1: QoS bandwidth (Kbps) drop at different places of the video stream of a 4-minute session.

(AVC) H.264 [86]. We consider two second segment duration for the video and network bandwidth (QoS) values are based on the 5G traces.

A bash script is designed that reads the trace value from an excel sheet and changes the downlink bandwidth parameter using Linux TC HTB. The experiments are run on an Intel Core i7, SSD Linux machine.A parameterized script is used to run each experiment, resulting in one *godash* logfile per client and a corresponding *pcap* file per run.

We force less QoS at certain positions motivated by use cases mentioned in Chapters 3 and 5 of this thesis, where we observe at least 40 to 60 seconds of congestion in the network. Therefore, we consider a sudden drop in QoS for 40 seconds to 1 minute at different places of a 4-minutes video stream. Figure 4.1 presents three mobility scenarios, where we consider QoS in i) QoS drop in the beginning, ii) QoS drop in the beginning and before completion, iii) QoS drop in the middle and before completion. We consider mobility use cases for the experiments, as mobility suffers from stalling and quality shifts (Chapters 3 and 5).

We select three state-of-art ABS, i) BBA-2 — Buffered, ii) Conventional — Throughput, iii) Elastic — Hybrid. The adaptation mechanism of each of the selected algorithms is explained in [13] and in Chapter 3. We select BBA-2 and protocol TCP as a reference to show QoS impact on QoE. Moreover, the default behavior of godash is to consider streaming on a PC with a display size of 1920x1080, and a viewing distance is 152cm. QoE values will differ based on devices, screen sizes, and viewing distance, ultimately affecting model performance.

In Figure 4.2 and 4.3 we show four QoS metrics (Throughput, CUSUM, DEMA, EMA). We select three cases in which we drop QoS bandwidth at different positions: i) QoS drop in the middle, ii) QoS drop in the beginning and before completion, iii) QoS drop in the middle and before completion, see Figure 4.1 (A, B, C) respectively.

QoS feature (throughput) in Figure 4.2 (a, b, c) shows relationship with the scenarios as shown in Figure 4.1 (A, B, C). We see a sudden drop in the middle followed by a throughput drop in the beginning and before completion and finally, we observe a similar pattern when we drop QoS bandwidth in the middle and before completion. Next, in Figure 4.2 (d, e, f), we observe CUSUM for all the cases. We notice CUSUM is showing similar patterns followed by DEMA in Figure 4.2 (g, h, i). Finally, we have our last shift metrics named EMA, see Figure 4.2 (j, k, l). We observe a few peaks of spikes when QoS bandwidth is dropped. We observe similar behavior for QoS features, the number of packets and packet sizes greater than 100B. Now move on to another video name Sintel in Figure 4.3. We see similar patterns for these four QoS metrics see Figure 4.3 (a, b, c) for throughput, followed by (d, e, f) for CUSUM. Next, we have DEMA in (g, h, i), and finally, our last shift feature EMA in Figure 4.3 (j, k, l).

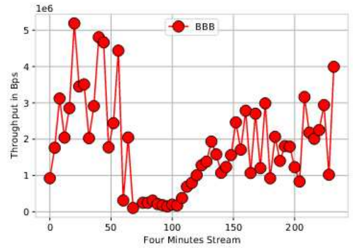
4.4.3 Results and Discussion

We provide further insights into the performance of different types of machine learning classifiers. For a comprehensive analysis of the dataset that requires less pre-processing, such as scaling and normalization we select Artificial Neural Network (ANN), K-nearest neighbour (KNN), Decision Tree (DT) and Random Forests (RF). We find QoS features using the filter method, the filtering is commonly done using Pearson correlation. The correlation coefficient has values between -1 to 1. A value closer to 0 implies a weaker, and 1 implies a stronger positive correlation. We select two subsets of QoS features to feed into machine learning models. The objective of dividing the QoS features into two sets is to reduce computational complexity. For instance, (`'Throughput'`, `'Total Packets'`, `'Total Packet GT100'`, `'Throughput 10-90P'`, `'Single EMA'`, `'CUSUM'`, `'Double EMA'`). These are fifteen QoS features; by using these QoS, we achieve similar accuracy as shown in Table 4.7. Whereas on 24 QoS features which also include packet size distribution into 10-90 percentile the accuracy is nearly equal to the first set of features, there is trade-off ± 2 %. Therefore, we can rely only on those QoS features derived from IPGs and other basic metrics such as throughput and packet count.

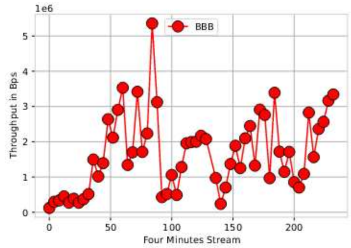
During the feature engineering process, we find (IPG average (ipg_avg), IPG average of packets size greater than 100B (ipg_avg_gt100) and Inter Arrival Time (IAT_w) are interrelated in a few network settings. For example, IPG average has a positive relationship in BBA, 3 seconds window but doesn't provide good results with other combinations.

QoE predictions throughout the research work are done through 5-fold cross-validation using 30 % and 70 % ratios for train test split. We also ensure data samples in each class are balanced. For ANN, we used 3-layer neural network. Each layer has 100 neurons and we used a batch size of 128. However, we tried other combinations as well, such as changing the batch size and changing the number of layers and neurons on each layer. For DT, the maximum depth is 5. However, we check with other settings, i.e., checking with depth (1-10). In KNN we use the maximum number of neighbors 12-15 for different time windows. Finally, for RF we use different settings of n_estimators (10, 30, 50, 100, 150, 200, 250, 300, 350) which are the number of trees you want to build before taking the maximum voting or average predictions. Model parameters are calibrated through standard grid search optimization. We use TensorFlow on GPU for NN and scikit-learn library for the remaining models.

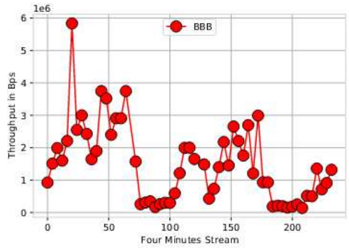
For QoE labeling, we leverage video player (godash) logs at (selected time window) and take aggregated values of QoE model ITU-P.1203 which were running at that moment



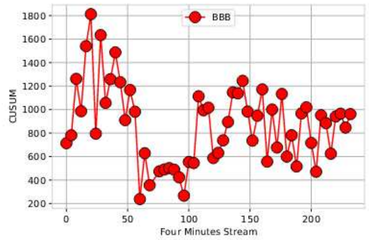
(a) Throughput: Scenario in figure 4.1 (A)



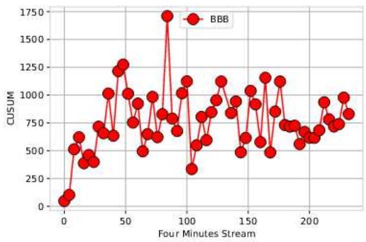
(b) Throughput: Scenario in figure 4.1 (B)



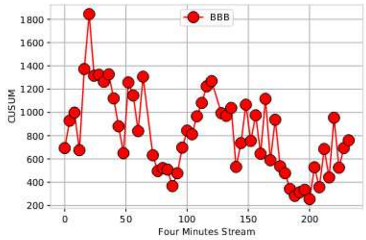
(c) Throughput: Scenario in figure 4.1 (C)



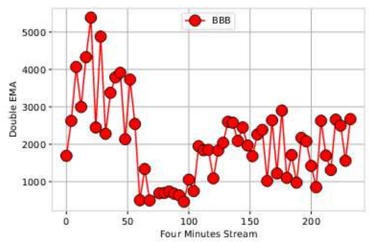
(d) CUSUM: Scenario in figure 4.1 (A)



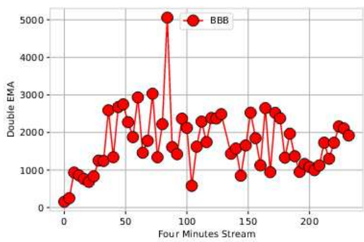
(e) CUSUM: Scenario in figure 4.1 (B)



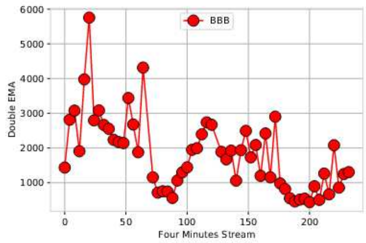
(f) CUSUM: Scenario in figure 4.1 (C)



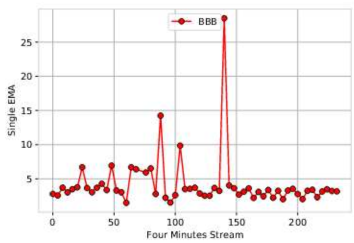
(g) DEMA: Scenario in figure 4.1 (A)



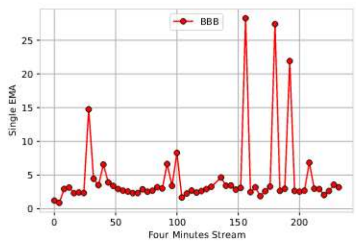
(h) DEMA: Scenario in figure 4.1 (B)



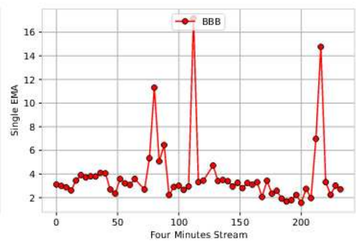
(i) DEMA: Scenario in figure 4.1 (C)



(j) EMA: Scenario in figure 4.1 (A)

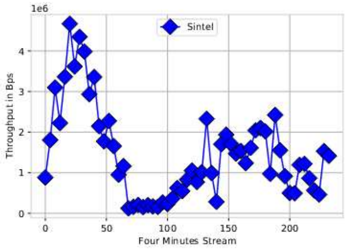


(k) EMA: Scenario in figure 4.1 (B)

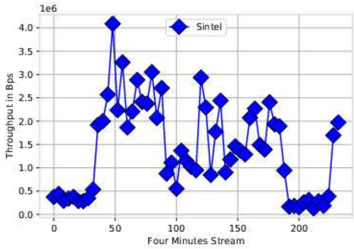


(l) EMA: Scenario in figure 4.1 (C)

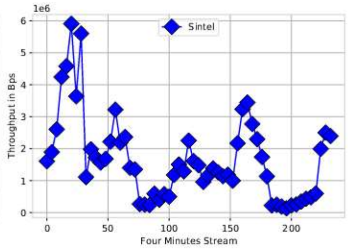
Figure 4.2: Video Big Buck Bunny: Throughput, CUSUM, DEMA, and EMA for three cases as shown in figure 4.1 (A, B, C).



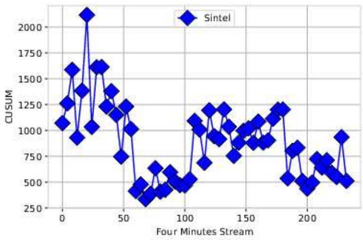
(a) Throughput: Scenario in figure 4.1 (A)



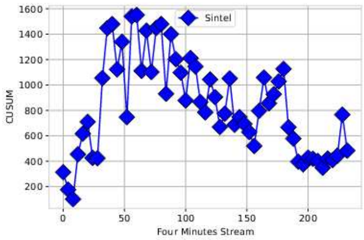
(b) Throughput: Scenario in figure 4.1 (B)



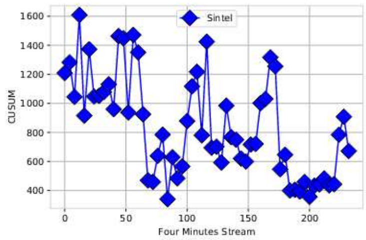
(c) Throughput: Scenario in figure 4.1 (C)



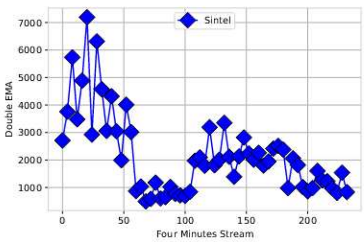
(d) CUSUM: Scenario in figure 4.1 (A)



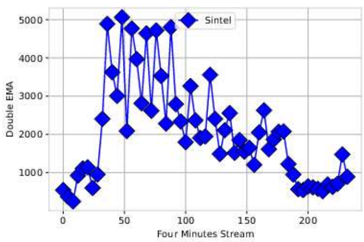
(e) CUSUM: Scenario in figure 4.1 (B)



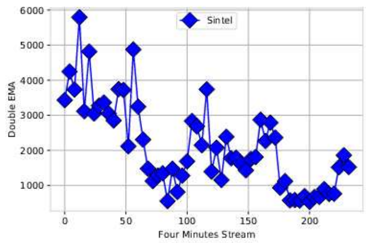
(f) CUSUM: Scenario in figure 4.1 (C)



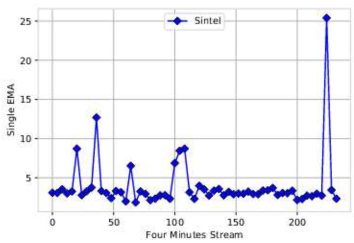
(g) DEMA: Scenario in figure 4.1 (A)



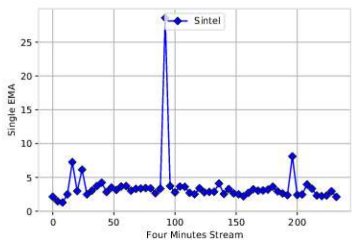
(h) DEMA: Scenario in figure 4.1 (B)



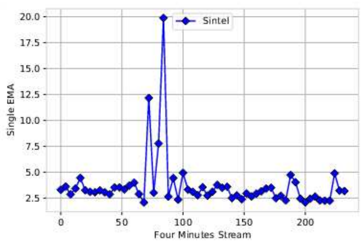
(i) DEMA: Scenario in figure 4.1 (C)



(j) EMA: Scenario in figure 4.1 (A)



(k) EMA: Scenario in figure 4.1 (B)



(l) EMA: Scenario in figure 4.1 (C)

Figure 4.3: Video Sintel: Throughput, CUSUM, DEMA, and EMA for three cases as shown in figure 4.1 (A, B, C).

using the 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 1 and 3, Good, if it fits between 3 and 4, and Excellent if it is between 4 and 5.

We achieve the highest accuracy of 79 % for TCP and 82 % for QUIC (see Table 4.7) where 24-QoS features forwarded to ML classifiers. We also provide per time slot accuracy for all the ABS see Table 5.7, Table 4.9 for TCP and QUIC respectively. In our results we observe, BBA has maximum accuracy. Next, we see Elastic has good accuracy in all time slots. Therefore, QoE using hybrid ABS has a good relationship with the proposed QoS features. The Conventional performs better when the time window is (4-5)-seconds for all classifiers.

1lightgray

Table 4.7: Accuracy of different classifiers using all ABS in percentage (%) - TCP and QUIC.

Protocol	**Window/seconds**	**ANN**	**KNN**	**DT**	**RF**
TCP	1	73	71	72	72
	2	77	75	76	77
	3	79	79	73	79
	4	79	77	71	76
	5	78	77	76	76
QUIC	1	73	77	78	76
	2	78	78	77	80
	3	77	78	76	80
	4	79	77	71	76
	5	77	82	77	81

Limitations While conducting experiments and generating the datasets by emulation-based testbed, we consider a few limitations of this work. The dataset was collected using a headless `godash` player. However, in reality, end users can stream video content from different platforms (e.g, Netflix, YouTube, Amazon, etc.). Therefore, QoS impact on QoE may vary. Moreover, a user can use different devices as well, such as Mobile – iOS, Android, PC – Laptop, and Tower with different viewports. Therefore, the proposed QoS to QoE inference technique would be unable to manage such massive diversity. The proposed technique requires practical deployment for the evaluation. We are unaware of the computational complexity, such as CPU, Memory, and Storage, for large deployments.

Table 4.8: Accuracy of different classifiers on each ABS in percentage (%) TCP.

Window/seconds	**ABS**	**ANN**	**KNN**	**DT**	**RF**
1	Conv	60	60	59	60
2	Conv	67	67	64	66
3	Conv	68	67	62	70
4	Conv	70	65	55	64
5	Conv	67	69	61	66
1	BBA	73	72	72	71
2	BBA	84	81	80	84
3	BBA	84	83	83	83
4	BBA	85	88	82	83
5	BBA	84	83	83	83
1	Elastic	74	74	73	71
2	Elastic	75	75	71	74
3	Elastic	75	79	76	78
4	Elastic	80	80	74	80
5	Elastic	75	79	74	79

Table 4.9: Accuracy of different classifiers on each ABS in percentage (%) QUIC.

Window/seconds	**ABS**	**ANN**	**KNN**	**DT**	**RF**
1	Conv	70	70	65	66
2	Conv	68	67	68	63
3	Conv	64	63	65	70
4	Conv	72	70	68	70
5	Conv	67	62	68	67
1	BBA	89	88	90	86
2	BBA	85	81	85	82
3	BBA	83	82	77	81
4	BBA	82	83	77	84
5	BBA	86	81	87	86
1	Elastic	81	82	73	80
2	Elastic	86	85	81	83
3	Elastic	88	87	82	85
4	Elastic	85	85	80	84
5	Elastic	82	78	73	79

4.5 Summary

Deep packet inspection is no longer available [22, 1, 5]. The research community is striving hard to find QoE indicators from packet-level statistics. In this chapter, we present QoS features relationship with QoE derived from Inter Packet Gap (IPG).

These features are lightweight and can be used to estimate QoE efficiently by using various Machine Learning Techniques. Moreover, these features do not require chunk detection, which adds extra computational complexity to existing approaches. We propose a window based (Time Window) QoS features extraction technique, which is ideal in real-time QoE prediction.

We choose various Time Windows, i.e., (0.5, 1, ...,5), and show the interrelation of each window with QoE for TCP and QUIC. We find that for TCP, a time window of 3 and 4 provides more interdependence with QoE, thus achieving an accuracy of up to 80 %. For QUIC, we observe higher Time Window – Higher accuracy. Thus we achieve an accuracy of up to 82 % using KNN and 81 % using Random Forests.

Chapter 5

YouTube goes 5G: QoE Benchmarking and ML-based Prediction

5.1 Introduction

Mobile video traffic is continuously growing, thus increasing an additional challenge for MNOs to manage this exponential growth [3]. Applications utilizing social media, gaming, and recent advances in Augmented/Virtual Reality and UHD videos have accelerated the demands for the next generation of networks, 5G. 5G technology New Radio (NR) is developed to address high bandwidth, low latency and massive connectivity requirements of enhanced Mobile Broadband (eMBB) compared to Fourth Generation (4G) Long Term Evolution (LTE). In order to provide a 5G network while addressing compatibility with previous cellular systems, there are two 5G deployment options, Non-Standalone (NSA) and Standalone (SA). In NSA, 5G control plane relies on a pre-existing 4G core network while SA on a dedicated 5G core network [87]. Both architectures require the deployment of a 5G NR Radio Access Network (RAN) composed of Next Generation Node Bs (gNBs), i.e., the 5G equivalent of 4G Evolved Node Bs (eNBs). The basic Radio Resource Management (RRM) measurements in LTE system are Channel Quality Indicator (CQI), Reference Signal Received Power (RSRP) and Reference Signal Received Quality (RSRQ). RSRQ and RSRP are used to make Handoff (HO) decision. When the RSRP and/or RSRQ of the serving cell fall(s) below the RSRP and/or RSRQ of the neighbor cell by a predefined HO margin for a certain period of time, handover occurs [88]. Both are mainly used to rank different candidate cells according to their signal quality. Signal to Interference plus Noise Ratio (SINR) is measured by UE on Resource Block (RB) basis and converts it to CQI and reports it to eNodeB. CQI is a quantized and scaled version of the experienced SINR [88], i.e., higher SINR means higher throughput and vice versa as it indicates data rate that could be transmitted over a channel [24].

QoE of YouTube video streaming from a Mobile Network Operator (MNOs) perspective is ideal and challenging compared to 4G/LTE networks. In addition to that, evaluating mobile carriers' end-to-end network performance in the wild is known to be difficult and complicated [24, 43, 44, 45, 46]. The research community is now continuously looking for an alternative solution to deal with end-users QoE and provide adequate methods to manage increased video traffic demands and provide a satisfactory experience [45]. Moreover, to understand the benefits that 5G brings to video streaming, we need fair comparisons with 4G. However, both technologies have very different characteristics making it difficult to experimentally compare them in a fair, efficient and representative way. For instance, 4K/8K video streaming, interactive 360 and volumetric video streaming, cloud gaming, and Augmented Reality/Virtual Reality (AR/VR), among others. Thus in this chapter of the thesis, the main intuition behind the work is to see the performance of 5G while streaming YouTube videos with different Frame Per Second (FPS), Context – i) Mobility, iii) Static Indoor, iv) Static Outdoor, Videos – Nature, Animation, Movie, Brand Promotions, etc. compared to 4G. Next, we study the CLM impact on the objective QoE of YouTube. We proposed a QoE interruption (Stall) mechanism where we can only use CLM to classify them. We also experience the greedy nature of the 5G, where we observe YouTube playing video in HD2160, even there are stalling events. We carry out a massive 4G and 5G dataset collection campaign using a commercial 4G and 5G network, where we consider YouTube as a baseline for video streaming to collect CLM and YouTube QoE logs with 1-second granularity. Thus, if we summarize the contributions:

- Collection of 4G and 5G datasets to support our framework as discussed in Chapter 3 of this thesis. Use cases for the collection of the dataset are i) Mobility, ii) Pedestrian, iii) Static – Indoor, Static Outdoor – Bus and Railway Terminals. The dataset considers YouTube streaming of more than 10 videos. All videos are selected from different categories such as Sports, Animated, Movies, Nature, etc. In addition, we consider all the videos with 4K quality and a few videos with 60fps. We provide detail of each video in Table 5.2

- We provide YouTube QoE and CLM with 1-second granularity. The channel level metrics include Timestamp, Longitude, Latitude, Velocity, Operator Name, Cellid, Network Mode, Download bitrate, Upload bitrate, RSRQ, RSRP, SNR, RSSI, CQI and RSRQ, RSRP and SNR values for the neighboring cell [23].

- We have released the functional artifacts (both dataset and tools) on GitHub: https://github.com/razaulmustafa852/youtubegoes5g.

- In our work, we derive a CLM relation with interruptions using a time based method. We checked with different time sequences (1, 3, 5, 7, 9)-seconds to interrelate stalling events of YouTube streaming, where we found that a 7-second window is ideal for predicting stalling events of YouTube video streaming, achieving high accuracy in Binary Classification of Stall vs. No Stall.

- Binary Classification of real YouTube Traffic by using only CLM such as CQI, RSRQ, RSRP, SNR and their distributions. Such methods are ideal for MNOs to take necessary measures before experiencing a bad MOS score from end-users.

5.2 Background and Related Work

5G promise to provide maximum speed to deal with current user demands. In addition, 5G supports thousands of devices simultaneously to provide all users with maximum QoE. Focusing on the data plane, i.e., how a 5G-capable User Equipment (UE) connects to the available 4G/5G Radio Access Technology (RAT) for data exchange. There are two main categories of HO events. i) Intra-RAT HO and ii) Inter-RAT HO. In Intra-RAT HO, UE switched from a 4G cell to another 4G cell or from a 5G Cell to another 5G cell. However, it remains in same technology. On the other end, Inter-RAT HO does the opposite, UE is instructed to rearrange its data plane from 5G to 4G or from 4G to 5G.

Over the last few years, several studies have been published aiming to empirically characterize 5G NSA, and SA performance from different perspectives, including application performance (e.g., web browsing and HTTP download and video streaming), coverage and latency, and power consumption and QoE [43].

The work very close to our work is [24]. Authors provide CLM with a 1-second granularity of a total of 83 traces in 5G use cases. They played two well known videos – animated (circa 200m) and live-action (circa 400m) and their key findings are more related to providing a dataset for 5G Mobility and Static using file download, Netflix, and Prime video streaming.

Another work that is close to our campaign is [46]. The authors tried to find the performance footprint of the current state-of-the-art ABR algorithms under 5G and how it compares with 4G. What are the major factors that impact ABR streaming performance over 5G, and finally, what new mechanisms are needed to make future ABR algorithms 5G-aware? Moreover, they also worked on power characteristics [46]. However, in our case, we consider YouTube as a baseline for QoE KPIs, whereas, in [46], custom settings are used to run diverse experiments. We use 10+ videos for streaming compared to one video with a duration of 2.38-minutes. Motivated by the research question, "What new mechanisms are needed to make future ABR algorithms 5G-aware?" we concluded streaming in 5G is more greedy as compared to 4G, i.e., 5G selects the highest bitrate chunk. Even though the decision is wrong, 5G remains at a higher bitrate, thus causing stalling events. However, in 4G, we see quick shifts from higher resolutions to lower, with many quality shifts.

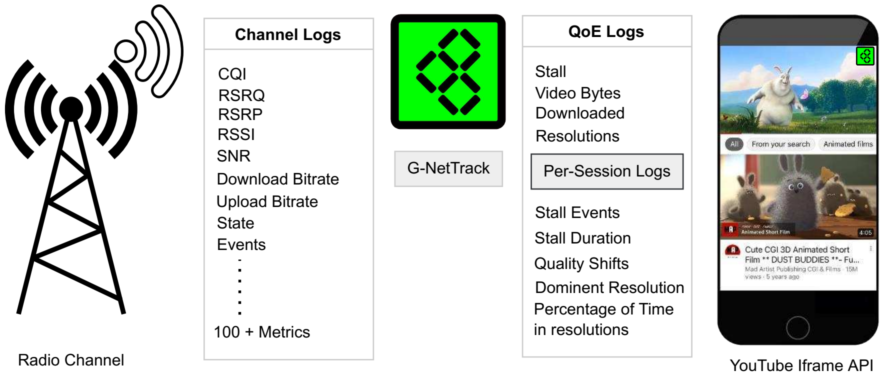


Figure 5.1: Block diagram of 4G and 5G dataset collection methodology.

Authors in [43] provide a dataset to understand the HO events by playing online video games. The total duration of the study is 7 weeks, but most of the work focuses on studying the HO while playing video games. They consider two use cases for collecting datasets, i) Pedestrian and ii) Mobility - Car. In another study, we see 5G dataset collection under very high mobility, such as Train. The authors reveal the key characteristics of 5G and LTE in extreme mobility in terms of throughput, RTT, loss rate, signal quality, and physical resource utilization [44]. They provide a dataset for a duration of six months with custom settings using DASH.js player and controlled experimentation. Next, in [45] worked on 5G aware streaming to avoid stalls and predict throughput. However, they only consider the case of Pedestrians with 20 days of dataset collection.

Table 5.1: Comparison of state-of-the-art work in 5G dataset collection.

Ref	Use Case	Settings	YouTube Logs	Total Traces	Video Played
[24]	Mobility – Car, Static	File Download Netflix Amzon Prime	No	83 Traces	animated (circa 200m) live-action (circa 400m)
[43]	Mobility – Car, Pedestrian	Online Video Gaming	No	7 Weeks	No
[44]	High Mobility – Train	Controlled Video Streaming DASH.js	No	6 Months	Custom Settings
[45]	Pedestrian	Controlled experiments TCP/IP stack and C++	No	20 Days	Custom Settings
[46]	Mobility – Pedestrian – Static	Controlled experiments Using 5G Traces	No	N/A	Custom Settings
This work	Mobility – Pedestrian – Indoor – Outdoor	YouTube	*	6 Months	Videos 4K 4K - 60fps

* Yes i) Events ii) Buffer iii) Stall iv) Quality

In comparison to previous work, we summarize the goal of this research work in Table 5.1.

5.3 Methodology

The proposed methodology leverages two software components:

1. YouTube IFRAME API [1] – IFRAME API provides YouTube player logs. The Iframe player API lets you embed a YouTube video player on web-based applications and

[1] https://developers.google.com/youtube/iframe_api_reference

control the player using JavaScript. We designed a custom web-based application and embedded the YouTube IFrame. Then, using Javascript, we define `functions` to save player events in MySQL database after every 1-second interval. Different JavaScript functions used for collecting YouTube Logs are listed in the Appendix of this thesis.

2. G-NetTrack-pro – It is a network monitor and drive test tool application for 5G/4G/3G/2G networks. It allows monitoring and logging of mobile network parameters without using specialized equipment. It provides 2G/3G/4G/5G serving and neighbors cells information measurement and save it in logfiles (text and kml format).

The block diagram of the data collection methodology is presented in Figure 5.1.

5.3.1 YouTube IFRAME API

We designed a custom web application using YouTube IFRAME API to collect player statistics such as Stalls and Quality shifts. Quality shifts refer to the change in resolution from lower to higher and vice versa. The application interface requires i) A unique ID to link Channel Level Metrics and ii) a Video to Play. We select 10 videos for the collection of Logs. The video statistics are shown in Table 5.2. The resolutions for the videos are 144p, 240p, 360p, 480p, 720p, 1080p, 1440p, 2160p for the first seven videos, however for remaining three resolutions or available qualities are same, but with 60/FPS, i.e., 1080p/60FPS, 1440p/60FPS, 2160p/60FPS. YouTube IFRAME API invokes **onStateChange** event, where the information of stall, along with other features is available. There are 6 states available for player. 3 – Stall, 1 – Playing, 0 – Ended, 2 – Paused, 5 – Cued, -1 – Unstarted. We designed a script to save the QoE KPIs of the YouTube player every 1-second using AJAX. For instance, we are saving i) Current Quality, ii) Video Bytes Downloaded, iii) Loaded Percentage, iv) Available Qualities, v) Time. These QoE KPIs can further provide per-session Objective QoE (i) Total Stalling Event, ii) Stalling Ratio, iii) Stalling Time, iv) Quality Shifts or Percentage of Time in a single Resolution, v) Dominant Resolution, etc.

5.3.2 G-NetTrack Pro

We used G-NetTrack Pro version for the collection of Channel Level Metrics. We set 1-second granularity for logs in the setting. The most valuable metrics include CQI, RSRQ, RSRP, SNR, and application download bitrate, among other 100 + features.

Table 5.2: Videos played for the collection of YouTube and channel metrics logs.

No.	Video ID	Category	Duration
1	Baur2Ypgd60	Mix	3.24m
2	JMbBjKnUoC4	Animated	6.30m
3	jAa58N4Jlos	Science	3.37m
4	9ZfN87gSjvI	Science	4.17m
5	9jrP7460a4o	Nature	5.02m
6	oh904_HdkwY	Tourism	3.11m
7	VMs0yEVC00A	Trailer	4.30m
8	LXb3EKWsInQ	Nature	5.13m
9	CHSnz0bCaUk	Nature	4.27m
10	pSnB7Uh8dvQ	Trailer	5.44m

5.3.3 Data Collection Approach

We use two smartphones of user equipment (UE) with 4G and 5G support, i) Samsung Galaxy S21 – 5G, and ii) Samsung Galaxy S8 – 4G. We selected a 15 km driving route that includes busy downtown regions and freeways with driving speeds (Mobility use case) ranging from 0 to 80 Km/h. For the use case – Pedestrian, we run 4G and 5G campaigns in busy downtown at different times and days. Static Indoor consider streaming video sessions inside work areas, whereas for static outdoor we selected various locations, i) Bus Terminal, ii) Railway Terminal and iii) Shopping Malls.

The data collection follows two methods, i) Standalone and ii) Comparison. In the first case, a single 4G device is used to collect the 4G dataset, and the same follows for 5G, and in the later case, two UE, 4G, and 5G, both at the same time to draw a quick comparison of both the technologies. 5G experiments are done mostly in 5G covered area, i.e., downtown, malls, and bus/railway terminals. We made the dataset anonymous to make the process ethical. We first name the video sessions, say **S1, S2** from UE 4G and 5G. The G-NetTrack Pro is running in the background, whereas we open our web application to collect player logs. The process of stopping and playing the session is manual. However, the logs and file saving process are automatically uploaded to the server. The video demonstration of the whole process is explained. [2]

[2] https://bit.ly/3elgSkT

Table 5.3: 4G and 5G dataset statistics.

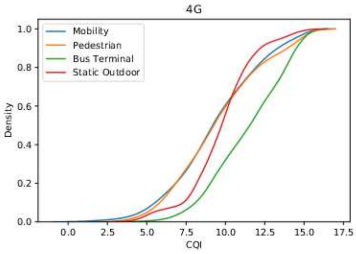
Parameter	**Statistics**
Mobility – Total Kilometers	1000+ (Approx)
Pedestrian – Total Kilometers	250+ (Approx)
Number of Videos	10
Total Video Sessions	300 +, 1500 + Minutes Streaming
4G and 5G Data Consumed	300+ GB
5G Smartphone	Samsung Galaxy S21 5G
4G Smartphone	Samsung Galaxy S8

5.4 Statistics of the Data Collected Using Commercial Carriers: 4G and 5G

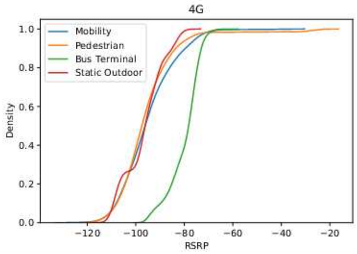
Since 5G commercial launch, it expanding and evolved. In our measurement study, we select a commercial 4G and 5G operator in France. The operator we select provides low-band (3.4-3.8 GHz range) 5G service using NSA modes. The dataset collection is conducted in Nice, France over a period of 6-months. Key statistics of the datasets collected are summarized in Table 5.3. We covered approximately 1000+ Km of mobility experiments during this time. We selected a 15 Km route that includes both busy downtown and less crowded during the use case – Mobility. For use case – Pedestrian, we mostly run the campaign in the downtown area. A total of 10 videos were used during this study, and each video's characteristics are listed in Table 5.2.

5.5 Real 4G vs. 5G Performance Footprint Using YouTube as a Baseline

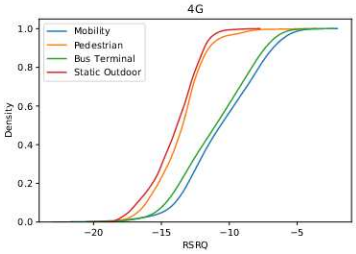
We observe better CQI (CQI is feedback provided by UE to eNodeB), RSRP (RSRP is used for measuring cell signal strength/coverage and therefore cell selection (dBm)) and RSRQ (RSRQ Indicates quality of the received signal, and its range is typically -19.5dB(bad) to -3dB (good)) at Bus and Railway Terminals in 4G and 5G. Moreover, we experience better CQI in 5G as compared to 4G see Figure 5.2. We observe better Signal to Noise Ratio (SNR) at Bus Terminals followed by Mobility in 4G. 80 % of SNR remains near 20 (db), while streaming at Terminals. SNR is the ratio of signal power to the noise power. In 5G, we observe Terminals and Pedestrian outperforming other use cases; see Figure 5.3. 4G experiences more stalling events as compared to 5G. However, 5G shows greedy behavior; even there are stalling events, the player remains in higher resolutions



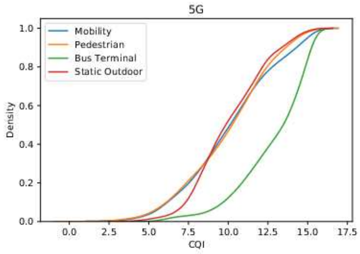
(a) 4G: CQI



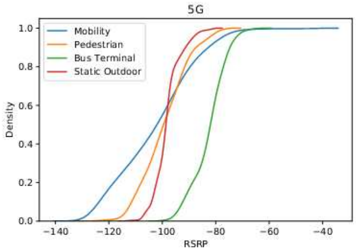
(b) 4G: RSRP



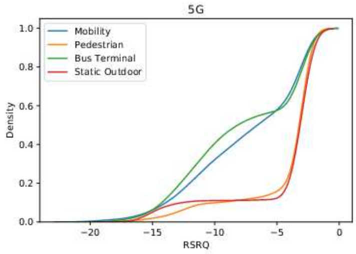
(c) 4G: RSRQ



(d) 5G: CQI

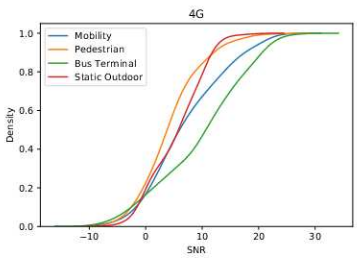


(e) 5G: RSRP

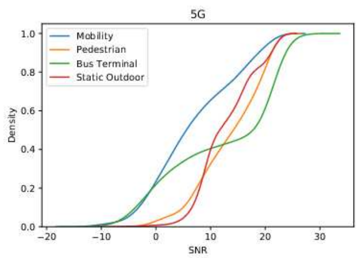


(f) 5G: RSRQ

Figure 5.2: 4G vs. 5G CQI.

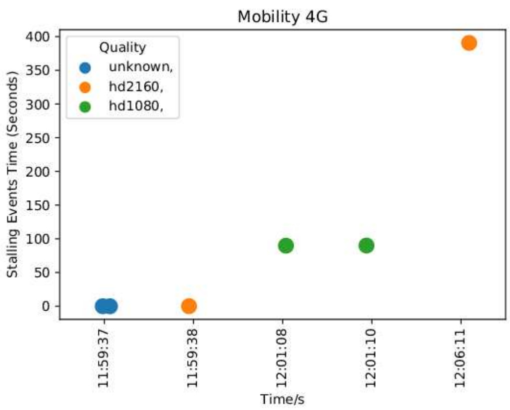


(a) 4G: SNR

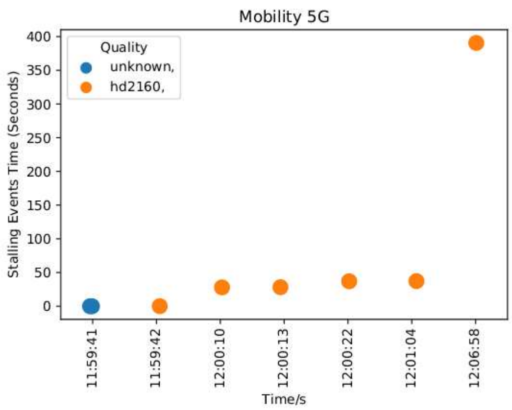


(b) 5G: SNR

Figure 5.3: 4G vs. 5G SNR.



(a) 4G – Stalls



(b) 5G – Stalls

Figure 5.4: 4G vs. 5G stalls in mobility.

instead of choosing a segment with lower resolution and bitrate to avoid stalls see Figure 5.4. However, in 4G we experience more quality shifts as compared to 5G.

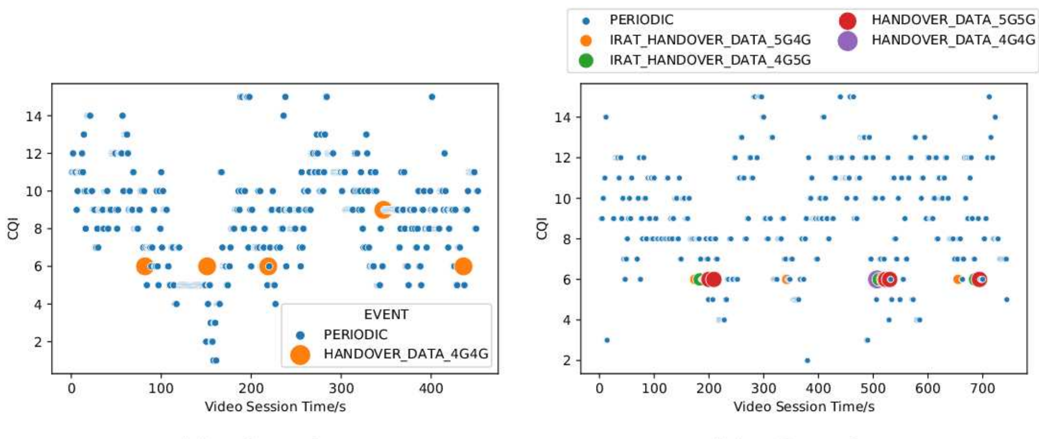
We experience Quality Shifts for use cases – Mobility and Pedestrian in both technologies 4G and 5G. In other use cases – Indoor and Outdoor, there are negligible shifts. 4G experiences more quality shifting as compared to 5G both in High and Low Mobility see Table 5.4 and Table 5.5 respectively. During the use case – High Mobility, 5G remains 99.8 % in hd2160 resolutions, whereas in 4G 63.2 %. Next, we experience 95.3 % of streaming in hd2160 in 5G whereas, 37.9 % in 4G for low Mobility – Pedestrian. Static cases experience more stability in both technologies, i.e., hd1440 (10%) and hd2160 (90%) in 4G and in 5G hd2160 (100 %).

Table 5.4: 4G vs. 5G quality shifts in % case – mobility.

Technology	144p	240p	360p	480p	hd720	hd1080	hd1440	hd2160
4G	-	-	-	7.7	-	3.2	25.9	63.2
5G	-	-	-	-	-	-	0.2	99.8

Table 5.5: 4G vs. 5G quality shifts in % case – pedestrian.

Technology	144p	240p	360p	480p	hd720	hd1080	hd1440	hd2160
4G	-	-	-	0.4	3.7	33.2	24.8	37.9
5G	-	-	-	-	0.6	3.7	0.4	95.3



(a) 4G – HO

(b) 5G – HO

Figure 5.5: 4G vs. 5G HO and CQI impact.

5G experiences more handover events. Most handover occurs at CQI value 6. In Figure 5.5 we show use case – High Mobility. Most of the HO events occur when the CQI value is 6 see Figure 5.5 (a). However, In 5G see Figure 5.5 (b) there are multiple HO events – i) Intra-RAT HO and ii) Inter-RAT HO.

5.6 Stall Prediction Using Channel Metrics

This section explains the use of machine learning classifiers to predict Stall vs. No Stall. For the proposed method, we use **Player Events** metrics, for instance, we saved the player states, 3 – Stall, 1 – Playing, 0 – Ended, 2 – Paused, 5 – Cued, -1 – Unstarted. Moreover, we also know the time of the events. We check previous *times (window)*, i.e., (1, 3, 5, 7, 9) seconds to see if there is any interrelation of CLM with stalling events. Thus, from the current time of interruption, say t, we see t - *window* where a window is (1, 3, 5, 7, 9) seconds. An example of CQI values of the past seven observations is mentioned in Table 5.6. Similarly, we also derive the same set of features for other metrics. We took the previous **n** observations of CLM, i.e., CQI, RSRP, RSRQ, SNR, along with their distribution and standard deviations. Other features derived from these metrics include, i) Majority of a window, ii) Standard deviation, iii) 25, 50, and 75 percentile of a window. We named interruptions instances as **Stall**, it is also a target class for the classifiers to differentiate from non-interruptions **(Non-stalling)** instances. Thus we have a binary classification problem where we have Stall – Class Yes, and No Stall – Class No. Next, we use different classification algorithms such as Decision Trees, Random Forests, KNN and Neural Networks. The results of each classifier are listed in Table 5.7. Regarding the settings of different classifiers, we used 5 neighbours for KNN, 500 Estimators for Random Forest, Decision Tree depth as 3 and 3 layers for ANN. On each layer of ANN we used 50, 100, 150 neurons with RULE activation function on Hidden layers followed by Sigmoid Activation Function on last layer with Binary Cross Entropy for Binary Classification. We use a batch size of 50 and 1000 epochs are used for training the classifier. Each classifier's Confusion Matrix is shown in Figure 5.6, 5.7, 5.8 and 5.9 for window 3s, 5s, 7s and 9s respectively.

The confusion matrix is a combination of four characteristics, i) True Positive (TP), ii) True Negative (TN), iii) False Positive (FP) and iv) False Negative (FN). With our use case of interruptions (Stall), TP represents the number of YouTube Interruptions that have been properly classified, meaning there are interruptions, and the classifier differentiates them from non-interruptions instances. Whereas TN represents the number of correctly classified non-stalling events. FP represents misclassified video sessions with no stalling events. However, they have stalls. It is also called a Type I error; finally, we have FN, a type II error, in which the classifier outputs the class "No Stalls – Interruption", but there are interruptions. Performance metrics of an algorithm are accuracy, precision, recall, and F1 score, which are calculated based on the above-stated TP, TN, FP, and FN.

Random Forest at 7s is the best classifier to differentiate Stalling vs. No Stalling events when 7s features are forwarded as input. After that, 9s second window, we observe better accuracy when we use ANN however, a decline in accuracy by other classifiers. The accuracy of an algorithm is represented as the ratio of correctly classified cases (TP+TN) to the total number of cases (TP+TN+FP+FN) see equation 5.1.

Table 5.6: CQI values of 7-second window for `Target Class` – stall, Yes/No.

Resolution	CQI-1	CQI-2	CQI-3	CQI-4	CQI-5	CQI-6	CQI-7	Stall
tiny	7	7	4	4	4	4	8	Yes
hd2160	7	7	5	5	4	5	5	Yes
hd2160	5	5	5	4	8	8	5	Yes
hd2160	8	5	5	5	4	4	4	Yes
hd2160	4	6	5	5	5	4	4	Yes
hd2160	15	15	14	14	13	13	9	No
hd2160	15	14	14	13	13	9	9	No
hd2160	14	14	13	13	9	9	13	No
hd2160	14	13	13	9	9	13	13	No
hd2160	13	13	9	9	13	13	14	No

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

Table 5.7: Performance of different classifiers in percentage (%).

Windows	ANN	KNN	DT	RF
3	76	80	76	85
5	79	81	77	89
7	82	81	76	91
9	83	82	76	90

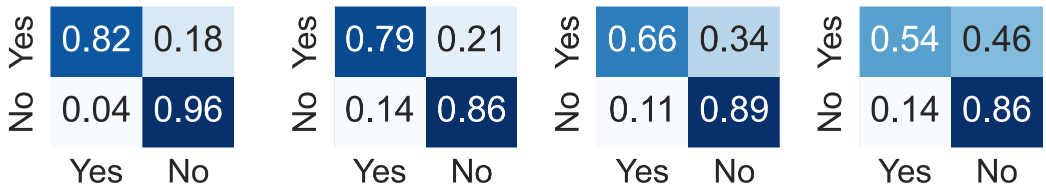
Limitations. The dataset collected in this work is collected using a web-based application, which uses YouTube IFRAME API. While the campaign, we used the browser to open the application. Therefore we believe there might be a chance of a different QoE than YouTube Android Application. Moreover, the dataset collection is done using two android devices, one for 4G and one for 5G. However, we do not consider multi-user streaming of the same content simultaneously. Moreover, during the dataset collection campaign, we consider the full width of YouTube player, which automatically adjusts to the viewport of the device. However, different screen sizes may influence QoE.

(a) RF (b) ANN (c) KNN (d) DT

Figure 5.6: Confusion matrix 3s - RF, ANN, KNN and DT.

(a) RF (b) ANN (c) KNN (d) DT

Figure 5.7: Confusion matrix 5s - RF, ANN, KNN and DT.



(a) RF (b) ANN (c) KNN (d) DT

Figure 5.8: Confusion matrix 7s - RF, ANN, KNN and DT.

(a) RF (b) ANN (c) KNN (d) DT

Figure 5.9: Confusion matrix 9s - RF, ANN, KNN and DT.

5.7 Summary

MNOs continuously strive hard to find relations between QoE from various methods. The main objective is to provide end-user maximum QoE, ultimately increasing satisfaction and trust in MNOs. Therefore, the contribution of this work is i) we provide a rich dataset with various features and metrics to explore for various other objectives, ii) A rich dataset with different use cases to run in `EFFECTOR` with real 4G and 5G dataset, iii) An approach where only channel level metrics can predict stalling events in YouTube Streaming. Future work could be done in many directions, i) Finding highly correlated features to draw a relationship between Quality Shifts and Channel Metrics as a first step, followed by triggers to avoid stalls in 5G-aware streaming, ii) A new 5G-aware ABR algorithm for a famous video streaming platform YouTube, iii) We see stalling events when the use-case is Mobility; therefore, for 5G-aware video streaming, there is a need for triggers from CLM to provide end-user maximum QoE.

Chapter 6

Conclusions & Future Work

6.1 Conclusions

Multimedia traffic is increasing with an exponential increase over time, where a large share of traffic is video traffic. Furthermore, the recent trend in 5G has created new norms for Mobile Network Operators (MNOs) to provide end-users with maximum Quality of Experience (QoE).

Throughout this thesis, the entire line of reasoning fits according to finding Quality of Service (QoS) metrics to map QoE followed by Machine Learning (ML) techniques. Thus increasing end-user satisfaction by various methods and ultimately reaching an acceptable Service Level Assurance (SLA) level.

We proposed DASH QoE estimation technique in a time window manner, i.e., (0.5, 1, 2, 3, 4, 5) seconds by using only QoS metrics based on Packet Time and Packet Size. This method overcomes the problem of finding chunk level statistics to infer QoE, which is heavyweight and expensive. Moreover, in these time windows, we derive various QoS features to infer QoE, i.e., throughput and packet size distribution in a time window. We also derive features from Inter Packet Gap (IPG) analytics, i.e., Exponentially Weighted Moving Average (EMA), Double Exponentially Weighted Moving Average (DEMA) and Commutative Sums of IPGs (CUSUM). We show their interrelation with objective QoE KPI (Stalls), Shifts and QoE model P.1203 in encrypted network scenarios, where MNOs have limited visibility to Network QoS. Such real-time estimation is ideal for network management, for instance, taking reactive performance diagnosis and resource allocation [1, 5]. Moreover, the QoE estimation technique will impact MNOs to review the SLA for proactive network capacity and configuration. Thus real-time (window-based or time-window) QoE inference technique enables proactive and reactive QoE-aware network traffic management.

Moreover, we released a framework named `EFFECTOR` [18], a reproducible framework to run real 4G and 5G use cases. To run real use cases, we provide commercial 4G and 5G datasets collected in the wild over six months in various regions [23]. The QoE evaluation framework is ideal for investigating QoS features with other objective KPIs such as – i) throughput prediction, ii) choice of segment, iii) quality shifts, iv) Adaptive Bitrate Streaming (ABS) algorithm selection, v) the choice of segment size, etc.

We also talk about the performance footprint of 5G with different use cases, protocols TCP and QUIC, and by using different ABS, i) Throughput, ii) Buffered iii) Hybrid. From different scenarios, we conclude that in case of congestion in the network, BBA quickly jumps to lower resolutions, then gradually increases to higher resolutions and bitrates over time. As it takes time to shift towards a high bitrate, there are less stalling events, and user experience a smooth transition from one quality to another. Throughput ABS follows a more greedy approach; therefore, it suffers from stalling events. Moreover, it has more similarity in terms of quality shifts to the popular streaming platform YouTube. On the other end, in Hybrid – Elastic, sudden congestion causes to switch to a lower resolution. However, Elastic remains in lower resolutions compared to BBA, ultimately with very few chances of stalling events.

Furthermore, we build datasets over TCP and QUIC for DASH video content and use various Machine Learning techniques. The dataset is composed of various time windows in seconds, i.e., (0.5, 1, 2, 3, 4, 5). We conclude that Random Forests and Artificial Neural Network (ANN) provides the best results. We also find that few windows are ideal for some ABS. For instance, in encrypted video traffic, a 4-second window provides the best results, and the accuracy is near 78 % in TCP and 77 % in QUIC while using ANN. The best performing classifier in TCP on a 4-second window, whereas in QUIC, it is KNN with 82 % accuracy in 5 seconds. We also talk about the importance of QoS features. We find that QoS features, i.e., packet size and throughput distribution into 10-90 percentile along with features derived from IPG provide a strong relationship with objective QoE KPIs, i.e., stalls and QoE Model P.1203. Therefore, the QoS features, which are lightweight and easy to compute, are efficient in processing real-time QoE at Edge hosts, thus saving processing time and memory consumption.

Apart from finding QoS features to estimate QoE, in the 5th chapter of this thesis, we provide real 4G and 5G datasets collected in the wild in different regions. From commercial 4G and 5G datasets and with the objective of comparing the performance footprints of 5G compared to 4G, we concluded that 5G outperforms 4G in video streaming. However, this is not the true case most of the time. 5G requires a stable connection to provide maximum perceived video quality. However, it suffers from stalling events in the case of Mobility due to frequent Intra-RAT HO and Inter-RAT HO.

Moreover, we also conclude that Channel Level Metrics (CLM), i.e., CQI, SNR, RSRQ, and RSRP, have a relationship with stalling events in real YouTube traffic. We propose a window based stalling event prediction technique to predict the binary classification of Stall vs. No Stall. We evaluated the window for up to 9 seconds and found the best time

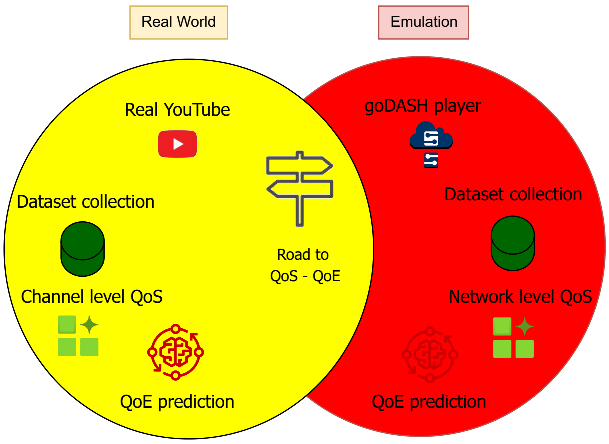


Figure 6.1: Road to QoE, real world and emulation – EFFECTOR.

window.

6.2 Road to QoE

In conclusion, the research work exhibits a common theme, namely the "Road to QoE", from real world and emulation based use cases. In Figure 6.1, both areas of research are depicted based on four stages each.

1. Environment – YouTube, Emulation
2. QoE logs collection
3. QoS logs collection
4. QoE prediction

YouTube: For YouTube, we consider CLM logs to predict stalling events. We saved CLM logs with 1-second granularity and find a relation between objective QoE stalls and channel logs. Next, ML classifiers are used to classify Stall vs. No Stall, objective QoE KPIs. QoE KPI stall strongly influences QoE [89]. In this phase of research work, we conclude that by looking only at CQI, RSRQ, RSRP and SNR and features derived from them, ML classifiers can predict QoE KPI stall. We used different ML classifiers, i.e., Decision Tree, Random Forests, ANN, KNN, for the prediction. We looked up to 9-seconds CLM metrics and found that a 7-second time window provides the best results.
Emulation: In Emulation – `EFFECTOR`, we consider real use case logs collected from the above phase of real YouTube experiments. We collect both QoE logs provided by goDASH player and network level QoS features. We conclude that time based QoS features extraction method provides promising results for estimating QoE using network level QoS features. We derive QoS features from Packet Time and Size and find that IPGs provide new venues for mapping QoS and QoE in DASH videos.

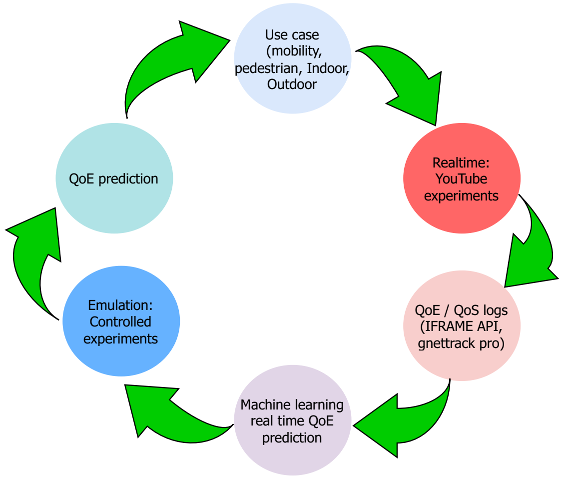


Figure 6.2: Closing the DASH video QoE loop.

6.3 Closing the DASH Video QoE Loop

Throughout this research work, the main objective has been to map QoS and QoE. However, emulation based QoS to QoE prediction requires real use cases to unveil convincing relationships between QoE metrics given the QoS features. Our DASH video QoE research loop consisted of six stages see Figure 6.2.

1. Definition of use case to create a realistic emulation environment;
2. Experiments with a real and popular streaming platform YouTube;
3. Collection of real QoE and QoS logs, with the smallest granularity of 1-second;
4. Application of ML techniques to predict objective QoE KPIs;
5. Investigation of QoS and QoE logs in the emulation environment and draw more complex relationships between QoS and objective QoE;
6. Prediction of objective QoE metrics.

We made all the source code and Frameworks reproducible to carry out more experimentation with more complex use cases and conditions [18].[1] [2]

6.4 Future Work

Finally, as prominent future work, a set of elaborated shortcomings was identified in the core chapters of this thesis. Among them, we can highlight a few objectives:

[1] https://github.com/razaulmustafa852/youtubegoes5g
[2] https://github.com/razaulmustafa852/EFFECTOR

- Such as the extension of `EFFECTOR` with SQAPE[3] reference topology alike complex network scenario, `Mininet-WiFi` access node to replicate 4G and 5G trace's channel condition (e.g., SINR, RSRP/Q, CQI) and deliver the framework in a resource-efficient, ready to use container format.

- More extensive experimental evaluation using more complex network use cases or scenarios to draw and find more lightweight QoS features such as Progressive Mean and EMA-CUSUM (mix) using IPGs as a baseline.

- In real-time YouTube QoE estimation, future work can be done in many directions. For instance, quality shifts are well known QoE metrics that influence the Mean Opinion Score (MOS). Therefore, we would like to investigate the CLM factors affecting the shifts – i) Up, ii) Down. Up refers to when a YouTube player chooses higher resolutions, i.e., hd720 to hd1080, and Down refers to a change in resolution from hd1080 to hd720.

- For 5G-aware streaming, research can be devoted to a recommender system or smart triggers to avoid stalls. For instance, when CLM meets certain values, an application on the UE device can generate an alarm. As discussed in Chapter 5, we observe stalling events in 5G under mobility conditions. Therefore, an application could continuously monitor location and mobility patterns to avoid stalls, thus, maximizing QoE.

- New 5G-aware ABR algorithms could be designed, leveraging triggers from CLM to enhance end-user video QoE.

[3] https://github.com/rtcostaf/INFOCOM2018

References

[1] Sarah Wassermann, Michael Seufert, Pedro Casas, Li Gang, and Kuang Li. Vicrypt to the rescue: Real-time, machine-learning-driven Video-QoE monitoring for encrypted streaming traffic. *IEEE Transactions on Network and Service Management*, 2020.

[2] Nisha Panwar, Shantanu Sharma, and Awadhesh Kumar Singh. A survey on 5G: The next generation of mobile communication. *Physical Communication*, 18:64–84, 2016.

[3] Kamila Ragimova, Vyacheslav Loginov, and Evgeny Khorov. Analysis of YouTube dash traffic. In *2019 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pages 1–5. IEEE, 2019.

[4] Diego Figueiredo, Rodrigo Dutra, Ilan Sousa, Aldebaro Klautau, and Pedro Batista. Predicting video bitrate from encrypted streaming traffic in SDN-based 5G networks with ml. In *2021 IEEE Latin-American Conference on Communications (LATINCOM)*, pages 1–6. IEEE, 2021.

[5] Craig Gutterman, Katherine Guo, Sarthak Arora, Trey Gilliland, Xiaoyang Wang, Les Wu, Ethan Katz-Bassett, and Gil Zussman. Requet: Real-time QoE metric detection for encrypted YouTube traffic. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 16(2s):1–28, 2020.

[6] 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)*, pages 81–87. IEEE, 2020.

[7] Abdelhak Bentaleb, Bayan Taani, Ali C Begen, Christian Timmerer, and Roger Zimmermann. A survey on bitrate adaptation schemes for streaming media over http. *IEEE Communications Surveys & Tutorials*, 21(1):562–585, 2018.

[8] Rongpeng Li, Zhifeng Zhao, Xuan Zhou, Guoru Ding, Yan Chen, Zhongyao Wang, and Honggang Zhang. Intelligent 5G: When cellular networks meet artificial intelligence. *IEEE Wireless communications*, 24(5):175–183, 2017.

[9] Sandeep Chinchali, Pan Hu, Tianshu Chu, Manu Sharma, Manu Bansal, Rakesh Misra, Marco Pavone, and Sachin Katti. Cellular network traffic scheduling with deep reinforcement learning. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[10] Thomas Stockhammer. Dynamic adaptive streaming over HTTP–: standards and design principles. In *Proceedings of the second annual ACM conference on Multimedia systems*, pages 133–144. ACM, 2011.

[11] Lam Dinh-Xuan, Michael Seufert, Florian Wamser, and Phuoc Tran-Gia. 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)*, pages 999–1004. IEEE, 2017.

[12] Itu-t recommendation p.1203: Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport. https://www.itu.int/rec/T-REC-P.1203/en/, 2016. [Online]. accessed 19-July-2018.

[13] Raza Ul Mustafa, Simone Ferlin, Christian Esteve Rothenberg, Darijo Raca, and Jason J. Quinlan. 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*, pages 57–64, 2020.

[14] Francesco Bronzino, Paul Schmitt, Sara Ayoubi, Guilherme Martins, Renata Teixeira, and Nick Feamster. Inferring streaming video quality from encrypted traffic: Practical models and deployment experience. *ACM SIGMETRICS Performance Evaluation Review*, 48(1):27–28, 2020.

[15] Iman Akbari, Mohammad A Salahuddin, Leni Ven, Noura Limam, Raouf Boutaba, Bertrand Mathieu, Stephanie Moteau, and Stephane Tuffin. A look behind the curtain: traffic classification in an increasingly encrypted web. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 5(1):1–26, 2021.

[16] Raza Ul Mustafa, David Moura, and Christian Esteve Rothenberg. Machine learning approach to estimate video QoE of encrypted dash traffic in 5G networks. In *2021 IEEE Statistical Signal Processing Workshop (SSP)*, pages 586–589. IEEE, 2021.

[17] Raza Ul Mustafa and Christian Esteve Rothenberg. Machine learning assisted real-time dash video QoE estimation technique for encrypted traffic. In *Proceedings of the 1st Conference on Mile-High Video*, MHV '22, page 123, New York, NY, USA, 2022. Association for Computing Machinery.

[18] Raza Ul Mustafa, Tariq Islam, Christian Esteve Rothenberg, and Pedro Henrique. EFFECTOR: DASH QoE and QoS Evaluation Framework For EnCrypTed videO tRaffic. In *36th IEEE/IFIP Network Operations and Management Symposium*

(NOMS 2023). IEEE, 2023.

[19] 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, pages 466–471, 2018.

[20] 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.

[21] Raza Ul Mustafa, Md Tariqul Islam, Christian Rothenberg, Simone Ferlin, Darijo Raca, and Jason J Quinlan. DASH QoE performance evaluation framework with 5G datasets. In *2020 16th International Conference on Network and Service Management (CNSM)*, pages 1–6. IEEE, 2020.

[22] Muhammad Jawad Khokhar, Thibaut Ehlinger, and Chadi Barakat. From network traffic measurements to QoE for internet video. In *2019 IFIP Networking Conference (IFIP Networking)*, pages 1–9. IEEE, 2019.

[23] Raza Ul Mustafa, Christian Esteve Rothenberg, and Chadi Barakat. Youtube goes 5G: Benchmarking youtube in 4G vs 5G, 2022.

[24] Darijo Raca, Dylan Leahy, Cormac J Sreenan, and Jason J Quinlan. Beyond throughput, the next generation: a 5G dataset with channel and context metrics. In *Proceedings of the 11th ACM Multimedia Systems Conference*, pages 303–308, 2020.

[25] Emna Hajlaoui, Aida Zaier, Abdelhakim Khlifi, Jihed Ghodhbane, Mouna Ben Hamed, and Lassâad Sbita. 4G and 5G technologies: A comparative study. In *2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, pages 1–6. IEEE, 2020.

[26] Daniel Calabuig, Sokratis Barmpounakis, Sonia Gimenez, Apostolos Kousaridas, Tilak R Lakshmana, Javier Lorca, Petteri Lunden, Zhe Ren, Pawel Sroka, Emmanuel Ternon, et al. Resource and mobility management in the network layer of 5G cellular ultra-dense networks. *IEEE Communications Magazine*, 55(6):162–169, 2017.

[27] James Nightingale, Pablo Salva-Garcia, Jose M Alcaraz Calero, and Qi Wang. 5G-QoE: QoE modelling for ultra-hd video streaming in 5G networks. *IEEE Transactions on Broadcasting*, 64(2):621–634, 2018.

[28] Ulrich Reiter, Kjell Brunnström, Katrien De Moor, Mohamed-Chaker Larabi, Manuela Pereira, Antonio Pinheiro, Junyong You, and Andrej Zgank. Factors influencing quality of experience. In *Quality of experience*, pages 55–72. Springer, 2014.

[29] Péter András Kara, László Bokor, Andreas Sackl, and Mariana Mourão. What your

phone makes you see: Investigation of the effect of end-user devices on the assessment of perceived multimedia quality. In *2015 Seventh International Workshop on Quality of Multimedia Experience (QoMEX)*, 2015.

[30] Christian Esteve Rothenberg, Danny Alex Lachos Perez, Nathan F. Saraiva de Sousa, Raphael Rosa, Raza Ul Mustafa, Md Tariqul Islam, and Pedro Henrique Gomes. Intent-based control loop for DASH video service assurance using ML-based edge QoE estimation. In *2020 6th IEEE International Conference on Network Softwarization (NetSoft) (NetSoft 2020)*, Ghent, Belgium, June 2020.

[31] M Hammad Mazhar and Zubair Shafiq. Real-time video quality of experience monitoring for HTTPS and QUIC. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 1331–1339. IEEE, 2018.

[32] Fan Zhang, Long Xu, and Qian Zhang. Maximum-likelihood visual quality based on additive log-logistic model. In *2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)*, pages 470–475. IEEE, 2013.

[33] Margaret H Pinson, Nicolas Staelens, and Arthur Webster. The history of video quality model validation. In *2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)*, pages 458–463. IEEE, 2013.

[34] Yining Qi and Mingyuan Dai. The effect of frame freezing and frame skipping on video quality. In *2006 international conference on intelligent information hiding and multimedia*, pages 423–426. IEEE, 2006.

[35] Quan Huynh-Thu and Mohammed Ghanbari. Temporal aspect of perceived quality in mobile video broadcasting. *IEEE Transactions on Broadcasting*, 54(3):641–651, 2008.

[36] Blazej Lewcio, Benjamin Belmudez, Amir Mehmood, Marcel Wältermann, and Sebastian Möller. 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)*, pages 1–6. IEEE, 2011.

[37] Michael Zink, Jens Schmitt, and Ralf Steinmetz. Layer-encoded video in scalable adaptive streaming. *IEEE Transactions on Multimedia*, 7(1):75–84, 2005.

[38] Muhammad Jawad Khokhar, Thierry Spetebroot, and Chadi Barakat. An online sampling approach for controlled experimentation and QoE modeling. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2018.

[39] Florian Wamser, Michael Seufert, Pedro Casas, Ralf Irmer, Phuoc Tran-Gia, and Raimund Schatz. Yomoapp: A tool for analyzing QoE of YouTube HTTP adaptive streaming in mobile networks. In *2015 European Conference on Networks and*

Communications (EuCNC), pages 239–243. IEEE, 2015.

[40] Parikshit Juluri, Louis Plissonneau, and Deep Medhi. Pytomo: a tool for analyzing playback quality of YouTube videos. In *2011 23rd International Teletraffic Congress (ITC)*, pages 304–305. IEEE, 2011.

[41] Xi Liu, Florin Dobrian, Henry Milner, Junchen Jiang, Vyas Sekar, Ion Stoica, and Hui Zhang. A case for a coordinated internet video control plane. In *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*, pages 359–370, 2012.

[42] Florin Dobrian, Vyas Sekar, Asad Awan, Ion Stoica, Dilip Joseph, Aditya Ganjam, Jibin Zhan, and Hui Zhang. Understanding the impact of video quality on user engagement. *ACM SIGCOMM Computer Communication Review*, 41(4):362–373, 2011.

[43] Konstantinos Kousias, Mohammad Rajiullah, Giuseppe Caso, Ozgu Alay, Anna Brunstrom, Luca De Nardis, Marco Neri, Usman Ali, and Maria-Gabriella Di Benedetto. Implications of handover events in commercial 5G non-standalone deployments in Rome. In *Proceedings of the ACM SIGCOMM Workshop on 5G and Beyond Network Measurements, Modeling, and Use Cases*, pages 22–27, 2022.

[44] Yueyang Pan, Ruihan Li, and Chenren Xu. The first 5G-LTE comparative study in extreme mobility. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 6(1):1–22, 2022.

[45] Eman Ramadan, Arvind Narayanan, Udhaya Kumar Dayalan, Rostand AK Fezeu, Feng Qian, and Zhi-Li Zhang. Case for 5G-aware video streaming applications. In *Proceedings of the 1st Workshop on 5G Measurements, Modeling, and Use Cases*, pages 27–34, 2021.

[46] Arvind Narayanan, Xumiao Zhang, Ruiyang Zhu, Ahmad Hassan, Shuowei Jin, Xiao Zhu, Xiaoxuan Zhang, Denis Rybkin, Zhengxuan Yang, Zhuoqing Morley Mao, et al. A variegated look at 5G in the wild: performance, power, and QoE implications. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, pages 610–625, 2021.

[47] Darijo Raca, Maëlle Manifacier, and Jason J Quinlan. goDASH-GO accelerated HAS framework for rapid prototyping. 2020.

[48] Ricky KP Mok, Edmond WW Chan, and Rocky KC Chang. Measuring the quality of experience of HTTP video streaming. In *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*, pages 485–492. IEEE, 2011.

[49] Tobias Hoßfeld, Raimund Schatz, and Udo R Krieger. QoE of YouTube video stream-

ing for current internet transport protocols. In *International Conference on Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance*, pages 136–150. Springer, 2014.

[50] Giorgos Dimopoulos, Ilias Leontiadis, Pere Barlet-Ros, and Konstantina Papagiannaki. Measuring video QoE from encrypted traffic. In *Proceedings of the 2016 Internet Measurement Conference*, pages 513–526. ACM, 2016.

[51] Zahaib Akhtar, Yun Seong Nam, Ramesh Govindan, Sanjay Rao, Jessica Chen, Ethan Katz-Bassett, Bruno Ribeiro, Jibin Zhan, and Hui Zhang. Oboe: Auto-tuning video ABR algorithms to network conditions. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '18, page 44–58, New York, NY, USA, 2018. Association for Computing Machinery.

[52] Luca De Cicco, Giuseppe Cilli, and Saverio Mascolo. Erudite: A deep neural network for optimal tuning of adaptive video streaming controllers. In *Proceedings of the 10th ACM Multimedia Systems Conference*, MMSys '19, page 13–24, New York, NY, USA, 2019. Association for Computing Machinery.

[53] Y. Chien, K. C. Lin, and M. Chen. Machine learning based rate adaptation with elastic feature selection for HTTP-based streaming. In *2015 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2015.

[54] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. Neural adaptive video streaming with pensieve. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '17, page 197–210, New York, NY, USA, 2017. Association for Computing Machinery.

[55] Y. Sani, D. Raca, J. J. Quinlan, and C. J. Sreenan. SMASH: A supervised machine learning approach to adaptive video streaming over HTTP. In *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6, 2020.

[56] Sarah Wassermann, Nikolas Wehner, and Pedro Casas. Machine learning models for YouTube QoE and user engagement prediction in smartphones. *ACM SIGMETRICS Performance Evaluation Review*, 46(3):155–158, 2019.

[57] Neeraj Bhargava, Girja Sharma, Ritu Bhargava, and Manish Mathuria. Decision tree analysis on j48 algorithm for data mining. *Proceedings of International Journal of Advanced Research in Computer Science and Software Engineering*, 3(6), 2013.

[58] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[59] Tasnim Abar, Asma Ben Letaifa, and Sadok El Asmi. Machine learning based QoE

prediction in SDN networks. In *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 1395–1400. IEEE, 2017.

[60] Bekir Karlik and A Vehbi Olgac. Performance analysis of various activation functions in generalized mlp architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems*, 1(4):111–122, 2011.

[61] Berndt Müller, Joachim Reinhardt, and Michael T Strickland. *Neural networks: an introduction*. Springer Science & Business Media, 1995.

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

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

[64] L. De Cicco et al. ELASTIC: A Client-Side Controller for Dynamic Adaptive Streaming over HTTP (DASH). In *2013 20th International Packet Video Workshop*.

[65] 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, pages 187–198, New York, NY, USA, 2014. ACM.

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

[67] Z. Li et al. Probe and Adapt: Rate Adaptation for HTTP Video Streaming At Scale. *IEEE Journal on Selected Areas in Communications*, 32(4):719–733, April 2014.

[68] Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell, and Mark Watson. A buffer-based approach to rate adaptation: Evidence from a large video streaming service. In *ACM SIGCOMM Computer Communication Review*, volume 44, pages 187–198. ACM, 2014.

[69] Jian Qiao, Xuemin Sherman Shen, Jon W Mark, and Lei Lei. Video quality provisioning for millimeter wave 5G cellular networks with link outage. *IEEE Transactions on Wireless Communications*, 14(10):5692–5703, 2015.

[70] Muhammad Ismail and Weihua Zhuang. Statistical QoS guarantee for wireless multi-homing video transmission. In *2013 IEEE Global Communications Conference (GLOBECOM)*, pages 4615–4620. IEEE, 2013.

[71] Darijo Raca, Maëlle Manifacier, and Jason J. Quinlan. goDASH - GO accelerated HAS framework for rapid prototyping. In *Proceedings of the 12th International Conference on Quality of Multimedia Experience*, 2020.

[72] Ramon R Fontes, Samira Afzal, Samuel HB Brito, Mateus AS Santos, and Christian Esteve Rothenberg. Mininet-WiFi: Emulating software-defined wireless networks. In *2015 11th International Conference on Network and Service Management*

(CNSM), pages 384–389. IEEE, 2015.

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

[74] Doru Gabriel Balan and Dan Alin Potorac. Linux htb queuing discipline implementations. In *2009 First International Conference on Networked Digital Technologies*, pages 122–126. IEEE, 2009.

[75] John O'Sullivan, Darijo Raca, and Jason J Quinlan. godash 2.0-The Next Evolution of HAS Evaluation. In *2020 IEEE 21st International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, pages 185–187. IEEE, 2020.

[76] C Rothenberg S Ferlin D Raca JJ Quinlan RU Mustafa, MT Islam. Dash QoE performance evaluation framework with 5G datasets. https://drive.google.com/drive/folders/1y4HZ7sYxzCi_yXTpAnZwMQlQy5na04b?usp=sharing, 2020.

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

[78] Dashframework. Github repository. https://github.com/sajibtariq/dashframework, 2020.

[79] Suneet Kumar Singh, Christian Esteve Rothenberg, Marcelo Caggiani Luizelli, Gianni Antichi, Pedro Henrique Gomes, and Gergely Pongrácz. HH-IPG: Leveraging inter-packet gap metrics in P4 hardware for heavy hitter detection. *IEEE Transactions on Network and Service Management*, pages 1–1, 2022.

[80] Dashframework. Github repository. https://github.com/razaulmustafa852/EFFECTOR, 2022.

[81] Michael Seufert, Pedro Casas, Nikolas Wehner, Li Gang, and Kuang Li. Stream-based machine learning for real-time QoE analysis of encrypted video streaming traffic. In *2019 22nd Conference on innovation in clouds, internet and networks and workshops (ICIN)*, pages 76–81. IEEE, 2019.

[82] Irena Orsolic and Lea Skorin-Kapov. A framework for in-network qoe monitoring of encrypted video streaming. *IEEE Access*, 8:74691–74706, 2020.

[83] Craig Gutterman, Katherine Guo, Sarthak Arora, Xiaoyang Wang, Les Wu, Ethan Katz-Bassett, and Gil Zussman. Requet: Real-time QoE detection for encrypted YouTube traffic. In *Proceedings of the 10th ACM Multimedia Systems Conference*, pages 48–59, 2019.

[84] Sarah Wassermann, Michael Seufert, Pedro Casas, Li Gang, and Kuang Li. I see what you see: Real time prediction of video quality from encrypted streaming traffic. In *Proceedings of the 4th Internet-QoE Workshop on QoE-based Analysis and*

Management of Data Communication Networks, pages 1–6, 2019.

[85] Sana Aroussi, Thouraya Bouabana-Tebibel, and Abdelhamid Mellouk. Empirical QoE/QoS correlation model based on multiple parameters for VoD flows. In *2012 IEEE Global Communications Conference (GLOBECOM)*, pages 1963–1968. IEEE, 2012.

[86] Jason J Quinlan, Ahmed H Zahran, and Cormac J Sreenan. Datasets for AVC (H. 264) and HEVC (H. 265) evaluation of dynamic adaptive streaming over HTTP (DASH). In *Proceedings of the 7th International Conference on Multimedia Systems*, pages 1–6, 2016.

[87] Guangyi Liu, Yuhong Huang, Zhuo Chen, Liang Liu, Qixing Wang, and Na Li. 5g deployment: Standalone vs. non-standalone from the operator perspective. *IEEE Communications Magazine*, 58(11):83–89, 2020.

[88] Farhana Afroz, Ramprasad Subramanian, Roshanak Heidary, Kumbesan Sandrasegaran, and Solaiman Ahmed. SINR, RSRP, RSSI and RSRQ measurements in long term evolution networks. *International Journal of Wireless & Mobile Networks*, 2015.

[89] Florin Dobrian, Asad Awan, Dilip Joseph, Aditya Ganjam, Jibin Zhan, Vyas Sekar, Ion Stoica, and Hui Zhang. Understanding the impact of video quality on user engagement. *Communications of the ACM*, 56(3):91–99, 2013.

Appendix A

4G and 5G Dataset Description Collected in the Wild

We made public 4G and 5G datasets collected using YouTube as a baseline on GitHub: https://github.com/razaulmustafa852/youtubegoes5g. The description of the dataset is as:

A.1 Channel Logs

In "Channel Logs" folder there is 1 file for each experiment. The name of the file is the same as his Experiment ID (Eid). In each file, there are many fields such as Timestamp, Longitude, Latitude, NetworkTech, NetworkMode, Level, Qual, SNR, CQI, LTERSSI, DL bitrate, UL bitrate, Altitude, Height, State, EVENT, Eid.

- Timestamp - Sequence of characters or encoded information identifying when a certain event occurred, giving date and time of day
- Longitude in decimal format
- Latitude in decimal format
- NetworkTech - Current Broadband cellular network technology – 2G, 3G, 4G, 5G
- NetworkMode - Current network mode. Ex: LTE, NR
- Level - Received Signal Receive Power (RSRP)
- Qual - The signal quality of the network. Received Signal Received Quality (RSRQ)
- SNR - Signal to noise ratio

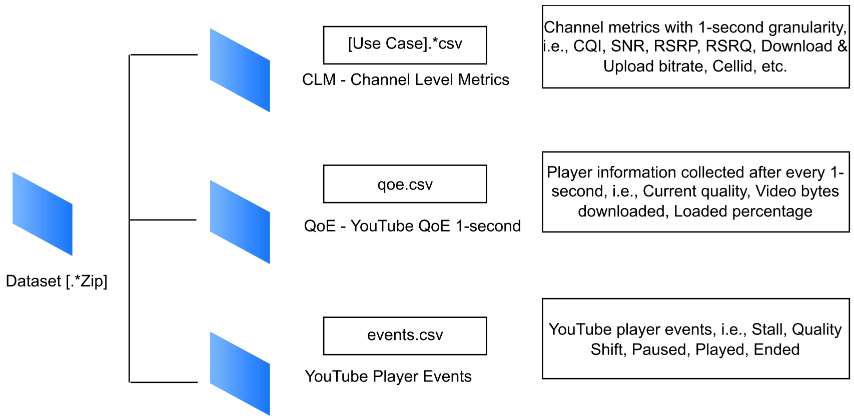


Figure A.1: Files overview.

- CQI - Channel quality indicator.
- LTERSSI - Received power in the whole band – 4G only
- DL bitrate - The current downlink data transfer speed in kbps
- UL bitrate - The current uplink data transfer speed in kbps
- Altitude - The GPS measured altitude
- Height - The difference between Altitude and Ground values
- State - IDLE, CALL, DATA – The current phone state – if it is idle or in active data transfer or in active voice call.
- EVENT - Current event name Ex: Periodic, Log Start, Log End
- Eid - Experiment ID

A.2 QoE Logs

YouTube QoE logs are of two types i) Events and ii) QoE
Events: For Events, we saved 6 events as: -1 – unstarted, 0 – ended, 1 – playing, 2 – paused, 3 – buffering, 5 – video cued
QoE: For QoE, we save player current state after every 1-second such as: Current Quality, Video Bytes Downloaded, Loaded Percentage, and Time For both Events and QoE, we have "EID", which is exactly the same as of CLM log file name. Therefore, you can extract QoE of YouTube and CLM by using EID - Experiment ID. For example: "5A12.csv" is CLM log file, and its corresponding QoE and Events of YouTube are available in csv files by using EID as "5A12".

A.3 Files Overview

The file hierarchy of the dataset is explained in Figure A.1. In the zip folder, there are three folders.

- **CLM:** In `CLM` folder, we provide [Use case.*csv] files.
  - Files starting with 4 are - 4G experiments, 4G cell phone, 4G technology
  - File starting with 5 are - 5G experiments, 5G cell phone, 5G technology
  - For example, 5Po30 - The experiment is done with 5G cell phone with 5G technology and the use case is - Pedestrian. Mostly the capital Letter (M, P, A, I, O) in the second position represents use case, i.e., M - Mobility, P - Pedestrian, A - Terminals, I - Indoor, O - Outdoor. For example, 5Or29 - 5G technology, use case - Outdoor.
  - Few use cases starting with `b, m` are Mobility and use cases starting with `w, c, s` are Pedestrian.
- **QoE:** In folder `QoE`, In the file [qoe.csv], we provide 1-second player information of YouTube. During the video session, we made a JavaScript function that stores current player statistics, i.e., Current quality, Video bytes downloaded, Loaded percentage, Timestamp, Available qualities, and Experiment ID as `Eid`.
- **Events:** In folder `Events`, the file [events.csv], we provide each video session events, i.e., Stalls – Buffering, Quality, Event time, Experiment ID as `Eid`.

Appendix B

Python Scripts & VMs

We provide all scripts and datasets for the reproducibility of the proposed work. Scripts can be downloaded from publicly available repositories on Github.

B.1 Per-segment QoS Features Extractions

For per-segment QoS features extraction method, use the following public repo:

- https://github.com/razaulmustafa852/5G

B.2 Time Slot QoS Features Extraction

For packet level statistics of encrypted DASH video stream use the following scripts:

- https://github.com/razaulmustafa852/EFFECTOR

B.3 Encrypted and Unencrypted VM for large scale experiments in 4G and 5G

Per-segment QoS features extraction VM and framework

- https://drive.google.com/drive/folders/1y4HZ7sYxzCi_yXTpAnZwMQlQy5na04b?usp=sharing
- https://drive.google.com/file/d/14fyG88dO9LthucnSw19_5QYyijtoFh17/view?usp=sharing